In the beginning there was an Intelligent Query Interface. . .
. . . that needed Natural Language Support. (Part II)

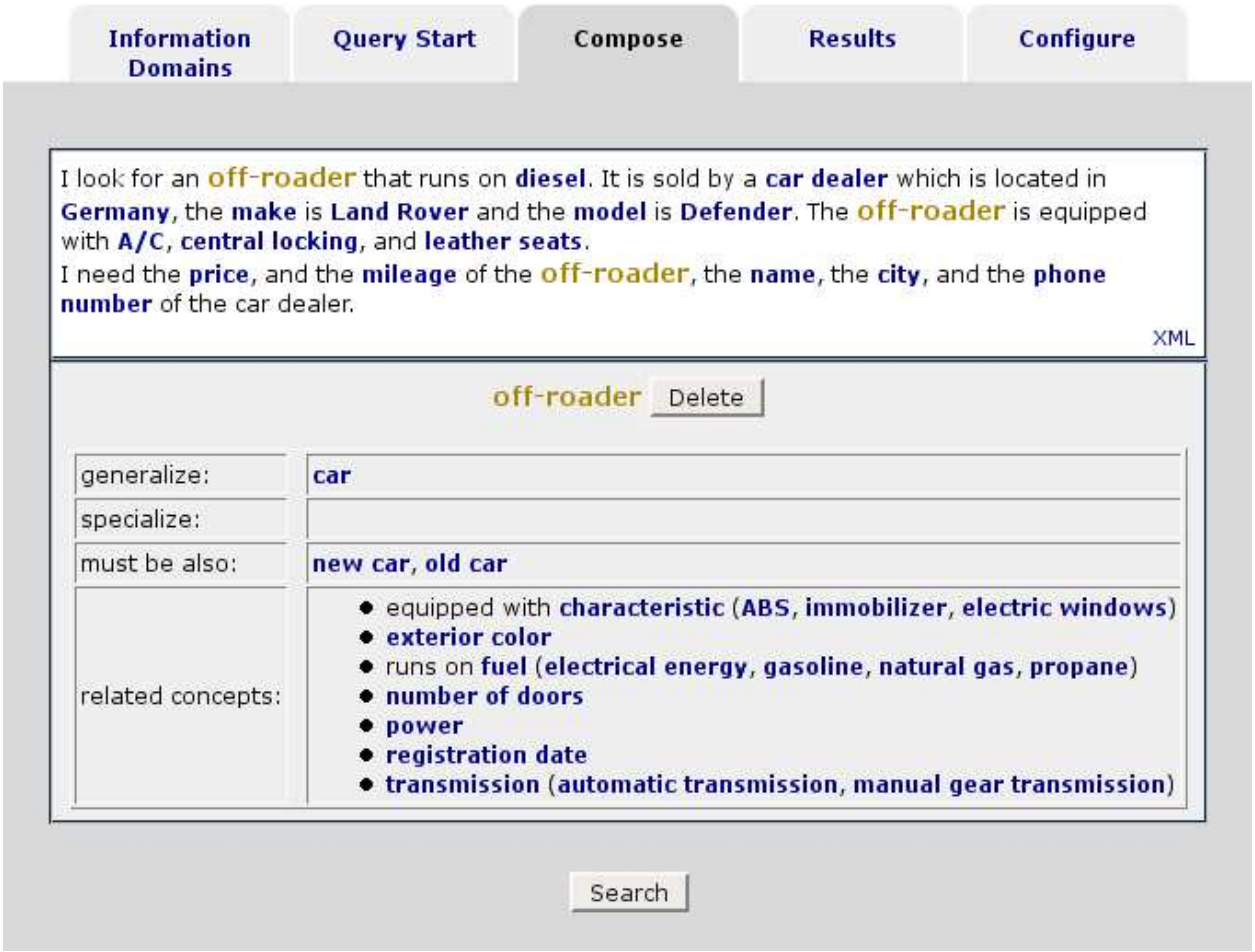
Paolo Dongilli

KRDB Centre
Faculty of Computer Science
Free University of Bozen-Bolzano, Italy

KRDB Seminar – May 24, 2006

Outline

The New Query Interface



What's Natural Language Generation?

- A subfield of Artificial Intelligence and Computational Linguistics concerned with building computer software systems that can produce meaningful texts in one or more human languages from some underlying nonlinguistic representation of information.
- NLG Systems use knowledge about language and the application domain to automatically produce documents, reports, help messages and other kinds of text.

What's Natural Language Generation?

- A subfield of Artificial Intelligence and Computational Linguistics concerned with building computer software systems that can produce meaningful texts in one or more human languages from some underlying nonlinguistic representation of information.
- NLG Systems use knowledge about language and the application domain to automatically produce documents, reports, help messages and other kinds of text.

Tasks of Natural Language Generation

- Text Planning
  - Content determination
  - Discourse structuring (e.g. RST [Mann and Thompson, 1988])

  $\Longrightarrow$ Text Plan
- Sentence Planning
  - Lexicalization
  - Aggregation
  - Referring expression generation

  $\Longrightarrow$ Sentence Plan
- Text Realization

  $\Longrightarrow$ Sequence of words

Tasks of Natural Language Generation

- Text Planning
  - Content determination
  - Discourse structuring (e.g. RST [Mann and Thompson, 1988])

  $\Longrightarrow$ Text Plan
- Sentence Planning
  - Lexicalization
  - Aggregation
  - Referring expression generation

  $\Longrightarrow$ Sentence Plan
- Text Realization

  $\Longrightarrow$ Sequence of words

Tasks of Natural Language Generation

- Text Planning
  - Content determination
  - Discourse structuring (e.g. RST [Mann and Thompson, 1988])

  $\Longrightarrow$ Text Plan
- Sentence Planning
  - Lexicalization
  - Aggregation
  - Referring expression generation

  $\Longrightarrow$ Sentence Plan
- Text Realization

  $\Longrightarrow$ Sequence of words

Approaches to Text Realization

Four basic approaches proposed by [Hovy, 1997]:

1. Canned Text
2. Template-Based Realization
3. Phrase-Based Realization
4. Feature-Based Realization

Approaches to Text Realization

Four basic approaches proposed by [Hovy, 1997]:

