Introducing maximal variation in text planning for small domains

Erwin Marsi
Department of Language and Speech
University of Nijmegen

January 28, 1998

Abstract

This work describes a method for text planning that is suitable to small domains like train table information. Our aim is to introduce maximal variation in the packaging of information and in the linear order of its presentation. To this end, we regard text planning as a goal-driven process that dynamically constructs a text plan. The goal is a state where all information in the input is shared with the user; the means to achieve this goal are utterances. The application of utterances is limited by constraints that refer to the user's current state of knowledge. This approach to text planning can be conveniently implemented as a functional unification grammar. In addition, we show how optional or inferable information can be accounted for, how focus can be distributed, and how the generation of anaphoric expressions can be constrained by looking at the form and content of a previous utterance.

1 Introduction

This work on text planning is part of a project that is concerned with investigating Dutch prosody by implementing a concept-to-speech system. The project focuses on how the pitch accents and the prosodic boundaries of an utterance are related to (or predictable from) its semantic and syntactic structure and its discourse context. The basic idea is that a natural language generator, as opposed to a parser, generates extensive and reliable information about the linguistic structure of an utterance, and is therefore particularly suitable to provide input to a prosodic module. This approach places at least two constraints on the generator. First, it ought to generate all information that is required by the prosodic module. Second, it ought to generate as much variation as possible, in order to put the prosodic module to the test. Given a fairly conventional architecture consisting of a text planner followed by a surface generator, these constraints affect the text planner. For instance, it should keep track of the information status of concepts, because the distinction between old and new information is important for pitch accent placement. Also, it should be able to paraphrase one and the same conceptual structure as different semantic structures, which are in turn realized as different sentences by the surface generator.

This paper describes a text planner that meets these requirements. It is described on the basis of an application of concept-to-speech in which train table information is taken as input to generate a spoken description, in Dutch, of how to get from one place to another by train. The approach, however, is easily adaptable to similar domains. Given our main goal of introducing variation at the level of text planning, the text planner is rather uncomplicated and completely ignores other aspects like rhetorical structuring of the text or tailoring information to the user. Its interesting points, however, are that it employs a constraint-based approach to produce

$$
\begin{bmatrix}
\text{CONCEPT} & \text{ROUTE} \\
\text{SECTIONS} & \begin{bmatrix}
\text{CONCEPT} & \text{SECTION} \\
\text{DATA} & \begin{bmatrix}
\text{DEP-PLACE} & \textit{Nijmegen} \\
\text{DEP-TIME} & \textit{12:08} \\
\text{ARR-PLACE} & \textit{'s-Hertogenbosch} \\
\text{ARR-TIME} & \textit{12:38} \\
\text{CONVEYANCE} & \textit{sneltrein} \\
\text{DIRECTION} & \textit{Roosendaal} \\
\text{PLATFORM} & \textit{4b}
\end{bmatrix} \\
\text{NEXT} & \begin{bmatrix}
\text{CONCEPT} & \text{SECTION} \\
\text{DATA} & \begin{bmatrix}
\text{DEP-PLACE} & \textit{'s-Hertogenbosch} \\
\text{DEP-TIME} & \textit{12:42} \\
\text{ARR-PLACE} & \textit{Geldermalsen} \\
\text{ARR-TIME} & \textit{12:59} \\
\text{CONVEYANCE} & \textit{stoptrein} \\
\text{DIRECTION} & \textit{Utrecht centraal station} \\
\text{PLATFORM} & \textit{3b}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

Figure 1: Example of an input structure

variation and that its implementation is completely grammar-based within the framework of Functional Unification Grammar.

