15

Natural Language Processing

Enrico Franconi

Abstract

In most natural language processing applications, Description Logics have been used to encode in a knowledge base some syntactic, semantic, and pragmatic elements needed to drive the semantic interpretation and the natural language generation processes. More recently, Description Logics have been used to fully characterise the semantic issues involved in the interpretation phase. In this Chapter the various proposals appeared in the literature about the use of Description Logics for natural language processing will be analysed.

15.1 Introduction

Since the early days of the Kl-One system, one of the main applications of Description Logics has been for *semantic interpretation* in Natural Language Processing [Brachman *et al.*, 1979]. Semantic interpretation is the derivation process from the syntactic analysis of an utterance to its *logical form*—intended here as the representation of its literal deep and context-dependent meaning. Typically, Description Logics have been used to encode in a knowledge base both syntactic and semantic elements needed to drive the semantic interpretation process. A part of the knowledge base constitutes the *lexical semantics* knowledge, relating words and their syntactic properties to concept structures, while the other part describes the *contextual* and *domain* knowledge, giving a deep meaning to concepts. By developing this idea further, a relevant part of the research effort has been devoted to the development of linguistically motivated ontologies, i.e., large knowledge bases where both concepts closely related to lexemes and domain concepts coexist together. Logical forms and various kinds of internal semantics representations based on Description Logics may also provide the basis for further computational processing such as representing common meanings in Machine Translation applications, generating coherent text starting form its semantic content, answering database queries, and for dialogue management.

After a big success in the eighties and in the beginning of the nineties (see, e.g., the paper collection in [Sowa, 1991]), the interest of the applied computational linguistic community towards Description Logics began to drop, as well as its interest in well founded theories on syntax or semantics. At the time of writing this chapter, there is no major applied project in Natural Language Processing making use of Description Logics. This is due to the positive achievements in real applications of the systems based on shallow analysis and statistical approaches to semantics, initiated by the applications in the message understanding area.

In this Chapter the basic uses of Description Logics for Natural Language Processing will be analysed, together with a little bit of history, and the role of Description Logics in the current state of the art in computational linguistics will be pointed out. Obviously, space constraints will lead to several omissions and over-simplifications.

15.2 Semantic interpretation

In order to understand the role of Description Logics for semantic interpretation, let us first introduce a general setting for the process of deriving a logical form of an utterance.

A basic property of a logical form as a semantic representation of a natural language constituent—such as a noun phrase (NP) or a verb phrase (VP)—is *compositionality*, i.e., the semantic representation of a constituent is a function of the semantic interpretation of its sub-constituents. Thus, a close correspondence between syntactic structure and logical form is allowed. In this way, a parser working according to some grammar rules can incrementally build up the semantic interpretation of an utterance using the corresponding lexical semantic rules of logical composition—specifying how the logical terms associated to the sub-constituents are to be combined in order to give the formula for the constituent. Thus, each lexeme has associated a (possibly complex) logical term, which forms its contribution to the meaning of the utterance it is part of.

In the context of such a formalism, an effective semantic lexical discrimination process could be carried on during parsing, by cutting out the exponential factor due to the explicit treatment all the possible derivations. Semantically un-plausible interpretations can be discarded, by checking—whenever the parser tries to build a constituent—the inconsistency of the logical form compositionally obtained at that stage. This leaves out many syntactically plausible but semantically implausible interpretations. Such a discrimination step is highly effective in restricted domain applications, where the world knowledge considerably reduces the number of possible models. Clearly, the more the contextual and domain knowledge is taken into consideration when evaluating a logical form, the more effective is the discrimination process. Thus, consistency checking of logical forms plays the role of a generalised selectional restrictions mechanism.

But which is the relationship between a syntactic constituent and its range of possible lexical semantic contributions? The conceptual content of a lexeme should convey both the lexical relations—such as, for example, synonymy, hyponymy, incompatibility—and the sub-categorisation information about the expected arguments (aka complements) of the lexical entry. For example the verb *paint* may be conceptualised as an event having an *agent* thematic role corresponding to the *subject* syntactic argument with a specified selectional restriction being the concept *animate*. It is important to distinguish the syntactic information—such as the lexical relations and the sub-categorisation frame constraining the complements to have specific syntactic structures—from the semantic information—such as the thematic roles and their selectional restrictions. A semantic lexical entry will specify the appropriate mappings between the syntactic structure of the lexeme and the conceptual information.