1. Canned Text
2. Template-Based Realization
3. Phrase-Based Realization
4. Feature-Based Realization

Approaches to Text Realization

Four basic approaches proposed by [Hovy, 1997]:

1. Canned Text
2. Template-Based Realization
3. Phrase-Based Realization
4. Feature-Based Realization

Approaches to Text Realization

Four basic approaches proposed by [Hovy, 1997]:

1. Canned Text
2. Template-Based Realization
3. Phrase-Based Realization
4. Feature-Based Realization

Approaches to Text Realization

Four basic approaches proposed by [Hovy, 1997]:

1. Canned Text
2. Template-Based Realization
3. Phrase-Based Realization
4. Feature-Based Realization

Linguistic Formalisms for Realization

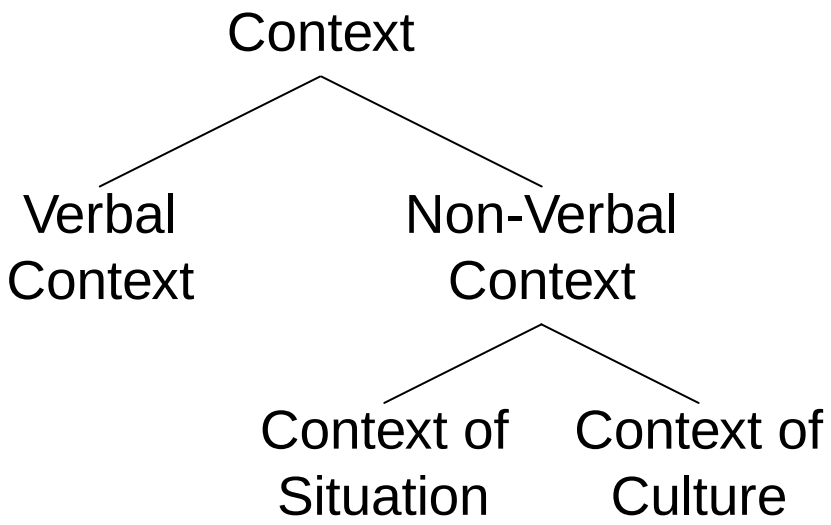
- Systemic Functional Grammar (SFG)
- Meaning-Text-Theory (MTT)
- Functional Unification Grammar (FUG), progenitor of the Functional Unification Formalism (FUF)
- Tree Adjoining Grammars (TAG)
- Categorial Unification Grammar
- Lexical Functional Grammar
- Government and Binding Theory
- Head-driven Phrase Structure Grammar (HPSG)

Origins of SFL

- Systemic-Functional Linguistics (SFL) is a major linguistics theory

Origins of SFL (Malinowski)

- Rooted in anthropology (Bronislaw Kasper Malinowski, 1935)
- important contributions to early modern linguistics from an anthropological perspective: meaning as function in context



Origins of SFL (Firth)

- Earliest formulation of SFL as linguistic teory dates back to John Rupert Firth (1957), founder of modern British linguistics and the first Professor of General Linguistics in the UK
- Firth disagreed with the American structuralists of his time (led by Bloomfield), because they were concerned only with the “anatomy” of language.

Origins of SFL (Firth)

- need for linguistics to give equal importance to both the "anatomy" and "physiology"

anatomy	**physiology**
chain	choice
syntagmatic	paradigmatic
structural	systemic
formal	functional
logical	rhetorical

Origins of SFL (Halliday)

- Further developed by Michael A. K. Halliday since 1961
- Michael Halliday, Firth's pupil and successor at London, disagreed with the American formalists (led by Chomsky).
- Birth of "Neo-Firthian linguistics", or the "London school of linguistics"
- Halliday developed a systematic and comprehensive theory of language, with a new terminology of its own. This theory, expounded in Halliday's many publications, became known as Systemic Functional Grammar.

Systemic-Functional Grammar

Why systemic? Why functional?

- systemic because of his development of detailed system networks for many areas of the English grammar and of other languages' grammars.
- functional because of his development of the theory of the ideational, interpersonal and textual metafunctions.

System networks

A system consists of an entry condition and a set of output features.

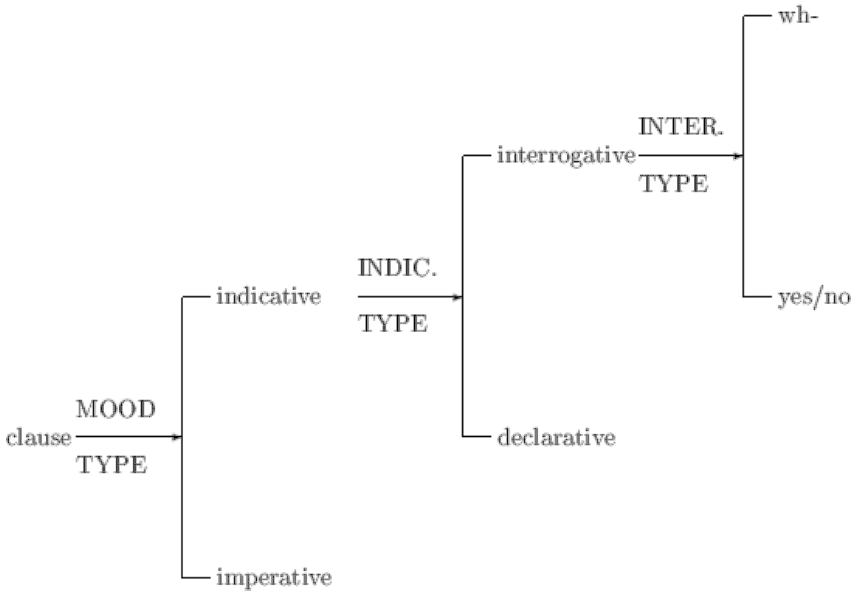


Figure: A fragment of a system network diagram

System networks

More than one system may share the same entry condition.

