UNIVERSIDADE DE SÃO PAULO

On the role of text generation in knowledge based systems interfaces

CLARISSE SIECKENIUS DE SOUZA

MARIA DAS GRAÇAS VOLPE NUNES

Nº 90

NOTAS

Instituto de Ciências Matemáticas de São Carlos

ISSN 0103-2577

On the role of text generation in knowledge based systems interfaces

CLARISSE SIECKENIUS DE SOUZA

MARIA DAS GRAÇAS VOLPE NUNES

Nº 90

N O T A S D O I C M S C

São Carlos

Set./1991

On the Role of Text Generation in Knowledge Based Systems Interfaces

Clarisse Sieckenius de Souza*and Maria das Graças Volpe Nunes†

July 1991

Abstract

This paper discusses aspects of the design of Knowledge Based Systems interface languages, suggesting that, in view of the abstract nature of objects they manipulate, Natural Language is a highly desirable means for human-computer communication. It presents a method for deriving Natural Language text plans from Natural Deduction-based structures. This method is shown to contribute for the specification of the generation component of the interface system, whose interpretation counterpart should be driven by the same grammatical knowledge.

KEYWORDS: Human-Computer Interaction, Natural Language Interfaces, Text Generation.

1 Introduction

The role of interfaces in computer systems is essentially that of providing the adequate medium in wich users and systems can achieve interactive problem solving. Interfaces must include facilities for: clear presentation of systems functionality, user's formulation of problems to be solved or solutions to be implemented, user's understanding of system's responses, and effective error handling and communicative failure treatment. It all amounts to two fundamental issues: (a) representing various types of knowledge and (b) communicating agents intentions, beliefs and actions.

One of the most distinguishing features of Artificial Intelligence (AI) systems is knowledge representation and processing. However, a more precise characterization reveals that even conventional programming involves knowledge representation (KR), in so far as a *programmer has in mind a systematic correspondence by which the contents of certain*

*PUC-Rio, SCT. e-mail: clarisse@inf.puc-rio.br. This author's research is being supported by a grant from the National Council for Technological and Scientific Development - CNPq.

†Universidade de São Paulo/São Carlos. e-mail: mdgvnune@brusp.bitnet

storage cells represent objects and relations within a subject domain [WINO86, p.84]. In fact, Winograd and Flores claim that computers are linguistic devices which allow for cascaded levels of representation. AI differs from conventional computing in that it separates types of knowledge and takes full advantage of the independence among KR levels.

The prime component of an interface system is the language it embodies for computer-human interaction. The semantic model of this language is ultimately the application model itself. In conventional programming, problem solving methods and domain structure are represented at the same level, forcing interface languages to introduce an arbitrary cut between them in order to ensure adequate processing of communicative intentions. These are typically speech acts [AUST62] that refer primarily to problem solving capacities (eg. commands in programming languages) and secondarily to the domain structure (eg. variables in programming languages). In knowledge based or artificial intelligence systems, the separation between problem solving and domain structure allows for a principled mapping between the communicative and representational functions of the interface languages and the KR formalisms of the backend system.

Interfaces for AI applications can be viewed as one more layer of knowledge on top of domain-specific knowldege. In fact, they incorporate a very special kind of knowledge – the management of communication between agents. Usually there are two different perspectives one can assume in designing interface systems. Either both system and user are viewed as equally competent in communicative terms, or the system is conceived as less competent than its human user. No matter which of the two perspectives is chosen by designers, interface systems are essentially knowledge based systems (KBS). In the first case, interfaces must incorporate sophisticated models of users beliefs and intentions, application domain structure, system's problem solving methods and conversational capacities. In the second, interfaces must reveal a precise picture of the system's functions and capacities, so that users can fully visualize the system's limitations and wield all of its available resources in search of adequate solutions.

Because systems models can never be as rich as human world models, we assume that the role of interfaces is that of harmonizing, in the most efficient and effective way, communication between two ontologically different agents. On one side stands a human being with a virtually unlimited background, and on the other side stands a formal system with limited and specialized knowledge about the world. Therefore this paper clearly assumes the second perspective on interface design: systems are communicatively less competent than humans. However, this does not mean that users can only communicate with systems by means of unnatural protocols. We claim that such protocols can be made easier for users if interface languages are adequately expressive and optimally understood by both agents.

The two basic paradigms of KR are structured (eg. semantic networks, frames, scripts) and non-structuring (eg. logic) representation models. Whereas the former have the advantadge of incorporating finer semantic features about the domain, the latter allow for general-purpose reasoning and ensure a greater degree of portability across domains.

Logic based systems, being syntactically driven, provide a formal representation language and inference mechanism as a semantic model for interface languages. Such languages can, therefore, be domain independent, though at the cost of shallower semantic knowledge.