The situation is, of course, a bit more complex, since, for example, there is no direct obvious conceptual content to lexemes belonging to particular syntactic categories like adjectives or adverbs. Moreover there is a distinction between complements (which are considered as internal arguments) and *adjuncts* (which are considered as modifiers). It is outside the scope of this chapter to analyse the correspondence between syntax and semantics and its compositional nature (see, e.g., [Jackendoff, 1990; Pustejovsky, 1988]).

For example, the sentence "A painter paints a fresco", involves the concepts Painter, Fresco, and Paint, where the concept Paint has two thematic roles associated to it, an agent and a goal, with the concepts Animate and Inanimate as respective selectional restrictions. Moreover, the conceptualisations should include the facts that a Painter is a sub-concept of Animate, a Fresco is a sub-concept of Inanimate, and the concepts Animate and Inanimate are disjoint. This information is enough, for example, to validate the above sentence, while it would discard as semantically implausible the sentence "A fresco paints a painter". This conceptualisation and its relationship with the lexical knowledge can be encoded in a Description Logics knowledge base.

Many studies have been done about building a good Description Logics knowledge base for natural language processing (also called *ontology*) [Bateman, 1990; Hovy and Knight, 1993; Knight and Luk, 1994; Bateman *et al.*, 1995]—see also Chapter 14. A good linguistically motivated ontology ought to be partitioned into a language-dependent but domain-independent part (the *upper model*) and a language-independent but domain-dependent part (the *domain model*)—but this result is theoretically very hard to achieve [Bateman, 1990; Lang, 1991]. A good linguistically motivated ontology should be used both for semantic interpretation and for natural language generation (see Section 15.4). The conceptualisation in the ontology should be at a level of granularity which may depend on the application: if selectional restrictions are too specific, disambiguation is achieved, but probably many correct sentences will be discriminated (e.g., the sentences involving some form of metaphor, type shifting, or metonymy); if selectional restrictions are too general, the opposite problem may appear. In principle, a good linguistically motivated ontology should be abstract, large-scale, reusable. However, these goals are very hard to achieve since they conflict with the practical need to implement effective and discriminating ontologies in specialised domains.

The ideas just sketched form the theoretical background of any application of Description Logics for semantic interpretation, since the early works where KL-ONE was involved [Bobrow and Webber, 1980; Sondheimer *et al.*, 1984; Brachman and Schmolze, 1985; Jacobs, 1991]. Every realised system relies on the so called *multilevel semantics architecture* [Lavelli *et al.*, 1992], where a sequence of processing phases is distinguished:

- Lexical discrimination: whenever the parser tries to build a constituent, the *consistency* of the semantic part of such a constituent is checked. In parallel, a first logical form is built up—where references and quantifiers scoping are still ambiguous—expressing the meaning of the sentence in the most specialised way with respect to the semantic lexicon and the background knowledge. Heuristics is applied to the minimal form in order to obtain a preferential ordering of the semantically consistent but still lexically ambiguous interpretations.
- Anaphora and quantifier scoping resolution: the semantically plausible referents for linguistic expressions such as definite NPs, pronouns and deictic references are identified, and the scope of quantifiers is resolved by making explicit the different unambiguous interpretations. Syntactic-based heuristics are used to cut down the various derivations to a unique unambiguous one.
- Contextual interpretation: decides how to react in a given dialogic situation, considering the type of request, the context, the model of the interest of the user. It makes use of knowledge about the speech acts, the dialogue and the user model.

It has to be emphasised the fact that all the approaches aim at deriving a unique unambiguous logical form. For this purpose, the logical form is treated as a mere compositionally-obtained data structure on which to operate ad-hoc algorithms for solving ambiguities, with the support of the information represented in the knowledge base. There is no attempt to give a logic-based semantics to the "logical form" during the disambiguation phases. The role of Description Logics is thus limited to serve a lexically motivated knowledge base, which is used for building the logical form. Some approaches pretend to represent the logical form itself as Description Logics assertions, but in fact they use it just as a support for somehow computing the real logical form. Section 15.3 will discuss the few Description Logics based well founded approaches, where the whole semantic interpretation process has been given a logical foundation.

A number of recent important projects involving Description Logics for semantic interpretation are listed below.