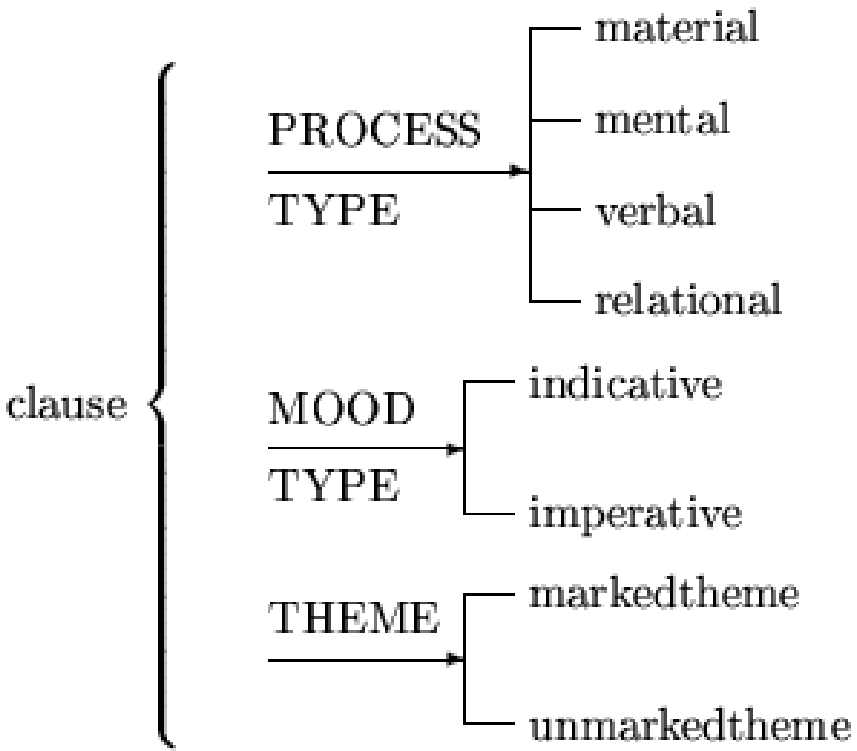


Figure: Simultaneous systems

Syntax and Semantics of a System Network (1/4)

$a \longrightarrow \left[\begin{array}{l} x \\ y \end{array} \right.$	system if a, then x or y (abbreviated as $a : x \vert y$)
$\left. \begin{array}{l} a \\ b \end{array} \right] \longrightarrow \left[\begin{array}{l} x \\ y \end{array} \right.$	disjunction in entry condition if $a \vert b$, then $x \vert y$

Syntax and Semantics of a System Network (2/4)

$\left.\begin{matrix} a \\ b \end{matrix}\right\} \longrightarrow \left[\begin{matrix} x \\ y \end{matrix}\right.$	conjunction in entry condition if a and b (abbreviated as $a\&b$), then $x \mid y$
$a \left\{\begin{matrix} \longrightarrow \left[\begin{matrix} x \\ y \end{matrix}\right. \\ \longrightarrow \left[\begin{matrix} m \\ n \end{matrix}\right. \end{matrix}\right.$	simultaneity if a, then simultaneously $x \mid y$ and $m \mid n$

Syntax and Semantics of a System Network (3/4)

$a \longrightarrow \left[\begin{array}{l} x \longrightarrow \left[\begin{array}{l} m \\ n \end{array} \right. \\ y \end{array} \right.$	delicacy ordering if a, then $x \vert y$; if x then $m \vert n$
$a \left\{ \begin{array}{l} \longrightarrow \left[\begin{array}{l} x * \longrightarrow \\ y \end{array} \right. \\ \longrightarrow \left[\begin{array}{l} m \longrightarrow * \\ n \end{array} \right. \end{array} \right.$	conditional marking if x, then also m

Syntax and Semantics of a System Network (4/4)

$a \left\{ \begin{array}{ll} \longrightarrow & \left[\begin{array}{l} y \\ x \end{array} \right\} m \\ \longrightarrow & \left[\begin{array}{l} f \\ n \end{array} \right. \end{array} \right.$	gate (one choice only) if x and f, then m
$a \left\{ \begin{array}{ll} \longrightarrow & \left[\begin{array}{l} x \\ y \end{array} \right. \\ \longrightarrow & \left[\begin{array}{l} // \\ \text{go on} \end{array} \right. \end{array} \right.$	recursive system (logical) if a , then $x \vert y$ and simultaneous option of entering and selecting from the same system again; // = stop

Realization

Realization rules show how the paradigmatic choices in the systems are expressed as syntagmatic chains in the structures of the language.

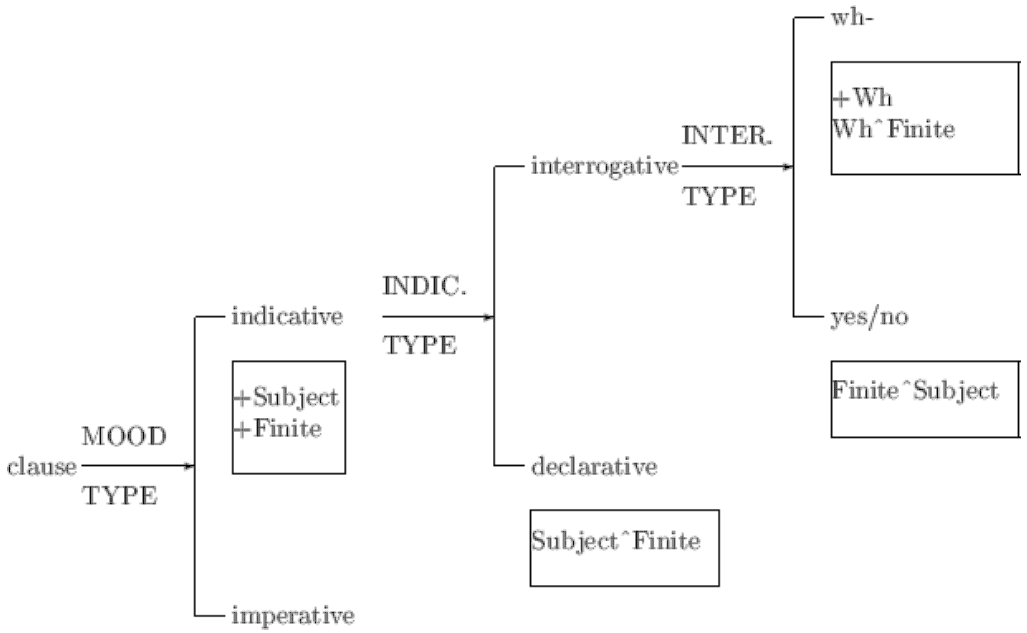


Figure: System network diagram with realization rules

Realization Statements in SFG

Name	Notation	Description
insert	`+Subject`	Function inserted as constituent of the structure of the unit being specified
order	`Subject` $\wedge$ `Finite`	One function ordered to precede another
expand	`Mood (Finite)`	One function expanded to have another function as constituent
conflate	`Subject / Agent`	One function conflated with another one to form the same constituent together
preselect	`Subject: singular`	A function preselected for a feature; the realization of the function is constrained to display that feature

