SEMANTIC STRUCTURES
Advances in Natural Language Processing

Edited by
David L. Waltz
Thinking Machines Corporation and
Brandeis University

LEA
1989 LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS
Hillsdale, New Jersey Hove and London

CHAPTER 5

Natural Language Description of Time-Varying Scenes

BERND NEUMANN
Fachbereich Informatik,
Universität Hamburg

OVERVIEW

This work explores the border area between vision and natural language with respect to a particular task: obtaining verbal descriptions of scenes with motion. The task involves image understanding as we assume that the time-varying scene to be described is represented by an image sequence. Hence, part of the problem is image-sequence analysis. We focus on high-level aspects: recognizing interesting occurrences that extend over time. Very little is said about lower level processes that constitute the scope of vision in a narrow sense. The concepts and representations proposed in this work can be viewed as extending the scope of a vision system beyond the level of object recognition. In this respect, our work is a contribution to the question raised by Waltz (1979): What should the output of a (complete) vision system be?

Another aspect of this work concerns the connection of vision and natural language. Both disciplines have been studied rather independently from each other. Hence, little is known about how the semantics of a verbal scene description relate to a description derived from visual input. This work shows that visual motion analysis

can lead to representations that easily map into deep case frames of natural-language utterances. Apart from the technical aspects, this is interesting because semantic categories developed in natural-language research turn out to have clear physical (and visual) connotations, computable from an image sequence.

The problem of generating a natural-language utterance from an appropriate deep structure is not our concern. Our work does however, touch upon the problem of composing a coherent description (i.e., selecting and ordering possible utterances). The general idea of our approach is to use the anticipated visualization of the hearer for speech planning.

These are, in brief, three major problem areas addressed by this contribution. We now give an overview of the system NAOS that implements our ideas.

The acronym stands for "*NA*tural language description of Object movements in a Street scene". This indicates our domain of interest: traffic scenes. In particular, we are concerned with the following scenario. Person A (looking out of a window) observes a street scene over a certain time span. Then A turns to some person B (who knows the street but cannot see it) and describes what he has seen. NAOS attempts to generate natural-language scene descriptions according to this scenario.

The raw input data are black-and-white TV images taken from a fixed viewpoint. Figure 5.1 shows 4 images out of a sequence of 64, covering a time span of approximately 13 seconds. The events of interest are pedestrians standing, walking, and crossing the street, cars starting and stopping, turning right, and so on.

Scenes like this have been used for many years as experimental data for image-understanding research at the Universität Hamburg, primarily for low-level motion analysis, object tracking, and motion stereo. In project NAOS we are interested in high-level interpretations. For this purpose, all low-level processing up to a complete recovery of the scene geometry (including classified objects) is simulated by human interaction. The output of this first stage of processing is called *geometrical scene description* (GSD). A precise definition of this intermediate-level scene representation is given in the next section.

The core of NAOS is a program that recognizes *events* in a GSD. An event is a subset of the scene that can be described by a certain verb of locomotion (e.g., "overtake"). A priori knowledge about event types is provided by *event models*. They consist of propositions about the scene that must be satisfied if an event can be said to have occurred. Event recognition is very much like proving the existence of an event based on facts provided by the GSD and rules provided by the event models. In implementing the proof procedure (using the programming language FUZZY), several techniques have been developed that may have relevance beyond this task. For example, relational matching has been extended to deal with constraints arising from time intervals. Event models are discussed in the third section; event recognition is discussed in the fifth section.

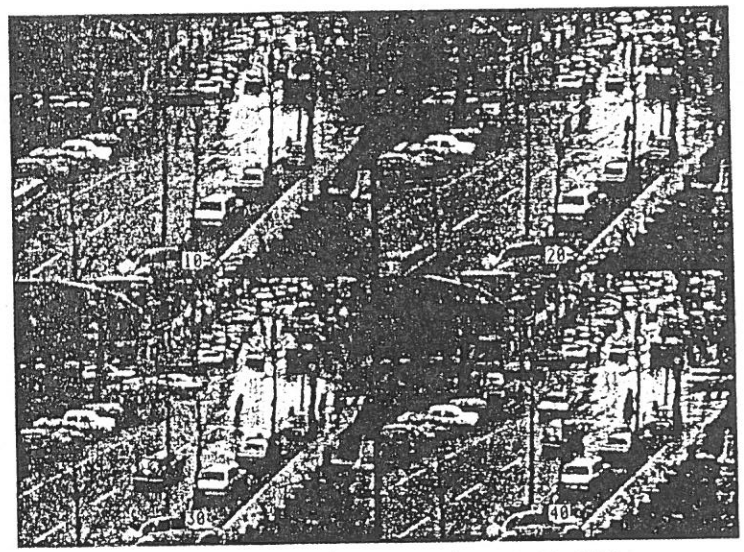

Figure 5.1. Images of a traffic scene to be described by NAOS

Events are conceptual units that are designed to capture the semantics of verbs of locomotion. The next step toward a natural-language scene description is filling the "case frames" associated with such verbs (Fillmore, 1968). For example, the agent case corresponds to a certain object in the event. Similarly source, path, and goal cases correspond to locations readily available from the event description. For verbalization, these objects that fill case roles have to be referenced according to certain rules of natural language use. For example, locations are referenced using spatial prepositions and nearby objects of reference ("at the traffic lights"). In our view, the construction of such references is the critical step from a visual to a verbal representation.

Although bottom-up scene description is the central goal of NAOS, we also consider question answering involving top-down processing. In this case, an inverse mapping is required: Natural-language input is transformed into a deep-case structure from which

a constrained event recognition task can be derived. This process is described in the third section. After event evaluation, a case frame is filled in a manner similar to bottom-up verbalization. Answer generation requires, however, several special processes (e.g., provisions for generating cooperative answers). This distinguishes the task from unconstrained verbalization. All issues concerning the mapping between events and case frames are discussed in the section on verbalization.

The core processes of natural-language understanding and generation were not developed as part of the NAOS-project. We make use of components of the natural language dialogue system HAM-ANS (Hoeppner et al., 1983, Hoeppner et al., 1984), in particular of a generator written by Busemann (1984). These components are not discussed in detail in this contribution. We are concerned, however, with another issue on the natural-language side: composing a coherent, natural description from a set of possible utterances. This is the theme of the fifth section. We assume that the computer is always trying to perform a single kind of "speech act": to inform its user of some situation or event. Many other "speech acts" (Searle, 1969) are possible: requesting, reminding, connecting, ordering, promising, apologizing, or many others. In order to perform appropriately, the system must anticipate the effect of each utterance on the hearer. We present a "standard plan" for scene description that is a first approximation to speech-act planning based on the hearer's anticipated visualization.

The final section of this contribution presents a discussion of related work and points out future directions planned for our research.

REPRESENTING THE SCENE

In this section, we describe the data that are used as input for the NAOS system. Eventually, we would like NAOS to generate verbal descriptions from the output of some existing vision system whose input would be raw images. This would clearly demonstrate the intended scope of our work: to extend vision to higher levels of representation that connect to concepts of natural language. Unfortunately, there do not yet exist vision systems that can analyze real-world image sequences with sufficient reliability and speed to provide the input for NAOS. Our group is indeed working on analyzing image sequences of traffic scenes (Dreschler and Nagel, 1981).

An intermediate level of representation has been defined that bypasses the problems of low-level vision. This level represents the output of a vision system in the narrow sense: it tells "what is where" (Marr, 1981). More specifically, this level provides a representation of the 3D scene geometry, and photometric scene properties, plus a classification of all objects of interest. This seems to be also in agreement with the intended output of a vision system as proposed in Ballard and Brown (1982): an explicit, meaningful description of physical objects. In NAOS, this representation is called the *geometrical scene description* (GSD) to emphasize the prevalence of geometrical information and the absence of high-level concepts.

Clearly, a vision system will hardly ever be able to recover the complete 3D geometry of a scene, as the shape of surfaces may remain guess-work, particularly if they are hidden. But as higher level scene interpretations seem to be based on what one knows about a scene rather than on what one does not know, it is appropriate to choose a canonical representation containing all information which could possibly be available. To really obtain such information requires considerable perceptual inference facilities, including viewpoint and light source geometry.

By similar reasoning, all photometric scene properties are assumed to be known (e.g., light source characteristics and surface reflectivity). Although these data are not essential for NAOS (except, perhaps, of object colors), they guarantee completeness of the GSD in the following sense: The data suffice, in principle, to regenerate the raw images. In other words, the scope of this representation does not presuppose loss of information along the way from raw images to the GSD.

In more detail, a GSD contains

—for each frame of the image sequence:

- instance of time
- visible objects
- viewpoint
- illumination

—for each object:

- 3D shape
- surface characteristics (color)
- class (automobile, person, tree)
- identity (VW1, Person1)
- 3D position and orientation in each frame

By far the most important information for NAOS is the list of positions and orientations attached to each object. Based on these data, high-level motion concepts are recognized (e.g., one car overtaking another). Position refers to some fixed reference point of an object coordinate system (usually the centroid) and is given with respect to a fixed world coordinate system. Similarly, orientation refers to a distinguished direction in the object coordinate system (usually the 'front').

Shape and surface information is provided by models based on polyhedra and cones (Brooks, 1981). The repertoire of possible shapes is in no way adequate for representing highly irregular bodies. Also there are only very crude provisions for encoding photometric surface properties. More sophistication, however, is currently not required in NAOS where shape information will be mainly used for computing qualitative spatial relations.

How is a GSD obtained for a real-world traffic scene? NAOS deals with traffic scenes observable from our laboratory window. The major stationary components of such scenes are known to the system: It has access to a model of the environment as part of its knowledge base.