This paper discusses aspects of the design of interface languages, suggesting that, in view of the abstract nature of objects manipulated in KBS, Natural Language (NL) is a highly desirable means for human-computer communication. In order to account for the limitations of full NL processing, we propose some guidelines for selecting the grammatical coverage of the interface and discuss its role in generating and interpreting user and system utterances. The discussion is substantiated by the presentation of a method for deriving text plans from Natural Deduction proof trees [PRAW65] in a logic-based system framework.

2 Issues in Designing Interfaces for KBS

Knowledge Based Systems can be used for a variety of applications. We will concentrate on question-answering systems, for which a possible metaphor underlying interface design is that of a desk attendant. Such systems are used to provide information on topics of their domain. The ability to expose reasoning is crucial in KBS. Even if systems are not expected to provide human-like explanations about their conclusions, they should provide an understandable depiction of the *train of thought* they follow to achieve conclusions.

The alternatives for conveying understandable depictions of reasoning are basically two: either the KBS formal language is used, or the interface can resort to the preferential language in which this activity is performed in reality – Natural Language. It does not seem reasonable to expect that users be competent in the KBS formal language. Thus, NL appears as an attractive choice. However, full NL processing is still a long range goal (if so much) in AI. We propose that a NL-like artificial language can be designed to meet the understandability requirement for users. In other words, we suppose NL-like texts can be generated from KBS knowledge representation.

Since we propose that the basic metaphor for the design of KBS interfaces is that of an information desk attendant, the only event taking place in this situation is conversation. Therefore, objects in the output text should be referrable by the input language. Not only should the input language be inter-referential with the output language [DRAP86], but in this particular case it should actually be the *same* language. Otherwise, commonly found conversation spans such as

– *Some types of whales are virtually extinct.*
– *Why do you say they are 'virtually' extinct?*

could not be processed.

The situation in which user and system are found is then one in which the system is competent in a language whose utterances all belong to a given NL but whose grammar does not cover all of this language's utterances. This NL subset should be used in the input side of that interface. Thus, there is a problem of control. Users must be kept

within the grammatical limits of an artificial language which looks like his/her language, but is not NL. The solution we propose for this problem is to design a menu-driven pattern of linguistic interaction for input, in which menus are generated from the same generative grammar as the output language. In this way, system and user can *speak* the same language in a natural way.

The concern about finding a natural conversational medium is due to the characteristics of the backend system. Direct translation between KR and NL may result in quite unnatural text. In the following, we will provide some details about how to go about problems related to KR/NL translation for a logical KBS. More specifically, we assume a KBS with a Natural Deduction (ND) inference machine [HAEU90] driving the system's reasoning. This implies that input and output of the inference machine must be logical formulae and ND proof trees, respectively. Figure I shows the overall architecture of such KBS.

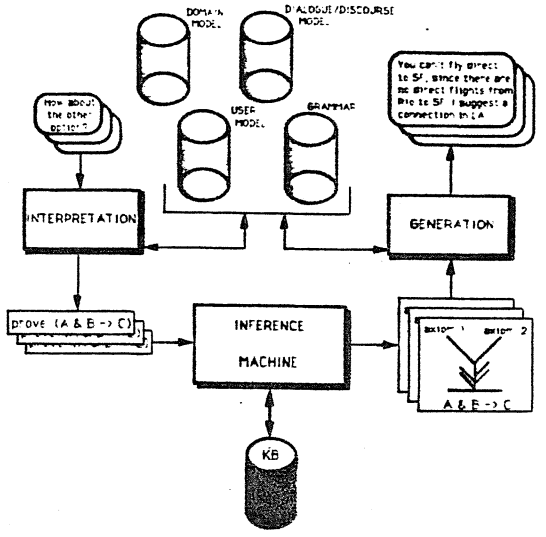

Figure I: KBS Overall Architecture

Notice that, as mentioned in section 1, AI systems allow for independent levels of knowledge representation. The backend system is composed of an inference machine and a knowledge base (facts and axioms), and the interface is composed of two major componentes – interpreter and generator – which share four different knowledge bases: a grammar, a domain model, a dialogue/discourse model and a user model.

Interpretation amounts to transforming interface language utterances into a set of logical formulae. Although one of the instances of this process consists of representing the semantics of sentences in terms of logical expressions, the overall meaning of utterances can only be achieved by the interplay of all knowledge bases.

The grammar contributes to interpretation by providing a set of mapping rules between sentences and logical formulae, lexico-syntactic structures and speech acts, and

text structure and logical processing schemata. The domain model provides the interpreter with domain objects/lexical structures mapping rules. The dialogue/ discourse model provides dialogue and focal structures to allow for the interpretation of conversational phenomena such as anaphora, for instance. Finally, the user model provides the interpreter with a picture of users beliefs and intentions to allow for cooperative behavior.