Metafunctions

Halliday analyzed lexicogrammar into three broad metafunctions:

- ideational metafunction is about the natural world in the broadest sense, including our own consciousness, and is concerned with clauses as representations.
- interpersonal metafunction is about the social world, especially the relationship between speaker and hearer, and is concerned with clauses as exchanges.
- textual metafunction is about the verbal world, especially the flow of information in a text, and is concerned with clauses as messages.

Metafunctions

Halliday analyzed lexicogrammar into three broad metafunctions:

- ideational metafunction is about the natural world in the broadest sense, including our own consciousness, and is concerned with clauses as representations.
- interpersonal metafunction is about the social world, especially the relationship between speaker and hearer, and is concerned with clauses as exchanges.
- textual metafunction is about the verbal world, especially the flow of information in a text, and is concerned with clauses as messages.

Metafunctions

Halliday analyzed lexicogrammar into three broad metafunctions:

- ideational metafunction is about the natural world in the broadest sense, including our own consciousness, and is concerned with clauses as representations.
- interpersonal metafunction is about the social world, especially the relationship between speaker and hearer, and is concerned with clauses as exchanges.
- textual metafunction is about the verbal world, especially the flow of information in a text, and is concerned with clauses as messages.

Metafunctions

Halliday analyzed lexicogrammar into three broad metafunctions:

- ideational metafunction is about the natural world in the broadest sense, including our own consciousness, and is concerned with clauses as representations.
- interpersonal metafunction is about the social world, especially the relationship between speaker and hearer, and is concerned with clauses as exchanges.
- textual metafunction is about the verbal world, especially the flow of information in a text, and is concerned with clauses as messages.

Metafunctional Layering

Example taken from [Matthiessen and Bateman, 1991]:

In this job	Anne	we	're	working	with silver	
Theme		Rheme				textual
		Mood				interpersonal
	Vocative	Subject	Finite			
Locative		Actor	Process		Manner	ideational

Figure: Metafunctional layering

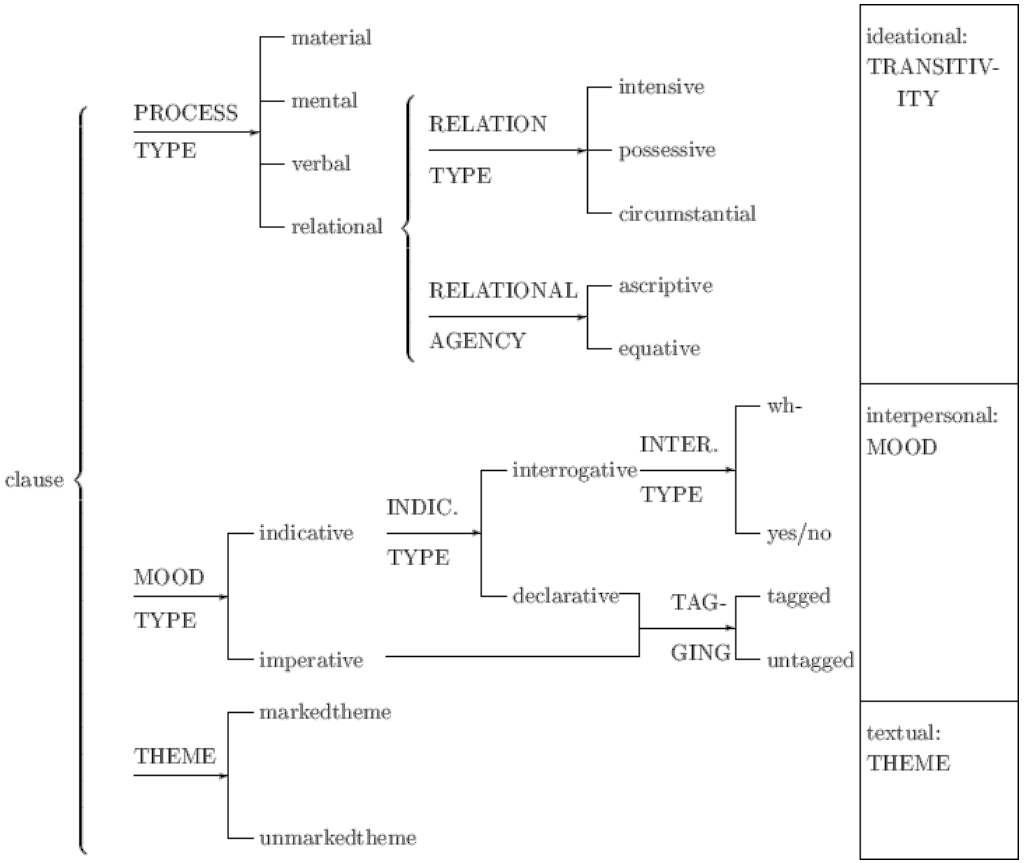
Principal Systems

	ideational	interpersonal	textual
clause	TRANSITIVITY	MOOD	THEME
verbal group	TENSE	MODALITY	VOICE
nominal group	MODIFICATION	PERSON	DETERMINATION

Figure: Principal Systems

[Matthiessen and Bateman, 1991]

Independence of Metafunctions



[Matthiessen and Bateman, 1991]

The Nigel Grammar of English (1/2)

- implementation of the SFG of English
- contains over 600 systems
- starts with the RANK system which selects from the rank scale:
  - clause
  - group/phrase
  - word
  - morpheme

The Nigel Grammar of English (2/2)

