Wolfgang Hoeppner, Helmut Horacek,
Johanna Moore (editors):

Principles of Natural Language Generation

Dagstuhl-Seminar-Report; 93
25.07.-29.07.94 (9430)

ISSN 0940-1121

Tel.: +49 - 6871 - 2458
Fax: +49 - 6871 - 5942

Das Internationale Begegnungs- und Forschungszentrum für Informatik (IBFI) ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.

Bezugsadresse: Geschäftsstelle Schloss Dagstuhl
Universität des Saarlandes
Postfach 15 11 50
D-66041 Saarbrücken, Germany
Tel.: +49 -681 - 302 - 4396
Fax: +49 -681 - 302 - 4397
e-mail: office@dag.uni-sb.de
www: http://www.dag.uni-sb.de/dagstuhl.html

Dagstuhl Seminar

Principles of Natural Language Generation

organized by:

Wolfgang Hoeppner (University of Duisburg, Germany)

Helmut Horacek (University of Bielefeld, Germany)

Johanna Moore (University of Pittsburgh, USA)

25.7.-29.7.1994

1 Introduction

This Seminar has been set up with the particular purpose in mind to provide the generation community with an opportunity to focus on problematic aspects of natural language generation that are under-reported in papers designed to be accepted into conferences, and to produce results requiring face-to-face activity of a significant portion of the community. The aim hereby was to strenghten the theoretical foundation of natural language generation by identifying reasons why a systematic approach is so difficult in this area, and by stating principles for this research field. As it has turned out, success in these directions has been partially achieved, the principles identified are yet tentative or formulated on an admittedly abstract level. Nevertheless, we believe to have done an important step in the right direction.

Unlike with most scientific meetings, the form of this Seminar was rather informal, which is quite in accordance with the overall goals. A few introductory talks have been prescheduled to prepare the ground for subsequent discussions. Some additional talks have been given according to actual course of development. The major part of the time available has been devoted to tasks to be addressed by relatively small working groups in parallel although, as it turned out, a lot of time was also spent in general discussions after talks and in connection with working group reports.

Four working groups have been formed altogether. Two of them have addressed issues of lexicalization, including architectural concerns, the other two working on text planning issues. Some comparably simple examples taken from working systems have been prepared for the Seminar to provide adequate material for concrete actions. These examples were taken from the areas of text (report generation) and dialog, augmented by a specialized environment (hypertext). Heavily use was made of these examples during group work, which is also reflected by the reports produced.

In the course of the Seminar, some reasons have turned out to be primarily responsible for the difficulty to address generation in a systematic way (they partially overlap in scope):

- Apart from a subtask commonly viewed as realization, it is almost impossible to structure the whole generation process in a principled way that is meaningful for a variety of different types of applications.

- The initial representations from which a generation process potentially starts differ significantly in type of data (e.g., visual data or tables) and depth of representation.

- The requirements on the functionality of a generator may vary significantly according to the scope and kind of task

- The perspective adopted by the motivation underlying the generation task (be it psychological, engineering or theoretical purposes) may cause crucial differences in

the approach taken (some consider this distinction a source for a separation of the whole field).

Nevertheless, we think that some tentative, abstract principles have been identified, which should be taken into account in the design of a generator:

- Clearly describe the task to accomplish by performing corpus anayses or comparable procedures, in order to identify the functionality of the system to build
- Design a model representation in such a way that the conceptual distinctions it entails are all needed and sufficient to get the envisioned behavior
- Select and adapt methods that exploit the conceptual distinctions made to produce the results according to these distinctions; unfortunately, classical divisions into types of application like dialog system, report generation, etc. are too course grained to provide a suitable basis for selecting methods.

2 Program overview

Monday, 25.7.

Introduction

Knowledge Sources for Natural Language Generation: how can they interact?
Barbara Di Eugenio

Architectures for Generation in Dialogue Scenarios
Norbert Reithinger

What does it take to lexicalise?
Manfred Stede

Towards an Abstract View of Lexicalization
Helmut Horacek

Tuesday, 26.7.

Working Group sessions

Preliminary reports from Working Groups

Wednesday, 27.7.

Working Group sessions continued

Thursday, 28.7.

Reports from Working Groups and discussion

A (Psycho)linguistically motivated Architectures for Generating Language
Michael Zock

Friday, 29.7.

Reports from Working Groups and discussion continued

Dragging Heavy Constituents Around
Koenraad De Smedt

3 Abstracts of talks

The following abstracts of presentations appear in temporal order of presentations.

Knowledge Sources for NL Generation: how can they interact?

Barbara Di Eugenio, Carnegie Mellon University

My main research interest so far has been the interpretation of NL instructions. One problem that seems central to me when interpreting instructions is the integration of different sources of knowledge: what I have worked on so far is the integration of linguistic knowledge and planning knowledge about the domain [Di 93].

NL generation requires a similar integration of several different sources of knowledge, but the problem has not really been addressed so far. In fact, after trying to solve many (all?) aspects of discourse planning with rhetorical schemata, possibly RST based [McK85, Hov91], more recently researchers have started to try to keep distinct different kinds of knowledge, especially intentional and informational [MP93, YM94, DHP+94], and to individuate what kinds of knowledge are really necessary [KKR91].

While the enterprise of distinguishing informational from intentional knowledge and of "cleaning up" the formalism is indeed laudable, even in this more recent body of work the interaction between the two kinds of knowledge is basically neglected, and the role of domain knowledge is in a sense subordinate. This is not to say that researchers in NL generation are not aware of this problem --- see for example [MP91] --- but that the structure of the domain knowledge is often not taken into account.

At the workshop I would like to explore how the interaction of different sources of knowledge can affect the discourse plan. I think that there are domains that are particularly amenable to this investigation, namely those where the domain knowledge can also be expressed in terms of planning operators; this is the case for domains concerned with task execution, such as those used to generate instructions (on representing planning knowledge necessary to generate instructions see for example [DHP+94]).

The three examples I would like to discuss are

1. Go into the kitchen to get me the coffee urn.
2. Go into the kitchen to wash the coffee urn.
3. To clean the appliance, disconnect the power plug, then wipe the surfaces with a wet cloth.

To discuss these examples, and to be concrete, I will loosely adopt the DPOCL plan operator formalism [YPM94,YM94], as the DPOCL planner is very flexible: it is able to generate plans that are DAGs, not just trees, and to deal with partial ordering constraints. Moreover, the representation of discourse operators in [YM94] is similar to the representation I adopt for actions.

References

[DHP+94]
Judy Delin, Anthony Hartley, Cecile Paris, Donia Scott, and Keith Vander Linden. Expressing Procedural Relationships in Multilingual Instructions. In *Seventh International Workshop on Natural Language Generation*, pages 61--70, Kennebunkport, Maine, 1994.

[Di 93]
Barbara Di Eugenio. Understanding Natural Language Instructions: a Computational Approach to Purpose Clauses. *PhD thesis*, University of Pennsylvania, December 1993. *Technical Report MS-CIS-93-91* (Also Institute for Research in Cognitive Science report IRCS-93-52).

[Hov91]
Eduard H. Hovy. Approaches to the planning of coherent text. In Cecile L. Paris, William R. Swartout, and William C. Mann, editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pages 83--102. Kluwer Academic Publishers, Boston, 1991.

[KKR91]
Richard Kittredge, Tanya Korelsky, and Owen Rambow. On the need for domain communication knowledge. *Computational Intelligence*, 7(4), 1991.

[McK85]
Kathleen R. McKeown. Text generation. Using discourse strategies and focus constraints to generate natural language text. *Cambridge University Press*, 1985.

[MP91]
Johanna D. Moore and Cecile L. Paris. Requirements for an expert system explanation facility. *Computational Intelligence*, 7(4):367--370, 1991.

[MP93]
Johanna D. Moore and Cecile L. Paris. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651--695, 1993.

[YM94]
R. Michael Young and Johanna D. Moore. DPOCL: A Principled Approach to Discourse Planning. In *Seventh International Workshop on Natural Language Generation*, pages 13--20, Kennebunkport, Maine, 1994.

[YPM94]
R. Michael Young, Martha E. Pollack, and Johanna D. Moore. Decomposition and causality in partial order planning. In *Second International Conference on Artificial*

Intelligence and Planning Systems, 1994. Also *Technical Report 94-1*, Intelligent Systems Program, University of Pittsburgh.

Architectures for Generation in Dialogue Scenarios

Norbert Reithinger, DFKI GmbH

Given the task to build a generation system, one should think about whether there is natural language interaction with the user. Interaction requires the generation of user adapted utterances which pay respect to the linguistic material the user produced earlier on. The linguistic phenomena can range from sentence ellipsis to whole texts, which of course must be processed by the analysis system. The resulting linguistic context provided by the user and processed according to the analysis" knowledge sources are the context for generation.

When a dialogue system is designed, special emphasis has to be on the bidirectional utilization of the knowledge sources within the system and perhaps even the algorithms, both by the analysis and the generation. There are currently only a few systems where this is realized. But, different knowledge sources, e.g. grammars, semantic formalisms or contextual knowledge, that contain redundant knowledge require translation procedures between generation and analysis, and raise the problems of inconsistencies.

Also important is the point of departure for the generation task. This might be the formalism the analysis delivers its results with. But it can also be the case that generation starts from a more language independent world knowledge source of the background system. If generation starts from the latter representation it has to do non linguistic inferences on contextual knowledge sources like discourse and user model to provide a user adapted message specification. In situations where the interaction is more or less stereotypical, no such reasoning is needed: using templates is then an appropriate way to generate utterances.

The ideas presented in the talk are demonstrated with three systems: XTRA, VERBMOBIL and EVAR, where the first is a fully developed dialogue system with bidirectional use of at least major knowledge sources. The last is a speech understanding system to access the DB time table that uses templates for generation.

What does it take to lexicalise?

Manfred Stede, FAW Ulm & University of Toronto

This talk addresses five "burning issues" in lexicalisation that should be dealt with to enable generation systems to perform more elaborate word choice than it is possible today. 1) What is a lexical unit? In addition to single words, a lexicon should account for phrasal items, which can vary substantially in terms of modifiability and extensibility. So far, no thorough classifications of the semantic and syntactic behavior of idioms and other phrasal items and their relationship to grammar have been compiled, which is a prerequisite for their treatment in generation. 2) Linking lexical items to concepts needs to be done more flexibly than with the straight 1:1 correspondences that most generators still employ today. A more general approach leads to choice among synonyms (one concept corresponds to n words), and to a mapping between lexical items and entire configurations of concepts. Several proposals to this end have been made in the literature, but a computationally tractable general solution for graph matching in this task needs yet to be found. 3) Research in lexical semantics has made quite some progress in recent years, but results have hardly been incorporated into generation systems, partly due to the fact that distinctions between conceptual and semantic knowledge have not been made. 4) What factors influence lexical choice? Several criteria for choosing among similar words have been investigated in isolation, but their interactions are poorly understood, and no attempt on unifying them into a more comprehensive framework has been undertaken. 5) The lexicalisation of non-content words, for example discourse markers or connectives, has received very little attention, yet they play a significant role in conveying meaning beyond sentence-level propositional content.

Towards an Abstract View of Lexicalization

Helmut Horacek, University of Bielefeld

Though the lexicalization issue is certainly recognized by the generation community as playing a crucial role in both theoretical models and working systems, little agreement is observable so far on its precise role in the generation process and on well principled computational models achieving the intended function. This article constitutes an attempt towards defining lexicalization in abstract, general terms, in accordance with the dimensions used for analysis in software and knowledge engineering: function, structure, and behavior. The function lexicalization has to accomplish is derived from the function of the overall generation task, and the way this function can be achieved is contrasted with the typical approach pursued in analysis. Structural and behavioral perspectives on lexicalization are adopted by expressing what lexicalization should be in an ideal sense,

thereby relating the achievements obtained so far, and identifying the most severe problems that inhibit progress. This approach should help researchers in assessing the achievements obtained and in identifying hard-wired assumptions incorporated in their systems, thereby providing a better basis for comparisons and increasing the reusability of components and the potential for developing standards.

Dragging Heavy Constituents Around

Koenraad De Smedt, Leiden University

Not only the content of the information conveyed in each part of an utterance, but also the amount of information in it influences the linguistic form of utterances that people generate. At sentence level, the quantity of linguistic material is often referred to as syntactic "weight". The weight of a constituent, usually measured simply as the number of words, is one of the factors influencing the position of that constituent in the sentence. For example, a "heavy" noun phrase as in (1a) tends to occur at sentence-final position, before the prepositional phrase, whereas a "lighter" one normally occurs before the prepositional phrase (1b).

(1a) Helmut threw into a lake the person who had cheated on him four times in a row.
(1b) Helmut threw that person into a lake.

This and several other, similar phenomena related to syntactic weight have so far hardly been accounted for in any computational model of natural language generation, although it is a factor which clearly has an effect on the acceptibility of utterances. A possible reason is that weight seems to be one of the phenomena that must be attributed to characteristics of the language processor, rather than to specific grammar rules.

A possible treatment of weight from a computational viewpoint consists of a kind of competition model, because it is the relative rather than the absolute weight that seems to cause these effects. There are strong indications that long constituents cause problems for the syntactic processor, especially when they occur in mid-sentence. Shorter constituents would therefore be preferred as early elements in the sentence, whereas longer ones would be deferred to a later moment, or, in a parallel system, simply take more processing time.