- The Janus system [Weischedel, 1989], where the consistency check of the selectional restrictions was implemented as double up-and-down subsumption check.
- The Xtra system [Allgayer *et al.*, 1989], proposing a clear distinction between the domain independent linguistically motivated part of the knowledge base (called Functional-Semantic Structure, FSS), and the domain dependent part (called Conceptual Knowledge Base, CKB) modelling the knowledge of an underlying expert system.
- The Pracma project [Fehrer *et al.*, 1994], in which an expressive Description Logic has been studied to support special inferences such as probabilistic reasoning, non-monotonic reasoning, and abductive reasoning.
- The Lilog project [Herzog and Rollinger, 1991], funded by IBM, a very ambitious research project for studying the logical foundations of the semantics of natural language, with an emphasis to computational aspects. The project belongs to the category of projects where the whole semantic interpretation process has been given a logical foundation—by means of a sorted first order logic. However, the role of Description Logics is again just as a knowledge server during the various interpretation and disambiguation phases.
- The AlFresco system, a multi-modal dialogue prototype for the exploration of Italian fourteenth century painters and frescoes [Stock *et al.*, 1991; 1993], and the natural language interface for the *concierge* of the system Maia, a mobile robot with intelligent capabilities in the domain of office activities [Samek-Lodovici and Strapparava, 1990; Lavelli *et al.*, 1992; Franconi, 1994]. These systems are characterised by the presence of natural language dialogues, so that logical form becomes central to convey the meaning for the evolving *behaviour* of the system.
- The Verbmobil project [Wahlster, 2000], a large speech-to-speech translation project, with translations in German, English, and Japanese. In Verbmobil, the role of Description Logics is limited to the off-line pre-computation of a taxonomy of concepts with thematic roles and selectional restrictions, which are then used by ad-hoc rules during the run-time disambiguation phase.
- The Ford's Direct Labor Management System (DLMS) [Rychtyckyj, 1996; 1999] is one of the few industrial level examples of a Description Logic based application involving natural language. DLMS utilises in a pretty standard way

a Description Logic knowledge base to build the semantic interpretation of *process sheets*—natural language documents containing specific information about work instructions—and to generate from them structured descriptions of the parts and the tools required for allocating labour at the car plant floor.

15.3 Reasoning with the logical form

Traditionally, the logical form has been considered in computational linguistics as only representing the literal—i.e., context independent—meaning of an utterance, as clearly distinguished from the representation of the surface syntactical constituent structure, and from a deeper semantic representation—function of discourse context and world knowledge. Thus, the logical form plays in these cases an intermediate role between syntax and the deep semantics, and it is therefore not intended to fully contain the meaning in context of the utterance. Moreover, quite often a further distinction is introduced among *quasi* logical forms—i.e., literal under-specified semantic representations—and proper logical forms—i.e., literal unambiguous derivations.

The reasons for having separated the literal under-specified, the literal unambiguous, and the deep meaning representations are mainly pragmatic rather than theoretical. Pure linguists would say that any sentence has just one unambiguous meaning, being the possible ambiguity introduced by under-constraining the interpretation process—e.g., by not adequately considering the context knowledge. In such a case, they would speak of different possible ending paths in the derivation (i.e., interpretation) process, each one of them being again unambiguous. Clearly, this approach is infeasible from a computational point of view: first, because the number of derivations might combinatorially increase; and secondly because the interdependencies among the derivations are lost.

On the other hand, computational linguists consider ambiguities as part of the meaning of utterances, with the ultimate goal of being able to reason with such under-specified expressions, in order to increase compactness in the representation and efficiency in the processing. Allen [1993] argues that

> ... one of the crucial issues facing future natural language systems is the development of knowledge representation formalisms that can effectively handle ambiguity.

We can identify two main approaches. The classical *computational* approaches—like the ones described above—rely on the modularity of the semantic analysis process—the multilevel semantics architecture—starting from the under-specified representation and ending up with an unambiguous and context-dependent representation. The *semantic-oriented* approaches usually propose a very expressive logical language—possibly with an expressivity greater than FOL—with the goal of giving a clear semantics to many NL phenomena, and in particular to ambiguities and under-specification. Ambiguities can be roughly classified as follows: lexical ambiguities introduced by, e.g., prepositions, nouns, and verbs; structural ambiguities such as PP-attachment ambiguities; referential ambiguities such as quantification scoping and anaphora.