2 A functional unification grammar for text planning

2.1 Input structures

The input to the text planner is derived from the response of a travel information system to a particular query by the user. It is a feature description (FD) of hierarchically structured concepts. For instance, the FD in Figure 1 represents a journey with one change. The top concept, representing the whole journey from departure place to arrival place, is called ROUTE.[1] It is composed of one or more SECTION nodes, each of which represents a partial journey from one place to another. A section node is accompanied by information about the place and time of departure, the place and time of arrival, the type of train, its direction, and the platform it leaves from. Notice that the attribute NEXT serves as a link to the subsequent section.

2.2 Text planning grammar

Text planning is regarded as the process of mapping the input structure to a sequence of semantic structures, which will ultimately be realized as spoken utterances. Evidently, not all the information in the input can be expressed in a single utterance, so the text planner must divide it into smaller packages. The information within a package should be coherent and the linear order of the packages should make sense. For instance, it is quite odd to start off the description of a section with the arrival place and arrival time, that is, without mentioning the departure place and departure time first. Ruling out certain ways of information packaging is

[1]It also contains information like the total amount of travelling time and the number of changes, which is used to generate a summary of the journey. This option will be ignored here.

of course a matter of common sense; it is always possible to come up with a context in which a very marked order of presentation is acceptable. The obvious solution is to use one or more templates that prescribe acceptable ways of presenting the information. However, as explained above, an important requirement is to generate as much variation as possible. Using a limited number of templates would severely restrict the amount of variation at the level of text planning.

The alternative is to adopt a dynamic approach to text planning, and consider it as an attempt to achieve a particular goal under certain constraints [5]. The goal is a transfer of all available information, i.e. a state where the user knows all the information that is in the input structure. This does not imply, however, that all data have to be explicitly expressed, because the listener may infer some of it from the situational context or from the previous discourse. For example, the departure place may be inferred, because it is the arrival place of the previous section. The means to achieve this goal are utterances. Generating a semantic structure for an utterance may be considered as performing a speech act that alters the user's state of knowledge [1]. According to this view, the application of a certain utterance is limited by constraints referring to the user's current state of knowledge, and the form and content of previous utterances. Within the boundaries of these constraints, planning is assumed to be a dynamic process directed by random choices. As a result, the output of the planner will vary considerably from one run to another. Thus, the text planner is not designed to generate a plan that will eventually transfer the information to the user optimally, but instead to generate as many plans as possible, which nevertheless transfer the information in an acceptable way.

The way this is implemented is as a functional unification grammar [6] in FUF [2]. The grammar is a feature description that consists of a number of alternatives, most of which represent an utterance with its constraints on application, its semantic structure and its effect of the user's knowledge. The process of text planning is a step-wise unification of the input with the grammar. The control mechanism of FUF traverses all concepts in the input structure (i.e. sub-FD's that contain the attribute CONCEPT), unifying them with suitable alternatives of the grammar. During this process, the input structure is enriched with new concepts, semantic structures and updates of the user's knowledge state.

We will trace this process on the basis of a simplified example. Suppose we take the FD in Figure 1 as input. Each SECTION concept in the input is unified with a corresponding grammar alternative. The grammar alternative for SECTION (see Figure 2) adds a feature UNITS that is used to store a number of nodes of type UNIT, corresponding to the utterances that together describe a section. A section typically contains between two and six units. A unit has a feature BMB, shorthand for 'belief-mutual-belief', which represents the text planner's belief about the current knowledge shared with the user. The alternative for SECTION initializes the knowledge state for its first unit: it is assumed that initially all information is unknown. The remaining features will be explained later on.