However, if we view weight as a surface feature, measured, for example, as the number of words, then the weight of a constituent cannot play a clear role before the whole surface structure of the constituent has been computed. This seems to be at odds with the demands of the incremental generation hypothesis, which supposes that the speaker may begin uttering a constituent before its computation is fully completed. An alternative

approach consists of measuring weight in terms of conceptual accessibility, which reflects the semantic complexity of a constituent as well as its givenness in the context.

A (Psycho)linguistically Motivated Architecture for Generating Language

Michael Zock, LIMSI

The processability of any complex task hinges critically upon the way in which the process is organized, that is, the success of accomplishing the task will depend upon the way in which the whole is decomposed into a series of manageable subtasks. Obviously, this line of reasoning applies to NLG, which can be viewed as a task that is solved by making certain choices at various levels, in a more or less well specified order. If one accepts this view various questions can be asked: (a) *How is the process decomposed ?* (what are the modules); (b) *What are the relevant knowledge sources* (pragmatic, conceptual, linguistic)?; (c) *How are the different modules interrelated* (hierarchically-heterarchically)? *How is the process organized as a whole* (control of information flow)?

There have been a number of proposals from researchers with divers backgrounds: psychology (Flower & Hays 1977; Garrett, 1988, Levelt, 1993), computer scientists and computational linguists (McKeown,1986; Appelt, 1985; Hovy, 1988).[1] What strikes me when looking at these models is not so much their diversity, – obviously psychologists, linguists and engineers have different goals, hence views – as their lack of precision concerning the *modules* (variety, number), the *knowledge* embodied in each one of them (what are their respective inputs and outputs?), and the *control of information flow* (dependencies between the modules - what is processed when?). In sum, most models are grossly undespecified, utterly rigid and hardly ever motivated on empirical or linguistic grounds. As a matter of fact, most architectures are motivated on engineering grounds.[2]

During my talk I've taken a different approach. Rather than being concerned with engineering considerations (maintenance, transparency, etc.), I was interested in psycholinguistic problems: what kind of characteristics should an architecture have to accomodate for the facts observed in natural settings? In order to get a handle at the problem one can look at constraints both in the language and in the mind. The results of such an analysis should allow us to plead for a specific architecture. I've spent most of

---

1 For a more information, see Ehrich (1987: 433-35) and de Smedt al. (1995).

2 For an exception, see Garrett's model (1988). Models proposed by psychologists are generally based on performance factors like *speed* (hesitations), *quantity* and *quality of output* (speech-errors), etc.

my talk looking at the linguistic constraints. Showing that there are interdependencies among nearly all the components or levels (word-sentence-text; word level <-> syntax, syntax <-> morphology; word level <-> text level)[3] I've concluded that, in the absence of connectionist models,[4] black-board architectures are at present probably the best approach. They allow for parallel-distributed processing and for opportunistic planning.[5] Furthermore, they can account for the different sort of interdependencies encountered. Processing need not be algorithmic, that is, processing order is not specified once and for all. it can be computed dynamically. [6]

This is in line with the fact that the order in which the relevant elements (e.g. words, or syntactic structure) become available in our mind may vary not only from individual to individual, but also within a given individual: the same sentence may be produced by taking different routes (Harley, 1982). Which route is taken may depend on such factors as the current *cognitive state*, *information available* in a given context, *salience* of relevant information (Bock & Warren 1985), or of features present in the environment, etc.

---

3 Psychological evidence for *lexical choice* preceding *syntactic structure* or the opposite is given in Aitchinson (1987, chapter 11)

4 Modelling within the connectionist framework is certainly a valuable candidate, but for the time being it is still too early to judge its adequacy. Despite the number of interesting features that these architectures allow to model in principle (interaction, learning), it is still unclear whether these models can scale up. What is clear though, is the fact that they are very hard to build. So, even if they turn out to be of the right sort, it remains to be shown that this approach is feasable, because reasonably expensive. For more information on generation done in the connectionist or interactive-activation framework, see Stemberger (1985), Schade (1992), Ward (1994).

5 For black-board models or distributed processing with opportunistic control in the context of machine translation, see (Nirenberg et al. 1989)

6 One could view the functioning of the mind, hence, the functioning of natural language in similar ways as the functioning of a complex society (oligarchy). The two systems are organized in a similar way : (a) problem solving is *decomposed* : the result is produced not by a superexpert, but by a team of specialists;(b) the different agents (components) contributing to the solution have a certain amount of *autonomy*; (c) the agents *negociate*, that is, they do not only communicate their results and draw on the results produced by their colleagues, but they can also adapt their behavior to allow for accomodation of the results produced by the other components.

The advantage of such a heterarchical kind of organization are multiple: (a) *freedom of processing*: various orders are possible to reach the solution; (b) *time-sharing* : each agent can work on its own without having to wait for an order coming from a supervisor; (c) *flexibility* : information flow is bidirectional; (d) *opportunistic planning* : as information becomes available at different moments and in unpredictable ways, and since the different components can accomodate, it is possible to have the different agents compete and to use the first result produced by any of them.

Architectures must allow to accomodate for these constraints, all the more as we are working under severe time constraints. Psychological constraints in general (memory, attention) impose specific constraints on language and its processing. In order to be processable, languages must be flexible (many-to-many correspondances), so must be the architecture in charge of producing it. As I have tried to show by giving exemples of the different interdependencies, this is clearly the case: there is clearly more than one way to reach the same result. The view I've tried to share is compatible with such fundamental notions as *competition* , [7] resource management, how much time to allocate to a given component at a particular moment, etc.These notions, I believe, are fundamental, though they clearly have not received yet the attention they deserve.

References

Aitchison, J. (1987). *Words in the mind: An introduction to the mental lexicon.* Oxford: Blackwell.

Appelt, D. (1985). *Planning English sentences.* Cambridge: Cambridge University Press.

Bock, J. & Warren (1985). *Conceptual Accessibility and Syntactic Structure in Sentence Formulation.* In Cognition, 21.

De Smedt, K., Horacek, H. & Zock, M. (1995). *Architectures for Natural Language Generation: problems and perspectives.* In Adorni, G. & Zock, M. (Eds.), Trends in Natural Language Generation: an Artificial Intelligence Perspective, Springer Verlag, New York.

Ehrich, V. (1987). *The Generation of Tense.* In: Kempen (Ed.), Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics, Martinus Nijhoff Pbs, Dordrecht.

Flower, L. & Hayes, J.(1977). *Problem Solving Strategies and the Writing Process.* In College English, 39.

Garrett, M. (1988). *Processes of Language Production.* In: F.J. Newmeyer (Ed.), Biological and Psychological Aspects of Language, Harvard University Press.

Harley, T. (1982). *There's more than one way...* In ECAI-82: Proceedings of the 5th European Conference on Artificial Intelligence (pp. 267–268).

---

[7] This kind of competition can take place both inside of a component (e.g. lexical choice: synonymes competing for each other) or between them (e.g. lexicon and syntax). Whoever wins, i.e. produces a result posts constraints, that is, forces the other components to accomodate.

Hovy, E. (1988). *Generating Natural Language under Pragmatic Constraints*. Lawrence Erlbaum Associates.

Levelt, W. (1993). *The architecture of normal spoken language use*. In Blanken G., J. Dittmann, H. Grimm, J. Marshall & C. Wallesch (eds.), Linguistic Disorders and pathologies. W. de Gruyter, Berlin, New York.

McKeown, K. (1986). *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press, London.

Nirenburg, S., V. Lesser & E. Nyberg (1989). *Controlling a Language Generation Planner*. In IJCAI-89.

Schade, U. (1992). *Konnektionismus*. Zur Modellierung der Sprachproduktion.

Stemberger, J. (1985). *An interactive activation model of language production*. In Ellis A.(Ed.), Progress in the psychology of language. Lawrence Erlbaum, Hillsdale, NJ., vol. 1, pp. 143-186

Ward, N. (1994). *A Connectionist Language Generator*. Ablex, N.J.

4 Reports from working groups

Report from Working Group 1: Lexicalization and Architecture

Group Members: John Bateman, Stephan Busemann, Koenraad De Smedt, Helmut Horacek, Kwee Tjoe Liong, Hajo Novak, Leo Wanner

1 Introduction

This report summarises the results of the discussions held in Working Group 1. The group essentially has worked on two major tasks:

- Variations in conceptual representations and consequences for lexicalization
- Improvements on texts produced by some generation systems, and measurements to achieve these improvements in an automated way

We present these issues in turn, and we summarize major insights.

2 Impacts of variations in conceptual representations

We discussed lexicalization matters in connection with two issues:

- Attributes measureable by quantities and units
- Realization of conceptual constellations by adjectives/adverbs

2.1 On expressing attributes measureable by quantities and units

2.1.1 The issue

We discussed the weather forecast as produced by Kittredge"s FoG system. We concentrated on the concept of "wind force", and we reached to an interesting set of (German) sample sentences. The problems relate to lexicalization of measurement units and of noun compounds (see the following examples):

1a) Der Wind hat Stärke 8 erreicht.

1b) Der Wind hat (50 km/h) erreicht.

* 2) Der Wind hat Stärke 50 km/h erreicht.

3a) Der Wind hat eine Stärke von 50 km/h erreicht.

* 3b) Der Wind hat eine Stärke von Stärke 8 erreicht.

4) Die Windstärke ist 8 (50 km/h\).

5) Der Wind hat Sturmstärke erreicht.

These sentences give rise to several observations. One interesting point is the interdependency between the realization of wind and its strength. We hardly say "Der Wind hat Windstärke 8 erreicht." since this is redundant. We may, however not realize wind at all if we use "Windstärke" to verbalize the measurement unit (see (4)). This shows, as well as (5), that not all nominal compounds can be lexicalized. Hence, it is probably more suitable to assume that "Stärke" is the realization of the attribute and "Wind" the realization of the phenomenon. Their relation may be expressed by an appropriate auxiliary or by building a compound noun. A second important point is that "Stärke" is in fact the measurement unit for the Beaufort scale in the sense "km/h" is a measurement unit (compare (1a) and (1b) and (2)). However, if we use "Stärke" in its literal sense, we cannot use the Beaufort units at all (3b).

In order to be able to generate correctly the variety of expressions listed above, a program must have control about:

- The underlying conceptual representation(s)
- Correspondences to lexical items and grammatical functions
- Exploiting context knowledge to select purposefully among available alternatives.

We concentrate our effort on the first two aspects.

2.1.2 Conceptual representation

The following considerations have to be taken into account for conceptualization which give rise to some alternative representations. Although it seems plausible to assume "force" to be an abbreviated form, derived from "speed", "wind force" probably was, historically seen, the intuitive, primitive, undefined concept, which much later was defined scientifically by exact measurement, in this case of speed (in kmph). Support for this view can be found in the fact that there are lexical expressions such as "strong" vs "weak" (and in Dutch "hard"/"stevig" vs "zacht", i.e. "firm" vs "soft") as opposed to the non-existence of "fast" vs "slow" (maybe these exist as jargon terms, e.g. in sailing?). This is comparable to other "natural" observable phenomena, like heat, or temperature (in case of environment as well as of human body), air pressure, air humidity, brightness (especially, of stars: magnitude!), and so on.

Another possibility is to assign an attribute "speed" to the wind, i.e., something measured in distance/time, and the Beaufort scale is treated as a derived measure. "Strength" is then considered the lexicalization for speed applying to wind (a matter of collocation).

Thus, basically two possibilities offer themselves as conceptual representations. "Force" certainly can be claimed to be an attribute of "wind", and that it depends on its definition

how it is to be measured, but that the latter (measurement in kmph) is then only derivative of the "primitive" concept. In the other alternative, derivation is expressed in opposite direction. The first of these alternative representations puts emphasis on the impact of wind on human senses, while the second alternative puts emphasis on the mere physical aspect. Whatever position is taken, there are certain prerequisites which an adequate representation has to accomplish so that the linguistic variations can be handled systematically:

- Access must be provided to both scales.
- Their relation must be expressed explicitly and declaratively.

2.1.3 Defining correspondences to lexical representations

In order to express these concepts lexically, mappings must be defined from (chunks of) conceptual elements to lexical items and grammatical functions in a compositional way. The relevant set of lexical items comprises "Wind", "Stärke", and appropriate numbers and units. Applicable grammatical functions are appositive and a "von"-prepositional attribute; moreover, the same conceptual chunks may be expressed by auxiliaries, function verbs, or noun-noun composition. A principled lexicalization of the process aspect expressed by "erreichen" is beyond the scope of our discussion.

Furthermore, conditions of lexical composition must be applied, so that certain alternatives fail to result in a coherent expression. For instance, "Windstärke" results from "wind" and the attribute "speed", as an alternative to auxiliary constructions. Therefore, the utterance "Der Wind hat die Windstärke ..." cannot result from lexicalization. However, expressing these conditions precisely is far from easy. Consider the sentences "Das Erdbeben hat eine Stärke von 6 erreicht", which is dubious, but "Das Erdbeben hat eine Stärke von 6 auf der Richter-Skala erreicht" is perfectly acceptable.