The generation process uses the same knowledge bases as the interpretation one, and each base provides essentially the same information to the generator as to the interpreter. This is a result of our assumption that both share the same linguistic competence. In methodological terms, the specification of the generator precedes that of the interpreter because the generative grammar is designed to account for all and only the subset of NL needed to expose the results of the backend system's reasoning (ie. truth values and proof trees). This is one of the major advantages we see in the approach we take to designing interfaces to KBS: the selection of the NL subset to support the system's linguistic competence is directly derivable from proof tree schemata/text plan mapping rules, and consequently from mapping rules between logical formulae schemata/sentence structures and atomic formulae/lexical items.

3 Issues on Text Generation for KBS

Generating text and generating sentences are different tasks. While generating sentences is essentially a matter of syntax and semantics, generating text involves, in addition to these, aspects of rhetoric and pragmatics. Texts consist of a coherent and cohesive multisentential structure, a hierarchical organization of elements whose terminals are informational units. The process of generation is carried out in two major steps: planning and realization. The planning step is traditionally defined as that in which decisions about what to say and when to say it are made. The realization step is where decisions about how to say it are made. Roughly, the division of the task lies in separating decisions about content from decisions about form. We have discussed elsewhere [SCOT90] the fact that form can affect cognitive processing, and thus the realization component can impact the understandability of messages to a relevant degree.

The generation component of a KBS interface must transform knowledge representation structures into text plans. Specifically, in our context, it must derive such text plans from ND proof trees. There are two major alternatives for representing text plans: the use of schemata (static pre-defined templates) [McKE85] or the generation based on a rhetorical theory (dynamic generation of text spans) [MOOR91]. Text planning with schemata involves (1) selecting an appropriate text schema and (2) filling it out with the appropriate information from the domain knowledge base. The selected text schema is fixed and therefore context-insensitive. Rhetorical Structure Theory (RST), alternatively, is a theory of text organization in which spans of texts are described in terms of rhetorical relations that hold between them [MANN87]. Elements of a relation are referred to as nuclei and satellites, with nuclei being semantically more primary to text than satellites.

Relations hold between elements of a text at all levels (ie. an RST structure is hierarchical). Unlike schemata, RST has the potential for being a *text grammar* [SOUZ89a].

Our choice of a ND inference system is not arbitrary. Previous research [HAEU88] has shown that the understandability of deductive systems proofs depends on such factors as: the potential of inference rules informing more than just the fact that conclusions logically follow from premises; the explicitness of the logical relationships among parts of proofs; the directness of the proof method. Unlike resolution [ROBI65], one of the most efficient inference mechanisms used in AI systems, ND renders more understandable proofs, since it has a variety of inference rules (one for each logical connective), thus providing a richer environment for the generation of explanatory texts.

The output of a ND-based automatic prover is a recursively generated tree, whose branches are qualified by possibly different inference rules. The limited number of rule types, together with explicit principles to combine them, allow for the identification of proof patterns. Therefore, the input to the text generator of the interface system presents a low degree of variability, thus favoring the use of the schemata approach to text planning. For each pattern of proof corresponds a pattern of text.

Some schemata-based text planners [McKE85, PARI87] generate texts without having access to the reasons why relations and structures they embody are chosen. This is due to the fact that schemata are pre-conceived by the system designer, and represented as a closed knowledge unit. In this approach, schemata represent ideal texts. Therefore, there is no concern to operate structural transformations to enhance rhetorical style. However, if the approach is not a one-shot achievement of the ideal text, rhetorical style can be fine-tuned for cognitive purposes [SOUZ89b]. In this case, the system must have access to an explicit representation of the elements which compose the text structure embodied by the schema.

We believe text generation should always be flexible and provide means of presenting a number of paraphrases for the same message content. This guarantees the possibility of adequating form to cognitive needs of the various users [SCOT90]. RST specifies rhetorical relations among elements of a text structure in terms of beliefs and intentions of readers and writers. This is easily applicable to human-computer interaction taking place at KBS interfaces: both agents, as they alternate their roles as readers and writers of interface language utterances, are actually exchanging information in terms of their own beliefs and intentions about each other.

4 Deriving Text Plans from Natural Deduction Proof Trees

What we will present in the following is our first approach to generating NL-like text from ND proof trees. At this stage of our research, our focus is not on generating alternative paraphrases for the same message structure, but rather on showing how to proceed in a

systematic way to achieve an acceptable text plan. Further stages should investigate ways of producing different output according to various types of constraints found in the user model, for example.

We build text plans in two independent steps: content selection (what to say) and rhetorical structuring (when to say it).

The content selection step consists of two operations performed on proof patterns: factorization and derivation of rules. Both operations prune the proof tree. Factorization applies to a sequence of inference rules involving the introduction or elimination of identical logical connectives.