The grammar contains many different alternatives for UNIT, of which the one in Figure 3 is an example. The value of the BMB is best viewed as a condition on the applicability of this alternative. For the current example, it states that the departure place, departure time and conveyance must be unknown. Notice that, due to the nature of unification, the condition is indifferent with respect to the status of other data; they can be either known or unknown. If the condition succeeds, the speech act under ACT can be performed, which amounts to sending a semantic structure to the surface generator. The template shown as the value of ACT is for expository reasons only; the value is actually an FD that is the semantic structure for an utterance. Semantic structures will be discussed later on. The slots in the template are filled by reference to the relevant values under DATA, which is the reason why this attribute is shared between a unit and a section (cf. Figure 2) and between units (cf. Figure 3). Now performing a speech act alters the knowledge state, which is modeled by the fact that in the subsequent

$$
\begin{bmatrix}
\text{CONCEPT} & \text{SECTION} \\
\text{DONE} & \langle \uparrow \text{UNITS DONE} \rangle \\
\text{CSET} & (\ \text{UNITS}\ ) \\
\text{UNITS} & \begin{bmatrix}
\text{CONCEPT} & \text{UNIT} \\
\text{DATA} & \langle \uparrow \uparrow \text{DATA} \rangle \\
\text{BMB} & \begin{bmatrix}
\text{DEP-PLACE} & \text{UNKNOWN} \\
\text{DEP-TIME} & \text{UNKNOWN} \\
\text{ARR-PLACE} & \text{UNKNOWN} \\
\text{ARR-TIME} & \text{UNKNOWN} \\
\text{CONVEYANCE} & \text{UNKNOWN} \\
\text{DIRECTION} & \text{UNKNOWN} \\
\text{PLATFORM} & \text{UNKNOWN}
\end{bmatrix}
\end{bmatrix} \\
\multicolumn{2}{l}{\left\{ \begin{array}{l}
\textit{:wait}\ \text{DONE} \\
\begin{bmatrix} \text{NEXT} & \text{NONE} \end{bmatrix} \\
\begin{bmatrix}
\text{CSET} & +\ (\ \text{NEXT}\ ) \\
\text{NEXT} & \begin{bmatrix} \text{CONCEPT} & \text{SECTION} \end{bmatrix}
\end{bmatrix}
\end{array} \right\}}
\end{bmatrix}
$$

Figure 2: Grammar alternative for a section

unit the values of the attributes DEP-PLACE, DEP-TIME and CONVEYANCE become known. The state of the other data is shared with the previous BMB, implementing the assumption that their status remains unaffected by the current speech act.[2]

The expansion of a unit into a speech act and a next unit is a recursive process. It continues until BMB reaches the point where all data have become known. This termination condition is modeled by a special unit that has neither a speech act nor a NEXT attribute; see Figure 4. It does however, provide the attribute DONE with its value TRUE, and because this value is shared between subsequent units as well as between a section and its first unit, it means that in the section node the attribute DONE becomes TRUE too. This in turn, triggers the alternatives in Figure 2, which had been frozen by means of the special option *:wait* until the feature DONE had received a value. FUF tries the alternatives in the order they are given in the grammar. The first alternative succeeds if no more sections are given in the input, i.e. this was the last section of the route. Otherwise, the second alternative is taken, which forces processing of the next section.[3]

The important thing to notice is that when the next unit must be added, there are in general multiple units whose conditions are compatible with the current knowledge state. At such points, the random choice of a unit introduces the variation that was sought after. However,

[2]The fact that two attributes share the same value is expressed by means of path. For instance, the path $\langle\uparrow\uparrow\uparrow$ BMB ARR-PLACE$\rangle$ that is the value of ARR-PLACE means: go up three levels (i.e. skipping the attributes ARR-PLACE, BMB and NEXT) and from there follow the attributes BMB and NEXT to arrive at the intended value. This value is not present yet in the grammar alternative of Figure 3, but will be present in the input structure it is unified with.