A disadvantage of the first approach is that there are no solid formal grounds for the proper use of the logical form, and in particular for the treatment of ambiguity, so that operations on the logical form are often based on heuristics and ad-hoc procedures. This can be justified by the fact that reasoning on logical forms including—among other things—domain knowledge, incomplete and ambiguous terms, unsolved references, under-specified quantifications, is considered a hard computational task. Computational linguists have devised structural processing techniques based on syntax, selectional restrictions, case grammars, and structured information such as frames and type hierarchies—carefully trying to avoid or to drastically reduce the inclusion in the computational machinery of logical inference mechanisms for treating ambiguities. Of course, these techniques often need ad-hoc mechanisms when such ambiguities come into play. The computational approach is an example of "*knowledge representation as engineering*".

On the other hand, a number of recent works in applying Description Logics to Natural Language Processing ([Quantz, 1995; Franconi, 1996; Ludwig *et al.*, 2000]) are getting closer to a semantic-oriented approach, but they follow a minimalist conceptualisation, and they emphasise the computational aspects. Instead of trying to solve sophisticated semantic problems of natural language, they try to logically reconstruct some *basic* issues in a general way, which is *compositional*, *homogeneous*, *principled*, and interesting from an applicative point of view. The main idea of these approaches is to take logical forms seriously: they do not only represent the literal meaning of the fragment, but also lexical ambiguities, represent unresolved referents via variables and equality, interpret plural entities and (generalised) quantifiers, and are linked to a rich theory of the domain. To that purpose, an expressive logical language should have a proper reasoning mechanism, and nonetheless be compositional.

In this Section an abstract overview will be given by means of examples, in a way that, we believe, common ideas will be captured.

Let us first try to understand how a logical form can be characterised in terms of proper logical constructs. It is observed that, assuming the widely accepted Davidsonian view on eventualities, natural language phrases—such as a NP or a VP—explicitly introduce discourse referents stating the existence of individuals or events of the domain model. Introduced referents are represented as existentially quantified variables. The possibility of having variables and constants allows for the representation of referential ambiguities. This is the basis of most works on logical formalisations of the logical form.

For example, the NP *A fresco of Giotto* might be given the following logical form

$$\exists b.\ \mathsf{Fresco}(b) \wedge \mathsf{of}(b, \mathsf{GIOTTO})$$

while the NP *A fresco painted by Giotto* might be given the logical form

$$\exists b, e.\ \mathsf{Fresco}(b) \wedge \mathsf{Paint}(e) \wedge \mathsf{agent}(e, \mathsf{GIOTTO}) \wedge \mathsf{goal}(e, b). \tag{15.1}$$

As we have pointed out above, consistency checking of a (partial) logical form corresponding to a constituent may help in the semantic discrimination process. Thus, in a restricted application domain, we would like to discard a sentence like *A fresco paints Giotto*, since its logical form

$$\exists b, e.\ \mathsf{Fresco}(b) \wedge \mathsf{Paint}(e) \wedge \mathsf{agent}(e, b) \wedge \mathsf{goal}(e, \mathsf{GIOTTO})$$

would be inconsistent with respect to a general domain theory of frescoes and animate things that we could attach to the lexicon:

$$\begin{array}{l}\forall x, y.\ \mathsf{Paint}(x) \rightarrow (\mathsf{agent}(x, y) \rightarrow \mathsf{Animate}(y))\\ \forall x.\ \mathsf{Animate}(x) \rightarrow \neg\mathsf{Inanimate}(x)\\ \forall x.\ \mathsf{Fresco}(x) \rightarrow \mathsf{Inanimate}(x).\end{array}$$

Such an axiomatic theory plays the role of *meaning postulates* for the predicates appearing in the logical form; they can be also considered as a set of *predicate level axioms*. Using a Description Logics based formalism, this will be written as the following theory:

$$\begin{array}{rcl}\mathsf{Paint} & \sqsubseteq & \forall\mathsf{agent.Animate}\\ \mathsf{Animate} & \sqsubseteq & \neg\mathsf{Inanimate}\\ \mathsf{Fresco} & \sqsubseteq & \mathsf{Inanimate}.\end{array}$$

This is the place where Description Logics play a formal role as general domain theories representing the basic ontological properties of common-sense domain Knowledge.