Under certain circumstances, the attribute, "Stärke", can be excluded from contributing to the lexical items involved in a composed expression. Its place in the lexical representation is then taken by the measurement (e.g., "Der Wind hat 50 km/h erreicht"). Building this kind of expressions can be implemented by techniques involving type shifting. Similar expressions can be found in other descriptions of measureable objects, in case the unit by itself already identifies the attribute.

A delicate case seems to be constituted by the expression "Sturmstärke". While one might argue that this expression originates from a comparison with a hypothetical measure attributed to a "typical storm", a simpler view is to derive a term of this kind from concepts associated with the stages of the beaufort scale (which already exist for this scale).

A major lesson we have learnt from this discussion is that the lexicalization of measureable attributes turns out to be more complex than it seems at first sight. A

principled approach requires correspondences between conceptual and lexical elements that may deviate from simple one-to-one mappings in some cases. The interplay between these correspondences, which depend on the conceptual representation chosen, and lexical composition constraints provides for the flexibility in generating descriptions and should guarantee correctness of the results produced.

2.2 Some issues concerned with the generation of adverbs/adjectives

2.2.1 The issue

The status of adjectives has been a hotly debated topic in linguistic circles - a few typical examples are:

1) a beautiful dancer (beautiful as a person, or dancing beautifully)

2) the alleged murderer (might even not be a murderer after all!)

3) the Brazilian soccer player (this might mean either of "playing for Brazil", "born in Brazil", "playing for a Brazilian club", "having a Brazilian passport", or "having Brazilian parents", and may have even other interpretations)

The problems exemplified here include 1) scope ambiguity, 2) context matters, and imprecision of the relation expressed. This problem is caused by the fact that, in many languages, very different kinds of "modifiers" can be expressed in an "abbreviated" adjectival form. Also, not every adjective that can be used predicationally, can be used attributively (with the same meaning), and vice versa. Maybe all adjectives are just shorthand abbreviations for more verbose descriptions ?

We started our discussion from a conceptual representation of the propositional content. A second input is a representation of the speaker"s perspective of the concepts to be verbalized, be that part of the propositional content or not.

We discussed a couple of examples (based on Marie Meteer"s ones):

1) He makes a quick (an important) decision.

2) He makes a decision about an important issue.

3) He decides on an important issue.

4a) He decides quickly.

* 4b) He decides importantly.

* 5a) It is quick to make a decision.

5b) It is important to make a decision.

Obviously, (4a) and (4b) show that "important" is an attribute of what is being decided on rather than of the decision making process, whereas "quick" is an attribute of the decision making process. Pustejovsky"s qualia structure would demonstrate how "quick" can syntactically go with the argument while nevertheless semantically modifying the decison making process (see (1)). However, (5a) and (5b) show that it is possible do qualify the decision-making process itself as "important".

2.2.2 Conceptual representations

Assuming a KL-ONE style knowledge base, we want to foresee the following set of TBox concepts (notation from Luck et al 1987):

thing	= rootconcept
human	= primconcept (specializes (thing))
process	= primconcept (specializes (thing))
agent	= primrole (domain-range (process, human))
quick-process	= defconcept (specializes process))
important-process	= defconcept (specializes (process))
object	= primconcept (specializes (thing))
important-object	= defconcept (specializes (object))
issue	= primconcept (specializes (object))
fix-altern	= defconcept (specializes (process))
content	= primrole (domain-range (fix-altern, object))
fix-altern-q	= defconcept (specializes (fix-altern), specializes (quick-process))
fix-altern-i	= defconcept (specializes (fix-altern), specializes (important-process))

The above definition cleanly assumes quick processes as a possible specialization of process. Subconcepts of "fix-altern" are then restricted to be quick or important. We want to avoid to attach a role to "fix-altern" that is value-restricted to some concept "quick" since this has under reasonable interpretations, the consequence that *all* alternatives are fixed quickly (a similar argument holds for "important" being a value restriction to a role for object). Alternatively, we might have chosen to represent qualifiers as concepts (e.g., "important-entity") and inherit from these, thus describing e.g., "fix-altern-i" as the intersection between "fix-altern" and "important-entity". That way we would express the commonality between "important decision makeing" and "important issue".

However, in both cases the definition is insufficient in that it does not express the relation between processes and their results; e.g., between "fix-altern" and "decision". Nevertheless, we have taken this representation as underlying material for our discussions, because it is formally clean and simple enough to be understood without much effort. Some problems and limitations will be mentioned in the course of subsequent discussions.

Clearly, the above definitions are already "derived concepts" that might not occur in some application knowledge base. However, KL-ONE requires us to underly a well-defined,

set-theoretic semantics, which can serve as a sort of general interface to any logic-based interpreter. Hence no idiosyncratic properties of NL items should be included. This semantics is interesting with respect to lexical choice as it represents meaning that needs to be linked to grammar-specific NL semantic expressions.

The input for NLG differs with respect to the TBox above. Here are three different ABox specifications (notation from Luck et al 87; uc = unique constant):

uc1 - fix-altern-q (agent = uc2)
uc2 - human (name = "Helmut")

This accounts for the examples "He makes a quick decision." and "He decides quickly."

uc1 - fix-altern-i (agent = uc2)
uc2 - human (name = "Helmut")

This accounts for the example "It is important that he decides.", or "His coming to a decision is important."

uc1 - fix-altern (agent = uc2, content = uc3)
uc2 - human (name = "Helmut")
uc3 - important-object

This accounts for the examples "He decides on an important issue.", "He makes an important decision.", and "He makes a decision about an important issue."

Which of the alternatives is selected depends on the speaker's perspective. If she is interested in the result, she might prefer "He makes a quick decision." over "He decides quickly." If she is more interested in communicating the process, it will come out the other way around. The above representations do not make the relevant distinctions for this.

It remains to be shown how the linguistic entities are actually associated with TBox concepts and how inheritance interacts with the combination of the respective entities. For instance, what linguistic objects are associated with "fix-altern", "fix-altern-i", and "important-process"? We believe that the clarity and systematicity of such an assignment motivates or refutes any model of derived concepts.

2.2.3 Lexical knowledge

Concepts can be associated with lexicalization knowledge. E.g.\ fix-altern can be realized as "make a decision" or "decide", process by "process", and quick-process by "quick". A crucial problem is the mapping between the lexical alternatives available (granted by the underlying ABox representation) and the text plan. By virtue of the TBox inheritance hierarchy we use lexicalization knowledge from superconcepts if the concept in question

has none associated with it. fix-altern-q, for instance, can use "quick" and "make a decision" or "decide".

Clearly, a suitable association of lexemes with concepts must interact with a general strategy for collecting relevant lexemes on the basis of an ABox. In a second step, lexical semantics and text planning interact to combine words on the basis of the speaker"s perspective. Here lexical collocations might further restrict the choice.
In the examples above, "quick" can modify the noun "decision" although it qualifies the process. This is not the case with "important". Thus we cannot produce "He makes an important decision" from the second ABox representation. And there are syntactic restrictions that prohibit the use of "importantly" as a noun modifier, which prohibits "He decides importantly".

In essence, the association of lexical knowledge with concepts drives the formation of correct compositions of lexical expressions, (together with syntactic restrictions, of course). A crucial distinction is made between the associations attributed to the subconcepts of process. Whereas simply "quick" is associated with quick-process, a pattern like "it is important" must be associated with important-process in order to express scope relations correctly. Hence, appropriately expressing these associations excludes the derivation of "important decision" from fix-altern-i. However, it may seem a bit awkward to associate not just lexemes but, eventually, also syntactic patterns with concepts. Nevertheless, this is the price to pay for having a compact representation like fix-altern-i on the conceptual level: the way how conceptual representations are expressed strongly influences the ease how associations with lexical knowledge can be formulated.

In an ideal system, the conceptual representation has to be as explicit as possible, and that it will be decided only at the lexicalisation stage (or later, at the lexical choice stage), whether the concept is expressed by means of an adjective or in another way (or not expressed at all, when it is not relevant in the particular context). In a realistic environment, however, one is always engaged in several compromises. Hence, conceptual representations must be as explicit as needed to provide for the relevant distinctions in a certain task. For instance, if "fast" is directly represented as a property of car, verbalizations cannot refer to "driving", but this may not be neccessary in the particular application (this is another example we have discussed in our working group).

3 Measurements to improve textual presentations

We have considered two example texts from different genres, which we have critized and to which we have applied changes aiming at improving these texts. These texts comprise a report originating from time series data, and a hypertext based presentation.

3.1 A Report generated from numerical data organized in tables

We have worked on a report presenting unemploiment data presenting differences in time, sex, and age, among others. We felt that the text was boring to read (at least for an ordinary newspaper user, which is, by the way, not the intended audience of this report). We applied changes to some portion of this text, mainly to avoid repetitions whereever possible. This led us to two variants which differ according to the basic grouping of facts to convey.

Original text:

> Employment among women aged 25 and over rose by 44000 and the employment / population ratio among women aged 25 and over increased by 0.5 to 52.3. Employment among men aged 25 and over fell by 12000 and the employment / population ratio among men aged 25 and over decreased by 0.3 to 72.5.

Improved texts:
(variant 1)

> Employment among women aged 25 and over rose by 44000 and their employment / population ratio increased by 0.5 to 52.3. In contrast (But, on the other hand), among men of the same age group, employment fell by 12000, the employment / population ratio decreasing by 0.3 to 72.5.

(variant 2)

> Employment among women aged 25 and over rose by 44000, but (in contrast,) among men of the same age group, it fell by 12000. The (respective) employment / population ratios increased by 0.5 to 52.3 and decreased by 0.3 to 72.3 (, respectively).

Several measurements have contributed to improve the original stereotype text (if this is desired for the application):

- Expressing discourse relations explicitly
- Building descriptions (e.g., "age group")
- Aggregation (by age groups, reordering facts)
- Pronominalization, eventually interleaved with aggregation ("it" in the context of "men" instead of "women", "respective"-construction)
- Using an "ing"-form instead of co-ordinating clauses by "and".

In order to produce similarly (improved) reports, a generator is facing (at least) the following difficulties:

- Applying simple, surface-oriented criteria, the pronouns used in the improved texts are ambiguous. Their controlled use requires the following lines of reasoning: if a pronoun may refer to a term which consists of a basic term (e.g., "women") including a restriction (e.g., an age group), the more precise referent is preferred; and if an actual context term (e.g., "men" may replace the restriction, this is the intended interpretation.

- Finding suitable descriptions to avoid repetitions (e.g., "age group").

The discussions in our working subgroup concerning the overall style of the text, which as a consequence resulted in much better versions of the fragments handled, led us to the conclusion that it might be worth while to assume yet another step in NLG of texts, i.e., a third stage, that of "revision", or polishing, or post-editing, after conceptualisation and lexicalisation-realisation. After all, that is the way many papers are written: first a preliminary, draft version, and then a definitive, final version. It is an open question whether it is easier to program a generator that produces (better) text versions by skillfully exploring its search space (using back-tracking, or pursuing several alternatives for some time), or by applying explicit rewrite techniques. Future systems will provide the answer.

3.2 Text generation in the context of hypertext

The text example examined in our working group stems from a system that prvides patients with personalised information about their problems and treatments in form of hypertext documentations. In our view, the example text discussed is bad in two respects:

- Thematic organization
- It carries wrong implications

We aimed at improving the text in these respects while also taking hypertext requirements into account.

Original text:

angina pectoris

Like acute myocardial infarction, (which is in your record) angina pectoris is a kind of ischaemic heart disease. "Angina pectoris" is the term for pain in the chest due to insufficient oxygen being carried to the heart muscle in the blood. According to your record, you are currently suffering from this diabetes-related problem. Its causes include ischaemia. Some common treatments include a coronary vasodilator.

Related text:

problems on record

- acute myocardial infarction Diagnosed: 1972. No longer active.
- angina pectoris Diagnosed: 1975. Still active.
- background diabetic retinopathy Diagnosed: 1988. Still active.
- neuropathy Diagnosed: 1988. Still active.

Improved text:

(1) Angina pectoris is pain in the chest due to insufficient oxygen being carried to the heart in the blood. (2) It is a kind of ischaemic heart disease, that is, a heart disease the causes of which include ischaemia. (3) Angina pectoris is also related to diabetes. (4) (According to your record, you are currently suffering from angina pectoris.) Your record also mentions two other problems which are related to (your) angina pectoris. (5) Formerly, you suffered from acute myocardial infarction, which is also a kind of ischaemic heart disease. You are still suffering from background diabetic retinopathy, which is another diabetes related problem. (6) Some common treatments of angina pectoris include a coronary vasodilator.

In the original text, the order in which the information is presented, is thematically awkward. For instance, generic knowledge about diseases is intermixed with partial references to the patient's problems (e.g., "which is in your record" in the first line), and vice versa (e.g., "its causes include ischaemia"). Even more serious, the original text carries two false implications:

- Acute myocardial infarction is mentioned as belonging to the patient's record without revising the associated default assumption by saying that acute myocardial infarction is no longer active.

- The patient's suffering from angina pectoris is introduced as a diabetes-related problem without mentioning background diabetic retinopathy, which is another diabetes-related problem the patient suffers from.