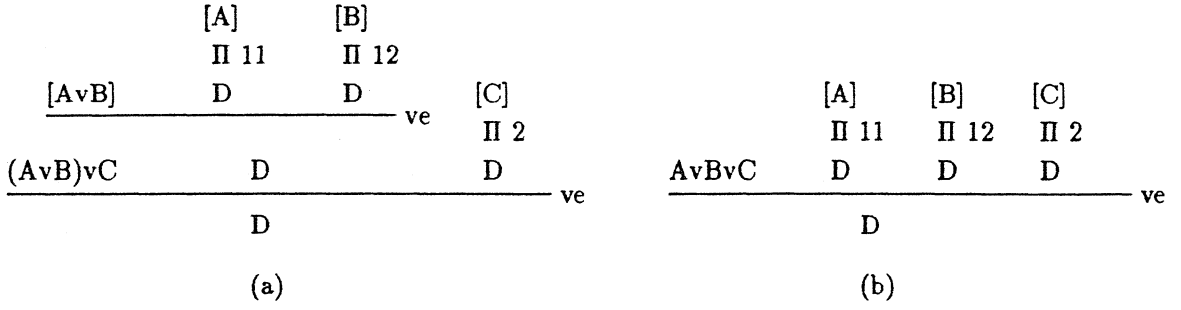

Figure II: Factoring or-elimination rule

Consider, for example, pattern (a) in Figure II. The need to repeat the application of an or-elimination rule is due to the fact that this operator is binary, which forces the rule to apply to every disjunction pair, even though all components of the disjunction play the same role in deriving the final conclusion. The idea behind pattern (a) is that if A is true, then D is true; if B is true, then D is true; if C is true, then D is true. In other words, no matter which of A, B or C is true, D is true since it is given that AvBvC is true. Factorized pattern (b) is just a linearization of pattern (a), preserving the same underlying notion. The motivation for the factorization is only to avoid that derived text plans reflect in NL expositions of the system's reasoning a structure which is due to a syntactic idiossincrasy of a logical language. So, for each logical connective, a factorization is proposed to reduce the impact of the language representation syntax on the final text.

Derivation rules affect content selection by detecting logic argumentation patterns usually found in common sense reasoning. Such patterns, however, do not belong to the set of inference rules of a ND system, but are naturally accepted without deeper details. An instance of such derivation rules is Modus Tollens. In Figure III (a), we see the cannonical ND derivation of $\sim$A from A$\rightarrow$B and $\sim$B. Rule (b) is derived from pattern (a) in a systematic way. In fact, we have formally defined abstract proof patterns which

cover the whole set of derived rules [NUNE91], including tautologies, De Morgan laws and syllogisms.

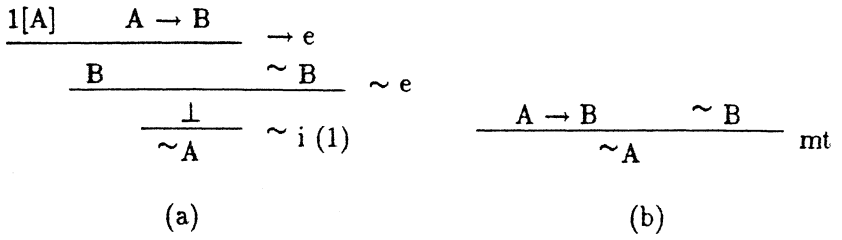

Figure III: Devering Modus Tollens

The rhetorical structuring step is carried out by means of mapping rules from ND subtrees to RST schemata. More specifically, a subtree corresponding to an inference rule application is mapped onto RST subtrees. Figure IV shows an example of one such mapping rule. Notice that, whereas ND subtrees have premises as leaves and a conclusion as the root, RST subtrees have information (ie. premises an conclusion) as leaves, and rhetorical relations as nodes. The mapping rules are guided by the type of the inference rule applied.

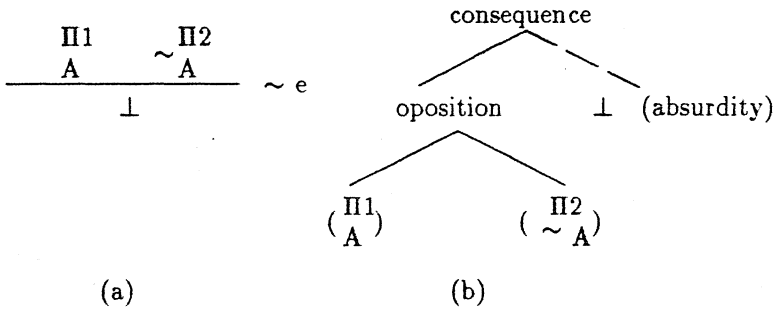

Figure IV: An Example of a Mapping Rule

We have made some changes in the original RST set of rhetorical relations which are used to describe text structures. *Consequence* and *Opposition* are relations we have re-defined for sake of exposing ND-based reasoning. RST presents some kernel concepts we should emphasize here. First, it proposes that rhetorical relations bind two hierarchically different units: nuclei (N) and satellites (S). Nuclei carry the most important portion of information to be conveyed in the text span, whereas satellites carry relatively secondary

information. The attribution of importance to informational content is made by the writer (W). Second, rhetorical relations are specified in terms of writers intentions and readers (R) expected reactions. In the definitions below, we can see such intentions and expected reactions specified in the *Constraints on N + S Combination* and *Effect* fields. Finally, text structuring is achieved by a construction of rhetorical relations among spans of text which, themselves, may contain rhetorical relations: in other words, RST structures are (rhetorical) trees. The overall tree ensures text cohesion in discourse.