[3]The *:wait* option forces goal-freezing and is one of the ways in which FUF extends the FUG formalism it is based on. Another extension is exemplified by the special attribute CSET. By default, the unifier identifies constituents (i.e. sub-FD's in the input that need to be unified with the grammar) by the presence of a special attribute (CONCEPT in our case) and traverses these constituents in a top-down breadth-first manner. The CSET attribute enables the grammar writer to overrule this default and explicitly specify the constituents. This options is used to force processing of al units (by CSET ==(UNITS)) before the next section is processes (by CSET +(NEXT)). See [2].

$$
\begin{bmatrix}
\text{CONCEPT} & \text{UNIT} \\
\text{BMB} & \begin{bmatrix} \text{DEP-PLACE} & \text{UNKNOWN} \\ \text{DEP-TIME} & \text{UNKNOWN} \\ \text{CONVEYANCE} & \text{UNKNOWN} \end{bmatrix} \\
\text{ACT} & \begin{bmatrix} \text{CONCEPT} & \text{CLAUSE} \\ \text{SEM} & \text{``}\textit{Take the } \langle \text{CONVEYANCE} \rangle \textit{ from } \langle \text{DEP-PLACE} \rangle \textit{ at } \langle \text{DEP-TIME} \rangle\text{''} \end{bmatrix} \\
\text{NEXT} & \begin{bmatrix} \text{CONCEPT} & \text{UNIT} \\ \text{DATA} & \langle \uparrow\uparrow \text{ DATA} \rangle \\ \text{DONE} & \langle \uparrow\uparrow \text{ DONE} \rangle \\ \text{BMB} & \begin{bmatrix} \text{DEP-PLACE} & \text{KNOWN} \\ \text{DEP-TIME} & \text{KNOWN} \\ \text{ARR-PLACE} & \langle \uparrow\uparrow\uparrow \text{ BMB ARR-PLACE} \rangle \\ \text{ARR-TIME} & \langle \uparrow\uparrow\uparrow \text{ BMB ARR-TIME} \rangle \\ \text{CONVEYANCE} & \text{KNOWN} \\ \text{DIRECTION} & \langle \uparrow\uparrow\uparrow \text{ BMB DIRECTION} \rangle \\ \text{PLATFORM} & \langle \uparrow\uparrow\uparrow \text{ BMB PLATFORM} \rangle \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

Figure 3: One of the grammar alternatives for a unit

$$
\begin{bmatrix}
\text{CONCEPT} & \text{UNIT} \\
\text{BMB} & \begin{bmatrix} \text{DEP-PLACE} & \text{KNOWN} \\ \text{DEP-TIME} & \text{KNOWN} \\ \text{ARR-PLACE} & \text{KNOWN} \\ \text{ARR-TIME} & \text{KNOWN} \\ \text{CONVEYANCE} & \text{KNOWN} \\ \text{DIRECTION} & \text{KNOWN} \\ \text{PLATFORM} & \text{KNOWN} \end{bmatrix} \\
\text{DONE} & \text{TRUE}
\end{bmatrix}
$$

Figure 4: The special unit that states the termination condition

not all choices lead to a solution, causing FUF to backtrack and revise its choice of units. Thus, the text planner can actually be considered a 'planner' in the AI sense of the word as a program that traverses a search space (a network of connected units) for a path (a sequence of units with associated speech acts and knowledge updates) that satisfies its goal (a state where the planner believes that all data is shared with the user).

Figure 5 shows an example of a part of the output of the text planner based on the input in Figure 1[4]. Notice that in the third unit the arrival place is repeated, although it was already mentioned in the first unit. This is possible because the grammar alternative for the third unit requires the departure time to be unknown, but does not constrain the value for arrival place. Therefore, it can be applied to introduce the arrival time only, or to introduce the arrival place as well. Either way, the arrival place is known after application of the unit. All units however, with the exception of the termination unit, require at least one piece of data to be unknown, since otherwise their application would be superfluous.

The planning grammar presented so far is simplified; the one actually used has a number

[4]The features DATA, CSET, FC and the second are left out to save space.

of extensions. For instance, the assumption that some information is optional is modeled by relaxing the termination condition. That is, if the feature [PLATFORM KNOWN] is removed from the FD in Figure 4, then processing of a section may finish without making mention of the platform. Furthermore, the assumption that the place of departure is inferable, because it is the arrival place of the previous section, is implemented by forcing the departure place to be known in the first unit of a non-initial section. Two other extensions, for generating anaphoric expressions and discourse markers, will be discussed next.

