renzmechanismus nichts anderes als die mentale Implementierung eines aussagenlogischen Kalküls, eine 'mental logic' sei (vgl. Johnson-Laird 1983, 24 f.). 'Reasoning is not a matter of recovering the logical forms of the premises and then applying rules of inference to them in order to derive a conclusion [...] The heart of the process is interpreting premises as mental models that take general knowledge into account, and searching for counter-examples to conclusions by trying to construct alternative models of the premises.' ['Schlußfolgern ist nicht eine Sache des Aufdeckens der logischen Form von Prämissen, der dann folgenden Anwendung von Schlußregeln auf diese Prämissen, um dann eine Konklusion abzuleiten [...] Das Kernstück des Prozesses ist es, Prämissen als mentale Modelle, die Allgemeinwissen hinzuziehen, zu interpretieren und nach Gegenbeispielen für Konklusionen zu suchen, indem versuchsweise zu den Prämissen alternative Modelle konstruiert werden.'] (Johnson-Laird 1983, 54)

Die zentrale Rolle, die dem Schlußfolgern im Verstehensprozeß und damit auch bei sprachverstehenden Systemen zuzuweisen ist, und die Feststellung, daß dieses Schlußfolgern im Alltag mit unvollständigem und unsicherem, weil erfahrungsabhängigen Wissen umgeht, sollte — neben allen 'technischen' Problemem — erkennen lassen, daß bei der Erforschung und Entwicklung von sprachverstehenden Systemen Sprache nicht nur als Transportmittel für Information fungiert, sondern auch — wenn auch oft nicht wahrgenommen oder aber verdeckt — zur Verständigung(ssicherung) im Rahmen der Kommunikation über einen Sachverhalt (in einem sehr allgemeinen Sinne) dient. Sachverhalte werden formuliert nicht nur in Abhängigkeit von Vorstellungen und Einschätzungen bzgl. des Sachverhalts, sondern auch in Abhängigkeit von den Vorstellungen und Einschätzungen bzgl. dessen, dem gegenüber sie formuliert werden. Das heißt in letzter Konsequenz, daß ein System, das dies modellieren will, nicht nur in die Lage versetzt sein muß, Inferenzen rückgängig zu machen, es muß darüber hinaus in die Lage versetzt sein, Inferenzregeln zurückzuziehen, Inferenzschemata zu suspendieren, Prämissen zu verwerfen, Inferenzprozesse zu modifizieren, Inferenzstrategien zu ändern, sein Partnermodell zu revidieren, d. h., jede Art von Annahme zurückzuziehen bzw. zu korrigieren. Aber was heißt dies alles für die Konstruktion von Sprachverstehenssystemen und ihre Modellbeziehung zu einer Theorie sprachverstehender Systeme? 'That is the nature of many problems about the mind: we are so familiar with the outcome of its operations, which are for the most part highly successful, that we fail to see the mystery.' (Johnson-Laird 1983, X)

6. Literatur (in Auswahl)

G. L. Berry-Rogghe/M. Kolvenbach/H. D. Lutz 1980 · W. Bibel 1985 · E. Braun/H. Rademacher (ed.) 1978 · E. Charniak 1976 · E. Charniak/D. McDermott 1985 · M. G. Dyer 1983 · R. H. Granger 1980 · R. H. Granger/K. P. Eiselt/J. K. Holbrook 1983 · C. U. Habel 1980 · C. U. Habel 1982 a · C. U. Habel 1982 b · C. U. Habel 1983 · C. U. Habel 1984 a · C. U. Habel 1984 b · C. U. Habel 1984 c · C. U. Habel 1985 a · C. U. Habel/C.-R. Rollinger 1981 · P. N. Johnson-Laird 1983 · S. J. Kaplan 1983 · P. Klahr 1977 · P. Klahr/L. Travis/ Ch. Kellog 1980 · K. Morik 1984 · B. L. Nash-Webber 1978 · U. M. Quasthoff 1985 · Ch. J. Rieger 1974 · C.-R. Rollinger 1980 · C.-R. Rollinger 1983 b · C.-R. Rollinger 1984 b · C.-R. Rollinger 1984 c · M. Saluveer/H. Õim 1986 · R. C. Schank (ed.) 1975 · R. C. Schank 1982 a · R. C. Schank/ Ch. J. Rieger 1974 · A. Schmiedel 1984 · H.-J. Schneider/M. Eimermacher/S. Günther et al. 1981 · S.-O. Tergan 1986 · W. Wahlster 1981 · W. Wahlster 1982 b · M. G. Wessells 1984 · R. Wilensky 1983 · T. Winograd 1980.