In the improved text, we included additional information to avoid these false implications. Moreover, we organized the information to convey by regrouping facts in the following way:

- Generic information about the disease (angina pectoris, points 1 to 3). Within this section, a definition of angina pectoris (1) is followed by information about its generalization (2) and further properties (3) - note the similarity to McKeown's text schemata.

- Specific information from the patient's record is presented (4 to 5). The disease in focus (4) is followed by related diseases (5).

- Finally, treatments are mentioned (like in the original text).

In order to produce better texts, the system must incorporate the following pieces of knowledge:

- Ordering the facts to convey appropriately, for instance by applying a suitable text schema,
- Take a (default) user profile into account, stating that the "active" slot of a disease in the patient's problem list is of primary importance
- Watching the generated text for implications concerning diseases mentioned in the context of the patient's record (e.g., you are suffering from x, a kind of y -> there is no z, which is a kind of y the patient is suffering from.

We believe that it should not be too hard to improve the system to meet these requirements. Although addressing the issue of watching for implications can become very difficult if pursued in a well-principled way, a much simpler mechanism seems to be adequate given the restrictions in the concrete environment.

Apart from the general improvements on the text, we have learned that hypertext is a mode of information presentation with interesting constraints for text generation. A high level choice to be made is whether the hypertexts should generally be self-contained, with further pointers to additional information, or should mainly be concise, with further pointers to essential information. We chose for the first option. If hypertext pointers are to be parts of normal sentences, this may put restrictions on the syntactic structure, not only due to the lexical and subcategorization properties of the item, but also its morphological properties (in the case of separable words which should be kept together).

4 Conclusions

These examples show that Even a small augmentation in the repertoire of content words (age group, Beaufort) may significantly increase the variability in lexical expressions. To exploit this potential, it is necessary to have:

- Command of non-trivial structure mapping operations, i.e., aggregation including the usage of "respectively" and "same" constructions, as well as context dependent (functional) use of pronouns ("it" in the example)
- Detailed knowledge representations that exhibit the necessary distinctions, express relations, and make assumptions explicit - i.e., relations between different units, separation of entities and properties ("Wind" and "speed")
- Understand and efficiently apply restrictions on compositions of lexical items, comprising:

- Syntactic constraints
- Stylistic criteria (e.g., avoiding boring repetitions)
- Pragmatic goals (e.g., avoiding redundancies, intolerable ambiguities)

In our view, a systematic approach to lexicalization primarily requires improvements in aggregation techniques (both conceptual and structural ones), dealing, in particular, with interdependencies to lexicalization and meeting pragmatic goals.

Unfortunately, a systematic approach is difficult for several reasons:

- The variety and complexity of phenomena touching the scope of lexicalization
- The limited merit of (partial) solutions, because they usually interface with each other rather badly
- Incomplete initial representations entailing implicit assumptions

The resulting mismatch between representational prerequisites of methods and explicitly expressed information in knowledge representations makes the reuse of methods hard and almost impossible on larger scales.

In the future, the interfacing aspect of methods should be strengthened. Moreover, the need for simplified, but not over-simplified methods arises; problem: what are reasonable assumptions concerning simplifications? The assumption that lexicalization maps concepts onto words in a widely one-to-one fashion penetrates almost all phases in most models and systems. Many architectures are based on that assumption, and this is also why systems work. Abandoning this assumption will cause big changes in both, models and systems.

It may seem that the power of generation techniques is very limited yet, but this is also the case in other (even related) areas. At the workshop on "Implemented Ontologies" at ECAI-94 in Amsterdam, the conclusion essentially was:

- We know why we are building ontologies (there are several purposes)
- We do not how how to do that (in a principled way)

Report from Working Group 2: Lexicalization and Architecture

Group Members: Robert Dale, Wolfgang Finkler, Richard Kittredge, Nils Lenke, Günter Neumann, Conny Peters, Manfred Stede

1 Introduction

This report summarises the results of the discussions held in Working Group 2. The group discussions focussed around three reasonably independent topics, and we have organised the report to reflect this.

In Section 2, we consider the problem of how, given an application that requires to have text be generated, we might go about determining the contents of the conceptual base that should be constructed to facilitate this process.

In Section 3, we consider the notions of canned text and templates, and try to elaborate on the proper role of these notions in a text generation system.

In Section 4, we look at the problem of deciding how a dialog system should determine the size of its contributions to a dialog.

A distinction that surfaced at various points during the workshop is that between what one might think of as different approaches to work in natural language generation. These might be characterised as follows:

NLG-PSY: This is natural language generation as a means to exploring and modelling psycholinguistic theory.

NLG-Lx: This is natural language generation as a means to developing theories of language structure and function.

NLG-App: This is natural language generation as a means to building applications.

Ideas from any one subfield of NLG may provide useful insights for the others, and many researchers would be unhappy about being classified as working in only one of the three subfields (or, indeed, may not even agree that the three subfields are distinct). However, it should be borne in mind that the discussions reported below fall in the first instance within the domain of NLG-App. This is particularly true of Sections 2 and 3, but perhaps less so of the material in Section 4.

2 Determining a Conceptual Base for an Application

2.1 Starting Points

We assume that the task of a generation system is to take some elements of an input representation, and to produce some text that somehow corresponds to those elements of

the input representation. This characterisation is broad enough to cover a very wide range of cases; for example, all of the following count as possible input representations:

- Sense data: this might be an appropriate input representation if we have a vision system and we want to describe what it can see;
- Tabular information: this covers the kinds of data dealt with by the report generation systems described by Dick Kittredge and Tanya Korelsky as a basis for workshop discussions; and
- The knowledge representation of an expert system.

2.2 Intermediate Representations: The Problem and Our Assumptions

In each case the problem is this: there may not be a straightforward mapping from elements and structures in the input representation to words and syntactic constructs in the target natural language. In such situations, we can think about constructing an intermediate representation – which we will refer to here, with some caution, as a conceptual base – that makes the generation task easier.

We come to this problem with certain assumptions in mind. One might take the view --- particularly if one is working in the mould of NLG-Psy or NLG-Lx --- that the conceptual representation to use is the one that people use, or the one that is indicated by intuitions about the nature of language. This is not the view we take here. Without taking a stance on conceptual realism, our position here is that even if such conceptual representations do exist, they are likely to be far more sophisticated and fine-grained than our applications require; and, indeed, that the cost of implementing such rich conceptual models rules them out as inappropriate for our particular task.

Our view, then, is that we should develop theories of the meanings or concepts underlying particular words or phrases such that those theories are already assumed to be domain and application dependent, and that we should not expend the effort required to develop any kind of universal conceptualisations. Thus, for example, we might develop a micro-theory of the concept of "increase" in the domain of employment statistics domain, without expecting to find that the same micro-theory will be useful or usable in another domain.

2.3 A Principled Approach to Developing a Conceptual Base

On the basis of these assumptions and observations, we can then propose a methodology that can be applied to derive an appropriate conceptual base for a given application. The steps of the procedure are as follows.

1. Collect the words and phrases that appear in an appropriate set of sample texts: this is effectively a corpus analysis task, and can make use of whatever tools, such as

concordances, seem appropriate. The aim here is to identify the concepts that need to be represented in order to generate texts in the domain. Of course, intuition can be used to extrapolate from the available data to other plausible concepts that might need to be included: if our weather report corpus indicates that the concepts underlying the words "north", "west" and "east" will be useful, but does not contain any instances of the word "south", it seems reasonable to suppose that this is only the case for accidental reasons.

2. Identify the relevant semantic distinctions in the domain: the point here is that the set of lexical items and phrases we have identified may be larger than the set of concepts required to underly these words and phrases. Thus, for example, if the words "increase" and "rise" seem to mean the same thing, and the variation between their use looks as if it is best explained by local contextual or stylistic factors, then this argues for having only one concept of "increase" that can be realised by means of two distinct lexical items; this abstraction simplifies any reasoning processes we might need to carry out at the conceptual level. Since we are not committed to embodying a particular conceptual model in our system, we can provide a conceptual base on ly for the distinctions we need to have, and no others. The nature of the application will determine what concepts it is useful to have: if we are building a system which produces reports in a number of languages, then it is likely to be useful to have a level of representation which abstracts across the lexical differences in the languages.

3. Having developed a conceptual base in this way, we can then address the problem of providing a mapping from the input representation to this more generator-friendly conceptual representation. This can possibly involve re-designing the distinctions made in the previous step, as the mapping will often be non-trivial, and working on this task can inform the previous one.

Of course, there's nothing new about this methodology: indeed, it's very close to what people do when they build interlingua-based machine translation systems. However, we feel it is important to state the obvious, if only because the obvious is easy to lose sight of: it is all to easy to fall into abstract arguments about sophisticated representations being required in order to allow certain kinds of reasoning, but if one's application never needs to carry out the kind of reasoning in question, then the subtle distinctions in the representation are likely to be of no value to the application in hand.

After approaching several domains via this methodology, it may be appropriate to look for generalisations across domains, with the hope that some scope for reuse can be found; but one shouldn't expect too much here. It is always, of course, possible to build generalisations by abstracting further and further away from the real data at hand; but to do so may not be of any real benefit.

This viewpoint is possibly over-pessimistic: we might find, for example, that lexical semantic information about verbs may be transferable although the connection to the

underlying conceptual base needs to be reworked for each domain. But our general point is that one should not assume from the outset that transfer will necessarily be easy.

Our principle, then, for determining an appropriate conceptual base for an application is this:

Don't introduce any more conceptual distinctions than you have to in order to meet the needs of the underlying application on the one hand, and to get the desired generation behaviour on the other.

This may sound rather trivial, but to the extent that this is true, it is unavoidable: if we really are only interested in building a working application, and are not playing at developing broad-coverage ontologies, then the particular set of conceptual distinctions we require will vary from one domain and application to the next.

3 The Appropriate Use of Canned Text

3.1 The Problem

An issue which was raised during the first few days of the workshop was the question of the advantages and disadvantages of using either canned text or templates in a generation system, in opposition to the use of a full-blown sentence planning mechanism. Our group decided to consider whether it was possible to determine principles that would help a system designer to decide on the correct balance between these different mechanisms.

3.2 Some Definitions

3.2.1 Templates and Canned Text

Here we found a problem: what exactly do we mean by the terms "canned text" and "template"? These terms are often used with the intended meaning being left to the reader's intuitions; pinning down precise definitions is less easy. We offer the following.

Template: A parameterised correspondence between input representations and output text, where the output text varies as the input parameters vary.

Canned Text: A template with no parameters.

These are made clearer by example. Suppose we have an input symbol of the form "greet", which should be realised always as the sequence of words "Good morning". This is an instance of canned text, or what we will call a "canned text item". As an alternative, suppose that our input form is parameterised to take an internal symbol that corresponds to the person to whom the greeting is addressed; then, given the input "greet(x1)" where x1 is the internal symbol corresponding to the person named Noam Chomsky, the output might be "Good morning, Mr Chomsky", with appropriate changes

to the name of the addressee as the parameter varies. This is then an instance of a "template item".

This, again, may sound rather trivial, but note that by these definitions, the items that populate the lexicons of many systems – where correspondences are drawn between semantic types and lexical items – are to be considered canned text items. In those cases where the lexical item is responsible for carrying out some morphological variation on the basis of some input parameter (such as tense or number, for example), then we have template items.

Of course, we don't typically think of such small elements as canned text or templates: there's an intuition that there ought to be more than one word, at least, for something to qualify as canned text or template. But this is misleading, since (a) some "lexical items" may indeed contain more than one word (consider phrasal verbs); and (b) the process by means of which our "greet(x)" template item produces variable output is formally no different from a lexical entry that allows us to produce either of the words "child" or "children".

One way of subcategorising these mechanisms would be to note that structures like the "greet(x)" template item combine the canned phrase with any arbitrary string, whereas the morphological derivation of a word complete with either singular or plural marking is carried out on the basis of a parameter which has a finite number of values.

It should also be noted that, given our definitions above, any entire generation system can be characterised as a template: it takes different inputs and produces different outputs. If we are to make sensible and precise use of the notions of template and canned text, it looks like we need to do a little more work to delimit different kinds of transductions.

3.2.2 Different Kinds of Canned Text

We did not address further the question of how one might carve up this space of possible transduction devices; however, we did find it useful to elaborate a little further on the different kinds of canned text we might want to make use of.

When we think of canned text, we typically think of ASCII strings whose internal structure is not available. However, this view conflates two aspects of cannedness. A piece of text can, indeed, be canned in the sense that we do not have access to its internal structure; so, for example, we might have a dialog system which includes in its repertoire of canned text items a mapping from some internal symbol to the ASCII string "the red switch in the corner of the display". In such a system, if the user asks a question like "Why did you describe the switch as being red?", the system has little hope of answering the question: quite apart from the unavailability of information that would allow the system to reason about its own reasoning here (note the parallel with discussions of the need for keeping track of intentions in discourse planning), the system does not even have a clue about the internal structure of the noun phrase it has just generated.

But this lack of structure is only one aspect of cannedness. The other facet of cannedness is that, because the text is canned, we do not have to carry out the work of building the text each time: we just look up the internal symbol in some table, and retrieve the canned text that corresponds to that symbol. But notice that this "precompilation" aspect of cannedness does not necessitate the absence of structure: we could just as easily return a fully-formed noun phrase structure from our table, perhaps even annotated with intentional information when this is invariant.