2.3 Semantic structures

As mentioned earlier, the value of an ACT attribute is not a string template, but an FD that is the semantic structure for an utterance. An example is given in Figure 6. It is passed on to a surface generator for Dutch that is similar to the SURGE surface generator for English [3]. Notice how the lemmas for participants and circumstances are instantiated by means of paths that refer to the relevant values within the unit's DATA feature.[5]

Figure 6 also illustrates the distribution of focus. A constituent that is focused is presented as important to the listener (as opposed to unfocused material that is presented as less important to the listener). In general, information of which the speaker assumes that the listener is unfamiliar with is unfocused, and vice versa.[6] The distinction has repercussions for both syntactic and prosodic realization. Focus affects the syntactic structure, because it is used by the surface generator to fix the word order of an utterance. In particular, it will strive for a canonical word order with unfocused material at the start of the utterance and focused material at its end. Focus affects the prosodic structure, because focused material will be marked by at least one pitch accent.[7] For current purposes, this means that checking the value of BMB provides a convenient way to determine if something is focused or not. This check is implemented as the option between parentheses in Figure 6. It states that if the departure place is known, then its realization must be unfocused. However, if the departure is unknown, the option fails and the value for FOCUS is left unspecified. The surface generator will make some default assumptions about focus, in this case, that content words are focused, while function words are unfocused. Hence, the text planner can limit itself to the exceptions, like the aforementioned case where the departure place is realized as a content word, but is nonetheless unfocused. Likewise, there is no need to explicitly specify the arrival time as focused, or the agent as unfocused.

In addition to the distribution of focus, the text planner is also responsible for generating anaphoric expressions. The range of possible anaphoric expressions within the present domain is quite small. First, the listener is situationally evoked and is always referred to by a personal pronoun. Second, the conveyance may be referred to by a relative pronoun if it has been mentioned before. Third, a departure place, arrival place, direction or platform may be referred to by a locative anaphoric adverb. The last type of reference is less trivial, because its use is restricted by word order. An case in point is (1) versus (2). The anaphoric expression *daar* ('there') is most naturally interpreted as referring to the place that was most recently mentioned. This leads to the intended interpretation (i.e. the departure place) in (1-b), but to a confusing or even unintended interpretation (i.e. the direction of the conveyance) in (2-b). Thus, in order to generate adequate anaphoric expressions of place, the text planner must keep track of the most recently mentioned place.

[5] At present, the text planner performs lexical choice, and is therefore responsible for variation at lexical level. This is not a not the only option however, since lexical choice might as well be performed in a separate module.

[6] The focused/unfocused distinction does not always coincide with the unknown/known. For example, old or known information may be focused to obtain a contrast.

[7] Apart from intonation, focus may be prosodically marked by a decreased speech rate, an increased loudness, and a more careful pronunciation. We do not use this options yet.