The first step in processing an image sequence is to determine the viewpoint (camera position, orientation, and focus) with the help of the street model. This is done by finding point correspondences (currently by hand) and then employing a standard calibration technique (Yakimovsky & Cunningham, 1978). Using the viewpoint information one can identify those stationary objects of the street model that are visible in the scene. To obtain the 3D trajectories of moving objects, automatic and human-aided change detection and tracking procedures developed for other motion analysis tasks can be employed (Nagel & Rekers, 1982). Figure 5.2 shows a synthetic view of trajectories obtained manually for a scene involving 3 cars

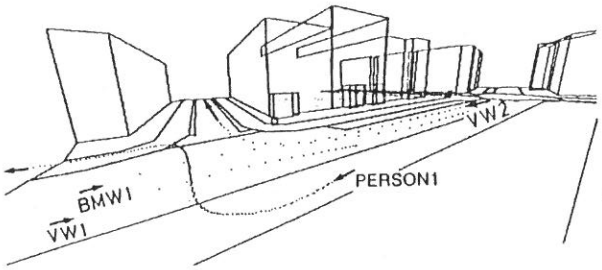

Figure 5.2. Synthetic view of a scene with 4 moving objects.

and a pedestrian. The intersection shown in Figure 5.1 is visible on the right.

EVENTS

Given an intermediate-level representation of a scene in terms of objects and their positions, it is not all clear where further processing should lead. One might be interested, for example, in finding out whether a certain object configuration is present or not (e.g., a parked red Mercedes). Or else one might want the system to communicate its observations to humans. There are clearly as many tasks as there are uses for visual data, and each task would suggest certain abstractions—high-level "conceptual units"—to be computed from the scene data. If one is finding a path for a robot, for example, it might be useful to compute an explicit representation of free space. As obvious guidelines for structuring high-level vision do not seem to be around the corner, some motivation for the approach taken in NAOS must be given.

We introduce conceptual units that are useful for talking about scenes with motion. Clearly, an intermediate-level representation of motion in terms of objects and their positions—the GSD introduced in the preceding section—would be inappropriate for this purpose. Natural language gives some indication of motion concepts that may be generally interesting, namely concepts for which succinct expressions are available.

In the remainder of this section, we discuss 'events' that are the conceptual units for motion description in NAOS. First, event models are introduced, and then procedures for event recognition are described.

Event Models

Events are interesting subspaces of the four-dimensional space–time continuum (much in accord with Webster's definition). We consider events that describe "changes of the kind people talk about" (Miller & Johnson-Laird, 1976). More specifically, an event is a subspace of a scene that can be described by a verb of change (in our domain: locomotion).

Events are organized into classes according to the verb that is associated with the event. Event classes are defined by event models. An event model is a schema that specifies what we are looking for in a scene. Events are particular instantiations of event models.

Event models consist of a head, which is a predicate about a scene, and a body, which specifies how to verify the predicate. The following is the model for 'overtake' events.

Head: (OVERTAKE OBJ1 OBJ2 T1 T2)
Body: (MOVE OBJ1 T1 T2)
(MOVE OBJ2 T1 T2)
(APPROACH OBJ1 OBJ2 T1 T3)
(BESIDE OBJ1 OBJ2 T3 T4)
(RECEDE OBJ1 OBJ2 T4 T2)

The semantics of OVERTAKE can be paraphrased as follows: "OBJ1 overtakes OBJ2 during the time interval (T1 and T2) if

- both objects are in motion throughout the time interval (T1 T2),
- OBJ1 approaches OBJ2 during the time interval (T1 T3),
- there follows a time interval (T3 T4) where OBJ1 is beside OBJ2,
- and finally OBJ1 recedes from OBJ2 throughout the remaining interval (T4 T2)."

It is not claimed that this definition captures the semantics of all 'overtake' situations that one might think of (for example, an airplane passing overhead another). The point is to demonstrate that the representational formalism is adequate for the street example.

Some comments on the syntax are in order. Predicates are written in a relational notation. The first element is a predicate identifier, the other elements are arguments. Arguments are usually variables which must be instantiated, but may also be constants, for example, numbers. (All arguments in this example are variables.) If there can be any doubt as to whether an identifier denotes a variable or a constant, a '?' will be attached to the variable identifier.

The variables T1 to T4 are time variables denoting interval boundaries. Events are taken to extend over a nonzero time interval in all but degenerate cases in accord with the notion of a 'four-dimensional subspace' of a time-varying scene. Hence, the head of an event model always involves a time interval. A predicate about one time interval does not necessarily imply anything about another time interval, even if the latter is a subinterval of the former. Nevertheless, there are many predicates that do allow the subinterval implication (e.g., MOVE, BEHIND). They are called *durative* corresponding to the linguistic notion. Durative predicates have also been introduced in Allen (1981) by means of the HOLDS predicate. To be precise, a predicate P is durative if

(P . . . T1 T2) => (P . . . T1' T2') for $T1 \leq T1' < T2' \leq T2$.

Similarly, there are *inchoative* and *resultative* predicates. The event model 'stop,' for example, is resultative. It is defined as follows:

Head: (STOP OBJ T1 T2)
Body: (MOVE OBJ T1 T2)
(STAND OBJ T2 T3)

For a resultative predicate P the following implication holds:

(P . . . T1 T2) => (P . . . T1' T2) for $T1 \leq T1' < T2$.

Inchoative predicates are discussed in the second part of this section.

The 'overtake' event is hierarchical (i.e., the body is composed of predicates that must be verified if the head predicate is to be true). The predicates of the body may be events (e.g., APPROACH) or other predicates (e.g., BESIDE). Predicates are called 'primitive' if they cannot be decomposed further. The body of each primitive predicate is a procedure to be evaluated with the GSD as data. The event model MOVE, for example, is primitive. So is the predicate BESIDE.

Some motion concepts are like events except that there is no verb available for describing such motion. For example, it proved useful to define the primitive predicate SYM-APPROACH ("symmetrical approach"):

Head: (SYM-APPROACH OBJ1 OBJ2 T1 T2)
Body: <test whether the distance between
OBJ1 and OBJ2 decreases>

Using this predicate, APPROACH may be defined as follows:

Head: (APPROACH OBJ1 OBJ2 T1 T2)
Body: (SYM-APPROACH OBJ1 OBJ2 T1 T2)
(MOVE OBJ1 T1 T2)
(IN-FRONT-OF OBJ2 OBJ1 T1 T2)

APPROACH is an event model that closely corresponds to the meaning of the natural-language verb 'approach.' Not only must the distance decrease, but OBJ1 is also required to move toward OBJ2. The predicate IN-FRONT-OF tests whether OBJ2 is located within a certain sector relative to the 'front' of OBJ1.

Similarly, RECEDE is defined in terms of SYM-RECEDE (increasing distance) and BEHIND.

Event models have clear logical interpretation. They specify that the head is logically equivalent to the body:

⟨head⟩ <=> ⟨body⟩

All variables are existentially quantified. The body is given by a conjunction of predicates or by a procedure that is equivalent to a single predicate. Event recognition can be viewed as inferring certain predicates about the scene using the GSD for facts and the event models for inference rules. For event recognition the implication

⟨body⟩ => ⟨head⟩

will be extensively exploited.[1]

Event models form an implication hierarchy. As customary, general events are considered "higher" than special events. Hence, the top of the hierarchy corresponds to 'happen' which is implied by all other events. The structure of the hierarchy is determined by the implications that follow from

⟨head⟩ => ⟨body⟩

when decomposing the body into the individual conjuncts. For example, we get

[1]The event models presented so far do not use the full power of predicate calculus notation. In particular, there are neither explicit disjunctions nor explicit universal quantifications. As it turns out, none of the other 50 verbs currently considered in NAOS (see Appendix A) requires an extension of the notation in this respect (there will be other extensions). Hence, one may wonder whether this must be so. The first thing to observe is that the body of predicates may be procedural, hence unrestricted. Thus, no limitations are imposed in principle. Second, universal quantifications are in fact part of the formalism as far as time intervals are concerned. Whenever a predicate is marked "durative," it implies a predication of all subintervals of the given interval. Regarding disjunctions, it is quite conceivable to define an event model in terms of alternatives. For example, 'turn off' could be broken down into 'turn off right' or 'turn off left'. As there are other intuitive predicates to express the alternative (e.g., 'change of direction'), there is no need to employ a disjunction. Ambiguity of verb meaning also gives rise to alternatives. These can (and should) be handled, however, outside event models, as these alternatives do not constitute a single conceptual unit. In summary, there are no deep reasons for choosing this representational formalism. It just happens to be adequate.

(OVERTAKE OBJ1 OBJ2 T1 T2) => (MOVE OBJ1 T1 T2)
(OVERTAKE OBJ1 OBJ2 T1 T2) => (APPROACH OBJ1 OBJ2 T1 T2)
(etc.)

As all arguments are existentially quantified, the rules can also be written in the weaker form:

OVERTAKE =) MOVE

This means: if there is an overtake event in the scene, there must be also a 'move' event. The hierarchy presented in Appendix A is based on such rules. It is useful for event recognition in the free verbalization task (as opposed to question answering). Events are recognized in their order of decreasing generality (i.e., from the top of the hierarchy to the bottom). If at any point in the hierarchy an event model cannot be instantiated, all descendents can also not be instantiated by virtue of the implication chains. For example, if nothing moves, no overtake may take place:

NOT (MOVE) => NOT (OVERTAKE)

VERBALIZATION

We are interested now in clarifying the connection between events and natural language. We first consider bottom-up verbalization (i.e., the task of obtaining a verbal scene description from a GSD). In this section, we restrict our discussion to single utterances.

The main contribution of this section is to describe how case frames can be filled by events and other scene data, and how event recognition is triggered by a filled case frame in the context of question answering. The remaining problem of generating surface strings from deep case frames and vice versa is handled by processes that have been developed for the natural language system HAM-ANS. They are not discussed further here.