Let us consider the *deep* meaning of *A fresco of Giotto*. The NP is ambiguous (at least) with respect to the two readings *A fresco painted by Giotto* and *A fresco owned by Giotto*. We could reformulate the ambiguous logical form, by enumerating the non ambiguous derivations, i.e., by disjoining the logical forms of the two readings. However, it is infeasible to explicitly enumerate all the (exponentially large) number of readings; moreover, this would not add any information to the logical form. Note however that traditional computational approaches pretend to always find a unique non ambiguous representation for the final logical form, based on syntactically and contextually motivated heuristics; in this case, the enumeration will be the basis for an ad-hoc preferential ordering. If the logical form is written instead as

$$\exists b.\ \mathsf{Fresco}(b) \wedge (\mathsf{paintedBy} \sqcup \mathsf{ownedBy})(b, \mathsf{GIOTTO}) \tag{15.2}$$

then each of the two readings clearly entails this *ambiguous* (or, better, underspecified) representation. Of course, the use of an explicit disjunction to encode the ambiguity requires a particular treatment of the natural language negation, which can not be represented as a classical negation in the logical form. In fact, derivations from the ambiguous content are independent traces and, for example, de Morgan's law would not hold anymore. The treatment of natural language negation has never been considered in description logic based approaches. So, we assume the logical form to be always positive; of course, this is not necessary for the description logic based domain theory.

In this way, the lexicon—which can be considered as an associated theory—may contain a meaning postulate for the relation of:

$$\forall x, y.\ \mathsf{of}(x, y) \leftrightarrow \mathsf{paintedBy}(x, y) \vee \mathsf{ownedBy}(x, y)$$

which can be rewritten using Description Logics as

$$\mathsf{of} \equiv \mathsf{paintedBy} \sqcup \mathsf{ownedBy}.$$

Moreover, by writing *reification* axioms (see [Franconi and Rabito, 1994]) of the kind

$$\forall x, y.\ \mathsf{paintedBy}(y, x) \leftrightarrow \exists z.\ \mathsf{Paint}(z) \wedge \mathsf{agent}(z, x) \wedge \mathsf{goal}(z, y)$$

then, the logical form (15.1) with the explicit event also entails the ambiguous representation (15.2). In Description Logics, this would be written as

$$\mathsf{paintedBy} \equiv \mathsf{goal}^{-}|_{\mathsf{Paint}} \circ \mathsf{agent}.$$

The ambiguity of *A fresco of Giotto* can be *monotonically* refined later on in the dialog by uttering, e.g., either *Giotto painted the fresco in Siena* or *Giotto sold his fresco*. The refinement process is monotonic, since it is not necessary to revise the knowledge asserted by means of the logical form (15.2).

Lexical ambiguities of nouns can also be represented, as in the example *The pilot was out*—where pilot can be a small flame used to start a furnace, or a person who flies airplanes. The sentence *He was on the toilet* monotonically refines the previous one, because the pronoun *he* may refer just to a person, thus leaving out the reading with flame. Of course, in order to make possible such a reasoning by cases, axioms at the predicate level having negation and, more generally, partitioning capabilities have to be added to the theory—specifying and reducing the possible models:

$$\begin{aligned}
\mathsf{Pilot} &\equiv \mathsf{Flame} \sqcup \mathsf{Aviator} \\
\mathsf{Flame} &\sqsubseteq \mathsf{Process} \\
\mathsf{Aviator} &\sqsubseteq \mathsf{Human} \\
\mathsf{Human} &\sqsubseteq \mathsf{Animate} \\
\mathsf{Animate} \sqcap \mathsf{Process} &\sqsubseteq \bot.
\end{aligned}$$

Verb ambiguity is also captured in the same manner. For example, it is possible to rule out the sentence *The door opens the door*, given the two senses of *open* as "cause to open"—transitive, with an animate agent—and "become open"—intransitive. According to these two senses, both the constituents "*The door opens*" and "*opens the door*" are consistent, but the whole sentence is inconsistent.

Talking briefly about structural ambiguities, a general theory of common-sense knowledge will allow only for one interpretation of *Giotto paints the fresco with a brush* where the PP attaches to the painting event—"*paints with a brush*"—ruling out the interpretation "*the fresco representing a brush*". An early detection of the semantic inconsistency solving the PP-attachment problem is very important in practical applications, since the non-deterministic choice among the different interpretations is usually left to the parser. Thus, the parser does not need to compute a combinatorial number of derivations. Clearly, any metaphoric aspect of language is excluded in these approaches.