RELATION: CONSEQUENCE
Constraints on the N: None
Constraints on the S: None
Constraints on the N + S Combination: N presents a situation that caused the situation presented in S; the presentation of N is more central than S for the W's purposes.
Effect: R recognizes that the situation presented in N has caused the situation presented in S.
Syntactic Markers[1]:
since ⟨N⟩, ⟨S⟩
⟨N⟩ then ⟨S⟩
⟨S⟩ because ⟨N⟩

RELATION: OPPOSITION
Constraints on the N: Multinuclei; two nuclei - N1 and N2.
Constraints on N1 + N2 Combination: The situations presented in N1 and N2 are contradictory, incompatible.
Effect: R recognizes the contradiction or incompatibility between the nuclei.
Syntactic Markers: ⟨Ni⟩, but ⟨Nj⟩

In our approach, nuclei are always the information corresponding to premises of inference rules, and satellites, the conclusions. For the purpose of exposing the system's reasoning, the information about facts and axioms that have caused the conclusion is more central than the conslusion itself. It is worth saying that intermediary conclusions in the proof tree have, in each step, the role of premises, due to the recursiveness of the structure.

Mapping inference rules onto RST structures results in a text plan which, if realized, is still far from acceptable. This is due to the method of building text plans in a recursive fashion: in traversing the proof tree, inference rules are locally translated into rhetorical relation schemata. Also, if realization took place in an interleaved way [HOVY88], the resulting text plan could be sufficient for acceptable text rendition. However, since a

[1]A study about rhetorical markers has been carried out for Portuguese. Markers which appear in the examples are just representative of a class; the complete set of possible markers for each rhetorical relation is not being considered in the present paper.

sequential model is assumed, enhancements to this initial version of the text plan have to be made [SOUZ89b].

So, since text planning is the result of recursive context-free rule application, some important structures that improve the understandability of the final textual exposition of the system's reasoning have to be dealt with in a special mode. In particular, this is the case of hypothesis assumptions in ND-based systems, whose appropriate structuring in rhetorical terms is context-sensitive. Once hypotheses are raised, ND proof proceeds according to the same set of rules as in any other case. Later on, when desired (partial) conclusions are achieved, those hypotheses are discarded. Of course, discarding only makes sense if raising is in topic. So, the text plan has to explicitly introduce the status of discardable premises as hypotheses, so that when discarding is mapped onto a rhetorical schema, the reader can refer it to the hypothesized information in topic. Figure V shows the effect of the proposed mapping.

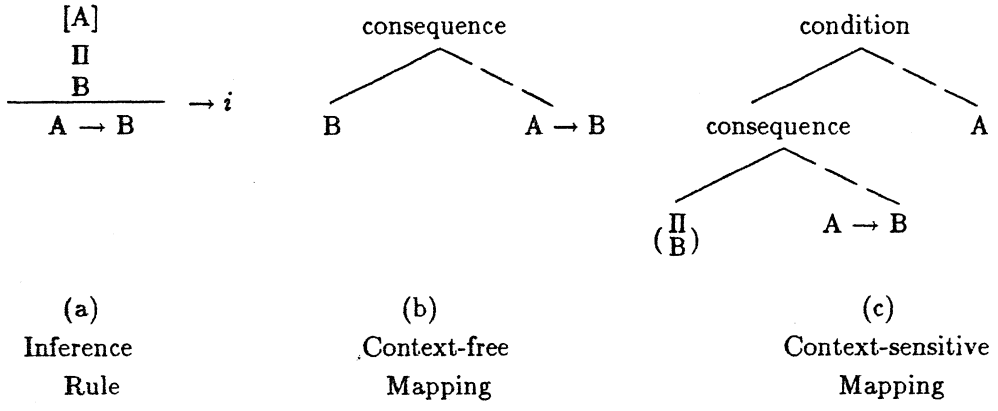

Figure V: Hypothesis Assumptions

In fact, cases as the above-mentioned are related to discourse structuring not at the level of informational content (ie. proof-bound) proper, but rather at the level of the writer's perlocutionary goals, such as clarity, understandability and the like. Clearly, proof derived mapping rules account for informational content, but cannot fully ensure the achievement of perlocutionary goals in writing. This is why, in situations like hypothesis assumptions and ordering, for example, special mechanisms are introduced. Another similar case is that of repeating (ie. re-introducing as topic) information corresponding to premises logically related to a conclusive information, but linearly far from the point where that information is presented. The necessity of this kind of *remembering* may be derived heuristicly from the height of the corresponding proof tree.