Figure 5.3 gives an overview of the major processing steps.

Filling Case Frames

The target structures into which the GSD and the events will be transformed for verbalization, are deep case frames for verbs of locomotion.

Given a GSD and an 'overtake' event recognized in the scene, how

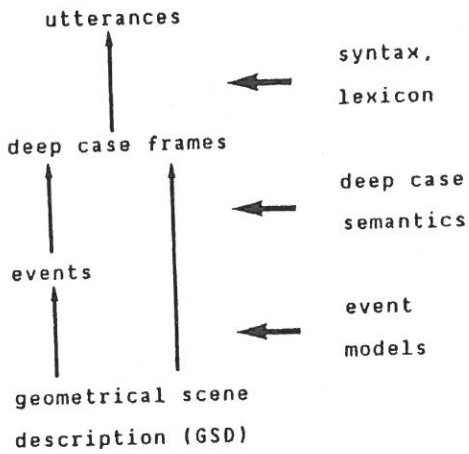

Figure 5.3. Overview of connection between vision and natural language system

can one obtain the deep case structure for a natural-language utterance that describes that event? Consider, for example,

"A yellow VW overtook a truck in front of the FBI."

(FBI is the German abbreviation of the Department of Computer Science.) The sentence is composed of the verb overtake, two noun phrases for the agent and objective cases, and a prepositional phrase for the locative case that also involves a noun phrase. In addition, there is temporal information expressed by the past tense. The referents of the deep cases can be easily expressed in terms of scene data. The verb refers to an 'overtake' event, the agent and objective cases to the two participating objects, the locative case to the locations taken up by the agent during the event, and finally the past tense to the temporal relation of the event interval to a reference time. Thus, filling a case frame for verbalization amounts to constructing references to the constituents of an event so that all constraints are satisfied.

Knowledge that governs this process is called *deep case semantics*. It is represented by case frame models as illustrated here for the example 'overtake.'

The case frame model for 'overtake' is as follows:

```
(VERB "overtake")
(OVERTAKE OBJ1 OBJ2 T1 T2)
(AGENT AGT-EXP)
(REF AGT-EXP OBJ1)
(OBJECTIVE OBJ-EXP)
(REF OBJ-EXP OBJ2)
(LOCATIVE LOC-EXP)
(LOC-REF LOC-EXP (LOC-PREP OBJ1 LOC-OBJ T1 T2))
(TENSE TNS-EXP)
(TIME-REF TNS-EXP T1 T2)
```


Each case description of the case-frame model consists of two parts: a declaration of an identifier (or constant in case of the verb) for the case expression on the language side, and a predicate (in general a list of predicates) relating the case expression to the scene data. The heart of the deep case semantics are the predicates REF and LOC-REF. They are now described in more detail.

REF relates a natural-language expression (in the format understood by the surface string generator or delivered by the parser) to a symbolic object identifier of the GSD. It works in two ways. Bottom-up, it generates suitable NL expressions for a scene object. This process is called *referencing*. Top-down, an NL expression is given and possible scene objects are generated. This is *dereferencing*. Referencing and dereferencing are well understood in NL research. We have adapted techniques developed in HAM-ANS. For example,

```
(REF AGT-EXP BUILDING1)
```

(where BUILDING1 is a certain scene object) will be evaluated as follows. First, the hearer model will be searched; the hearer model records all objects that are known to both speaker and hearer (e.g., by previous mention). If BUILDING1 is found and it has a name, for example "the FBI", its name can be used for reference. If BUILDING1 is found, but it has no name, then the definite article is used and referentiation is accomplished by retrieving class membership and possibly discriminating properties from the GSD. *If* BUILDING1 is not found in the hearer model, it must be newly introduced using the indefinite article, class membership, and discriminating properties similar to the definite case. For example, "a yellow VW" may be returned for a scene object VW1 which is a VW that has not been mentioned before. There are several other issues connected to referencing which are, however, outside the scope of this presentation (e.g., quantization, use of pronouns, and ellipses). The reverse process of dereferencing is discussed in the context of question answering.

LOC-REF is analogous to REF with the difference that an abstract location instead of an object is to be related to an NL expression. The locative case and other spatial deep cases such as source, path, and

goal are often misconstrued as referring to names of places or objects. With scene data as a referential data base the spatial deep cases can be defined concisely as follows. The locative case is the union of all positions of the agent during the event (i.e., the volume swept out by the agent's trajectory). To verbalize the locative case means to find a natural language expression referring to that volume. In NAOS we have only considered prepositional phrases so far. Hence, LOC-REF tries to find a reference object in the scene (LOC-OBJ) that is in a prepositional relation to the locative volume. Similarly, source and goal-deep cases are verbalized by relating the source or goal volume to a reference object using a suitable preposition. Note that after finding a reference object in the scene, REF is called to generate a natural language expression for this object.

The semantics of spatial prepositions are not trivial (see, for example, Boggess, 1979, Herskovits, 1985, Waltz & Boggess, 1979). In NAOS, we have currently implemented simplified versions of the most commonly used prepositions. For example, 'in-front-of,' 'behind,' and 'beside' simply test whether the second object is in the appropriate sector of the first object. Sectors originate at the centroid and extend in a fixed direction relative to the 'front' of an object.

We now turn to TIME-REF, which relates time intervals expressed in clock units of scene time to temporal expression in natural language. One effect of TIME-REF is the determination of tense. This is accomplished by comparing the interval boundaries with time marks separating the past from the present. The present time is held fixed and coincides with the end of the scene data. More sophistication will be required when the present time progresses as the description is generated. This is currently outside the scope of NAOS.

In addition to tense, temporal expressions can be used to specify the event interval. The problem of referencing interval boundaries is similar to referencing locations as there are in general no names attached. Hence, an indirect specification has to be generated by referring to a suitable item nearby—in this case in the temporal neighborhood. For example, a 'turn-off' event could be used to mark the beginning of the event in question:

"After the BMW turned off Schlueterstreet, the yellow VW . . ."

If events are described in chronological order, reference to preceding events is particularly easy, as one can use "then" or "after this" or rely on the implicit understanding that the end of one event marks the beginning of the next. This is elaborated further in the next section.

COMPOSING A DESCRIPTION

So far we have described methods for recognizing all events in a scene that match event models of a given repertoire, and for generating a natural-language utterance for a single event. We now consider the task of producing a coherent scene description. At several levels decisions have to be made that determine format and contents of the description. At the level of events, one has to select from a possibly large number of instances. For example, any of the events 'move,' 'slow down,' 'leave,' 'turn off,' or others could be selected for a description of roughly the same subspace of the scene. Furthermore, one also has to decide the order in which events should be presented.

At the case-frame level, there are choices concerning optional deep cases or alternative case fillers. For example, one could say

"A VW turned off"

(which may very well be adequate in certain scenarios) or

"A VW turned off from Schlueterstreet into Bieberstreet after the BMW had passed the FBI."

One might also consider adverbials or other modifiers, relative clauses, comparisons, and so on. Finally, decisions have to be made concerning such matters as voice, use of pronouns, connectives, ellipsis, and so on to obtain a pleasant and natural description.

Linguistic theories (see for example, Austin, 1962, or Cohen, 1978) view speech acts as purposeful and planned actions, designed to achieve certain goals. Planning such actions involves a hearer model, as the selection of an utterance must be done with regard to its effect on the hearer.

Speech-act theory is also a useful framework for the description task in NAOS. Our communication situation is very simple: A speaker/viewer 'informs' a hearer about a scene. Informing somebody means communicating things that are *true* and *new*. These are the two basic criteria on which speech act planning in NAOS is built. Other interesting criteria (e.g., focus and level of detail), are considered refinements that play a subordinate role.

What does it mean to convey 'truth'? We follow the semantics of Barwise and Perry (1983) where *meaning* is defined as a relation between utterances and situations. The interpretation of an utterance by a hearer is the set of possible situations connected to that

utterance via the meaning relation. Applied to scene description, we define an utterance to be true if its interpretation—the set of possible scenes—includes the actual scene. Thus, utterances leading to a misunderstanding are not true in this sense. Hence, if an utterance is to be true, the speaker must take into account the semantics of the hearer—he must take care that the hearer understands the right thing. In the uncomplicated world of NAOS it is assumed that the hearer always has the same semantics as the speaker, and as no false utterances are intended, the 'truth' criterion is always satisfied.

Considering the interpretations of the hearer is also fundamental for the second requirement, saying something new. As the interpretation of an utterance is defined in terms of possible scenes, a description that is composed of several utterances can be viewed as narrowing down the set of possibilities. This is—conceptually—carried out by set intersection. We define an utterance as new, if the set of possible scenes is strictly reduced by that utterance. One may equivalently speak about a partially specified scene instead of a set of possible scenes. Using this notion, an utterance is new if it conveys additional specifications for a partially specified scene.

In NAOS, a completely specified scene is available to the speaker in terms of the GSD. Thus, the speaker has a representation of what he tries to convey to the hearer. In addition, he needs a representation of what has been achieved at a given time (i.e., what the hearer has learned about the scene from the description received so far). We call this representation a *visualized geometrical scene description* (VGSD), as it can be considered the output of some kind of visualization process.

A comparison of the GSD and the VGSD is the heart of speech-act planning. Informing the hearer means causing his VGSD to approach the GSD. Planning a speech act that informs means anticipating its effect on the hearer's VGSD. This can be done by simulating the hearer's visualization process. We call this method of speech act planning *visualization anticipation*. The general idea is illustrated in Figure 5.4.