Because of this, we find it useful to distinguish two kinds of canned text:

Canned strings: A canned string has no internal structure, and is simply a sequence of ASCII characters (typically, but not necessarily, corresponding to more than one word in the natural language).

Canned structures: A canned structure has internal structure, but has been prebuilt so that this structure does not have to be rebuilt on each occasion of use. The specific contents of this structure can vary: an obvious type of content is syntactic structure, but one can imagine many other kinds of structure that could be build in, including semantic or intentional information, and information such as might be required for generating hypertext links.

In addition to this distinction, we might also attach annotations to a canned text item that determine the appropriate contexts for its use.

All of the above has been expressed in terms of canned text, but of course the same observations apply to templates, so that we can think of string templates and structured templates. A simple example of a structured template would be a phrasal lexical item for an idiom like "kick the bucket", where the template would allow the tense marking on "kick" to be specified by parameter while at the same time providing the sequence of words in the form of a fully analysed verb phrase.

3.3 A Principle for Deciding on the Mix

So much for our hunt for useful definitions of these terms. There remains the question: how does a system designer decide when to use canned text, when to use templates, and when to use full-blown sentence planning techniques? Our principle here is in the same minimalist mould as that which we offered in the case of developing conceptual representations:

> Adopt a mixed approach that yields maximum efficiency and/or elegance in terms of design and execution, depending entirely on the purpose of the system.

The point is, again, the same as in the case of our previous principle: there are no hard and fast rules here, since it all depends on the particular application. The component of a programming language compiler that generates error messages uses canned text, or in some circumstances, templates (in order to be able to include line numbers, for example); it would be quite absurd to generate these messages from first principles using a sentence planner. However, at some point in any application, a threshold is reached where the designer deems it more useful to abstract across some set of circumstances and deal with these by means of a paramterised procedure or rule, rather than on a case by case basis. This threshold is determined by many application-specific characteristics. If what we are interested in is building systems, rather than building a model of language, then principles such as efficiency, economy, and elegance of system design dictate the correct mix.

4 Chunking Text in Dialog

This section of our report differs from the foregoing two in that no "minimalistic" approach was used. Instead, we tried to investigate the space of possible parameters and solutions up to a certain depth. As a result, we can only give more questions than (very tentative) answers.

The topic of the following is: Imagine you have to provide a "large" contribution in a dialogue, e.g.. a longish route description. How should you organize it into chunks of information, e.g. divide it into several smaller contributions? Which structure should be imposed on that, and how could this structure be signalled to the user?

Of course, to a certain degree this can also be applied to small contributions.

4.1 How can the size of a contribution be measured?

You have to differentiate between its size on the conceptual level ("size of content") and on the surface level ("size of string").

Size of string is measured easiest but there is no 1:1-correspondence between size of content and size of string. The ratio of "informational unit per word" ("density") differs between dense language, e.g. in a scientific talk, and shallow language, e.g. in a politicians" talk :-). String size also depends on the expressibility of concepts in a certain language, which depends on the user model. So, it is difficult to predict the string size from the content size.

This could indicate a problem for serial generation architectures!

How can the size of content be measured? Number of propositions? But then, what is a proposition?

4.2 Which factors that impact on the size of a contribution can be identified?

Here are some:

- The amount of noise in the channel.
  If there is plenty of noise, it would be a wise idea to keep contributions small in order to give the hearer a chance of signalling trouble regularly and to keep repetitions in case of trouble small. On the other hand, Noise could increase the need of redundancy in the contributions, that is, increase their string size.

- The genre.
  In small talk, contributions should not be too large in order to give the other participants a chance. In Interviews it is quite O.K. for the interviewed to keep the floor most of the time.

- The predictability of the information (given vs. new).
  In normal speech, rest of words or sentences can be guessed if the beginning was understood. If predictability is lowered, contributions should become smaller. An extreme example: Giving names via the telephone. Size of chunks (= contributions) can be as small as one letter:

A: D
B: yes
A: a
B: yes
A: l
B: yes
A: e
....

- Certain rhetorical "rules" and schemata, e.g. making lists consist of THREE items rather than two ore four.

4.3 How can the informational structure of contributions be organized?

All texts have got some kind of (hierarchical) informational structure. This best be seen in argumentative texts, where it is the argument structure/tree. This structure has to be grasped by the reader/ hearer. The following points can be made about that:

- Structure can be signalled by "cue words/phrases", e.g. "first", "coming back to" and the like. Intonation and different voice levels can be used, too. "Advance organizers" can give hints to the hearer about the organization of the following contribution(s).

- In speech, the structure has to be somewhat "flatter" than in written texts, because of the limited processing capacity of participants (memory etc.) and due to the linear form of speech.
- This might lead to a loss of information (which was encoded in the original structure) if the tree is flattened for the purpose of the dialogue.
- The necessary degree of flattening might depend on your audience. (Computer scientists who are used to tree structures vs. your granny)
- In dialogues via computer terminals, more techniques for signalling structure are available, e.g. itemization and indentation.

4.4 Dialogue aspects.

Don't forget that in a dialogue there is a hearer that might be able to help you.

- So, after each contribution s/he should get a chance of signalling success or failure. To eliciting these "back channels" you can use pauses, tag questions and intonation.
- If the hearer interrupts you unexpectedly this might be a valuable hint for chunking: At least this chunk/contribution was too long! Your program should have methods to deal with this.

5 Conclusions

In conclusion, we would return to our initial statements, and encourage generation researchers to make clear, when they address questions of principle, which perspective on generation they are taking. It is our view that this will lead to much less confusion and disagreement in the field than has been evidenced so far.

Report from Working Group 3: Text Structure

Group Members: Elisabeth André, Ernst Buchberger, Alison Cawsey, Tanya Korelsky, Elena Not, Dietmar Rösner, Elke Teich

1 Introduction

Starting from a set of example texts the group first discussed – among others – the following questions:

- Where and how do intentions come into play in report generation?
- What aspects of aggregation and information clustering are relevant in report generation?

1.1 The material

The sample texts provided were for two different kinds of application (monologue and dialogue) from different genres:

- TEXT1 (Hypertext):

  register: medical (field); doctor-patient relationship (tenor); written, dialogical (mode)

  text type/goal: consultative

- TEXT2 (Dialogue):

  register: information about auto routes (field); client-expert (tenor); spoken, dialogical (mode)

  text type/goal: information-giving

- TEXT3 (Weather report)

  register: weather (field); ships-forecasters (tenor); written, monologue (mode)

  text type: report

- TEXT4 (Statistics)

  register: employment data (field); general public (tenor); written, monologue (mode)

  text type/goal: report

- TEXT5 (Project report)

  register: project data (field); company-internal (employer-employee, among colleagues) (tenor); written, monologue (mode)

  text type/goal: report

2 Report Generation

2.1 Motivation for Report Generation

It was pointed out that, in general, two reasons exist for the use of report generators:

- one, the volume, and
- two, the accuracy

of the data to be presented.

2.2 Conflicts in Report Generation

There are a number of typical conflicts in report writing, e.g.:

1. completeness vs. conciseness:

   On the one hand, the system has to ensure that all relevant information will be encoded. On the other hand, the text should be as concise as possible.

2. relevance vs.exhaustiveness:

   On the one hand, reports should emphasize the most relevant aspects and give them prominence in the presentation. On the other hand, an exhaustive coverage of the data may be wanted as well.

3. predictability vs. variability:

   On the one hand, text with a lot of variations is more appealing and also attracts more easily the reader's attention. On the other hand, it is easier to find relevant information in monotonous text with a fixed structure.

There are other possible sources of conflict, though: in management reports, a certain amount of euphemism is often expected which may conflict with the goal of accuracy.

2.3 Reports as schematized monologues

In reports (e.g. weather, unemployment, stock market, etc.) intentions seem to be compiled into schemata covering the typical structuring of the report.

When generating such monologues, schemata often offer a good basis for the selection and organization of the content to be communicated. The amount of work that can be solved exclusively by their use depends, of course, on the genre and on the context of the communicative situation. But even in those applicative domains and genres where schemata play the major role, we cannot avoid reasoning on the intentions precompiled in the schemata stages. The more the intentions and the reader's expectations are explicitly represented and exploited, the more the system gains generality and flexibility, since the system is able to identify dynamically – during the generation process – whether unwanted implicatures may occur.

Even in the case of report generation, where it could seem that intentions do not need to be explicitly considered, unwanted implicatures may occur. As an example from project reports: if the primary intention of the writer to give a positive and organic view over the project progress needs to be considered. In fact, if the schema simply extracts the data available in the knowledge base and organizes it in a coherent way, some unbalanced presentation may occur where one aspect is more extensively dealt with simply because of

the data available. The unbalanced presentation could suggest to the reader that that aspect is more important than the others.

2.4 Problems with implicatures

When carefully analyzing the sample generated texts, we found that they create wrong expectations or wrong implications in various places.

For example, in TEXT5, first paragraph:

"Progress Against Milestones (1) During the month of April, the preliminary design of Project Reporter and the analysis of Requirements for Project Reporter continued. (2) One programmer, Doug Weber, worked on the analysis of Requirements for Project Reporter."

After mentioning TWO tasks (design and analysis of requirements of/for Project Reporter) in sentence (1), sentence (2) elaborates on only ONE of the tasks (i.e., the requirements). Also, one particular programmer who worked on this task is mentioned by name. Providing no further information about the other task leaves the reader with the impression that the writer of the text wants to stress Doug Webers participation in the second task – an implication that may not have been intended.

Another example of unwanted implication can be found in TEXT2:

"Like acute myocardial infarction (which is in your record) angina pectoris is a kind of ischaemic heart disease ...."

Looking at the record itself, which is another hypertext page, the patient realizes that the importance attributed to 'acute myocaridal infarction' by being the first term mentioned in the text and related directly to the record is misleading because the record itself says that the acute myocardial infarction is no longer active (and therefore not as relevant as the text seems to suggest).

Generalizing from these few examples of undesired implication, we have found that the text structure in general is suggestive of the *intentions* a writer has when designing a text; these must come across to the reader – mediated by the particular text structure. This includes also decisions of expressing or not expressing information, if it is available, or choosing not to express information if it is only partly available. For example, in TEXT5, more information was available only about the requirements tasks, but there was no information about the analysis task. In order not to create the unwanted implications mentioned above, one could have decided NOT to express this information at all or mention that this was the only information available in the data base ("According to our records ... ").

The second example points to a different aspect of the 'intentions-problem'. It points to the problem of the non-availability of information about reader expectations, reader

knowledge etc., in the generation process. If it was clear, what the most relevant information for a reader (here: patient) is, the text could be re-phrased in a way that reflects this relevance (e.g., as the re-phrasing De Smedt suggested). This, of course, is not particularly a generation problem; the more fundamental question is: what makes a text a GOOD text for what kind of purpose?

3 The role of intentions

3.1 Examples of intentions

Some examples of intentions that are relevant for the example texts (R ... reader):

- in a tutoring application: R understands < topic>
- in a medical information application: R knows <how to take treatment> R is-persuaded-to <take treatment>
- in technical instructions: R able-to-perform <action> (in a way that is harmless to him and to the device)
- in reports: R knows <data> R confident that s/he got complete information

3.2 Derived guidelines

In order to conform to these intentions, some general guidelines can be formulated:

- stress positive aspects
- increase reader happiness
- stress progress
- do not hide, but do also not highlight negative facts
- instruction: allow reader to perform a certain action in a way that is harmless to him/her and to the device

These guidelines are quite general; others may be derived from them:

- do not be too verbose
- be complete
- be concise

To put it quite general, the main goal is to make the reader happy, e.g. by allowing him/her to understand e.g. terms, treatment, etc. This depends on the level of presentation (Gricean maxims).

A complication although is brought about by the fact that intentions can also be in conflict.

3.3 Keeping the Reader's Attention

An important aspect of conciseness is that it is a means for keeping the reader's attention.

How to keep the reader's attention:

- conciseness
- variation in wording

Example for variation in statistical report: with certain numbers use 'rise', with others 'increase'.

Another example for variation is to (*but this is not strictly variation in wording - is this another means of keeping the reader's attention?*): include small anecdotes. In medical domains this also increases the reader's confidence (Mrs.X had the same type of illness, and she did Y, and now she is much better) (Cawsey). Such anecdotes/examples are quite useful – they increase authenticity.

3.4 Classification of Intentions

Intentions can refer

- to the effects of an utterance on the reader's mental state (i.e. ability, belief, desire, ..., and R's actions or decisions) or
- to the communicative act itself and the rhetorical means (e.g. keep R's attention, minimize R's effort, maximize understandability)

Examples of the first kind of intention are: the reader understands a certain topic in tutoring systems, the reader is willing to take a medical treatment in a medical application, the reader is able to operate a technical device in an instructional setting or the reader knows all relevant data in a report. Examples of the second kind of intention are: to inform the reader about a certain event in a newspaper report or to describe an action in instructions for use.

In addition to that, we also have to distinguish between the *primary* and the *secondary effects* of a presentation. Primary effects relate to immediate effects on the reader's mental state, e.g., the reader knows a certain fact. Secondary effects relate to the indirect effects of a presentation, e.g., to attract the reader's attention or to distract the reader.