$$
\begin{bmatrix}
\text{CONCEPT} & \text{ROUTE} \\
\text{SECTIONS} & \begin{bmatrix}
\text{CONCEPT} & \text{SECTION} \\
\text{UNITS} & \begin{bmatrix}
\text{CONCEPT} & \text{UNIT} \\
\text{DONE} & \langle \uparrow \text{NEXT DONE} \rangle \\
\text{BMB} & \begin{bmatrix}
\text{DEP-PLACE} & \text{UNKNOWN} \\
\text{DEP-TIME} & \text{UNKNOWN} \\
\text{ARR-PLACE} & \text{UNKNOWN} \\
\text{ARR-TIME} & \text{UNKNOWN} \\
\text{CONVEYANCE} & \text{UNKNOWN} \\
\text{DIRECTION} & \text{UNKNOWN} \\
\text{PLATFORM} & \text{UNKNOWN}
\end{bmatrix} \\
\text{ACT} & \begin{bmatrix}
\text{CAT} & \text{CLAUSE} \\
\text{SEM} & \textit{``U gaat van Nijmegen naar 's-Hertogenbosch} \\
 & \textit{met de sneltrein richting Roosendaal.''}
\end{bmatrix} \\
\text{NEXT} & \begin{bmatrix}
\text{CONCEPT} & \text{UNIT} \\
\text{DONE} & \langle \uparrow \text{NEXT DONE} \rangle \\
\text{BMB} & \begin{bmatrix}
\text{DEP-PLACE} & \text{KNOWN} \\
\text{DEP-TIME} & \langle \uparrow \uparrow \uparrow \text{BMB DEP-TIME} \rangle \\
\text{ARR-PLACE} & \text{KNOWN} \\
\text{ARR-TIME} & \langle \uparrow \uparrow \uparrow \text{BMB ARR-TIME} \rangle \\
\text{CONVEYANCE} & \text{KNOWN} \\
\text{DIRECTION} & \text{KNOWN} \\
\text{PLATFORM} & \langle \uparrow \uparrow \uparrow \text{BMB PLATFORM} \rangle
\end{bmatrix} \\
\text{ACT} & \begin{bmatrix}
\text{CAT} & \text{CLAUSE} \\
\text{SEM} & \textit{``Die vertrekt vanaf perrron 4b om twaalf uur acht.''}
\end{bmatrix} \\
\text{NEXT} & \begin{bmatrix}
\text{CONCEPT} & \text{UNIT} \\
\text{DONE} & \langle \uparrow \text{NEXT DONE} \rangle \\
\text{BMB} & \begin{bmatrix}
\text{DEP-PLACE} & \text{KNOWN} \\
\text{DEP-TIME} & \text{KNOWN} \\
\text{ARR-PLACE} & \text{KNOWN} \\
\text{ARR-TIME} & \langle \uparrow \uparrow \uparrow \text{BMB ARR-TIME} \rangle \\
\text{CONVEYANCE} & \text{KNOWN} \\
\text{DIRECTION} & \text{KNOWN} \\
\text{PLATFORM} & \text{KNOWN}
\end{bmatrix} \\
\text{ACT} & \begin{bmatrix}
\text{CAT} & \text{CLAUSE} \\
\text{SEM} & \textit{``U arriveert in 's-Hertogenbosch} \\
 & \textit{om twaalf uur achtendertig.''}
\end{bmatrix} \\
\text{NEXT} & \begin{bmatrix}
\text{CONCEPT} & \text{UNIT} \\
\text{DONE} & \text{TRUE} \\
\text{BMB} & \begin{bmatrix}
\text{DEP-PLACE} & \text{KNOWN} \\
\text{DEP-TIME} & \text{KNOWN} \\
\text{ARR-PLACE} & \text{KNOWN} \\
\text{ARR-TIME} & \text{KNOWN} \\
\text{CONVEYANCE} & \text{KNOWN} \\
\text{DIRECTION} & \text{KNOWN} \\
\text{PLATFORM} & \text{KNOWN}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix} \\
\text{NEXT} & \ldots
\end{bmatrix}
\end{bmatrix}
$$

Figure 5: A part of the output that corresponds to the description of one section with three utterances. See example ((3)) for a gloss of the dutch sentences.