Hans Dieter Lutz, Koblenz (Bundesrepublik Deutschland)

36. Language Generation Systems

It is gradually being recognized that language generation is a very intricate type of behavior, comparable in complexity with language understanding. The capabilities of computer-based language generators are rapidly expanding in response to intensified theoretical efforts and growing needs for intelligent human-computer interaction. My survey of this multi-faceted research area is divided into three parts. In the first two Sections I present an overview of prominent projects, ideas and systems which saw the light during the past two decades. Sections 3 and 4 concentrate on theory: which knowledge sources are needed, how can language generation be decomposed into subtasks (modules), and what are their interrelationships? Basing on the insights gained here I will then, in the final Section, work out some prospects and priorities for future research.

Not surprisingly, students of language generation borrow numerous concepts and techniques from other fields within computational linguistics and from other disciplines. In view of the limited space available I will occasionally refer the reader to other Chapters of this volume or to other recent handbooks (e. g. Winograd 1983) for further information. This applies in particular to computational and linguistic formalisms at word and sentence level, which I cannot review here.

1. Goals and Applications

Language generators have been designed to fulfill a variety of functions. The following seven categories cover most of them:

(1) Planning natural language utterances (phrases, sentences, texts) in human-computer dialogues.
(2) Planning connected discourse (e. g. narrative or expository texts).
(3) Psycholinguistic modeling (i. e. building computational models of human language production processes).
(4) Evaluating linguistic grammars (e. g. checking the consistency and completeness of a proposed set of syntactic rules).
(5) Composing target language texts in machine translation systems.
(6) Language and grammar teaching (e. g. programs which can paraphrase or otherwise transform sentences).
(7) Linguistic functions in wordprocessors (e. g. programs which automatically compose semi-standard documents).

The first two categories represent the core functions that language generators serve: to act as speaker or writer in goal-directed communication. Together with the third category, they embody a rich body of theoretical knowledge which can be put to a multitude of practical uses in the area of human-computer interaction (question-answering systems, natural-language interfaces, the dialogue component of expert systems) and in the categories numbered 4 through 7.

2. Projects and Capabilities

Fig. 36.1 lists 16 prominent language generation projects carried out over the past 15 years.

Authors	Name
1.Simmons & Slocum (1972)	
2.Goldman (1975)	BABEL
3.Meehan (1976)	TALESPIN
4.Davey (1979)	PROTEUS
5.McDonald (1980)	MUMBLE
6.Swartout (1981)	XPLAIN
7.Mann (1983)	PENMAN
8.Kukich (1983)	ANA
9.Sigurd (1983)	COMMEN-TATOR
10.Kempen & Hoenkamp (1984)	IPG
11.Busemann (1984)	SUTRA
12.Hovy (1985)	
13.Appelt (1985)	KAMP
14.McKeown (1985)	TEXT
15.Danlos (1985)	
16.Jacobs (1985)	KING

Fig. 36.1: Language generation systems

The selection represents the major research issues addressed during this period. It does not include generators running in systems for machine translation (e. g. Laubsch/Roesner/Hanakata et al. 1984; Ishizaki 1983), for grammar testing (Friedman 1969 a), or language teaching (Bates/Ingria 1981). Most systems produce written English text, sometimes adding limited second-language capabilities. Systems primarily intended for other languages than English were developed by Sigurd (1983, Swedish), Kempen/Hoenkamp (1984, Dutch), Busemann (1984, German) and Danlos (1985, French).

Very little attention has been given to spoken output. Simply feeding a generator's written output into a text-to-speech system is not a very attractive solution even if the speech quality obtained would be high. (See Müller, article 48 in this volume, for a survey of speech synthesis research.) One reason is that substantial parts of the generator's work will be redone during wave-form computation. For example, the text-to-speech converter will have to reconstruct the syntactic structure of an input sentence for the purpose of computing acceptable intonation contours and other prosodic features (Sigurd 1984; Danlos/Emerard 1985). Such redundancies can be avoided by establishing an intimate connection between language generator and speech synthesizer. Van Wijk/Kempen (1985 a) have designed and implemented an algorithm for computing intonation contours for Dutch sentences. It presupposes the IPG (Incremental Procedural Grammar) language generator.

Most of the 16 projects have resulted in fairly sophisticated generators in the sense that they can produce a variety of sentences on the basis of conceptual or perceptual input data (intentions, visual scenes, logical formulae, semantic representations). Section 6 illustrates their capabilities by means of some samples of machine-generated texts. The fragments stem from systems developed around 1982—83 but were mostly published in more recent years. In Fig. 36.2 I have indicated, for each of the projects, the theoretical issues that were central to the generator's design.

At the top of Fig. 36.2 we find systems which produced simple stories (Meehan, Hovy, Danlos) or running commentaries of ongoing events (Davey, Sigurd). Meehan's program was capable of planning interesting plots. Davey, Sigurd and Danlos used elementary discourse grammars to obtain well-organized narratives. Hovy worked out strategies for taking into account the storyteller's relationship to his/her audience (the hearer's knowledge, interests, sympathies and antipathies, emotional state, etc.).