- clause complexes are also handled by this SFG (*paratactic* and *hypotactic* relations between two clauses)
- higher-level and more complex structures are handled outside Nigel by systems implementing RST (Rhetorical Structure Theory) [Mann and Thompson, 1988]
  [Taboada and Mann, 2006 (1)]
  [Taboada and Mann, 2006 (2)]

USC Nigel/Penman system

- example of a fully-developed SFG-based generation system
- developed at the University of Southern California Information Sciences Institute (USC/ISI) (1980's)
- by Bill Mann, with major contributions by Christian Matthiessen, Sandra Thompson, John Bateman, Robert Kasper, Eduard Hovy and others.
- personal participation of Halliday in the development of the Nigel systemic grammar

KPML System

- KPML (KOMET-Penman multilingual)
- development environment for developing and maintaining large-scale sets of multilingual SFL grammars
- multilingual text generation system
- successor of the Nigel/Penman system

The Chooser/Inquiry Interface

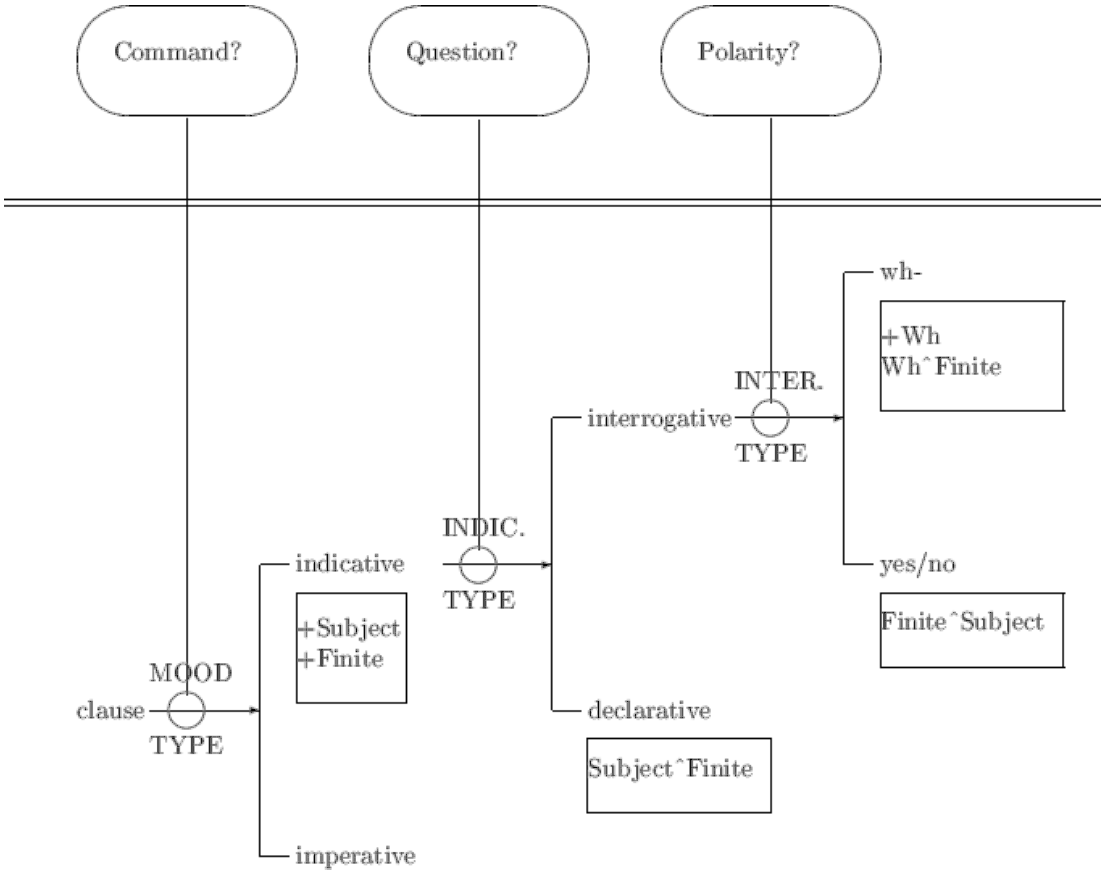


Figure: System network diagram with choosers and inquiries

Inquiry Semantics

- developed by William C. Mann (see [Mann, 1983])
- a choice must be made in each disjunctive choice system during grammar network traversal
- each choice system has an associated procedure called its chooser, which traverses a decision tree from its root to a single leaf node.
- each branching node has an associated inquiry, which obtains information from the external environment in which the grammar is embedded
- the chooser then selects which branch to take according to the reponse to the inquiry

The Chooser/Inquiry Interface

```
(system
   :name MOOD-TYPE
   :inputs CLAUSE
   :outputs ((INDICATIVE (INSERT SUBJECT) (INSERT FINITE))
             (IMPERATIVE))
   :chooser MOOD-TYPE-CHOOSER
   :region MOOD
   :metafunction INTERPERSONAL)

(system
   :name INDICATIVE-TYPE
   :inputs INDICATIVE
   :outputs ((DECLARATIVE (ORDER SUBJECT FINITE))
             (INTERROGATIVE))
   :chooser INDICATIVE-TYPE-CHOOSER
   :region MOOD
   :metafunction INTERPERSONAL)

(system
   :name INTERROGATIVE-TYPE
   :inputs INTERROGATIVE
   :outputs ((YES/NO (ORDER FINITE SUBJECT))
             (WH (INSERT WH) (ORDER WH FINITE)))
   :chooser INTERROGATIVE-TYPE-CHOOSER
   :region MOOD
   :metafunction INTERPERSONAL)
```

... two more ingredients missing

1. the Upper Model
2. an interface between application and generator

... two more ingredients missing

1. the Upper Model
2. an interface between application and generator

... two more ingredients missing

1. the Upper Model
2. an interface between application and generator

The Upper Model

- a particular domain model based on an ideational grammatical semantic typology for English
- a linguistically motivated ontology
- it reflects English lexical semantics (in fact, English lexicogrammatical semantics)
- it reflects the ideational metafunction, and it is called ideation base

SPL

- SPL (Sentence Plan Language) represents an interface between the application and the generator [Kasper, 1989]
- SPL input includes not only ideational content but also interpersonal and textual specifications.
- ```
  (p1 / class-ascription
      :domain (A2 / adder
                  :identifiability-q identifiable)
      :range (B1 / binary-operator
                  :identifiability-q notidentifiable))
  ```