A relevant point that has to be considered in planning long and deductive texts is guiding the reader along the reasoning path. An interesting result of previous research [NUNE91] is that ND normal proof patterns offer the possibility of identifying a special type of information (*minimal formulae*) which is uniquely related to both premises and

conclusions. This information is then used in text planning to anticipate reasoning steps. The rhetorical schema which applies to this information is of a different kind than all other RST schemata. In particular, it operates at some sort of meta-level if compared to them. See the example in Figure VI, where β has the role of guiding the deduction of α.

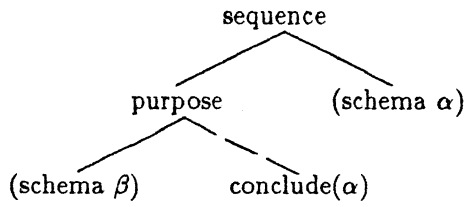

Figure VI: Antecipation of Reasoning Path

The result of the planning approach exposed here can be seen in the example below. First, the ND proof presents an instance of what is called *proof by case*. Then, we show, in Figure VII, the equivalent RST structure corresponding to the text plan.

$$\dfrac{\dfrac{\sim X \quad \sim X \rightarrow (A \vee B \vee C)}{(A \vee B \vee C)}\,ve \qquad \dfrac{\dfrac{[A]\ A \rightarrow D \qquad D}{D \rightarrow T}\,{\rightarrow}e}{T}\,{\rightarrow}e \qquad \dfrac{\dfrac{[B]\ B \rightarrow E \qquad E}{E \rightarrow T}\,{\rightarrow}e}{T}\,{\rightarrow}e \qquad \dfrac{\dfrac{[C]\ C \rightarrow F \qquad F}{F \rightarrow T}\,{\rightarrow}e}{T}\,{\rightarrow}e}{T}\,ve$$

In the following, we propose a possible realization of the plan in English, where the formulae indexed by capital letters are instantiated as NL sentences. Once again, it is worth noting that at the present moment we are investigating realization rules and processes for Brazilian Portuguese. Therefore, all examples are tentative in English.

X: There are environment-protection policies available.
A: Ozone is being depleted from the atmosphere.
B: Land is undergoing a desertification process.
C: Waste is accumulating on earth.
D: Human immune system is depressed.
E: The area of productive land on earth is getting smaller.
F: Toxic substances are accumulating on earth.
T: Human life is threatened.