The major new component of this scheme is the VGSD. Structure and contents of the VGSD are not at all obvious, as visualized data are in many respects different from the scene data represented by the GSD, but still should be comparable. Internal representations of visualized scene data have been the topic of a long-standing debate among philosophers, psychologists, linguists and, lately, researchers from cognitive science and AI. (See Block, 1981 or Yuille 1983, for recent contributions to this so-called 'imagery debate)').

It would be beyond the scope of this chapter to discuss our approach relative to the positions taken up in that debate. Our position is pragmatic. The nature of the internal scene representation follows from the tasks which this representation must support. In our case the emphasis is on making two representations comparable, the VGSD that is derived from a NL description, and the GSD that is derived from visual input. Although our task is speech-act planning, we believe that several other tasks such as spatial reasoning or remembering scenes will also require similar representations.

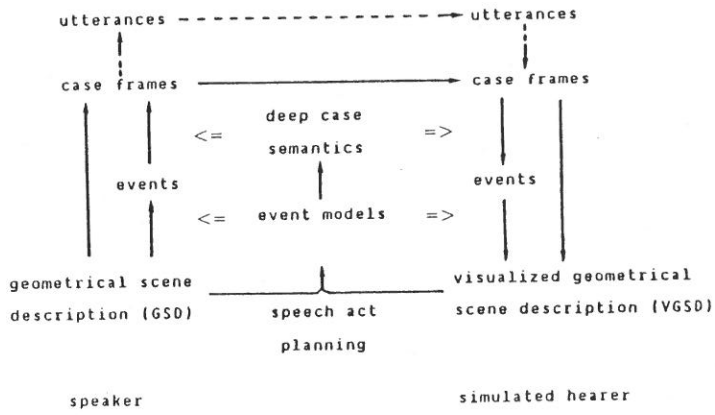

Figure 5.4. Visualization anticipation for speech-act planning

As indicated in Figure 5.4, the VGSD is generated, roughly speaking, by inverting the verbalization path. We assume that case frames can be completely recovered from utterances. Hence, the visualization process begins at the level of case frames. From a case frame, propositions about the scene can be derived using deep-case semantics and event models, following a procedure similar to the top-down event instantiation process described in the previous section. The major difference is, of course, that there is no referential database (the GSD) on which to instantiate the variables of a predicate. Instantiation takes place by creating ("visualizing") scene tokens which satisfy the predicates. Hence, from

"A yellow VW overtook a large truck"

the following primary VGSD propositions are derived:

(CLASS VW1 VW)
(COLOR VW1 YELLOW)
(CLASS TRUCK1 TRUCK)

```
(SIZE TRUCK1 LARGE)
(OVERTAKE VW1 TRUCK1 T1 T2)
        (MOVE VW1 T1 T2)
        (MOVE TRUCK1 T1 T2)
        (APPROACH VW1 TRUCK1 T1 T3)
                (SYM-APPROACH VW1 TRUCK1 T1 T3)
                (MOVE VW1 T1 T3)
                (IN-FRONT-OF TRUCK1 VW1 T1 T3)
        (BESIDE VW1 TRUCK1 T3 T4)
                *
                *
                *
```

Here, VW1 and TRUCK1 are visualized objects. The time variables are constrained according to the past tense of the utterance or whatever temporal information can be extracted from the context of the utterance. The predicates have been expanded down to the level of primitives, and it should be clear that knowledge about the procedural definition of primitives is also available. The primary propositions implicitly define possible scenes compatible with the verbal description.

This is not all there is to a VGSD. We believe that considerable detail must be added by tapping a body of knowledge that has not yet been mentioned in our discussion: knowledge about 'typical events.' A typical event is a scene description composed of an event and of information that is typical for the event but not implied by the event model. We have developed an analogical event representation for this purpose (Mohnhaupt & Neumann, 1988) which is associated with a propositional event model as introduced before and essentially provides typical trajectories in space and time.

There is much to say about typical events, their use in NL understanding and reasoning, their relationship to beliefs, their acquisition by experience, and other interesting issues. This is, however, outside the scope of this chapter. In NAOS, typical event knowledge is used to supply scene data —expectations— beyond the primary propositions implied by the event models. Expectations have a modality different from primary scene data because they need not be true. "Contradictory" primary information overrides (or replaces) expectations. Both primary scene and expectations are part of the VGSD.

We have outlined the process of 'visualization' now. With this process the task of 'informing' is well-defined: describe a scene in

such a way that the VGSD approaches the GSD in successive refinements. In the NAOS system we have not yet developed a speech-planning component that actually computes a VGSD. But we have devised a standard plan for the special task of giving a complete, coherent description of all moving objects. The plan is based on the general idea of anticipated visualization, as it monitors to which extent the trajectory of a moving object has been specified so far, and generates further utterances accordingly.

Standard Plan for Scene Description

The first utterance is a standardized summary of all moving objects, using the template:

> "There are ⟨N⟩ moving objects in the scene: ⟨N1⟩ ⟨class-1⟩, ⟨N2⟩ ⟨class-2⟩, . . . , and ⟨NM⟩ ⟨class-M⟩."

Then a chronological description of each individual moving object is given. The following example is a translation of a German language text generated automatically for the scene in Fig. 5.2 (Neumann & Novak 86).

> The scene contains four moving objects: three cars and one pedestrian.
>
> A VW drives from the old post office to the FBI. It stops.
>
> Another VW drives toward Dammtor station. It turns off Schlueterstreet. It drives on Bieberstreet toward Grindelhof.
>
> A BMW drives toward Hallerplatz. Thereby it overtakes the VW which has stopped at Bierberstreet. The BMW stops at the traffic lights.
>
> The pedestrian goes toward Dammtor station. He crosses Schlueterstreet in front of the FBI.

The description of an object is generated according to the following rules:

1. The trajectory of the object is described completely except where not visible.
2. The trajectory is described in terms of the "most special events" according to the event hierarchy. This is done by choosing the most special event covering a time interval from the beginning of the scene to some time point, and then proceeding from this time point in the same manner until the trajectory is completely covered.

3. The spatio-temporal location of an event in four-dimensional space-time is specified as follows:
   (a) Spatial deep cases are used except where the location has already been specified by the preceding event or the verb does not allow it. If necessary, additional—less special—events are verbalized to provide spatial information.
   (b) Time specifications are only exceptional. As a rule the temporal distribution of events is given as follows:

- The first event begins at the beginning of the scene if not specified otherwise.
- Events described thereafter immediately succeed the preceding event (chronological coverage).
- The duration of an event follows from location specifications and a typical velocity in case of durative events (e.g., 'walk'). For nondurative events a typical duration is assumed. Exceptional cases arise if no locomotion takes place ('stand') or standard values deviate too much from the actual ones. For the latter case one can use linguistic means to modify standard values, for example adverbials ('quickly'). If it is necessary to explicitly specify a time point, this is done by referring to time points that have already been specified. All start and end times of events that have been previously mentioned according to the standard plan, are considered specified.

Note that temporal specifications rely heavily on typical event data. Thus, knowledge of typical events—common to speaker and hearer—is shown to play a significant part in scene description. Further details about text generation in NAOS are given in Novak (1986).

Question Answering

The possibility of putting questions to a vision system and obtaining answers is certainly an interesting perspective, as questions may radically reduce the required processing compared to general-purpose scene analysis. In NAOS, top-down processing can only be demonstrated down to the level of the GSD. But other systems show the effectiveness of top-down constraints below this level (Brooks, 1983).

Consider the decision question:

"Did a yellow VW overtake a truck in front of the FBI?"

To answer this question, the following steps are executed. First, all noun phrases are dereferenced. That is, REF is called with a noun phrase as input and generates scene objects that fit the description. For example, a set of yellow VWs: VW1, VW3, . . . may be determined as the range of OBJ1. If no such objects exist, an answer such as

"There is no yellow VW in the scene"

will be generated. In general, the second step is a quantization test on referents. Consider the question:

"Did the BMW overtake two trucks?"

If a BMW has not been previously mentioned, the definite article implies that there is exactly one BMW in the scene. Also, there must be at least two trucks. If the quantization test fails, an appropriate answer will be generated.

Although dereferentiation of objects is performed *before* event recognition, all other constraints of a question are evaluated in connection with the event model. LOC-REF takes a locative expression (e.g., "in front of the FBI") and generates the appropriate scene constraint, for example, (IN-FRONT-OF OBJ1 LOC-OBJ T1 T2) with BUILDING1 (which is the FBI) bound to LOC-OBJ. Spatial deep case expressions supplied by the parser are transformed into constraints in a similar way. TIME-REF generates constraints for the interval boundaries. For example, from the past tense of the question one gets

$$\text{T-PAST-BEG} \leq T1 < T2 \leq \text{T-PAST-END}$$

where the boundary values are fixed time marks as mentioned earlier. Hence, from the first of the two questions one gets the predicates:

(OVERTAKE OBJ1 OBJ2 T1 T2)
(IN-FRONT-OF OBJ1 OBJ3 T1 T2)

with the following ranges attached to the variables:

OBJ1 = {VW1, VW3, . . .}
OBJ2 = {TRUCK1, TRUCK 2, . . .}
OBJ3 = {BUILDING1}
T1 = (T-PAST-BEG T-PAST-END−1)
T2 = (T-PAST-BEG+1 T-PAST-END)

These constitute a set of constraints on events that can satisfy the question.

Event recognition now takes place as described in the next section. A trace of this process is presented in Appendix B.

Event Recognition

We now turn to the process of event recognition, which is the search of the GSD for events that could match the constraints previously generated. For bottom-up scene description the search is unconstrained.

The GSD and all facts computed about the scene are kept in an associative database. The availability of an associative net was one of the reasons for selecting the programming language FUZZY for our implementation. The basic techniques for event recognition are hierarchical matching (Barrow, Ambler, & Burstall, 1972) and backtracking search. The scheme used in NAOS is particular in several ways as becomes apparent here. The following shows the search strategy of EVENTEVAL. This is the component of NAOS that tries to instantiate a list of predicates with range restrictions with the goal of making all predicates conjunctively true.