Much effort has recently been put into the generation of expository texts. Swartout was concerned with the problem of how expert systems can explain and justify their reasoning as opposed to simply describing it. His solution hinges on a special knowledge representation framework capturing the principles and strategies behind the reasoning steps. Knowledge items of this type provide the conceptual contents for understandable and informative explanatory texts. Kukich wrote a program which extracts interesting facts from a daily stock quotes database and produces a stock market summary as output. Mann and McKeown concentrated on the rhetorical organization of expository discourse. They designed 'rhetorical schemas' governing structure (and sometimes content) of multisentential text. McDonald used somewhat similar 'text plans' but added a layer of stylistic decisions. These suggest 'points of attachment' for new content items and thereby exert certain control over their linguistic expression (McDonald/Pustejovsky 1985 a). The stylistic component pre-

Issues			Authors
Discourse	Narrative	Content Planning	Meehan '76
		Organization	Davey '79, Sigurd '83, Danlos '85
		Speaker-Hearer	Hovy '85
	Expository	Content Planning	Swartout '81, Kukich '83
		Rhetorical	Mann '83, McKeown '85
		Style	McDonald/Pustejovsky '85 a
Dialogue	Speech-act Planning		Appelt '85
	Speaker-Hearer		Wahlster et al. '83, Jameson '83
Sentence	Lexical & Grammatical Knowledge		Goldman '75, Jacobs '85
	Grammatical Formalism	ATN	Simmons/Slocum '72, Goldman '75
		Functional Unification	Appelt '85, McKeown '85, Jacobs '85
		Systemic	Davey '79, Mann '83
		Psycholinguistic	McDonald '80, Kempen/Hoenkamp '84

Fig. 36.2: Central issues addressed in various language generation projects

scribes, for example, whether a new content item will be realized as a separate main clause or embedded in another sentence at a designated position.

The dialogue situation imposes special requirements upon the generator's capacity to interact with conversational partners. Appelt developed an approach to language generation based on the assumption that speaker and hearer are agents cooperating to satisfy goals. Language behavior is viewed as controlled by the same general planning mechanism which also regulates non-linguistic actions by the dialogue partners. The language generator built by Appelt is driven by a multiple-agent planning system which can draw upon a repertoire of linguistic actions: speech acts (illocutionary acts, surface speech acts), concept activations (resulting in referential descriptions), and utterance acts (determining choice of words and syntactic constructions for realizing the descriptors). The planner takes into account the knowledge that is available to each of the agents, and their goals. Its plans may also involve a mixture of linguistic and non-linguistic (physical) actions, for example, an utterance act combined with pointing to an object.

Other aspects of speaker-hearer interaction were modeled in the HAM-ANS question-answering system (Wahlster/Marburger/Jameson et al. 1983; Jameson 1983). The generator constructed as part of this project (SUTRA; Busemann 1984) attempts to anticipate the hearer's reactions to a conceived answer and to expand or modify the overt answer accordingly. For instance, the generator avoids elliptical answers which might be ambiguous. If certain follow-up questions are expected, it tries to precompute their answers and to include them in the current answer. Jameson was specifically concerned with subjective speaker bias ('impression formation'). The system he built can volunteer unsolicited comments meant to influence the user's evaluation of objects under discussion. In order for the HAM-ANS generator to foresee the effects of its utterances upon the user it cooperates with the parser/understander. This component interprets answers proposed by the generator and evaluates them from the user's point of view. The conclusions reached in this 'anticipation feedback loop' form the basis for adjusting the generator's overt answer (Wahlster/Kobsa 1985; see also Habel/Pribbenow, article 57 in this volume, for a discussion of the prominent role the parser/understander has to play here).

The large majority of publications in the language generation literature deals with issues at the sentence level, i. e. with computing individual sentences and two- or three-sentence paragraphs. Various linguistic formalisms for representing and manipulating linguistic knowledge have been applied. Most popular were Systemic Grammar and Functional Grammar (see Winograd 1983 for detailed explanations). Two recent grammar formalisms are being explored but have not yet been incorporated into large-scale generators: Gazdar's Generalized Phrase Structure Grammar (GPSG; Gazdar/Klein/Pullum et al. 1985) and Joshi's Tree Adjoining Grammar (TAG; Vijay-Shankar/Joshi 1985; see also McDonald/Pustejovsky 1985 b). Were ATNs (Augmented Transition Networks) the dominating computational mechanism in the seventies, in the eighties they were superseded by Functional Unification Grammar, a formalism developed by Kay (1979; 1984 a). Other computational models of grammar were inspired by psycholinguistic considerations. (Fig. 36.2 lists the grammar types used by most projects in Fig. 36.1.)

A central problem — hardly addressed by theoretical linguists — is lexicalization: choosing words and idioms which (a) render the speaker's intentions and (b) fit into the prevailing context (syntactic, discourse, conversational). Human speakers are remarkably flexible and creative wordfinders, thanks to their capacity of viewing an event or a state of affairs from many different angles (cf. metaphors). For the computer to mimick such versatility depends only in part on size and organization of its lexicon. Even more critical is the structure of the semantic representations in which the speaker's intentions are couched. Two important studies — about 10 years apart — were carried out by Goldman (1975) and Jacobs (1985). The former concentrated on lexical paraphrases, the latter on metaphors and idiomatic expressions.