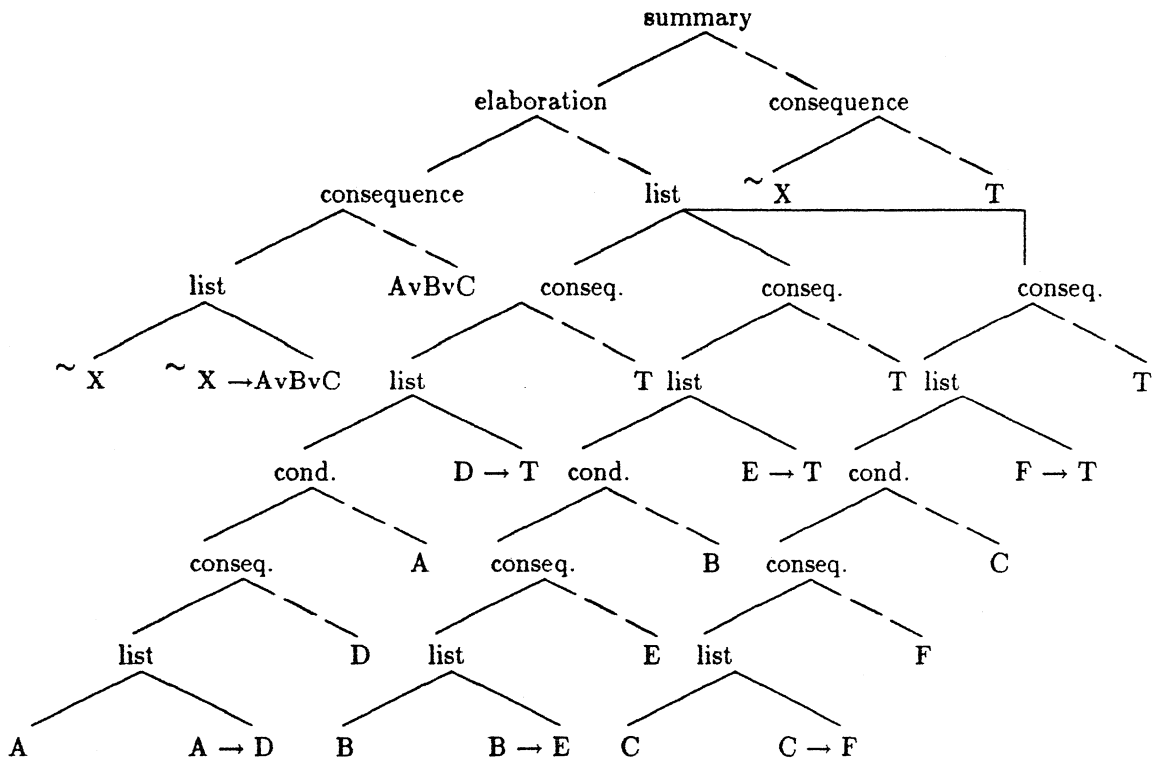

Figure VII: Equivalent RST Structure

There are not environment protection policies available, and this implies that either ozone is being depleted from the atmosphere, land is undergoing a desertification process, or waste is accumulating on earth. Suppose that there is ozone depletion in the atmosphere; if so, human immune system is depressed. Then, human life is threatened. Now, suppose that land undergoes desertification; if so, the area of productive land on earth is getting smaller. Therefore, human life is also threatened. Finally, suppose that waste accumulates on earth; if so, toxic substances accumulate too. Thus, again, human life is threatened. In other words, if there are no environment protection policies available, human life is threatened.

5 Discussion and Future Work

The generated text in section 4 shows how much knowledge a realization component must have to produce acceptable output. For example, anaphoric and elliptical structures are of prime importance for text cohesion. Expressions like *this*, *also*, *so*, bind the pieces of information together. Also, the ommission of whole informational units, which was done with the conclusion of the leftmost subtree in the RST structure (Fig. VII), prunes

the final text so that it does not repeat structures implied in current text spans. An interesting aspect of this process is that, like in derivation rules that transform ND proof patterns into inference paths closer to common-sense reasoning, again, in text planning, some RST subtrees are ommited during the realization process to account for common-sense implicatures existing in the current realized span.

A crucial issue detected in experiments made so far is that of selecting predicates, arguments and the overall KR strategy in translating real-world information into logical formulae. If knowledge bases are built independently of a text generation component to expose reasoning, the production of good quality text may be made difficult or ultimately impossible. This is not due to a lack of adequate grammatical structures, but to an arbitrary mapping of real-world entities into KR language components. In the specific case of logical languages, they can be translated into NL sentences in a syntactically-driven systematic way. Semantics in such languages is specified in terms of elements deprived of real-world meaning. However, the attribution of *labels*, or *names*, to predicates and arguments, for example, plays the role of relating structural items (*lexical items*) to objects of a domain. And these figure as words in text – ie. they are assigned meanings derived from NL. Thus, if the relation between predicate and argument labels and NL words is not made in a principled and generation-oriented way, the output of the interface system can be severely impacted. The tradeoff between the flexibility of logic languages – which are desirable for knowledge engineers, for example – and the cognitive constraints that apply to the interface language – which are of prime importance for end users – seems to be one of the greatest challenges in KBS interface design.

The underlying assumption in our approach to text generation is that the style of the final text has to be one that optimizes the reader's cognitive processing [SCOT90]. This lays a heavy burden on the generative grammar of the system, which has to be psycholinguisticly motivated. Grammatical structures should be selected according to their power to convey in the most explicit way semantic contents derived by the backend system. Such selection does not allow one to guarantee that all of grammatical structures of a given language are selected. In this sense, the generative grammar of the interface system is not likely to be that of the NL it originates from. However, the grammar should embody rules that ensure the rendition of *all and only* the semantic contents derivable by the backend system in an optimal way. In previous work [SOUZ89b] we suggest some grammatical structures – embedding and coordination – which can play that role.

The use of RST provides accessibility to the reasons why text is organized as it is. At the present moment, we are investigating general grammatical structures to realize the text plan and are not devoting much effort to customization of output to users needs. However, the KR to be adopted in our systems (hierarchical structure and belief/intention oriented specifications of RST relations) allows for further stylistic elaboration of text. In this way, users can be provided with an exposition of reasoning more adapted to their personal knowledge.

The approach described in section 4 should clarify the claims we have presented in

section 2, about the input and output languages of the interfaces being the *same*. In fact, if a menu-oriented input component is assumed, a generative grammar is needed to gear the parsing and interpretation of the alternative constructions of users utterances. Our generative grammar should be conceptually reversible, in that the specification of generation mechanisms amount to the specification of interpretation mechanisms. Moreover, it should be clear that the role of a menu-driven approach is that of constraining users to the system's subset of their natural language. The limits of such subset are not expected to be easily derivable from the texts presented as output.

Our research project includes theoretic investigations in both linguistics and computer science, specification of an interface system, the implementation of such interface for an existing KBS, and empirical testing of both software and theoretic assumptions. At the present moment, theoretic investigations are being carried out in the fields of lexical semantics and cognitively-motivated syntactic structures in Brazilian Portuguese. A specification of a text planner following suggestions presented in [NUNE91] is also in progress. Future work in the short term should investigate aspects of the specification of the realizer in the generation component, and aspects of the input interface design.