EVENTEVAL list of predicates:

- SELECT predicate from list.
- GENERATE all instances.
- Select instance and TEST for compatability.
- Backtrack if not compatible, else
- EVENTEVAL remaining predicates.

Before commenting on this procedure, it is necessary to consider the GENERATE component in more detail. The following steps are carried out:

GENERATE all instances of a predicate:

- Generate all instances of non-instantiated arguments except time variables. Each combination of such instances defines a predicate 'pattern.'
- Skip predicate pattern if generated before,
- EVENTUAL body if predicate is composite, else EVAL body.

GENERATE cycles through all patterns of a predicate by substituting possible instances for noninstantiated variables except for those de-

noting time intervals. The variable range restrictions are used to generate potential instances. There are provisions for avoiding duplicate computations by keeping a history of all patterns which have been tried before. Evaluation is either carried out by a recursive call of EVENTEVAL or by EVAL that deals with primitive predicates. Each evaluation of a pattern generates all time intervals for which the pattern is true. EVAL can be characterized as follows.

EVAL primitive predicate:

- Compute all maximal time intervals, for which the predicate is true.
- Enter instances into database.

The computations of EVAL are carried out using data of the GSD or facts of the knowledge base. In its simplest form the computation of a primitive predicate is a direct retrieval from the GSD (e.g., CLASS or COLOR). But there are also primitives which require more processing (e.g., SYM-APPROACH, BESIDE).

From the structure of EVENTEVAL and GENERATE one can see that event recognition proceeds in a doubly recursive manner: by recursively instantiating a list of predicates and by recursively decomposing predicates. This should be kept in mind when studying the trace of the 'overtake' example in Appendix B.

So far, very little has been said about instantiating time intervals. Time intervals are different from other data in that they are represented by constraints rather than fixed instances. Consider the proposition

(MOVE CAR1 10 30)

As MOVE is durative, the interval boundaries 10 and 30 have to be interpreted as constraints marking the range of possible subintervals. Hence if a predicate

(MOVE OBJ1 T1 T2)

is matched against the MOVE data, OBJ1 is instantiated to CAR1, whereas T1 and T2 are only constrained:

$$10 \leq T1 < T2 \leq 30.$$

As more predicates involving the same time variables are instantiated, more constraints accumulate.

For the following example we assume that all 'move' events have been computed in an initialization step and entered into the database. (This is the usual procedure in NAOS.) Consider the data

(MOVE CAR1 1 30)
(MOVE CAR2 7 13)
(MOVE CAR2 20 35)
(IN-FRONT-OF CAR2 CAR1 15 27)

and the list of predicates (taken from the event model 'overtake')

(MOVE OBJ1 T1 T2)
(MOVE OBJ2 T1 T2)
(IN-FRONT-OF OBJ2 OBJ1 T1 T3)

One possible instantiation would give rise to the inequalities:

$1 \leq T1 < T2 \leq 30$
$20 \leq T1 < T2 \leq 35$
$15 \leq T1 < T3 \leq 27.$

Assuming for the sake of simplicity that the three propositions amount to a complete 'overtake' event, what are the temporal constraints of this event? We can solve the inequalities by inspection and get

$20 \leq T1 \leq 26$
$21 \leq T2 \leq 30.$

The resulting 'overtake' event is recorded using the notation:

(OVERTAKE CAR1 CAR2 (20 26) (21 30))

Hence, the general form for writing a constrained time interval with beginning T-BEG and end T-END is

(T-BEG-MIN T-BEG-MAX) (T-END-MIN T-END-MAX)

For durative predicates, the minimal and maximal values coincide except of the fact that zero intervals are not allowed. Thus, the notation introduced earlier

(. . . T1 T2)

is equivalent to

(. . . (T1 T2-1) (T1+1 T2)).

Inchoative predicates are written

(. . . T1 (T1+1 T2))

which is equivalent to

(. . . (T1 T1) (T1+1 T2)).

Similarly, for resultative predicates

(. . . (T1 T2-1) T2)

is equivalent to

(. . . (T1 T2-1) (T2 T2)).

Note that another instantiation for the example data would result in an inconsistent set of inequalities. Checking the current time constraints for consistency is the task of the TEST component of EVENTEVAL. If the test fails, backtracking ensues and other instantiations are selected.

The fact that recognition of temporal events involves feasibility test and solution of a set of linear inequalities has also been observed in Malik and Binford (1982). They suggest linear programming, in particular the SIMPLEX method, to obtain the desired results. In NAOS, a much simpler procedure is employed. It is based on an inequality net that is maintained for all time variables. Each variable has a current minimum and maximum value and is linked to other variables according to the inequalities. This is shown in Fig. 5.5 for the example used earlier.

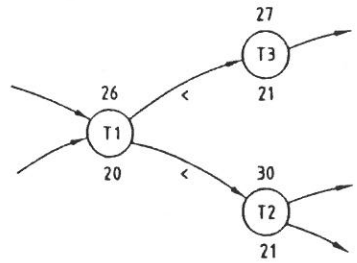

Figure 5.5. Inequality net for time variables

If a new inequality is encountered, new links are added, and the new constraints are propagated along the links, lower bounds upward and upper bounds downward. Whenever a minimum surpasses a maximum, the inequalities are inconsistent. Otherwise, minimum and maximum are valid bounds and provide the desired solution. Note that all entries are subject to backtracking.

We believe that this incremental constraint propagation method is a key element for dealing with temporal concepts in scenes. It reflects the fact that the basic building blocks of interesting concepts are scene properties extending over some time interval (i.e., durative properties). Taken together, they give rise to systems of inequalities as shown above, and to concepts which need not be durative (e.g., 'overtake' or 'stop').

In conclusion of the discussion of event recognition we describe the SELECT component of EVENTEVAL, as this is another unusual feature of NAOS. SELECT determines the order in which a list of predicates is instantiated. It should be intuitively clear that the size of the search tree and hence the computing costs depend on this order. For example, consider finding all 'stop' events. As 'stop' is composed of 'move' and 'stand,' the question is which to evaluate first. Assuming that no a priori knowledge about motility is available and that there are many more stationary objects than moving ones, it would be advisable to first find moving objects and then the standing ones that were moving earlier.

In NAOS the optimal order for evaluating predicates is determined based on the intrinsic branching factor associated with each predicate (this is a priori knowledge about the likelihood of its occurrence), the actual number of patterns of each predicate (given the variable ranges), and the cost for evaluating a predicate. From this a score is computed which favors predicates with the least chance of success and the smallest cost. After each predicate evaluation the score is recomputed for the remaining predicates (see Neumann, 1984 for details).

DISCUSSION

In this section we compare our work on natural-language motion description with related approaches. We also indicate in which direction current work in NAOS is proceeding.

Our work can be considered as a contribution to high-level interpretation of image sequences. Image sequences with motion have not yet been studied for a long time—the 1979 Workshop on Computer Analysis of Time-Varying Imagery in Philadelphia marked the beginning of general interest in this area. One of the early contributions is the pioneering work of Badler (1975). He investigated the recognition of motion concepts in computer-generated line drawings. His conceptual units are oriented toward natural-language verbs ('swing,' 'bounce') and directional adverbials ('forward,' 'upward'). The focus of Badler's work, however, is not on generating a natural language output but on segmenting trajectories into motion primitives which can be combined into complex motion patterns.

The work of Tsotsos (Tsotsos, 1980; Tsotsos, Mylopoulos, Covvey, & Zucker, 1980) builds on some of the motion concepts developed by Badler, but also improves on Badler's framework in several respects. Most importantly, a hierarchy of domain-independent motion concepts is defined (e.g., 'area-expand,' 'rotate') from which domain-specific motion concepts (e.g., 'posterior-rapid fill,' which is a special left ventricular motion in Tsotsos' domain) can be constructed. Tsotsos also presented a data-driven recognition strategy where hypotheses are generated according to conceptual proximity. Proximity is based on ISA, PART-OF, SIMILARITY, and TEMPORAL-NEXT links in the conceptual database. The implication hierarchy in NAOS plays a similar role: Implication links are used to prevent useless instantiation attempts.

It is interesting to note that Tsotsos, who did not attempt to provide natural language output, based his motion concepts on categories developed by Miller (1972) for motion verbs of the English language. Tsotsos realized that some of Miller's categories cannot be easily incorporated in a vision system (e.g., 'causative' or 'permissive' motion), whereas others have natural visual correlates (e.g., 'inchoative' motion). This was also observed by Marburger, Neumann, & Novak (1981), which is the first report on the NAOS project.

A different set of motion concepts underlies the system SUPP (Okada, 1980), which produces sentences from a short sequence of line drawings showing, for example a bird landing on a tree, a man entering a car, and so on. Okada used 20 semantic features (e.g., 'displacement,' 'deformation,' 'change in quality,' 'start and stop') to decide which of a set of about 1,200 primitive verb concepts applies to a given scene. Usually, many concepts qualify and give rise to as many simple sentences describing the same event. Okada's work seems to be influenced by the feature space paradigm of pattern recognition. It suffers from the lack of structure of feature vector semantics.

There is one issue common to all this work and also to NAOS that is worth emphasizing: the question of what motion concepts should

be considered a conceptual unit. Tsotsos stressed the role of taxonomical hierarchies (ISA, PART-OF), hence motion concepts are decompositions of a conceptual space. In NAOS, the conceptual units—events—are subspaces of scenes. They are considered units by virtue of corresponding natural language concepts. Taxonomical units are useful building blocks if parsimonious representations are desired. Events are useful for pragmatical reasons. A combination of both may be advantageous as exemplified by certain event models in NAOS.