3.5 Intentions and linguistic realization

Of course, intentions do not influence only the content selection and organization, but all the following steps of aggregation and linguistic realization.

3.6 Aggregation in reports

An important aspect for report generators is correct use of aggregation: instead of listing the employees by name, you would say "*all* employees". Instead of listing various minor accidents, you could say "*No* major accidents were reported". Reasons for using aggregation are conciseness, exhaustiveness, stressing importance: "*All* members worked on A, therefore B had to be postponed."

3.7 Deciding about aggregation

In particular, when deciding on the introduction of aggregations all the following factors have to be considered:

- conciseness;
- relevance and emphasis cues that can be introduced by means of quantifiers (e.g. *"One programmer, Doug Weber, worked on ...."*);
- exhaustiveness cues (e.g. *"All the people in the group ...."*);
- implications conflicting with the writer's intentions (e.g. *"No task was completed within the deadline"*).

The final decision about aggregating will be a compromise between all these factors.

4 Towards a methodology of natural language generation applications

4.1 A uniform view

Against the usual distinction, there are many commonalities between the setting of report generation and the setting of dialogue: monologue can be seen as dialogue with an imagined partner (Rösner) as opposed to the often held position of seeing monologue as one stretch in a dialogue. In the same vein, parametrized reports can be seen as dialogues in which one partner answers an (imagined) partner's questions.

4.2 A questionnaire

We concluded that, although schema-based approaches to (sub)language generation are useful, a detailed analysis of the particular text type should take place which considers not just the surface features and structure of the text, but also the context and intentions behind the text. This kind of analysis should both allow "better" schemata to be developed (that, for example, avoid unwanted implicature), and also allow these schemata to be re-used (to some extent) in systems for generating texts of a related type. Papers describing schema-based systems should ideally report on such an analysis, so that others

can determine how far the details of the approach reported can be applied to a new text type.

Issues to consider in this analysis include:

- Situation:
  - What activities will the readers be engaging in when reading the text?
  - How has the prior discourse/text developed?
- The writer:
  - What are the goals of the "writer".
- The reader:
  - What are their goals? What types of actions or decisions do they need to make?
  - What is their level of prior knowledge?
  - What are their interests?
  - How interested/motivated are they to look at the text?
- Their roles:
  - What are the roles of the participants in the exchange?
- Textual conventions and prior texts:
  - Are there any expectations set up by conventions or
  - previously read texts?
- Mode:
  - Is it written/spoken?
  - Is it dialogue or monologue? If the former, what degree of interaction is allowed.

We should consider how these features influence observable semantic features such as the use of aggregations. examples, anaphoric forms, tenses, person, theme etc; as well as the content and overall structure of the text.

Although there is much research on these sorts of issues, there may be a tendency for applications-oriented work to bipass the issues as working within a narrow sub-language. The suggestion therefore is that applications-oriented work can both benefit from and contribute to an analysis of such issues, while still working within a simple and practical schema-based framework.

4.3 A closer look

The flavour of these questions can be summed up in a sort of meta-level Gricean maxim: for every domain you define, be consistent with what the writer and reader can be happy with.

- Who is/are the writer/s the generator is substituting?

  This issue has a lot of relevance because it determines the viewpoint that influences the whole text. E.g. in the domain of Management Reports both the Project Manager and the officer that checks the project developments for the funding agency may have interest in writing a Progress Report. But they may wish to emphasize completely different aspects of the project.

- Who is/are the reader/s? Which is the motivation for reading the text or engaging in the dialogue?

  Here, what is important is not only the expertise level of the reader but also his social role in the communicative exchange. E.g. in the domain of Management Reports both a partner in the project consortium and the funding agent may have a very high level of expertise wrt the technical jargon and the financial data conventions used in the text. But the two potential readers look for different information when reading the text. The project partner wants to have an objective and, as much as possible, exhaustive view over the project workplan. While the funding agent focuses his attention on anomalous situations or delays with respect to the schedule.

- What type of actions has the reader to take? Which decisions to make? How do they combine with the writer's intentions?

  According to the type of actions or decisions the reader has to perform after reading the text, the text have to be tailored in order fulfill the writer's intentions. E.g., if the writer's intention is to give a positive image of the project progress to a reader that is to critize delays or problems occurred, the text is to include explanations that could help justifying the snags occurred.
- Which are the reader's expectations?

  This is a factor that can be very useful in deciding whether to include in the text some information retrievable by means of the schemata. E.g., if the reader expects that the Management Report is to be produced from a complete data record the following text will suggest him that something went wrong in the past project phase: *"During the month of April, the preliminary design of Project Reporter and the analysis of Requirements for Project Reporter continued. One programmer, Doug Weber, worked on the analysis of requirements for Project Reporter."* Here, the negative impression to the reader is given by the fact that only for one of the two tasks information about man power involved is given. With a closed world assumption, the reader might conclude that nobody worked on the first task. More in general, to avoid breaking reader's expectations, schemas have to be used consistently: if one stage of the schema has

been applied in a particular situation, it has to be applied in all similar situations or its unapplicability has to be explicitly signalled. For example, when describing some electric devices, if a warning has been introduced in the text for one of the components, a warning should be introduced for each other dangerous component. Otherwise the reader could conclude that they are not dangerous at all. As a general rule: the text should be predictable with respect to reader's expectations.

- Which is the reader's willingness to read the text / engage in the dialogue?

  It is important to know which is the level of interest of the reader in starting or continuing the reading of a text or his participation in a dialogue. This has relevance on all those discourse or linguistic factors that stimulate the reader's/hearer's attention, like for example: the content selection (whether to include an information or not), the thematic progression strategy, the choice of the level of conciseness, the degree of lexical variation and on the introduction of examples.

4.4 Problems with Schemata

For certain sublanguages, e.g., weather reports, it is obviously not necessary to maintain an explicit representation of the intentional structure of the discourse. In this domain, it seems to be possible to compile primary communicative goals, e.g., the reader knows all relevant events, into schemata. However, schemata have to be adapted to the individual reader, the conversational setting and the domain. Among other things, these parameters have influence on the organization of a text, the style, the degree of detail and the kind of the selected information. It can be said that schemata contain compiled information about a certain sublanguage, e.g., reports or instructions.

Although schemata often seem to provide a good basis for text generation, they also bear disadvantages:

- *Wrong implicatures*

  Since schemata contain a lot of compiled information, they may lead to wrong implicatures. To avoid wrong implicatures, we have not only to maintain an explicit representation of the intentions behind an utterance, we also have to consider possible inferences the reader may draw and to block unwanted inferences. Among other things, we discussed implicatures arising from aggregations and enumerations. For example, "All persons worked on A, therefore B had to be postponed." stresses the importance of A whereas "X, Y and Z worked on A, therefore B had to be postponed" is less plausible.

- *Reusability*

  It is difficult to transfer schemata to another domain because reusablity requires interpretation. In particular, a system has to understand the reasons behind the use of a schema. This is, however, only possible if the intentions underlying the discourse are explicitly represented.

- *Interactivity*

  We take a uniform view of monologues and dialogues and consider monologues as dialogue contributions to one or several imaginary conversational partners. In a dialogue situation, each conversational partner has the possibility to tailor its contributions on-line to the current situation, e.g., if the other dialogue partner complains or asks follow-up questions. In contrast to this, monologues have to be adapted off-line to the needs of a potential reader. In dialogues, intentions play a more central role than in monologues. As also shown by Paris, approaches relying on schemata have difficulties in handling follow-up questions because the rationale behind the use of a schema is not explicitly represented. A system using schemata can apply another schema if the reader complains, however, it is very hard to answer questions corresponding to a smaller part of a schema.

4.5 Towards reusability

The general problem all generators the sample generated texts of which we looked at have is an under-specification of information about intentions, reader expectations, reader knowledge etc...

This includes both global-level decisions in packaging information and knowledge of local-level text structure preferences (e.g., theme, reference etc.), for example, for a particular stage of a text.[8] For example, TEXT4 could be re-organized locally in terms of thematic development that respects the global organization in terms of schemas.

Furthermore, this general under-specification is a severe impediment for reusability. If reusability is an issue, it seems the wrong strategy to look at just one sublanguage because this will prevent a generalization over application contexts, domains and text types.

Similarly, looking at dialogue or monologue only, one will miss uncovering commonalities and not be able to make use of them in system design. It may be recommendable to consider monologue and dialogue on a cline rather than being two completely different things. For a monologue, there will always be a situation one imagines the discourse to take place in, and this must be represented as a crucial source of constraint on the whole generation process. As to the implication problem sketched above, representing the situational context of a discourse, be it monologic or dialogic, is indispensable, if one aims at determining which implications are to be created and which are not. One of the major differences then between a dialogue and a monologue then is that for a monologue we must assume a particular speaker-hearer relationship and this will hold throughout the discourse; for a dialogue, in contrast, the speaker-hearer relationship can be negotiated again and again – and thus also intentions and possible implications may vary as the discourse unfolds.

---

[8] The sample texts generally showed a lack of coherence and cohesion.

The general agreement in our working group was that that a full-fledge generation system must include knowledge about intentions as a major source of constraints on the well-formedness of a text.

4.6 Summary: A practitioner's perspective

The following quotation from Tanya Korelsky summarizes the discussion of the working group from the perspective of practitioners that design, implement and sell report generators:

4.6.1 The methodological importance of bringing intentions "back" into schema-based report generation

Since we at CoGenTex work on applications, we probably would never attempt to build a system where intentions could be stated declaratively and abstractly enough to be re-usable between different domains and compiled into text planning schemas automatically. I still assume that we as developers author text planning schemas.

I think, however, that a methodology should be developed that

(A) allows the developers to establish an adequate set of intentions for an application (intentional Domain Communication Knowledge, DCK)

- with respect to both
  - content ("the reader feels adequately informed") and
  - communication ("the reader does not feel tired, bored or lost")
- containing both
  - domain-independent (like the ones above), or rather their instantiations for a particular domain. Every domain will have its own structured definition of "adequately informed". In my opinion, the notorious "Doug Weber" report did not fulfill this particular intention and thus gave rise to wrong implicatures.
  - domain-dependent ones (e.g., "the reader should get the impression that some progress has been done", for the partner modeling domain)

(B) allows the developer to state explicitly how these intentions (intentional DCK, essentially, meanings of a particular kind) were implemented (realized) for this particular application by the means of all the levels of text generation process: the content selection, the rhetoric, sentence planning, and lexical choices. I am not sure whether mapping of intentions into observable phenomena can be "pipelined" through these levels in the Meaning-Text vein (similar to the approach Richard Kittredge and I tried to outline in our preliminary note "Stratification of RST" in the workshop at ACL-93), this requires

serious research. It should be and this mapping would represent a complete DCK for an application.

Describing such mappings (or at least providing an explicit statement of a strong correlation) is very important because it provides a rationale for all the decisions that are compiled into an application generator on all levels (not just into text planning schemata). This rationale could then be used by a maintainer of this appliciation who would respect the application's intentional set and its mapping into observable language phenomena when making changes during the lifetime of the application. This is not to say that the intentional set itself cannot change during the application lifetime; however, such changes (either to the intentional set or to the mapping system) will have to be recorded explicitly.

(C) allows the developer to avoid certain content, rhetorical and lexical choices which are known to be incompatible with some particular intentions.

Such a methodology and its repositories would, in my opinion, constitute a very much needed re-usable knowledge base for text generation. I repeat that I am not sure how much automation might be practical for application development, however, it would be very interesting to set up a research project which would probe into the automation aspects.

4.6.2 The methodological importance of an integrated approach to monologue and dialogue text generation.

Monological text generation genres (such as reports) can be seen as historically evolved, debugged and frozen (omitting questions) dialogues, i.e., as texts providing answers to an established (and changing only very slowly) collection of questions.

This approach is very fruitful, in my opinion, because it would allow application developers to analyze the reports from a dialogue participant's point of view: i.e., reconstruct how the current report's structure and style anticipate and pre-empt the reader's follow-up questions. This methodological approach would allow the developer to understand and analyze the sample texts in a deeper, intentions-oriented way. The intentional set for an application and its mapping into observable language phenomena, which I mentioned above, will result from such analysis.

The currently evolving genre of a restricted dialogue via hypertext buttons when the reader can choose among a limited number of follow-up questions (e.g., Alison's application, or Mellish and Reiter's IDAS system) provides an example of an interesting monologue/dialogue hybrid. It makes the job of the generator (text planner) substantially easier, and provides the reader with some autonomy in composing an answer which humans greatly appreciate.

Report from Working Group 4: Text Structure

Group Members: Barbara di Eugenio, Robin Fawcett, Wolfgang Hoeppner, Elisabeth Maier, Benoit Lemaire, Norbert Reithinger, Thomas Rist, Annely Rothkegel

1 Introduction

The discussion in our group started with identifying those subtopics of the general theme 'Text structure', which seemed to be most interesting to the participants:

- Is there a genuine notion of text structure both for monologues and dialogues?
- What are the merits of either a template-approach, or a so called principled approach? (This dichotomy might be rephrased with text schemata vs. planning, top-down vs. bottom-up strategy)
- How is the structure of the 'input data' related to the final text, especially to the purpose of the text?
- How can thematic progression be employed? Is there a general concept for (co-)referring expressions?
- Is there a general conception underlying different levels of detail?
- How important are user models, which possibilities do we really have (besides wishful thinking) to generate specific texts for a specific person?