$$\begin{bmatrix} \textsc{concept} & \textsc{unit} \\ \textsc{bmb} & \ldots \\ \textsc{data} & \ldots \\ \textsc{act} & \begin{bmatrix} \textsc{cat} & \textsc{clause} \\ \textsc{process} & \begin{bmatrix} \textsc{type} & \textsc{material} \\ \textsc{lemma} & \textit{vertrekken} \end{bmatrix} \\ \textsc{partic} & \begin{bmatrix} \textsc{agent} \mid \textsc{cat} & \textsc{relative-pronoun} \\ \textsc{source} \mid \textsc{comp} & \begin{bmatrix} \textsc{describer} \mid \textsc{lemma} & \textit{perron} \\ \textsc{lemma} & \langle \uparrow\uparrow\uparrow\uparrow\uparrow \textsc{data dep-place} \rangle \\ \textsc{proper} & \textsc{true} \\ \multicolumn{2}{l}{\left(\begin{bmatrix} \langle \uparrow\uparrow\uparrow\uparrow \textsc{bmb dep-place} \rangle & \textsc{known} \\ \textsc{focus} & \textsc{false} \end{bmatrix}\right)} \end{bmatrix} \end{bmatrix} \\ \textsc{circum} \mid \textsc{time} \mid \textsc{comp} & \begin{bmatrix} \textsc{cat} & \textsc{time} \\ \textsc{lemma} & \langle \uparrow\uparrow\uparrow\uparrow\uparrow \textsc{data dep-time} \rangle \end{bmatrix} \end{bmatrix} \\ \textsc{next} & \ldots \end{bmatrix}$$

Figure 6: A unit containing a semantic structure for an utterance

$$\begin{bmatrix} \textsc{concept} & \textsc{unit} \\ \textsc{fc} \mid \textsc{place} & \textsc{dep-place} \\ \textsc{act} & \text{“Daar \textit{vertrekt u om 12:08 van perron 4b}”} \\ \textsc{next} & \begin{bmatrix} \textsc{concept} & \textsc{unit} \\ \textsc{fc} & \begin{bmatrix} \textsc{human} & \textsc{listener} \\ \textsc{place} & \textsc{platform} \\ \textsc{object} & \textsc{conveyance} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 7: Example of the use of FC

(1) a. *U neemt de sneltrein richting Roosendaal in Nijmegen*$_i$.
you take the train towards Roosendaal in Nijmegen

b. *Daar*$_i$ *vertrekt u om 12:08 van perron 4b.*
there leave you at 12:08 from platform 4b

(2) a. *U neemt in Nijmegen de sneltrein richting Roosendaal*$_i$.
you take in Nijmegen the train towards Roosendaal.

b. **Daar*$_i$ *vertrekt u om 12:08 van perron 4b.*
there leave you at 12:08 from platform 4b

This is implemented by means of a feature FC[8] that tells a unit what the most recently mentioned items of type HUMAN, PLACE and OBJECT are. Therefore, a unit can consult the content of FC to decide whether or not to use an anaphoric expression in its accompanying utterance. Depending on the content and word order of its utterance, a unit projects similar information to the FC feature of the next unit. An example is given in Figure 7.

The difference between the features BMB and FC is that the former tells us whether something

[8] FC stands for 'forward centers', because its use shows some resemblance to the notion of a set of forward centers in centering theory [4]. However, the text planner is certainly not meant to be an implementation of centering theory.

is already *known* (either because it was mentioned in one of the previous sentences or because it could be inferred), wheras the latter tells us whether something was *mentioned* in the *latest* utterance.

This approach poses an interesting question regarding word order: is word order determined by the text planner, by the surface generator, or perhaps by both? One point of view is that a semantic structure is passed on to the surface generator, which determines the word order, which in turn determines the FC a unit projects to the next unit. This assumes that there is feedback from surface generator to text planner and that generation proceeds 'depth-first' (i.e. plan the first utterance, realize the first utterance, plan the second utterance, realize the second utterance, etc.) An alternative point of view is that the semantic structure contains restrictions on word order (like 'mention the departure place last'), depending on the FC a unit projects to the next unit. This requires no feedback and assumes that the generation process is 'breadth-first' (i.e. planning all utterances before sending them to the surface generator). So far, we have adopted the latter approach, because it less complicated, both in concept and in implementation.[9]

Finally, the text planner also inserts discourse markers. For the time being, this is just a provisional solution to improve the quality of the output; the implementation is not based on any theory. Since the nature of the domain is a small narrative in which the sections are described in chronological order, temporal continuity markers are suitable in most cases. For example, the first unit of non-initial section may add a temporal continuation marker like *next*, *then*, *after that* etc. It would be interesting to explore the possibilities for a more principled account of discourse markers, e.g. by using rhetorical relations as in [5].