The choice of verb-oriented event models has several consequences which are discussed in Neumann and Novak (1983). For example, one is led to deal with concepts that express more than actually can be observed in the subspace of a scene associated with the verb. A striking example is "continue walking," which is the translation of one of the meanings of the German verb "weitergehen." It denotes an uninterrupted walk where stopping has been expected (e.g., "He continued walking in spite of the approaching car"). To recognize such an event, one obviously has to generate expectations about the development of the scene. There are many other verbs of this kind and also other ways of expressing expectations in natural language, for example by negative statements ("The car did not stop"). A framework for generating expectations in NAOS has been devised (Retz-Schmidt 1985). The main idea is to employ script-like conceptual units that represent knowledge about typical sequences of events, for example about typical behavior at traffic lights. Scripts constitute a level of representation "above" events, much in accord with structures proposed in Waltz (1981). Partially instantiated scripts give rise to expectations. Event models may consist of expectations as well as other predicates about the scene.

On the natural-language side our work is influenced by the early NL dialogue system HAM-RPM (von Hahn, Hoeppner, Jameson, & Wahlster, 1980) and its successor HAM-ANS (Hoeppner, Morik, & Marburger, 1984). One of the domains of HAM-ANS is a street scene (in fact: the same street as in NAOS) which has been used to study various NL issues arising from a dialogue about a scene (e.g. focus of attention and referencing). Working close to the HAM-ANS research group was an interesting experience, as they approached NL scene description from the language side, whereas our work is vision oriented. On several occasions we found our ideas and concepts competing with existing linguistic notions, for example when defining deep cases. In the authors's opinion, vision—or for that matter: the physical nature of a scene—is the easier side from which to investigate NL scene description.

ACKNOWLEDGMENTS

The author wishes to acknowledge the valuable work of Hans-Joachim Novak who was the main collaborator in this part of project NAOS. The project has been partially supported by the German Science Foundation (DFG).

REFERENCES

Allen, J. F. (1981). *A general model of action and time.* Technical Report 97, University of Rochester, Rochester, NJ.

Austin, J. L. (1962). *How to do things with words.* New York: Oxford University Press.

Badler, N. I. (1975). *Temporal scene analysis: Conceptual descriptions of object movements.* Report TR 80, Department of Computer Science, University of Toronto, Toronto/Canada.

Ballard, D. H., & Brown, C. M. (1983). *Computer vision.* Englewood Cliff, NY: Prentice Hall.

Barrow, H. G., Ambler, A. P., & Burstall, R. M. (1972). Some techniques for recognizing structures in pictures. In J. K. Aggarwal, R. O. Duda, & A. Rosenfeld (Eds.), *Computer methods in image analysis,* (pp. 397–425). IEEE Press.

Barwise, J., & Perry, J. (1983). *Situations and attitudes.* Cambridge, MA: The MIT Press.

Block, N. (Ed.) (1981). *Imagery.* Cambridge, MA: The MIT Press.

Boggess, L. C. (1979). *Computational interpretation of English spatial prepositions.* University of Illinois-Urbana, Illinois.

Brooks, R. A. (1981). Symbolic Reasoning Among 3-D Models and 2-D Images. In J. M. Brady (Ed.), *Computer vision* (pp. 285–348). Amsterdam: North-Holland.

Brooks, R. A. (1983). *Model-based three-dimensional interpretations of two-dimensional images.* IEEE Trans. PAMI 5 (pp. 140-150).

Busemann, S. (1984). Surface transformations during the generation of written German sentences. In Bolc. L. (Ed.), *Natural language generation systems.* Heidelberg: Springer.

Cohen, P. R. (1978). *On knowing what to say: Planning speech acts.* Ph.D. Thesis, Dept. of Computer Science, University of Toronto.

Dreschler, L., & Nagel, H.-H. (1981). *Volumetric model and 3D-trajectory of a moving car derived from monocular TV-frame sequences of a street scene.* Proceedings of the IJCAI-81 (pp. 692–697). Vancouver, B.C., Canada: University of British Columbia.

Fillmore, C. J. (1968). The case for case. In Bach, E., & Harms, R. T. (Eds.) *Universals in linguistic theory* (pp. 1–88). New York: Holt, Rinehart, and Winston.

Herskovits, A. (1980). *On the spatial uses of prepositions.* Proceedings 18th Annual Meeting ACL, 1-5, Philadelphia, PA.

Herskovits, A. (1985). Semantics and pragmatics of locative expressions. *Cognitive Science*, 9, (pp. 314–378).

Hoeppner, W., Christaller, T., Marburger, H., Morik, K., Nebel, B., O'Leary, M., & Wahlster, W. (1983). *Beyond domain independence: Experience with the development of a German language access system to highly diverse background systems.* Proceedings of the IJCAI-83 (pp. 588–594). Los Altos, CA.: M. Kaufmann.

Hoeppner, W., Morik, K., & Marburger, H. (1984). *Talking it over: The natural language dialog system HAM-ANS.* (Report ANS-26) Hamburg: Research Unit for Information Science and Artificial Intelligence, Hamburg.

Malik, J., & Binford, T. O. (1982). *Representation of time and sequences of events.* Proceedings of a Workshop on Image Understanding (pp. 15-16). Palo Alto, CA.

Marburger, H., Neumann, B., & Novak, H.-J. (1981). *Natural language dialogue about moving objects in an automatically analyzed traffic scene.* Proceedings of the IJCAI-81 (pp. 49-51). Vancouver, B.C., Canada: University of British Columbia.

Marr, D. (1981). *Vision.* San Francisco, CA.: Freeman.

Miller, G. A. (1972). English verbs of motion: A case study in semantics and lexical memory. A. W. Melton & E. Martin (Eds.), *Coding Processes in Human Memory* (pp. 335–372). Washington D.C.: V. H. Winston and Sons.

Miller, G. A., & Johnson-Laird, P. N. (1976). *Language and perception.* Cambridge, MA.: Cambridge University Press.

Mohnhaupt, M., & Neumann, B. (1988). *Some aspects of learning and reorganization in an analogical representation.* Proceedings International Workshop on Knowledge Representation and Knowledge (Re)organisation in Machine Learning, K. Morik (Ed.). Heidelberg: Springer.

Nagel, H. H., & Rekers, G. (1982). *Moving object masks based on an improved likelihood test.* Proceedings of the ICPR-82 (1140-1142). Silver Spring, MD.: IEEE Computer Society Press.

Neumann, B., & Novak, H.-J. (1983). *Natural language oriented event models for image sequence interpretation: The issues.* (CSRG Technical Note 34) Toronto, Canada: Department of Computer Science, University of Toronto.

Neumann, B., & Novak, H.-J. (1986). NAOS: *Ein System zur natuerlichsprachlichen Beschreibung zeitveraenderlicher Szenen.* Informatik Forschung und Entwicklung, 1, 83–92. Heidelberg: Springer.

Neumann, B. (1984). *Natural language description of time-varying scenes.* (Report FBI-HH-B-105/84). Hamburg: Fachbereich Informatik, Universitaet Hamburg.

Novak, H.-J. (1986). *Generating a coherent text describing a traffic scene.* Proc. COLING (pp. 570-575).

Okada, N. (1980). *Conceptual taxonomy of Japanese verbs for understanding natural language and picture patterns.* Proceedings COLING-80 (pp. 127-135).

Retz-Schmidt, G. (1985). *Script-based generation and evaluation of expectations in traffic scenes.* (Report FBI-HH-M-136/85) Hamburg: Fachbereich Informatik, Universitaet Hamburg.

Searle, J. (1969). *Speech acts: An essay in the philosophy of language.* Cambridge, MA: Cambridge University Press.

Tsotsos, J. K. (1980). *A framework for visual motion understanding.* (TR CSRG-114) Department of Computer Science, University of Toronto, Toronto, Canada.

Tsotsos, J. K., Mylopoulos, J., Covvey, H. D., & Zucker, S. W. (1980). *A framework for visual motion understanding.* Proceedings IEEE Trans. Pattern Analysis and Machine Intelligence PAMI-2 (pp. 563-573).

von Hahn, W., Hoeppner, W., Jameson, A., & Wahlster, W. (1980). The anatomy of the natural language dialogue system HAM-RPM. In L. Bolc (Ed.) *Natural language based computer systems,* (pp. 119–253). Muenchen: Hanser/McMillan.

Waltz, D. L. (1979). *Relating images, concepts, and words.* Philadelphia, PA.: Proc. NSF Workshop on the Representation of Three-Dimensional Objects, R. Bajcsy (Ed.). Philadelphia, PA.

Waltz, D. L. (1981). *Toward a detailed model of processing for language describing the physical world.* Proceedings of the IJCAI-81 (pp. 1-6). Los Altos, CA.: M. Kaufman.

Waltz, D. L., & Boggess, L. C. (1979). *Visual analog representation for natural language understanding.* Proceedings of the IJCAI-79. Los Altos, CA.: M. Kaufman.

Yakimovsky, Y., & Cunningham, R. (1978). *A system for extracting three-dimensional measurements from a stereo pair of TV cameras.* Computer Graphics and Image Processing, 7, 195–210.

Yuille, J. C. (Ed.) (1983). *Imagery, memory, and cognition.* Hillsdale, NJ: Lawrence Erlbaum Associates.