  $\longrightarrow$ *The adder is a binary operator.*
  ```

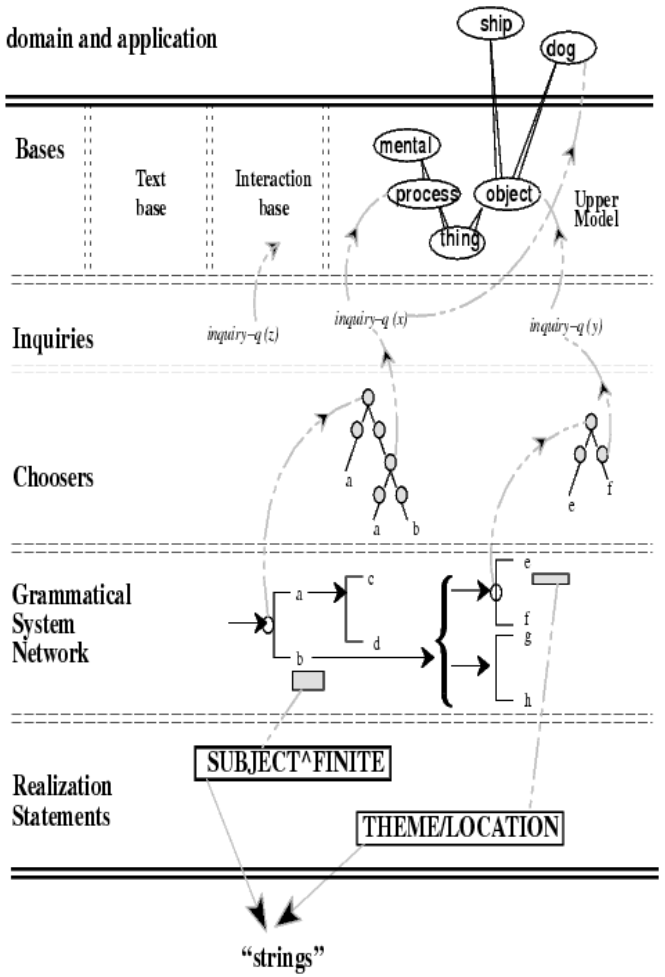
SPL

- SPL (Sentence Plan Language) represents an interface between the application and the generator [Kasper, 1989]
- SPL input includes not only ideational content but also interpersonal and textual specifications.
- ```
  (p1 / class-ascription
      :domain (A2 / adder
                  :identifiability-q identifiable)
      :range (B1 / binary-operator
                  :identifiability-q notidentifiable))
  ```
  $\longrightarrow$ *The adder is a binary operator.*
  ```