2.4 Final output

(3) gives an example of a travel description in Dutch, generated by the combination of text planner and the surface generator, and based on the input in Figure 1.

(3) a. *U gaat van Nijmegen naar 's-Hertogenbosch met de sneltrein richting Roosendaal.*
you go from Nijmegen to 's-Hertogenbosch with the express-train towards Roosendaal
'You take the Roosendaal express train to 's-Hertogenbosch.'

b. *Die vertrekt van perron 4b om twaalf uur acht.*
that leaves from platform 4b at twelve hour eight
'Which leaves from platform 4b at 12.08'

c. *U arriveert in 's-hertogenbosch om twaalf uur achtendertig.*
You arrive in 's-Hertogenbosch at twelve hour thirty-eight
'Which gets you to 's-Hertogenbosch at 12.38'

d. *Vervolgens neemt u daar de stoptrein richting Utrecht Centraal Station.*
next take you there the local-train towards Utrecht central station
'Next, take the local train to Utrecht central station.'

e. *Die vertrekt in 's-Hertogenbosch van perron 3b om twaalf uur tweeenveertig.*
that leaves in 's-hertogenbosch from platform 3b at twelve hour fortytwo

[9] A possible argument against the 'breadth-first' approach is that the text planner needs to know in advance what kind of semantic structures the surface generator can handle. Theoretically, the text planner might decide to stuff the whole description of a section into one, large semantic structure. However, under present circumstances, it is the grammar writer who makes up the greater part of a semantic structure, whereas the contribution of the text planner is limited to the insertion of lexical items and the distribution of focus. Therefore, this seems to be a weak argument as long as the semantic structures are not dynamically constructed during the generation process.

'Which leaves in 's-hertogenbosch from platform 3b at 12.42.'

f. *Dan bent u in Geldermalsen om twaalf uur negenenvijftig.*
then are you in Geldermalsen at twelve hour fiftynine
'Which gets you to Geldermalsen at 12.59.'

3 Summary

We have described a rather uncomplicated method for text planning that is suitable to small domains like travel information. Our aim was to introduce maximal variation in the packaging of information and in the linear order of its presentation. To this end, it proved useful to view text planning as a goal-driven process, in which utterances are used to alter the knowledge state of the user, and their application is restricted only by constraints that refer to the user's knowledge state. This can be conveniently implemented as a functional unification grammar. In addition, we showed how optional or inferable information is accounted for, how focus is distributed, and how the generation of anaphoric expressions is constrained by the form and content of the latest utterance. Future work may address the need for feedback from surface generator to text planner, and the incorporation of a more principled account of generating discourse markers.

References

[1] P.R. Cohen and C.R. Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, (3):177–212, 1979.

[2] Michael Elhadad. *FUF: The Universal Unifier - User Manual, version 5.2.* New York, 1993. Technical Report CUCS-038-91.

[3] Michael Elhadad and Jaques Robin. An overview of SURGE: A reusable comprehensive syntactic realization component. Technical Report 96-03, Ben Gurion University, Department of computer Science, Beer Sheva, Israel, 1996.

[4] Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. Centering: a framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225, 1995.

[5] Eduard H. Hovy. Approaches to the planning of coherent text. In W. Swartout and W. Mann, editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pages 83–102. Kluwer Academics Publishers, Boston, 1991.

[6] Martin Kay. Functional unification grammar: A formalism for machine translation. In *Proceedings of COLING-84*, pages 75–78. ACL, Stanford University, 1984.