APPENDIX A

Event Hierarchy in NAOS

P = parents (or predecessors)
S = sons (or successors)

abbiegen (turn off)
P: drehen (turn)
S: —

abfahren (depart)
P: beschleunigen (accelerate), halten (halt)
S: —

anfahren (start driving)
P: beschleunigen (accelerate), halten (halt)
S: —

anhalten (stop)
P: bremsen (slow down), halten (halt), stehenbleiben-1 (stop)
S: einparken (park)

ankommen (arrive)
P: herankommen (come near)
S: —

ausweichen (yield, avoid)
P: bewegen (move)
S: —

begegnen (meet)
P: naehern-r (approach)
S: treffen-r (meet)

beschleunigen (accelerate)
P: fahren (drive)
S: abfahren (depart), anfahren (start driving), losfahren (start driving)

betreten (tread on)
P: gehen (walk)
S: —

bewegen (move)
P: existieren (exist)
S: ausweichen (yield, avoid), drehen (turn), entfernen-r (recede), fahren (drive), folgen (follow), gehen (walk), kommen (come), laufen (run), naehern-r (approach), rasen (speed), stehenbleiben-1 (stop), ueberqueren (cross)

bremsen (slow down)
P: fahren (drive)
S: anhalten (stop), stoppen (stop)

drehen (turn)
P: bewegen (move)
S: abbiegen (turn off), einbiegen (turn into), umdrehen (turn round)

einbiegen (turn into)
P: drehen (turn)
S: —

einholen (catch up with)
P: naehern-r (approach)
S: —

einparken (park)
P: anhalten (stop), parken (park), stoppen (stop)
S: —

entfernen-r (recede)
P: bewegen (move)
S: passieren (pass), verlassen (leave), vorbeifahren (drive past), vorbeigehen (go past), vorueberfahren (drive past), voruebergehen (go past), wegfahren (drive off)

erreichen (reach)
P: naehern-r (approach)
S: ueberqueren (cross)

existieren (exist)
P: —
S: bewegen (move), stehen (stand)

fahren (drive)
P: bewegen (move)
S: beschleunigen (accelerate), bremsen (slow down), hinterherfahren (drive behind, follow), rasen (speed), vorbeifahren (drive past), vorueberfahren (drive past), wegfahren (drive off), weiterfahren-2 (continue driving)

folgen (follow)
P: bewegen (move)
S: hinterherfahren (drive behind, follow)

gehen (walk)
P: bewegen (move)
S: betreten (tread on), losgehen (start walking), vorbeigehen (go past), voruebergehen (go past), weggehen (go off), weitergehen-2 (continue walking)

halten (halt)
P: stehen (stand)
S: abfahren (depart), anfahren (start driving), anhalten (stop), losfahren (start driving), parken (park), stoppen (stop), wegfahren (drive off)

herankommen (come near)
P: kommen (come)
S: ankommen (arrive)

hinterherfahren (drive behind, follow)
P: fahren (drive), folgen (follow)
S: —

kommen (come)
P: bewegen (move)
S: herankommen (come near)

laufen (run)
P: bewegen (move)
S: rennen (run fast)

losfahren (start driving)
P: beschleunigen (accelerate), halten (halt)
S: weiterfahren-1 (resume driving)

losgehen (start walking)
P: gehen (walk), stehen (stand)
S: weitergehen-1 (resume walking)

naehern-r (approach)
P: bewegen (move)
S: begegnen (meet), einholen (catch up with), erreichen (reach), passieren (pass), vorbeifahren (drive past), vorbeigehen (go past), vorueberfahren (drive past), voruebergehen (go past)

parken (park)
P: halten (halt)
S: einparken (park)

passieren (pass)
P: naehern-r (approach), entfernen-r (recede)
S: vorbeifahren (drive past), vorbeigehen (go past), vorueberfahren (drive past), voruebergehen (go past)

rasen (speed)
P: bewegen (move), fahren (drive)
S: —

rennen (run fast)
P: laufen (run)
S: —

stehen (stand)
P: existieren (exist)
S: halten (halt), losgehen (start walking), stehenbleiben-1 (stop), stehenbleiben-2 (remain standing), warten (wait), weggehen (go off), weitergehen-1 (resume walking)

stehenbleiben-1 (stop)
P: bewegen (move), stehen (stand)
S: anhalten (stop), stoppen (stop)

stehenbleiben-2 (remain standing)
P: stehen (stand)
S: —

stoppen (stop)
P: bremsen (slow down), halten (halt), stehenbleiben-1 (stop)
S: einparken (park)

treffen-r (meet)
P: begegnen (meet)
S: —

ueberholen (overtake)
P: vorbeifahren (drive past), vorueberfahren (drive past)
S: —

ueberqueren (cross)
P: bewegen (move), erreichen (reach), verlassen (leave)
S: —

umdrehen (turn round)
P: drehen (turn)
S: umkehren (return), wenden (turn, make a u-turn)

umfahren (drive round)
P: vorbeifahren (drive past), vorueberfahren (drive past)
S: —

umgehen (walk round)
P: vorbeigehen (go past), voruebergehen (go past)
S: —

umkehren (return)
P: umdrehen (turn round)
S: —

verlassen (leave)
P: entfernen-r (recede)
S: ueberqueren (cross)

vorbeifahren (drive past)
P: entfernen-r (recede), fahren (drive), naehern-r (approach), passieren (pass)
S: umfahren (drive round), ueberholen (overtake)

vorbeigehen (go past)
P: entfernen-r (recede), gehen (walk), naehern-r (approach), passieren (pass)
S: umgehen (walk round)

vorueberfahren (drive past)
P: entfernen-r (recede), fahren (drive), naehern-r (approach), passieren (pass)
S: umfahren (drive round), ueberholen (overtake)

voruebergehen (go past)
P: entfernen-r (recede), gehen (walk), naehern-r (approach), passieren (pass)
S: umgehen (walk round)

warten (wait)
P: stehen (stand)
S: —

wegfahren (drive off)
P: entfernen-r (recede), fahren (drive), halten (halt)
S: —

weggehen (go off)
P: gehen (walk), stehen (stand)
S: —

weiterfahren-1 (resume driving)
P: losfahren (start driving)
S: —

weiterfahren-2 (continue driving)
P: fahren (drive)
S: —

weitergehen-1 (resume walking)
P: losgehen (start walking), stehen (stand)
S: —

weitergehen-2 (continue walking)
P: gehen (walk)
S: —

wenden (turn, make a u-turn)
P: umdrehen (turn round)
S: —

APPENDIX B

Overtake Example

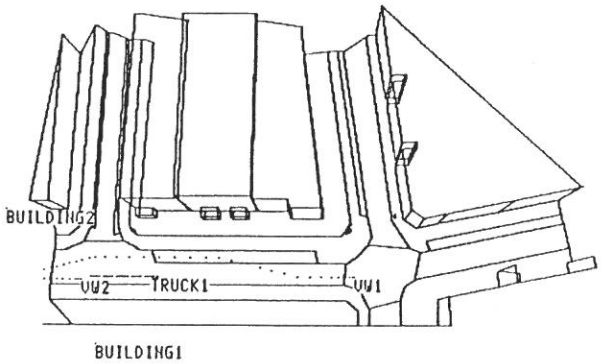

Figure 5.6. Synthetic view of 'overtake' example

Geometrical scene description (GSD) for 'overtake' example:

((CLASS BUILDING1 BUILDING) . 1)
((CLASS BUILDING2 BUILDING) . 1)
((CLASS TRUCK1 TRUCK) . 1)
((CLASS VW1 VW) . 1)
((CLASS VW2 VW) . 1)
((NAME BUILDING1 FBI) . 1)
((NAME BUILDING2 (OLD POST)) . 1)
((SIZE TRUCK1 LARGE) . 1)
((COLOR VW1 YELLOW) . 1)
((COLOR VW2 BLACK) . 1)
((LOCATION BUILDING1 (100 -60 70) (0 1 0) 1 40) . 1)
((LOCATION BUILDING2 (-200 350 80) (0 -1 0) 1 40) . 1)
((LOCATION TRUCK1 (50 50 15) (1 0 0) 1 2) . 1)
((LOCATION TRUCK1 (60 50 15) (1 0 0) 2 3) . 1)
((LOCATION TRUCK1 (70 50 15) (1 0 0) 3 4) . 1)
((LOCATION TRUCK1 (80 50 15) (1 0 0) 4 5) . 1)
((LOCATION TRUCK1 (90 50 15) (1 0 0) 5 6) . 1)