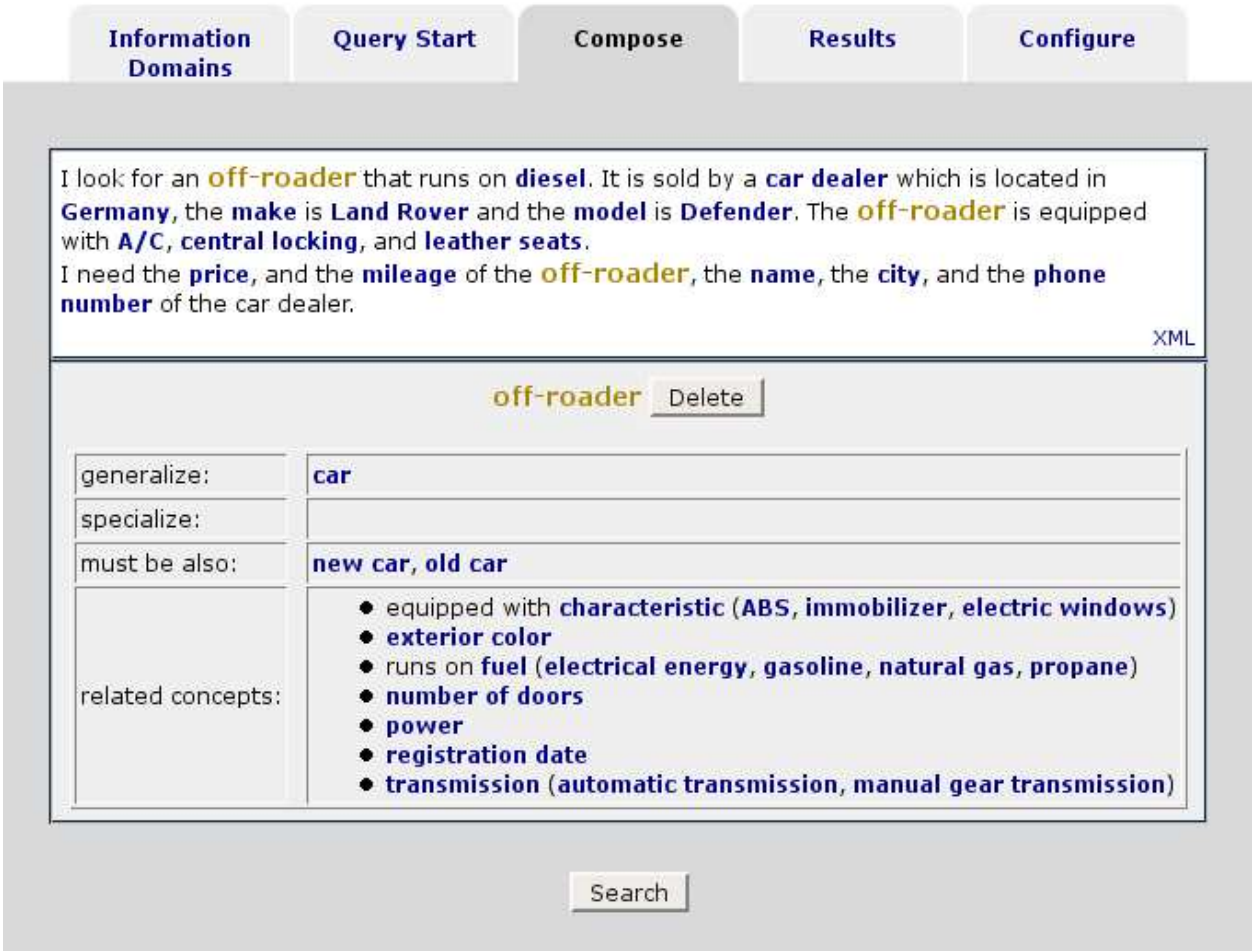
SPL

- SPL (Sentence Plan Language) represents an interface between the application and the generator [Kasper, 1989]
- SPL input includes not only ideational content but also interpersonal and textual specifications.
- ```
  (p1 / class-ascription
      :domain (A2 / adder
                  :identifiability-q identifiable)
      :range (B1 / binary-operator
                  :identifiability-q notidentifiable))
  ```
  $\longrightarrow$ *The adder is a binary operator*.
  ```

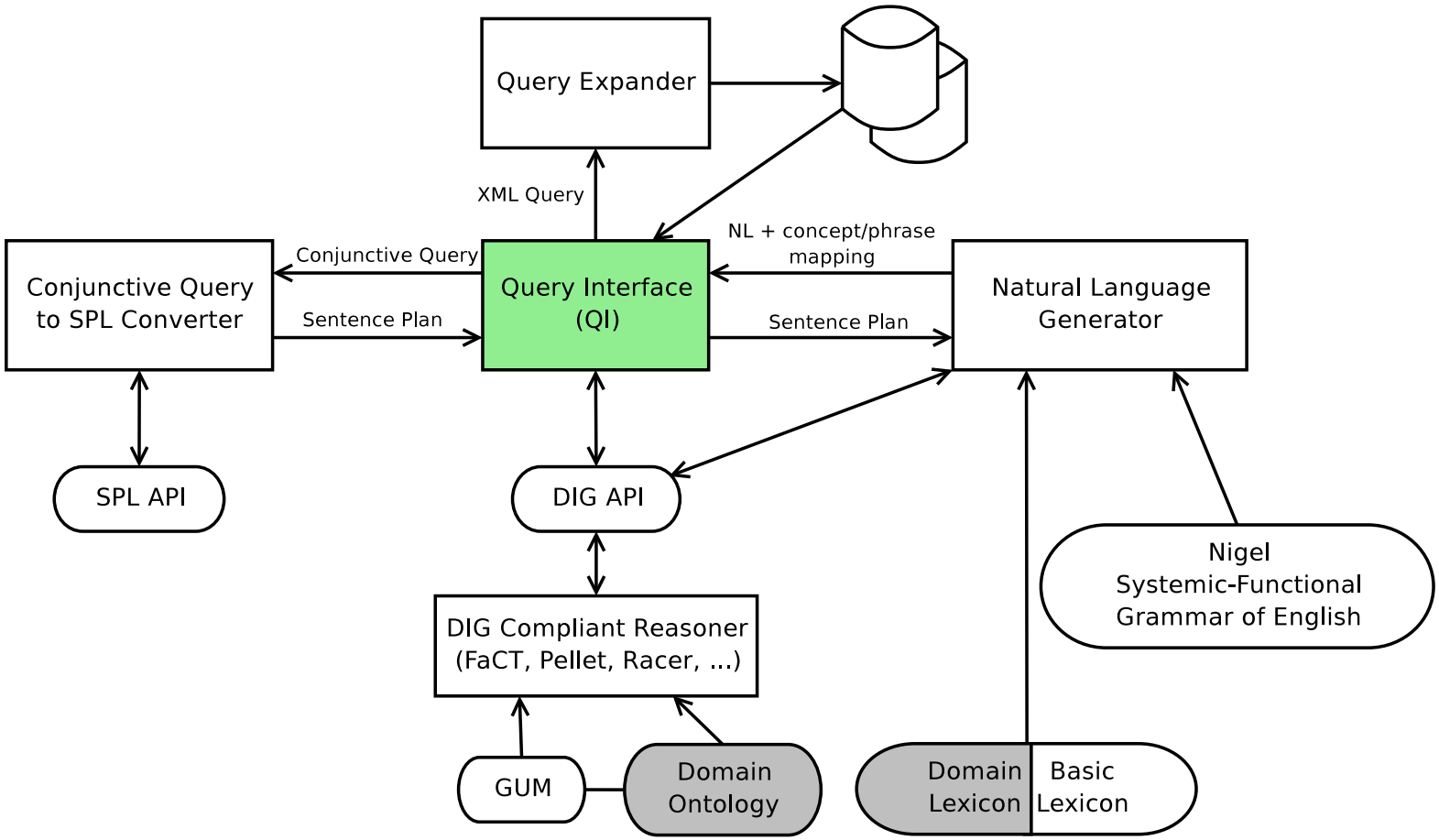
Penman-style Architecture



The New Query Interface



Inside the New Query System



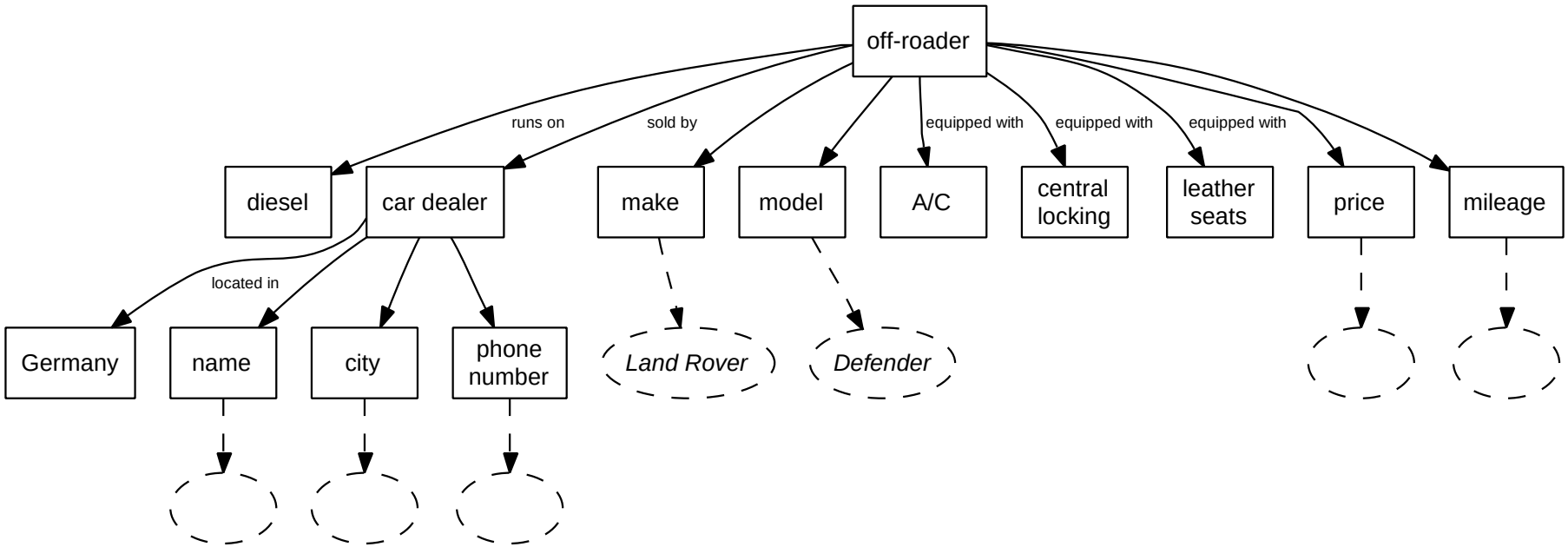
Conjunctive Query to Sentence Plan

$$\{x_1 |\ \text{car}(x_1),\ \text{run-on}(x_1, x_{1,1}),\ \text{diesel}(x_{1,1}),\ \text{make}(x_1, x_{1,2}),\ x_{1,2} \in \{'\mathit{LandRover}'\}\} \quad (1)$$