3. Knowledge Sources

The capabilities exhibited by present-day language generators are based upon a variety of knowledge and information sources. Fig. 36.3 shows 14 different categories. In this Section I will elaborate on this list, particularly on those items which I did not cite so far. This survey of knowledge sources also provides the opportunity to cover several important projects which examined individual aspects of the language generation process rather than building a full-scale generator.

Fig. 36.3:

A. Discourse Structure
   1. Domain Knowledge
   2. Intentional Structure
   3. Discourse Segment Structure
   4. Focusing Structure

B. Model of the Hearer
   1. Informational State
   2. Affective State
   3. Social Relationship

C. Grammatical Structure
   1. Semantics
   2. Lexicon
   3. Syntax
   4. Morpho-phonology
   5. Intonation/Punctuation
   6. Articulation/Orthography

Fig. 36.3: Knowledge and information sources needed in language generators

The division under heading A in Fig. 36.3 derives from recent work by Grosz/Sidner (1985). These authors lay out the foundations for a computational theory of discourse which builds on their extensive earlier studies of focusing in discourse and on task-structure in task-oriented dialogues. Discourse structure, according to Grosz/Sidner, consists of three interacting components: linguistic structure, intentional structure, and attentional state. The term linguistic structure, which applies to sequences of utterances rather than to single sentences, refers to the aggregation of utterances into *discourse segments*. Special linguistic devices are available to the speaker for marking boundaries between segments (e. g. phrases such as 'in the first place', or 'by the way', and more subtle cues such as a change of tense or of intonation). Discourse segments are not simply juxtaposed but enter into certain relationships to each other and to the overall discourse (cf. the relation between phrasal constituents and the sentence they belong to).

Associated with each discourse segment are a *discourse segment purpose* (DSP) and a *focus space* (FS). A DSP is an intention (of the speaker) which directly or indirectly helps attaining the purpose (intention) of the overall discourse (DP = *discourse purpose*). Some of Grosz/Sidner's examples are:

— intend that some agent intend to do some physical act (e. g. intend that Ruth intend to fix the flat tire);

— intend that some agent believe some fact (e. g. intend that Ruth believe the campfire is started).

There are no principled differences between DPs and DSPs.

A focus space is a collection of entities (objects, attributes, relations, intentions, etc.) which are related to the D(S)P of the current discourse segment. The D(S)P itself also belongs to the focus space. For example, if speaker intends that hearer believe some facts about John (e. g. his coming by and leaving the groceries), then the focus space contains at least the following items: the speaker's intention, John, groceries, and John's coming by. The discourse segment produced by the speaker might read 'John came by and left the groceries. I put them away'.

As the discourse unfolds, more and more focus spaces are subtended. However, most of these are quickly relegated to oblivion. The mechanism determining which FSs are saved and which are discarded, makes use of a push-down stack. Each time a new discourse segment is entered, its FS is pushed onto the stack. But first the relation between the new DSP and that of the preceding discourse segment is checked. If the new DSP contributes to (i. e. helps satisfy) the current top FS, the latter is allowed to stay on the stack (now in second position). If, on the other hand, the new DSP contributes to a D(S)P lower in the stack, all intermediate FSs are popped and discarded. A set of FSs saved on the stack constitutes the *focusing structure* of that point in the discourse. Entities forming part of an FS high on the stack are more salient and more readily available than entities occupying lower stack positions. Entities belonging to FSs which have been removed from the stack are inaccessible until being pushed onto the stack again.

The *attentional state,* which records the continually changing focusing structure as the discourse proceeds, controls important features of the linguistic structure. One specimen, already hinted at, is the insertion of special markers signaling discourse segment boundaries. Another example concerns anaphorical — in particular pronominal — references. The entities they refer to must belong to the current focusing structure, preferably to the most salient (topmost) FS. (Take for instance the interpretation of 'them' in the above example.) A further focus phenomenon has become known by the name *centering*. The center of an utterance is 'the single entity that the utterance most centrally concerns' (Grosz/Joshi/Weinstein 1983). Typically the center is realized by an NP. When going from one sentence to the next, the speaker may continue talking about the same entity or move the center to another entity in the focusing structure. There are fairly strict rules governing the movement of the center between consecutive sentences in a discourse segment. In addition, other rules constrain the choice of expressions referring to the centered entity. For example, if the center of two successive sentences is the same, then the second sentence preferably uses a pronoun to refer to it (see also Sidner 1983 a).