Following a semantic-oriented approach as sketched in this Section, Quantz [1993; 1995] proposes a preferential Description Logic based approach to disambiguation in Natural Language Processing. He gives a particular emphasis to the problem of anaphora resolution, showing that an adequate disambiguation strategy has to be based on factors which take globally into account heterogeneous information (e.g., from syntax, semantics, domain knowledge) and yield *preferences* with varying degree of relevance. For this purpose, Quantz introduced and developed a sound and complete proof theory for a *preferential Description Logic*, including a non-monotonic extension with weighted defaults. In his approach, a Description Logics theory comprises syntactic, semantics, domain, and pragmatic knowledge, which globally contributes to the preferential disambiguation process, following the proposal by [Hobbs *et al.*, 1993].

Franconi [1996] proposes a formalism based on an expressive Description Logics complemented with the ability to express logical forms as *conjunctive queries* [Calvanese *et al.*, 1998a], i.e., formulas in the conjunctive existential fragment of FOL. The formalism allows for both under-specified semantic representations and encapsulation of contextual and domain knowledge in the form of meaning postulates. In particular, lexical ambiguities, structural ambiguities, and quantification scoping ambiguities [Franconi, 1993] are considered, and an account to the structure of events and processes in terms of tense and aspect is given [Franconi *et al.*, 1993; 1994]. It is shown how to apply this logic for lexical discrimination based on semantic knowledge.

Ludwig *et al.* [2000] present a modified version of Discourse Representation Theory (DRT) and show that its Discourse Representation Structures (DRS) may be expressed as assertional statements in a Description Logic. This allows for lexical discrimination during the parsing process based on the domain model. In order to capture situations where the available information is incomplete to characterise the meaning of an utterance, a partial logic (called *first order ionic logic*) is introduced to represent and reason with the logical form. The approach combines in an elegant way linguistic and contextual semantics—both represented in the Description Logic domain model.

15.4 Knowledge-based natural language generation

In the previous Sections an architecture for semantic interpretation was introduced, where Description Logics were used to build a knowledge base with lexical and conceptual information. The knowledge base encodes the necessary data for building the logical form from the analysis of some natural language text. In this Section we mention another task which makes use of the same body of knowledge expressed in a Description Logics based ontology, but with the dual goal of generating a coherent (multi-sentential) natural language text, starting from an abstract non-linguistic specification of its meaning. Examples are in the context of dialogues (see, e.g., [Stock *et al.*, 1991; 1993]), of natural language instructions (see, e.g., [Moore and Paris, 1993; Di Eugenio, 1994; 1998; Paris and Vander Linden, 1996a; 1996b]), of language translation (see, e.g., [Dorr, 1992; Dorr and Voss, 1993; 1995; Knight *et al.*, 1995; Quantz and Schmitz, 1994; Wahlster, 2000]), or of multimedia presentations (see, e.g.,[Wahlster *et al.*, 1993; André and Rist, 1995; André *et al.*, 1996]).

The lexical and conceptual knowledge base classifier is the main driving component for the algorithms used to solve the problem of *lexical choice*, i.e., the task of choosing an appropriate target language term in generating text from an underlying logical form [Dorr *et al.*, 1994; Stede, 1999]. The lexicalisation problem is a non-trivial one, since it is possible to have alternative lexical choices covering various (overlapping) parts of the content representation—a translation *divergence*—or it may be necessary to change the information content to convey in order to find a viable lexical choice—a translation *mismatch*. The problem is usually solved by using ad-hoc algorithms which make use of the classifier for determining which lexical units can potentially be used to express parts of the logical form representing the content.

The choice and the realisation of the most appropriate verbalisation should be made in the context of the previous utterances (in the case of a dialogue), of the surrounding environment (in the case of multimedia presentation), and of the overall goal of the ongoing communicative act. For these tasks, it is not enough to have an underlying representation of the content of the text to be generated, but a *pragmatical* aspect has to be considered as well. The pragmatic knowledge about the *rhetorical* interrelationships which occur among the various parts of the broader communication linguistic and extra-linguistic context is needed to generate a coherent presentation in agreement with its communicative goals. In other words, on the one hand there is the content to be presented, on the other hand there is the style of its presentation which should use the most appropriate linguistic expressions to convey the message.

In order to generate a text satisfying the communicative goals and the coherence requirements, a planning algorithm is used to generate an overall structured text (or discourse) strategy, giving the general shape of the text. Using the lexical and conceptual information in the knowledge base, the planner converts the text plans into a specialised non ambiguous representation of the semantic and syntactic information—by taking into account the grammar of the target language—necessary to select the appropriate target language terms [Moore and Paris, 1993; Paris and Vander Linden, 1996b].