```
((S1 S2)
  (S1 / run-on
      :MODALITY must
      :ACTOR (C1 / car)
      :ACTEE (C2 / diesel)
      )
  (S2 / PROPERTY-ASCRIPTION
      :MODALITY must
      :DOMAIN (C3 / make)
      :RANGE
      (C4 / QUALITY
          :LEX 'Land Rover')))
```

The car must run on diesel, and the make must be Land Rover.

Critical Task: CQ Partitioning



Summary

- Evolution steps of an intelligent query tool
- Information access through ontology navigation
- Introduction to NLG based on computational SFL
- Representation of conjunctive queries in natural language via a SFG-based generator

- Outlook
  - Mapping algorithm: conjunctive query (CQ) to sentence plan (SPL)
  - Refactoring of the generation engine (KPML): J2EE technologies, DIG standard compliance

Summary

- Evolution steps of an intelligent query tool
- Information access through ontology navigation
- Introduction to NLG based on computational SFL
- Representation of conjunctive queries in natural language via a SFG-based generator

- Outlook
  - Mapping algorithm: conjunctive query (CQ) to sentence plan (SPL)
  - Refactoring of the generation engine (KPML): J2EE technologies, DIG standard compliance

Summary

- Evolution steps of an intelligent query tool
- Information access through ontology navigation
- Introduction to NLG based on computational SFL
- Representation of conjunctive queries in natural language via a SFG-based generator
- Outlook
  - Mapping algorithm: conjunctive query (CQ) to sentence plan (SPL)
  - Refactoring of the generation engine (KPML): J2EE technologies, DIG standard compliance

Summary

- Evolution steps of an intelligent query tool
- Information access through ontology navigation
- Introduction to NLG based on computational SFL
- Representation of conjunctive queries in natural language via a SFG-based generator

- Outlook
  - Mapping algorithm: conjunctive query (CQ) to sentence plan (SPL)
  - Refactoring of the generation engine (KPML): J2EE technologies, DIG standard compliance

Summary

- Evolution steps of an intelligent query tool
- Information access through ontology navigation
- Introduction to NLG based on computational SFL
- Representation of conjunctive queries in natural language via a SFG-based generator

- Outlook
  - Mapping algorithm: conjunctive query (CQ) to sentence plan (SPL)
  - Refactoring of the generation engine (KPML): J2EE technologies, DIG standard compliance

Summary

- Evolution steps of an intelligent query tool
- Information access through ontology navigation
- Introduction to NLG based on computational SFL
- Representation of conjunctive queries in natural language via a SFG-based generator

- Outlook
  - Mapping algorithm: conjunctive query (CQ) to sentence plan (SPL)
  - Refactoring of the generation engine (KPML): J2EE technologies, DIG standard compliance

For Further Reading I

Paul Piwek, Roger Evans, Lynne Cahill, and Neil Tipper.

Natural Language Generation in the MILE System.
In *Proceedings of the IMPACTS in Natural Language Generation Workshop*, Schloss Dagstuhl, Germany, 2000.

William C. Mann and Sandra A. Thompson.

Rhetorical Structure Theory: Toward a Functional Theory of Text Organization.
Text, 8(3):243–281, 1988.

Eduard Hovy.

Language generation: overview
Survey of the state of the art in human language technology, Chapter 4.1, pages 139–146.
Cambridge University Press, New York, NY, USA, 1997.

Maite Taboada and William C. Mann

Rhetorical Structure Theory: Looking Back and Moving Ahead.
Discourse Studies, 8(3), 2006 (to appear).
[Pre-publication version: `http://www.sfu.ca/rst/pdfs/Taboada_Mann_RST_Part1.pdf`]

Maite Taboada and William C. Mann

Applications of Rhetorical Structure Theory.
Discourse Studies, 8(3), 2006 (to appear).
[Pre-publication version: `http://www.sfu.ca/rst/pdfs/Taboada_Mann_RST_Part2.pdf`]

For Further Reading II

C. M. I. Matthiessen and J. A. Bateman, editors.

Text Generation and Systemic-Functional Linguistics: Experiences from English and Japanese.
Pinter, London, 1991.

William C. Mann.

Inquiry semantics: a functional semantics of natural language grammar.
In *Proceedings of the first conference on European chapter of the Association for Computational Linguistics*, pages 165–174, Morristown, NJ, USA, 1983. Association for Computational Linguistics.

R. T. Kasper.

A flexible interface for linking applications to PENMAN's sentence generator.
In *Proceedings of the DARPA Workshop on Speech and Natural Language*, 1989.