This brief and simplified outline of Grosz/Sidner's discourse theory provides some background for entries A2—A4 in Fig. 36.3. I have singled out knowledge about the domain under discussion (Al) as a separate knowledge source which can be consulted by the components responsible for planning intentional and discourse segment structure. Sophisticated knowledge representation systems in these areas make part of the KAMP, TEXT and PENMAN generators. TEXT and PENMAN make use of *rhetorical techniques,* i. e. more or less conventional plans for designing (aspects of) an intentional structure and its concomitant discourse segment structure. TEXT and KAMP contain rules controlling focus movements.

Entries B1—B3 in Fig. 36.3 name three knowledge sources needed in a model of the hearer. The hearer's informational state summarizes his/her current knowledge and the kinds of questions or interests which motivated him/her to engage in the conversation or to read a text (Wahlster/Marburger/Jameson et al. 1983, Bunt 1981). The affective state refers to hearer/reader's sympathies, antipathies, attitudes, feelings, etc. The social relationship between speaker and hearer concerns their relative status, their goals vis-à-vis each other, their beliefs about each other's goals, etc. (Jameson 1983, Hovy/Schank 1984). The development of *hearer modeling* components in language generation systems has barely begun. Work on user modeling — currently a very active research area — is likely to provide theoretical impulses (cf. Wahlster/Kobsa 1985).

The third rubrique in Fig. 36.3 lists six types of knowledge which hardly need further elaboration. They are relevant to computing individual sentence forms, spoken as well as written. Entry C1 (semantics) is intended to cover not only the representation of sentence meanings but also the selection of referential expressions (Granville 1984) and various determinants of style (influencing, for instance, choice of words and the combination of meaning units into more or less complex sentences; cf. McDonald/Pustejovski 1985 a).

The next Section addresses the problem of integrating such a heterogeneous collection of knowledge sources into the blueprint for a fluent language generator.

4. Architectures

When comparing the organization of the language generation process in the systems discussed here one discovers a remarkable variability. Beyond a global partitioning into a strategic and a tactical component (Thompson 1977) — the former determining 'what to say', the latter 'how to say it' — little similarity is discernible. And even if the modules used in two generators overlap to some degree, then the flow of control between them may be widely different. This section highlights four design issues which are being discussed in the literature.

Danlos (1985) argues that conceptual decisions (what information should be expressed, and in what order) and linguistic decisions (what lexical items to select, and what syntactic constructions) are strongly dependent upon each other. One-way traffic from a strategic to a tactical component will not lead to a sufficiently flexible and powerful generator. Similar views have been expressed by Appelt (1985 b) and Hovy (1985). Other researchers, notably McDonald (1983), Kempen/Hoenkamp (1984) and McKeown (1985), depart from the opposite standpoint. Their systems were built on the assumption that feedback from the tactical to the strategic component would be exception rather than rule.

An allied issue concerns homogeneity versus heterogeneity of the generator's architecture. The clearest example of a homogeneous generator design is KAMP: one general planning device takes care of both conceptual planning and shaping linguistic utterances. Reasoning about syntactic, lexical, etc., choices is fully integrated with planning communicative and physical actions. On the other hand, MUMBLE and TEXT exemplify heterogeneous designs because the workings of their strategic and tactical components are very different. *Within* the tactical component one finds a comparable situation. For instance, PENMAN has one general mechanism (based on Systemic Grammar) for making choices at the syntactic and morphological levels. Yet IPG has autonomous syntactic and morphological modules which operate according to very different principles. KING applies a uniform hierarchical knowledge representation system for encoding different types of knowledge (conceptual, syntactic, lexical, etc.). But IPG has at least two different representation schemes (one procedural, one declarative) even within the syntactic domain.

The homogeneity versus heterogeneity issue relates to J. Fodor's (1983) distinction between 'modular' and 'isotropic' cognitive systems. Modular systems consist of components whose modes of operation may be radically different. These components have no access to each other's inner workings (e. g. to informational structures used) and communicate through narrow input-output channels. In isotropic systems, however, the components are transparent and strongly interactive. In terms of this contrast one can characterize KAMP and KING as more isotropic and less modular than MUMBLE, TEXT and IPG. Fodor does not consider language generation directly but produces extensive empirical arguments in favor of the position that natural language *understanding* in humans is based on a modular system. This suggests that — to the extent that language generation shares modules with language understanding — a modular approach to the construction of a fluent language generator will be more profitable in the long run.

A third dividing line running across generator designs concerns *incremental generation* (Kempen/Hoenkamp 1982; 1984). The strategic and tactical components in human speakers operate in parallel rather than in sequence. Conceptual and linguistic structures are elaborated and refined simultaneously, and the production of overt output need not wait till the conceptual structure is rounded off by the strategic component. Utterances can thus be generated incrementally, in a piecemeal fashion. Most artificial generators, however, first have the strategic component compute the conceptual structure in full detail before activating the tactical component. Only IPG, MUMBLE and KAMP are capable of generating incrementally, be it in very different ways.