Reading through these questions one is inclined to think that each of them would justify a complete workshop on its own. Our working group, therfore, was only able to touch some of these issues. Most of the time was devoted to the first question, and this is, where we reached some conclusions, which will be reported below. On the other hand, the six topics we thought to be urgent questions in the area of natural language generation obviously are not monolithic research objectives, but rather are interconnected in a non-trivial way.

2 Structure of Monologues

Our discussion on the structure of monological texts started with the analysis of a report on employment developments in Canada, which was provided by Dick Kittredge and Tanya Korelski as an example for an application-oriented generation task (cf. fig. 1). The data to be verbalized were a tabular representation of various numbers of employed and unemployed people, subdivided into males, females and age groups.

It turned out that the structure of the generated text follows a schematic approach (as could be described e.g. by McKeowns text schemata). This is not astonishing, since the

Commentary

Overview

Estimates from Statistics Canada's Labour Force Survey for November 1989 show that the seasonally adjusted level of employment rose by 32000 and that the level of unemployment increades by 30000. The unemployment rate increased by 0.2 to 7.6.

Employment

For the week ended November 6, 1989, the seasonally adjusted level of employment was estimated at 12568000, up 32000 from October. The increase was concentrated among women aged 25 and over. The employment/ population ration remained virtually unchanged.

Employment among women aged 25 and over rose by 44000 and the employment/population ratio among women aged 25 and over increased by 0.5 to 52.3.

Employment among men aged 25 and over fell by 12000 and the employment/population ratio among men aged 25 and over decreased by 0.3 to 72.5.

Part-time employment increased by 25000. The increase was evenly distributed between men and women.

Full-time employment remained virtually unchanged.

Employment fell by 10000 in agriculture, by 12000 in transportation, communication and other utilities and by 12000 in other primary industries. Employment rose by 68000 in services and by 20000 in trade. It remained virtually unchanged on the other sectors.

Emploiyment rose by 11000 in Quebec, by 8000 in Alberta, by 6000 in British-Columbia and by 5000 in Ontario. Employment decreased by 4000 in Saskatchewan. It remained virtually unchanged in the other provinces.

Unemployment and Participation Rate

The seasonally adjusted level of unemployment was estimated at 1032000 for November 1989, up 30000 from October. The unemployment rate rose and the participation rate increased by 0.3 to 67.2.

The increase in unemployment was concentrated among men aged 25 and over.

Unemployment among men aged 25 and over increased by 24000 while it remained virtually unchanged among women aged 25 and over.

The unemployment rate among men aged 15 to 24 increased by 0.7 to 12.9.

The participation rate among men aged 15 to 24 increased by 0.5 to 73.4 and it remained virtually unchanged among women aged 15 to 24.

Unemployment increased by 24000 in Ontario. It remained virtually unchanged in the other provinces.

The unemployment rate rose by 0.8 to 15.7 in Prince Edward Island, by 0.5 to 12.8 in New Brunswick, by 0.4 to 5.3 in Ontario, by 0.3 to 8.0 in Manitoba and by 0.2 to 7.5 in Saskatchewan. The unemployment rate in the other provinces remained virtually unchanged.

Fig. 1

VERY tentative 'structure potential' for genre: report (sub-genre: government employment) (based on evidence of ONE text! - plus guesses based on experience!)

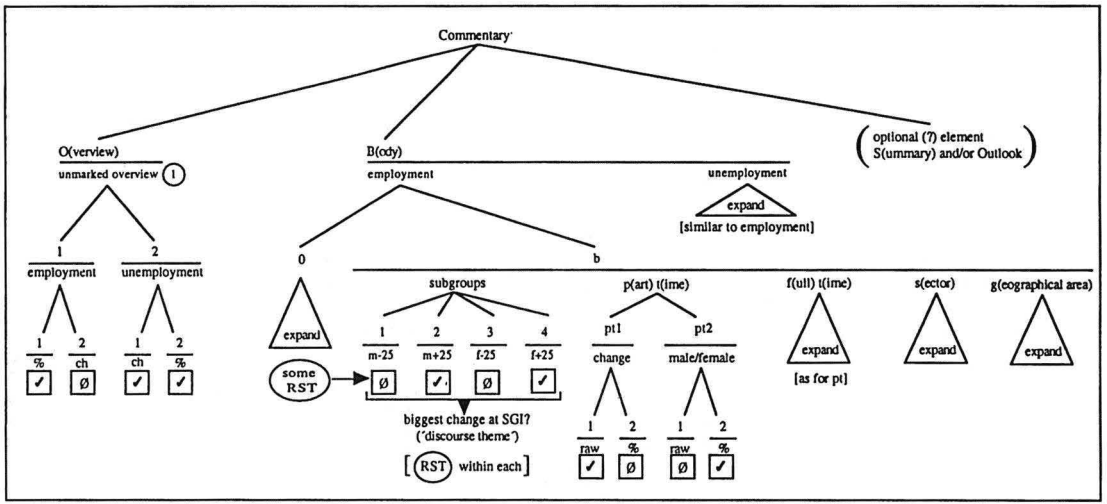

Fig. 2

Key:
raw = raw figure
ch = no. of change
% = % of change

(1) A 'marked overview' would be generated if a strongly newsworthy were to be 'promoted' from its unmarked element of structure (within B) tp precede this.

raw data are entries in a data base indicating the numerical distribution of employment classes on a monthly basis. The task of transforming these data into a natural language

THE GENRE AND EXCHANGE STRUCTURE OF A SIMPLE AUTOROUE DIALOGUE

Robin P Fawcett and Yuen Q Lin (last revised 22.3.94)

The class of genre is TELEPHONE INFORMATION SERVICE. The sub-genre is MOTOR ROUTE PLANNING. Exchanges are separated by a space. (The class of exchange is directly derivable from the class of Initiate move.) The genre and exchange grammars from which the structure below is generated also cover, or are adaptable to, many other variants of structure. 'Ivy' is the system's name.

INTERACTANT	TEXT	
User:	makes Ivy's telephone ring	
	Ivy picks up phone; Ivy discourse-ellipts R with h of 'signify attention'.	
Ivy:	good mordning/T2	
	Ivy discourse-ellipts R with h of 'rtn greeting'.	
	this is the Autoroute route/T1 planning service	
	how can I help/T1	
User:	I would like you to plan a route from Malvern to Tenby/T 1 please 2	
Ivy:	oh kay/T 2-	
Ivy.	Ido you want to stop off anywhere on the way/T 2	
User:	no I don't/T 1	
Ivy:	and what time do you want to arrive/T 1	
User:	I must be there by five pee em/T 1	
Ivy:	right/T 1	
request:	please wait for a moment/T 2 while we work out the route/T 1	
[Music (Albinoni?) is faded repidly in, then is faded out ten seconds later when the calculation is completed]		
Ivy.	right/T 1	
	we have worked out your route/T 1	
	the quickest route from Malvern to Tenby is a hundred and sixty-one miles/T 2	
	and the journey will take about three hours/T 1	
	would you like me to send you a printout of the route-plan/T 2	
User:	yes please/T 2-	
Ivy:	oh kay/T 2	
	could you give me your name and your address/T 1- please/T 2	
User:	yes/T 1	
	my name is George Edwards/T 2	[pause of 2 seconds to record it)
	and my address is fifty-eight/T 2	[pause of 1 second]
	Saint Andrews Road/T 2	[pause of 2 seconds]
	Malvern/T 2	[pause of 1 second]
	and the postcode is double-you are fourteen/T 2 two eh ee/T 1	[pause of 2 seconds]
Ivy:	thanks/T 1	
	the printout will be posted to you immediately/T 1	
User:	thank you very much indeed/T 1	
Ivy:	you're welcome/T2-	
	bye/T 2 1	
User:	bye/T 1	[both put phone down]

Fig. 3

report can be seen as an instantiation of a generic employment-report pattern. Selection between different items to include into the text candirectly be associated with the values found in the input data. If, for instance, only a marginal change in a specific data set is encountered, this might be reflected in the text by not mentioning it, or by a standard phrase ("It remained virtually unchanged in the other sectors" – this was, what the system did).The schematic structure we 'invented' for this real-world text, which was actually generated by machine, is shown in fig. 2. It was emphasized during the discussion, that the quality of such schema-based generated texts can be improved by applying aggregation techniques (likewise as data aggegation before schema instantiation, or as a kind of text restructuring after the instantiation).

3 Structure of Dialogues

The empirical basis for looking at dialogues was given by an interactive route description system provided by Robin Fawcett (cf. fig. 3). The goal of this dialogue was to request

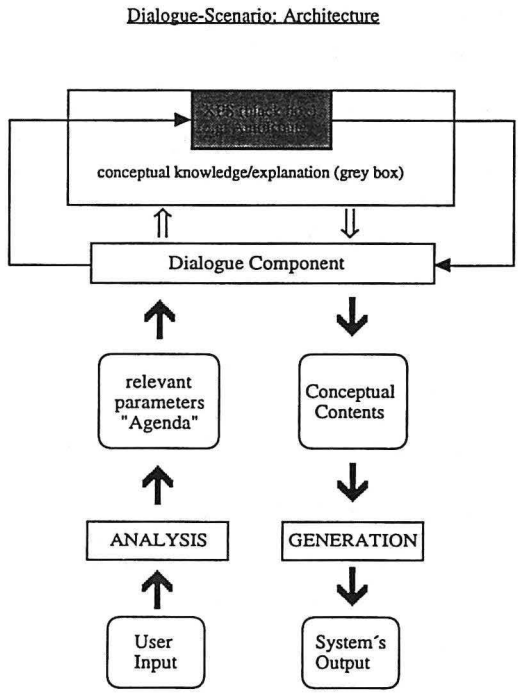

Fig. 4

the necessary input parameters for the Expert System AutoRoute, which would then produce a detailed (tabular) description of the road segments making up the overall route. This description would then be sent to the user by mail. One might even think of generating a natural language description of the expert system's results, which, of course, would require further research, for instance to determine which of the road segments might be comprised.

Robin's proposal of describing the structure of the example dialogue by means of conversational moves and speech-act classification was not completely supported in our working group (though his descriptive apparatus is fine grained). We rather advocated in favor of a more flexible solution by specifying a general architecture for an interactive system (cf. fig. 4). Knowledge of how to continue a dialogue and how to interpret user input should be contained in a 'dialogue component'. This was regarded as more flexible than a grammar of conversational moves, because it allows access to diverse knowledge sources (e.g. dialogue history, user model, conceptual and referential knowledge).

For an interactive natural language system as a means of activating the problem solving capabilities of an expert system (XPS) we discriminated between a 'black box', which would be the XPS per se, and a 'grey box', which would contain conceptual knowledge, thus allowing, among other things, facilities for explanation. The interaction of the dialogue component with the black box would consist in providing necessary data for the XPS to do its job and interpreting results of the XPS. Interaction with the grey box, however, would be a major task of the natural language processing components, deciding about what questions can be sensibly asked to the user, determining what is conversationally implicated by previous exchanges, by world knowledge, or assumptions about the user (class), and many other subtasks as well.

4 Summary

Trying to identify possible differences between generation in a dialogue setting and that in a pure monological situation we reached the following positions:

- There is a big difference between the addressees: dialogues usually are with an individual user, texts are produced for a set of (potentially quite heterogeneous) persons. Exploitation of a user model clearly has to take this difference into account. It seems as if generation for a specific person should be easier than for various addressees with quite different backgrounds. But: where do we get the specific knowledge about the specific user in a dialogue?

- In a dialogue setting, the update of the system's listener/user model has to take into account both, information conveyed so far by the system (i.e. in the previous discourse), and new input/ feedback from the real user. In monologues, however, all of the user's wishes, needs and reactions, which in dialogues can be acquired during the interaction, have to be hypothesized and integrated right from the start.

- Dialogues require problem solving during the generation task (e.g. by consulting the black or grey box while making decisions), For monologues, the problem solving usually is completed before generation starts.

- Dialogues are more likely to require flexibility in text structure (optional elements, change of order of elements, loops, embeddings etc.). Monologues can operate with a relatively fixed text structure, i.e. the template approach might be more adequate for these generation tasks.

- In dialogues there is a broad continuum between the information-seeking & -giving type and real consultation. Such a spektrum does not seem to be applicable for monologues.