Aknowledgements

We would like to thank Donia R. Scott for her fundamental contribution to the development of our research in text generation.

References

AUST62 Austin, J.L. *How to do Things with Words.* New York. Oxford University Press. 1962.

DRAP86 Draper, S.W. Display Manager as the Basis for User–Machine Communication. In Norman, D.A. and Draper, S.W. (eds) *User Centered System Design.* Hillsdale, NJ. Lawrence Erlbaum Associates, Publishers. 1986.

HAEU88 Haeusler, E.H. Automatic Theorem Proving: An Attempt to Improve Readability of proofs Generated by Resolution. In *Comtemporary Mathematics.* No. 69, pp. 179-188. 1988.

HAEU90 Haeusler, E.H. *Prova Automática de Teoremas em Dedução Natural: Uma Abordagem Abstrata.* PhD Dissertation. Departamento de Informática. PUC-Rio. Rio de Janeiro. 1990.

HOVY88 Hovy, E.H. Two types of Planning in Language Generation. In *Proccedings of the 26th. Meeting of the Association for Computational Linguistics.* Buffalo, NY. 1988.

MANN87 Mann, W.C. and Thompson, S.A. *Rhetorical Structure Theory: Description and Construction of Text Structures.* Technical Report ISI/RS-86-174. Information Sciences Institute. University os Southern California. 1987.

McKE85 McKeown, K.R. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.* Cambridge. Cambridge University Press. 1985.

MOOR91 Moore, J.D. and Swartout, W.R. A Reactive Approach to Explanation: Taking the User's Feedback into Account. In Paris, C.L.; Swartout, W.R. and Mann, W.C. (eds) *Artificial Intelligence and Computational Linguistics.* Kluwer Academic Publishers. 1991.

NUNE91 Nunes, M.G.V. *A Geração de Respostas Cooperativas em Sistemas Baseados em Lógica.* PhD Dissertation. Departamento de Informática. PUC-Rio. Rio de Janeiro. 1991.

PARI87 Paris, C.L. and McKeown, K.R. Discourse Strategies for Describing Complex Physical Objects. In Kempen, G. (ed) *Natural Language Generation.* Dordrecht. Martinus Nijhoff Publishers. 1987.

PRAW65 Prawitz, D. *Natural Deduction.* Stockholm. 1965.

ROBI65 Robinson, J.A. A machine-oriented logic based on the resolution principle. *Journal of the ACM* 12, 1, 1965.

SCOT90 Scott, D.R. and Souza, C.S. Getting the Message Across in RST-based Text Generation. In Dale, Mellish and Zock (eds) *Current Research in Natural Language Generation.* London. Academic Press. 1990.

SOUZ89a Souza, C.S. and Scott, D.R. Pragmatic Text Generation. Unpublished Manuscript.

SOUZ89b Souza, C.S.; Scott, D.R. and Nunes, M.G.V. Enhancing Text Quality in a Question-Answering System. In Martins, J.P. and Morgado, E.M. (eds) *EPIA-89.* Lecture Notes in Artificial Intelligence. No.390. Heidelberg. Springer-Verlag. 1989.

WINO86 Winograd, T. and Flores, F. *Understanding Computers and Cognition.* Reading, Ma. Addison-Wesley. 1986.

N O T A S D O I C M S C

Nº 89/91 - MORABITO, R.N.; ARENALES, M.N. - On solving large two-dimensional guillotine cutting problems

Nº 88/91 - MICALI, A.; VILLAMAYOR, O.E. - Algèbres de Clifford sur un corps de caractéristique 2

Nº 87/91 - RAPOPORT-CAMPODÔNICO, D.L. - On the construction of Lie-isotopic relativistic sto - chastic mechanics and Lie-isotopic potential theory from the Lie-isotopic geometry associated to a torsion potential

Nº 86/91 - ACHCAR, J.A. - Inferences for the Birnbaun-Saunders fadigue life model using Bayesian methods

Nº 85/91 - ACHCAR, J.A.; LOUZADA NETO, F. - Accelaerated life tests with one stress variable : a Bayesian analysis of the eyring model

Nº 84/90 - RODRIGUES, J. - Bayes predective likelihood function for the accelerated life tests via the orthogonal parameters

Nº 83/90 - RODRIGUES, J. - A Bayesian analysis of the generalized least-square procedure to func tional relationship

Nº 82/90 - LIZANA PEÑA, M. - Exponential dichotomy for singularity perturbed linear functional differential equations with small delays

Nº 81/90 - BERGAMASCO, A.P. - Perturbations of globally hipoelliptic operators

Nº 80/90 - MARAR, W.L. - On the image of $\mathcal{A}$-simple map-germs from $\mathbf{R}^2$ to $\mathbf{R}^3$