The last issue to be raised in this Section concerns *monitoring*. Very few generator programs enlist the services of a monitor or editor component evaluating utterances which have been or are about to be produced. The only exceptions are Gabriel (1981) and Wahlster/Marburger/Jameson et al. (1983; see the anticipation feedback loop discussed in 2.). This contrasts sharply with the detailed attention given to monitoring and self-correction in the psycholinguistic literature on language production (cf. Levelt 1983, Van Wijk/Kempen 1985 b). Nevertheless, the addition of a monitor may contribute to the solution of practical and theoretical problems significantly. Take for example the above issue of one-way versus two-way traffic between strategic and tactical components. Suppose the monitor can intercept the linguistic output from the tactical component (preferably before the point of speech) and feed it into a parser/understander. The latter evaluates the generator's utterances from relevant viewpoints and informs (via the monitor) the strategic component of its diagnosis (Hoenkamp 1980). This would establish the line of communication postulated by Danlos and others without complicating the generator's design — the parser is needed anyway.

In this Section we have touched upon some of today's active research topics. Let us now take a glance at tomorrow's trends.

5. Prospects and Priorities

Language generation research is in good shape. From all relevant scientific disciplines (linguistics, logic, psychology, phonetics, Artificial Intelligence, computer science) competent investigators are moving into the field. Their combined efforts will rapidly raise it to the standards of language understanding research. Practical applications in machine translation systems, natural language interfaces, intelligent word and text processors, language teaching programs, etc., are on the horizon.

What are high-priority research aims for the near future? At the risk of falling victim to my own biases I have tried to identify the most urgent needs:

— Topping the list is a computational model of discourse — along the lines of Grosz/Sidner (1985) or otherwise — embodied in an efficient implementation.

— At the sentence level high-performance syntactic, morphological and lexical processors are needed which can do their job many times faster that current versions.

— Practical applications presuppose a user modeling component optimizing the interplay between generator and hearer/reader.

— The relative merits of homogeneous versus heterogeneous architectures should be explored, preferably in conjunction with one or more of the preceding topics.

It will come as no surprise that these recommendations are not unlike a list compiled a few years ago by a group of American experts (Mann/Bates/Grosz et al. 1982). The most striking difference lies in my emphasis on performance factors versus their accentuating knowledge representation and structure. I hope this reflects an advance during the past period rather that a personal bias.

6. Output Samples

From six projects I have selected brief output samples which illustrate the capabilities of the generators. The first four samples are in English, the other ones in French and German respectively. The texts represent the state of the art in 1982—1983. Samples published later were hardly available at the time of writing (December 1985).

(1) TEXT (McKeown 1985) was implemented as part of a natural language interface to an ONR (Office of Naval Research) database containing information about vehicles and destructive devices.

Question: *What is a guided missile?*

Answer: *A guided projectile is a projectile that is self-propelled. There are 2 types of guided projectiles in the ONR database: torpedoes and missiles: The missile has a target location in the air or on the earth's surface. The torpedo has an underwater target location. The missile's target location is indicated by the DB attribute DESCRIPTION and the missile's flight capabilities are provided by the DB attribute ALTITUDE. The torpedo's underwater capabilities are provided by the DB attributes under DEPTH (for example, MAXIMUM OPERATING DEPTH). The guided projectile has DB attributes TIME TO TARGET & UNITS, HORZ RANGE & UNITS and NAME.*

Question: *What is the difference between a destroyer and a bomb?*

Answer: *A destroyer is a surface ship with a DRAFT between 15 and 222. A ship is a vehicle. A bomb is a free falling projectile that has a surface target location. A free falling projectile is a lethal destructive device. The bomb and the destroyer, therefore, are very different kinds of entities.*

(2) MUMBLE (McDonald 1983) generated the following description of the Barber Proof.

line 1:	premise $\exists x\ (barber(x) \wedge \forall y(shaves(x,y) \leftrightarrow \neg shaves\ (x,y)))$
line 2:	existential instantiation $barber(g) \wedge \forall y(shaves(g,y) \leftrightarrow \neg shaves(y,y))$
line 3:	tautology $\forall y\ (shaves(g,y) \leftrightarrow \neg shaves(y,y))$
line 4:	universal instantiation $shaves(g,g) \leftrightarrow \neg shaves(g,g)$
line 5:	tautology $shaves(g,g) \wedge \neg shaves(g,g)$
line 6:	conditionalization $\exists x\ (barber(x) \wedge \forall y(shaves(x,y) \leftrightarrow \neg shaves(y,y))) \supset (shaves(g,g) \wedge \neg shaves(g,g))$
line 7:	reductio ad absurdum $\neg \exists x\ (barber(x) \wedge \forall y(shaves(x,y) \leftrightarrow \neg shaves(y,y)))$

MUMBLE's version:

Assume that there is some barber who shaves everyone who doesn't shave himself (and no one else). Call him Giuseppe. Now, anyone who doesn't shave himself would be shaved by Giuseppe. This would include Giuseppe himself. That is, he would shave himself, if and only if he did not shave himself, which is a contradiction. This means that the assumption leads to a contradiction. Therefore, it is false, there is no such barber.