Dagstuhl-Seminar 9430:

List of Participants

Elisabeth **Andre**
Deutsches Forschungszentrum
für Künstliche Intelligenz
Stuhlsatzenhausweg 3
66123 Saarbrücken
Germany
andre@dfki.uni-sb.de
tel.: +49-681-302 5267

John **Bateman**
GMD-IPSI
Dolivostr. 15
D-64293 Darmstadt
Germany
bateman@gmd.de
tel.: +49-6151-869-826

Ernst **Buchberger**
Austrian Research Institute for
Artificial Intelligence
Schottengasse 3
A-1010 Wien
Austria
ernst@ai.univie.ac.at
tel.: +43-1-535-32 810

Stephan **Busemann**
Deutsches Forschungszentrum
für Künstliche Intelligenz
Stuhlsatzenhausweg 3
66123 Saarbrücken
Germany
busemann@dfki.uni-sb.de
tel.: +49-681-302-52 86

Alison **Cawsey**
University of Glasgow
Department of Computing Science
17 Lilybank Gardens
Glasgow G12 8QQ
Great Britain
alison@dcs.glasgow.ac.uk
tel.: +44-41-339-88 55 x60 39

Robert **Dale**
Microsoft Institute of Advanced
Software Technology
GS Epping Road
North Ryde Nsw 2113
Australia
rdale@microsoft.com
tel.: +61 2 870 2370

Koenraad **De Smedt**
University of Leiden
Exper. Psychology Unit
P. O. Box 9555
NL-2300 RB Leiden
The Netherlands
desmedt@rulfsw.leidenuniv.nl
tel.: +31-71-27 34 07

Barbara **Di Eugenio**
Carnegie Mellon University
Department of Philosophy
5000 Forbesave
Pittsburgh PA 15213
USA
dieugeni@lcl.cmu.edu
tel.: +1-412-268 8149

Robin **Fawcett**
University of Wales
Computational Linguistics Unit
Aberconvay Building
Cardiff CF1 3 TL
Great Britain
fawcett@uk.ac.cardiff
tel.: +44-222 874293

Wolfgang **Finkler**
Deutsches Forschungszentrum
für Künstliche Intelligenz
Stuhlsatzenhausweg 3
66123 Saarbrücken
Germany
finkler@dfki.uni-sb.de
tel.: +49-681-302-52 69

Wolfgang **Hoeppner**
Universität-GH Duisburg
Fachbereich 3 - Computerlinguistik
Lotharstr. 65
D-47048 Duisburg
Germany
he232ho@unidui.uni-duisburg.de
tel.: +49-203-379-20 06/2008

Helmut **Horacek**
Universität Bielefeld
Fakultät für Linguistik und
Literaturwissenschaft
Postfach 10 01 31
D-33501 Bielefeld
Germany
horacek@techfak.uni-bielefeld.de
tel.: +49-521-106-36 78

Richard **Kittredge**
Université de Montréal
Linguistique et Traduction
CP 6128
Montreal H3C 3J7
Canada
kittredg@iro.umontreal.ca
tel.: +1-514-343-2114

Tanya **Korelsky**
CoGenTex Inc.
Village Green
840 Hanshaw Rd
Ithaca NY 14850-1589
USA
tanya@cogentex.com
tel.: +1-607-266-0363

Tjoe-Liong **Kwee**
University of Amsterdam
Computational Linguistics
Spuistraat 134
NL-1012 VB Amsterdam
The Netherlands
tl@alf.let.uva.nl
tel.: +31-20-5 25- 20 72

Benoit **Lemaire**
Swedish Institute of Computer Science
Box 12 63
S-164 28 Kista
Sweden
lemaire@sics.se
tel.: +46-8-752-15 15

Nils **Lenke**
Universität-GH Duisburg
FB Sprach- u. Literaturwissenschaften
FG Computerlinguistik
Lotharstr. 65
47048 Duisburg
Germany
he232le@unidui.uni-duisburg.de
tel.: +49-203-379-2007

Elisabeth **Maier**
DFKI
Stuhlsatzenhausweg
D-66123 Saarbrücken
Germany
maier@dfki.uni-sb.de
tel.: +49-681-302-53 47

Günter **Neumann**
DFKI Saarbrücken
Stuhlsatzenhausweg 3
66123 Saarbrücken
Germany
neumann@dfki.uni-sb.de
tel.: +49-681-302-52 83

Elena **Not**
Istituto Per la Ricerca
Scientifica e Tecnologica
Loc. Pante di Povo
I-38050 Povo
Italy
not@irst.it
tel.: +39-461-31 43 45

Hajo **Novak**
IBM Deutschland GmbH
Entwicklung und Forschung
AE SAT/7030-91
Hans-Klemm-Str. 45
71003 Böblingen
Germany
novak@vnet.ibm.com

Conny **Peters**
Universität Bielefeld
Fakultät für Linguistik und
Literaturwissenschaft
Postfach 10 01 31
D-33501 Bielefeld
Germany
peters@lili.uni-bielefeld.de
tel.: +49-521-106-35 14

Dietmar **Rösner**
FAW Ulm
Helmholtzstr. 16
89081 Ulm
Germany
roesner@faw.uni-ulm.de
tel.: +49-731-501-530

Norbert **Reithinger**
DFKI Saarbrücken
Stuhlsatzenhausweg 3
66123 Saarbrücken
Germany
bert@dfki.uni-sb.de
tel.: +49-681-302-53 46

Thomas **Rist**
Deutsches Forschungszentrum
für Künstliche Intelligenz
Stuhlsatzenhausweg 3
66123 Saarbrücken
Germany
rist@dfki.uni-sb.de
tel.: +49-681-302 5266

Annely **Rothkegel**
Universität des Saarlandes
FR 8.7 Computerlinguistik
Postfach 15 11 50
66041 Saarbrücken
Germany
rothkegel@coli.uni-sb.de

Manfred **Stede**
FAW Ulm
Helmholtzstr. 16
89081 Ulm
Germany
stede@faw.uni-ulm.de
tel.: +49-731-501-89 69

Elke **Teich**
GMD-IPSI
Dolivostr. 15
64293 Darmstadt
Germany
teich@darmstadt.gmd.de
tel.: 0681-302-4485

Leo **Wanner**
Universität Stuttgart
Institut für Linguistik
Keplerstr. 17
70174 Stuttgart
Germany
wanner@ims.uni-stuttgart.de
tel.: +49-711-121-24 46

Michael **Zock**
LIMSI
B.P. 30
F-91406 Orsay
France
zock@limsi.fr
tel.: +33-1-69 85 80 05

Zuletzt erschienene und geplante Titel:

Ph. Flajolet, R. Kemp, H. Prodinger (editors):
"Average-Case"-Analysis of Algorithms, Dagstuhl-Seminar-Report; 68; 12.07.-16.07.93 (9328)

J.W. Gray, A.M. Pitts, K. Sieber (editors):
Interactions between Category Theory and Computer Science, Dagstuhl-Seminar-Report; 69; 19.07.-23.07.93 (9329)

D. Gabbay, H.-J. Ohlbach (editors):
Automated Practical Reasoning and Argumentation, Dagstuhl-Seminar-Report; 70; 23.08.-27.08.93 (9334)

A. Danthine , W. Effelsberg, O. Spaniol, (editors):
Architecture and Protocols for High-Speed Networks, Dagstuhl-Seminar-Report; 71; 30.08.-03.09.93 (9335)

R. Cole, E. W. Mayr, F. Meyer a.d.Heide (editors):
Parallel and Distributed Algorithms, Dagstuhl-Seminar-Report; 72; 13.09.-17.09.93 (9337)

V. Marek, A. Nerode, P.H. Schmitt (editors):
Non-Classical Logics in Computer Science, Dagstuhl-Seminar-Report; 73; 20.-24.09.93 (9338)

A. M. Odlyzko, C. P. Schnorr, A. Shamir (editors):
Cryptography, Dagstuhl-Seminar-Report; 74; 27.09.-01.10.93 (9339)

J. Angeles, G. Hommel, P. Kovács (editors):
Computational Kinematics, Dagstuhl-Seminar-Report; 75; 11.10.-15.10.93 (9341)

T. Lengauer, M. Sarrafzadeh, D. Wagner (editors):
Combinatorial Methods for Integrated Circuit Design, Dagstuhl-Seminar-Report; 76; 18.10.-22.10.93 (9342)

S. Biundo, R. Waldinger (editors):
Deductive Approaches to Plan Generation and Plan Recognition, Dagstuhl-Seminar-Report; 77; 25.10.-29.10.93 (9343)

P. Gritzmann, D. Johnson, V. Klee, Ch. Meinel (editors):
Counting Issues: Theory and Application, Dagstuhl-Seminar-Report; 78; 06.12.-10.12.93 (9349)

B. Endres-Niggemeyer, J. Hobbs, K. Sparck Jones (editors):
Summarizing Text for Intelligent Communication, Dagstuhl-Seminar-Report; 79; 13.12.-17.12.93 (9350)

Ch. Brink, G. Schmidt (editors):
Relational Methods in Computer Science, Dagstuhl-Seminar-Report; 80; 17.01.-21.01.94 (9403)

A. Arnold, H. Seidl, B. Steffen (editors):
Algorithms in Automata Theory, Dagstuhl-Seminar-Report; 81; 07.02.-11.02.94 (9406)

K. Ambos-Spies, S. Homer, U. Schöning (editors):
Structure and Complexity, Dagstuhl-Seminar-Report; 82; 14.02.-18.02.94 (9407)

H. Noltemeier, T. Ottmann, D. Wood (editors):
Data Structures , Dagstuhl-Seminar-Report; 83; 28.02.-04.03.94 (9409)

J. P. Finance, S. Jähnichen, J. Loeckx, D. Smith, M. Wirsing (editors):
Logical Theory for Program Construction, Dagstuhl-Seminar-Report; 84; 07.03.-11.03.94 (9410)

R. Klette, W.G. Kropatsch, F. Solina (editors):
Theoretical Foundations of Computer Vision, Dagstuhl-Seminar-Report; 85; 14.-18.03.94 (9411)

A. Buchmann, S. Chakravarthy, K. Dittrich (editors):
Active Databases, Dagstuhl-Seminar-Report; 86; 21.03.25.03.94 (9412)

F. Meyer a.d. Heide, H.J. Prömel, E. Upfal (editors):
Expander Graphs, Random Graphs and Their Application in Computer Science, Dagstuhl-Seminar-Report; 87; 11.04.-15.04.94 (9415)

J. van Leeuwen, K. Mehlhorn, T. Reps (editors):
Incremental Computation and Dynamic Algorithms, Dagstuhl-Seminar-Report; 88; 02.05.-06.05.94 (9418)

R. Giegerich, J. Hughes (editors):
Functional Programming in the Real World, Dagstuhl-Seminar-Report; 89; 16.05.-20.05.94 (9420)

H. Hagen, H. Müller, G.M. Nielson (editors):
Scientific Visualization , Dagstuhl-Seminar-Report; 90; 23.05.-27.05.94 (9421)

T. Dietterich, W. Maass, H.U. Simon, M. Warmuth (editors):
Theory and Praxis of Machine Learning, Dagstuhl-Seminar-Report; 91; 27.06.-01.07.94 (9426)

J. Encarnação, J. Foley, R.G. Herrtwich (editors):
Fundamentals and Perspectives of Multimedia Systems, Dagstuhl-Seminar-Report; 92; 04.07.-08.07.94 (9427)

W. Hoeppner, H. Horacek, J. Moore (editors):
Prinzipien der Generierung natürlicher Sprache, Dagstuhl-Seminar-Report; 93; 25.07.-29.07.94 (9430)

A. Lesgold, F. Schmalhofer (editors):
Expert- and Tutoring-Systems as Media for Embodying and Sharing Knowledge, Dagstuhl-Seminar-Report; 94; 01.08.-05.08.94 (9431)

H.-D. Ehrich, G. Engels, J. Paredaens, P. Wegner (editors):
Fundamentals of Object-Oriented Languages, Systems, and Methods, Dagstuhl-Seminar-Report; 95; 22.08.-26.08.94 (9434)

K. Birman, F. Cristian, F. Mattern, A. Schiper (editors):
Unifying Theory and Practice in Distributed Systems, Dagstuhl-Seminar-Report; 96; 05.09.-09.09.94 (9436)

R. Keil-Slawik, I. Wagner (editors):
Interdisciplinary Foundations of Systems Design and Evaluation, Dagstuhl-Seminar-Report; 97; 19.09.-23.09.94 (9438)

M. Broy, L. Lamport (editors):
Specification and Refinement of Reactive Systems - A Case Study, Dagstuhl-Seminar-Report; 98; 26.09.-30.09.94 (9439)

M. Jarke, P. Loucopoulos, J. Mylopoulos, A. Sutcliffe (editors):
System Requirements: Analysis, Management, and Exploitation, Dagstuhl-Seminar-Report; 99; 04.10.-07.10.94 (9440)

J. Buchmann, H. Niederreiter, A.M. Odlyzko, H.G. Zimmer (editors):
Algorithms and Number Theory, Dagstuhl-Seminar-Report; 100; 10.10.-14.10.94 (9441)

S. Heinrich, J. Traub, H. Wozniakowski (editors):
Algorithms and Complexity for Continuous Problems, Dagstuhl-Seminar-Report; 101; 17.10.-21.10.94 (9442)

H. Bunke, T. Kanade, H. Noltemeier (editors):
Environment Modelling and Motion Planning for Autonomous Robots, Dagstuhl-Seminar-Report; 102; 24.10.-28.10.94 (9443)

W. Maass, Ch. v.d. Malsburg, E. Sontag, I. Wegener (editors):
Neural Computing, Dagstuhl-Seminar-Report; 103; 07.11.-11.11.94 (9445)

G. Berry, W.P. de Roever, A. Poigné, A. Pnueli (editors):
Syncronous Languages, Dagstuhl-Seminar-Report; 104; 28.11.-02.12.94 (9448)