Compare this text to the version created by an early generation program written by Chester (1976):

Suppose that there is some barber such that for every person the barber shaves the person iff the person does not shave himself. Let A denote such a barber. Now he shaves himself iff he does not shave himself, therefore a contradiction follows. Therefore if there is some barber such that for every person the barber shaves the person iff the person does not shave himself then a contradiction follows. Thus there is no barber such that for every person the barber *shaves the person iff the person does not shave himself.*

(3) Hovy's generation system can produce different versions of a story in response to the speaker's goals and to characteristics of the hearer. The following story about IRA terrorism in Northern Ireland was told to two characters, a British soldier (John Brown) and an IRA terrorist (Sean O'Flanahan).

O'Flanahan:	interests: British losses
	sympathies: IRA successes
	antipathies: anything British, authorities
	emotion: tense, angry
	knowledge: terrorism in detail
Brown:	interests: weaponry
	sympathies: British successes, authorities
	antipathies: IRA successes
	emotion: tense, irritated
	knowledge: military action in detail

The four versions were published, along with a few others, in Hovy/Schank (1984).

Hearer is: (John Brown)
Input concepts are: (SHOOTING1)
Concentrating on: INTERESTS

A seventy calibre semi-automatic Belgian rifle was used by two gunmen to shoot and to kill a forty year old British soldier from Leeds, a city, Bill Smith. He will be buried tomorrow. The rifle was used by them to shoot and to wound a twelve year old girl, Mary, yesterday at Belfast Football Stadium. She was taken to Mary Magdalene Hospital and treated and discharged. She is an orphan at present. The rifle was used by them to shoot and to kill a twenty-five year old passerby woman, Rosie Areas. She will be buried in Belfast Cemetery three days from now. An Austin car was used by them to escape.

Hearer is: (John Brown)
Input concepts are: (SHOOTING1)
Concentrating on: SYMPATHY

A forty year old Englishman from Leeds, a city, Bill Smith, was shot and killed with a seventy calibre rifle at Belfast Football Stadium yesterday. The soldier will be buried tomorrow. A twelve year old girl, Mary, was shot and wounded. She was taken to Mary Magdalene Hospital and treated and discharged. She is an orphan at present. A twenty-five year old passerby woman, Rosie Areas, was shot and killed. She will be buried in Belfast Cemetery three days from now. An Austin car was used by two gunmen to escape.

Hearer is: (Sean O'Flanahan)
Input concepts are: (SHOOTING1)
Concentrating on: INTERESTS

A forty year old British soldier from Leeds, a city, Bill Smith, was shot and killed with a seventy calibre rifle at Belfast Football Stadium yesterday. He will be buried tomorrow. Two freedom-fighters shot and wounded a twelve year old girl, Mary. She was taken to Mary Magdalene Hospital and treated and discharged. She is an orphan at present. A twenty-five year old passerby woman, Rosie Areas, was shot and killed. She will be buried in Belfast Cemetery three days from now. They escaped with an Austin car.

Hearer is: (Sean O'Flanahan)
Input concepts are: (SHOOTING1)
Concentrating on: SYMPATHY

An Austin car was used by two freedom-fighters to escape from Belfast Football Stadium yesterday. A twelve year old girl, Mary, was shot and wounded with a seventy calibre rifle. She was taken to Mary Magdalene Hospital and treated and discharged. She is an orphan at present. A forty year old British soldier from Leeds, a city, Bill Smith, was shot and killed. He will be buried tomorrow. A twenty-five year old passerby woman, Rosie Areas, was shot and killed. She will be buried in Belfast Cemetery three days from now.

(4) ANA (Kukich 1983 a) generated the following stock market summary (2 versions) in response to actual data submitted for January 12, 1983.

DOW JONES INDUSTRIALS AVERAGE — 01/12/83

01/12 CLOSE	*30*	*INDUS*	*1083.61*
01/12 330PM	*30*	*INDUS*	*1089.40*
01/12 3PM	*30*	*INDUS*	*1093.44*
01/12 230PM	*30*	*INDUS*	*1100.07*
01/12 2PM	*30*	*INDUS*	*1095.38*
01/12 130PM	*30*	*INDUS*	*1095.75*
01/12 1PM	*30*	*INDUS*	*1095.84*
01/12 1230PM	*30*	*INDUS*	*1095.75*
01/12 NOON	*30*	*INDUS*	*1092.35*
01/12 1130AM	*30*	*INDUS*	*1089.40*
01/12 11AM	*30*	*INDUS*	*1085.08*
01/12 1030AM	*30*	*INDUS*	*1085.36*
01/11 CLOSE	*30*	*INDUS*	*1083.79*

CLOSING AVERAGE 1083.61 DOWN 0.18

Version 1:

After climbing steadily through most of the morning, the stock market was pushed downhill late in the day. Stock prices posted a small *loss, with the indexes turning in a mixed showing yesterday in brisk trading.*

The Dow Jones average of 30 industrials surrendered a 16.28 gain at 4pm and declined slightly, finishing the day at 1083.61, off 0.18 points.

Version 2:

Wall Street's securities markets rose steadily through most of the morning, before sliding downhill late in the day. The stock market posted a small loss yesterday, with the indexes finishing with mixed results in active trading.

The Dow Jones average of 30 industrials surrendered a 16.28 gain at 4pm and declined slightly, to finish at 1083.61, off 0.18 points.

(5) The French language generator implemented by Danlos (1985) produced five versions of a newspaper story.

Version 1: *Aujourd'hui à Paris, un policier a été tué et quatre autres blessés par des anarchistes qui ont fait exploser une bombe télécommandée sous le camion dans lequel ils allaient de leur bureau à un restaurant. La bombe contenait deux kilos de dynamite.*

Version 2: *Un policier a été tué et quatre autres blessés aujourd'hui à Paris. Des anarchistes ont fait exploser une bombe télécommandée sous le camion dans lequel ils allaient du bureau à un restaurant. La bombe contenait deux kilos de dynamite.*

Version 3: *Un policier a été tué et quatre autres blessés par des anarchistes aujourd'hui à Paris. Ils ont fait exploser une bombe télécommandée sous le camion dans lequel ils allaient du bureau à un restaurant. La bombe contenait deux kilos de dynamite.*

Version 4: *Aujourd'hui à Paris, des anarchistes ont tué un policier et en ont blessé quatre autres en faisant exploser une bombe télécommandée sous le camion dans lequel ils allaient du bureau à un restaurant. La bombe contenait deux kilos de dynamite.*

Version 5: Des anarchistes ont tué un policier et en ont blessé quatre autres aujourd'hui à Paris. Ils ont fait exploser une bombe télécommandée sous le camion dans lequel ils allaient du bureau à un restaurant. La bombe contenait deux kilos de dynamite.

(6) Busemann's (1984) German language generator SUTRA can deliver sentences such as the following. (They are unrelated, except for the first pair: pronominalisation.) Chess is the content domain.

— *Der Grossmeister gab dem Gegner ungewollt einen grossen Zeitvorteil, weil er den Partiebeginn verschlafen hatte.*

— *Hat er ihn ihm wirklich unfreiwillig gegeben?*

— *Der gefesselte Springer auf der langen Diagonale wird von der gegnerischen Dame erbeutet werden.*

— *Der gespielte Zug gewann am einfachsten, obwohl elegantere Möglichkeiten in der Stellung steckten.*

—

Er behauptete nach der Niederlage, dass die Zeitnot die Ursache des Debakels gewesen sei.

7. Literature (selected)

D. E. Appelt 1985 b · M. Bates/R. Ingria 1981 · H. C. Bunt 1981 · St. Busemann 1984 · D. Chester 1976 · L. Danlos 1985 · L. L. Danlos/F. Emerard 1985 · A. Davey 1979 · J. A. Fodor 1983 · J. Friedmann 1969 · J. Friedmann 1971 · R. P. Gabriel 1981 · G. Gazdar/E. Klein/G. Pullum et al. 1985 · N. M. Goldman 1975 · R. Granville 1984 · B. J. Grosz/A. K. Joshi/S. Weinstein 1983 · B. J. Grosz/C. L. Sidner 1985 · E. Hoenkamp 1980 · E. Hovy 1985 · E. H. Hovy/R. C. Schank 1984 · Sh. Ishizaki 1983 · P. S. Jacobs 1985 · A. Jameson 1983 · M. Kay 1979 · M. Kay 1984 a · G. Kempen/E. Hoenkamp 1982 · G. Kempen/E. Hoenkamp 1984 · K. Kukich 1983 a · J. Laubsch/D. Roesner/K. Hanakata et al. 1984 · W. Levelt 1983 · W. C. Mann 1983 · W. C. Mann/M. Bates/B. J. Grosz et al. 1982 · D. D. McDonald 1980 · D. D. McDonald 1983 · D. D. McDonald/J. D. Pustejovsky 1985 a · D. D. McDonald/J. D. Pustejovsky 1985 b · K. R. Mckeown 1985 · J. R. Meehan 1976 · C. L. Sidner 1983 a · B. Sigurd 1983 · B. Sigurd 1984 · R. F. Simmons/J. Slocum 1972 · W. Swartout 1981 · H. Thompson 1977 · K. Vijay-Shankar/A. K. Joshi 1985 · C. van Wijk/G. Kempen 1985 a · C. van Wijk/G. Kempen 1985 b · W. Wahlster/A. Kobsa 1985 · W. Wahlster/H. Marburger/A. Jameson et al. 1983 · T. Winograd 1983.

Gerard Kempen, Nijmegen (The Netherlands)