((LOCATION TRUCK1 (100 50 15) (1 0 0) 6 7) . 1)
((LOCATION TRUCK1 (110 50 15) (1 0 0) 7 8) . 1)
((LOCATION TRUCK1 (120 50 15) (1 0 0) 8 9) . 1)
((LOCATION TRUCK1 (130 50 15) (1 0 0) 9 10) . 1)
((LOCATION TRUCK1 (140 50 15) (1 0 0) 10 11) . 1)
((LOCATION TRUCK1 (150 50 15) (1 0 0) 11 12) . 1)
((LOCATION TRUCK1 (160 50 15) (1 0 0) 12 13) . 1)
((LOCATION TRUCK1 (170 50 15) (1 0 0) 13 14) . 1)
((LOCATION TRUCK1 (180 50 15) (1 0 0) 14 15) . 1)
((LOCATION TRUCK1 (190 50 15) (1 0 0) 15 16) . 1)
((LOCATION TRUCK1 (200 50 15) (1 0 0) 16 17) . 1)
((LOCATION TRUCK1 (210 50 15) (1 0 0) 17 18) . 1)
((LOCATION TRUCK1 (220 50 15) (1 0 0) 18 19) . 1)
((LOCATION TRUCK1 (230 50 15) (1 0 0) 19 20) . 1)
((LOCATION TRUCK1 (240 50 15) (1 0 0) 20 21) . 1)
((LOCATION TRUCK1 (250 50 15) (1 0 0) 21 22) . 1)
((LOCATION TRUCK1 (255 50 15) (1 0 0) 22 23) . 1)
((LOCATION TRUCK1 (260 50 15) (1 0 0) 23 30) . 1)
((LOCATION TRUCK1 (255 50 15) (1 0 0) 30 31) . 1)
((LOCATION TRUCK1 (250 50 15) (1 0 0) 31 32) . 1)
((LOCATION TRUCK1 (245 50 15) (1 0 0) 32 33) . 1)
((LOCATION TRUCK1 (240 50 15) (1 0 0) 33 34) . 1)
((LOCATION TRUCK1 (238 50 15) (1 0 0) 34 40) . 1)
((LOCATION VW1 (-100 70 8) (4 1 0) 1 2) . 1)
((LOCATION VW1 (-80 75 8) (4 1 0) 2 3) . 1)
((LOCATION VW1 (-60 80 8) (4 1 0) 3 4) . 1)
((LOCATION VW1 (-40 85 8) (4 1 0) 4 5) . 1)
((LOCATION VW1 (-20 90 8) (4 1 0) 5 6) . 1)
((LOCATION VW1 (0 95 8) (4 1 0) 6 7) . 1)
((LOCATION VW1 (20 100 8) (4 1 0) 7 8) . 1)
((LOCATION VW1 (40 105 8) (4 1 0) 8 9) . 1)
((LOCATION VW1 (60 110 8) (1 0 0) 9 10) . 1)
((LOCATION VW1 (90 110 8) (1 0 0) 10 11) . 1)
((LOCATION VW1 (125 110 8) (1 0 0) 11 12) . 1)
((LOCATION VW1 (165 110 8) (1 0 0) 12 13) . 1)
((LOCATION VW1 (210 110 8) (1 0 0) 13 14) . 1)
((LOCATION VW1 (260 110 8) (1 0 0) 14 15) . 1)
((LOCATION VW1 (310 110 8) (1 0 0) 15 16) . 1)
((LOCATION VW1 (360 110 8) (1 0 0) 16 17) . 1)
((LOCATION VW1 (410 110 8) (1 0 0) 17 18) . 1)
((LOCATION VW1 (450 110 8) (4 -1 0) 18 19) . 1)
((LOCATION VW1 (490 100 8) (4 -1 0) 19 20) . 1)
((LOCATION VW1 (540 90 8) (4 -1 0) 20 21) . 1)
((LOCATION VW1 (580 80 8) (4 -1 0) 21 22) . 1)
((LOCATION VW1 (620 70 8) (4 -1 0) 22 23) . 1)
((LOCATION VW1 (660 60 8) (4 -1 0) 23 24) . 1)
((LOCATION VW1 (700 50 8) (1 0 0) 24 25) . 1)
((LOCATION VW1 (740 50 8) (1 0 0) 25 26) . 1)
((LOCATION VW1 (775 50 8) (1 0 0) 26 27) . 1)

```
((LOCATION VW1 (805 50 8) (1 0 0) 27 28) . 1)
((LOCATION VW1 (830 50 8) (1 0 0) 28 29) . 1)
((LOCATION VW1 (850 50 8) (1 0 0) 29 30) . 1)
((LOCATION VW1 (865 50 8) (1 0 0) 30 31) . 1)
((LOCATION VW1 (875 50 8) (1 0 0) 31 32) . 1)
((LOCATION VW1 (880 50 8) (1 0 0) 32 40) . 1)
((LOCATION VW1 (-100 55 7) (1 0 0) 32 33) . 1)
((LOCATION VW1 (-80 55 7) (1 0 0) 33 34) . 1)
((LOCATION VW1 (-60 55 7) (1 0 0) 34 35) . 1)
((LOCATION VW1 (-40 55 7) (1 0 0) 35 36) . 1)
((LOCATION VW1 (-20 55 7) (1 0 0) 36 37) . 1)
((LOCATION VW1 (0 55 7) (1 0 0) 37 38) . 1)
((LOCATION VW1 (10 55 7) (1 0 0) 38 39) . 1)
((LOCATION VW1 (20 55 7) (1 0 0) 39 40) . 1)
```

The initialization phase yields the following additional entries:

(MOVE VW2 32 39)
(MOVE VW1 1 32)
(MOVE TRUCK1 1 23)
(MOVE TRUCK1 29 34)

In the following example all OVERTAKE events are to be instantiated. The range of the variables OBJ1? and OBJ2? is {VW1 VW2 TRUCK1}. The trace markers have the following meanings:
G: = generate all instances
T: = test instance
>> = enter proposition into database

```
SEARCH:

        (OVERTAKE OBJ1? OBJ2? (1 40) (1 40))

    G:  (OVERTAKE OBJ1? OBJ2? (1 40) (1 40))
       G:  (APPROACH OBJ1? OBJ2? 1 40)
          G:  (SYM-APPROACH OBJ1? OBJ2? 1 40)
          >> (SYM-APPROACH VW2 VW1 32 39)
          >> (SYM-APPROACH VW2 TRUCK1 32 39)
          >> (SYM-APPROACH VW1 TRUCK1 1 12)
          T:  (SYM-APPROACH VW2 VW1 32 39)
             G:  (IN-FRONT-OF VW1 VW2 32 39)
             >> (IN-FRONT-OF VW1 VW2 32 40)
             T:  (IN-FRONT-OF VW1 VW2 32 40)
                G:  (MOVE VW2 32 39)
                T:  (MOVE VW2 32 39)
                   >> (APPROACH VW2 VW1 32 39)
                   T:  (SYM-APPROACH VW2 TRUCK1 32 39)
             G:  (IN-FRONT-OF TRUCK1 VW2 32 39)
             >> (IN-FRONT-OF TRUCK1 VW2 32 40)
```



```
             T:  (IN-FRONT-OF TRUCK1 VW2 32 40)
                G:  (MOVE VW2 32 39)
                T:  (MOVE VW2 32 39)
                   >> (APPROACH VW2 TRUCK1 32 39)
                   T:  (SYM-APPROACH VW1 TRUCK1 1 12)
          G:  (IN-FRONT-OF TRUCK1 VW1 1 12)
          >> (IN-FRONT-OF TRUCK1 VW1 1 11)
          T:  (IN-FRONT-OF TRUCK1 VW1 1 11)
             G:  (MOVE VW1 1 11)
             T:  (MOVE VW1 1 32)
                >> (APPROACH VW1 TRUCK 1 1 11)
                T:  (APPROACH VW2 VW1 32 39)
       G:  (RECEDE VW2 VW1 1 40)
          G:  (SYM-RECEDE VW2 VW1 1 40)
          T:  (APPROACH VW2 TRUCK1 32 39)
       G:  (RECEDE VW2 TRUCK1 1 40)
          G:  (SYM-RECEDE VW2 TRUCK1 1 40)
          T:  (APPROACH VW1 TRUCK1 1 11)
       G:  (RECEDE VW1 TRUCK1 1 40)
          G:  (SYM-RECEDE VW1 TRUCK1 1 40)
          >> (SYM-RECEDE VW1 TRUCK1 12 34)
          T:  (SYM-RECEDE VW1 TRUCK1 12 34)
             G:  (BEHIND TRUCK1 VW1 12 34)
             >> (BEHIND TRUCK1 VW1 14 40)
             T:  (BEHIND TRUCK1 VW1 14 40)
                G:  (MOVE VW1 14 34)
                T:  (MOVE VW1 1 32)
                   >> (RECEDE VW1 TRUCK1 14 32)
                   T:  (RECEDE VW1 TRUCK1 14 32)
          G:  (MOVE VW1 1 32)
          T:  (MOVE VW1 1 32)
             G:  (MOVE TRUCK1 1 32)
             T:  (MOVE TRUCK1 29 34)
             T:  (MOVE TRUCK1 1 32)
                G:  (BESIDE VW1 TRUCK1 2 22)
                >> (BESIDE VW1 TRUCK1 9 14)
                T:  (BESIDE VW1 TRUCK1 9 14)
                   >> (OVERTAKE VW1 TRUCK1 (1 10) (15 23))
                   T:  (OVERTAKE VW1 TRUCK1 (1 10) (15 23))
FOUND:

      (OVERTAKE VW1 TRUCK1 (1 10) (15 23))
```

The next trace is for the example "Did a yellow VW overtake a truck in front fo the FBI?" The variable X1?, X2? and X3? are bound to VW1, TRUCK1 and BUILDING1 respectively. The database has been reinitialized.

```
SEARCH:

        (OVERTAKE X1? X2? (1 40) (1 30)
        (IN-FRONT-OF X1? X3? 1 30)

    G:  (OVERTAKE X1? X2? (1 40)(1 30)
       G:  (APPROACH OBJ1? OBJ2? 1 40)
          G:  (SYM-APPROACH OBJ1? OBJ2? 1 40)
          >> (SYM-APPROACH VW1 TRUCK1 1 12)
          T:  (SYM-APPROACH VW1 TRUCK1 1 12)
             G:  (IN-FRONT-OF TRUCK1 VW1 1 12)
             >> (IN-FRONT-OF TRUCK1 VW1 1 11)
             T:  (IN-FRONT-OF TRUCK1 VW1 1 11)
                G:  (MOVE VW1 1 11)
                T:  (MOVE VW1 1 32)
                   >> (APPROACH VW1 TRUCK1 1 11)
                   T:  (APPROACH VW1 TRUCK1 1 11)
          G:  (RECEDE VW1 TRUCK1 1 40)
             G:  (SYM-RECEDE VW1 TRUCK1 1 40)
             >> (SYM-RECEDE VW1 TRUCK1 12 34)
             T:  (SYM-RECEDE VW1 TRUCK1 12 34)
                G:  (BEHIND TRUCK1 VW1 12 34)
                >> (BEHIND TRUCK1 VW1 14 40)
                T:  (BEHIND TRUCK1 VW1 14 40)
                   G:  (MOVE VW1 14 34)
                   T:  (MOVE VW1 1 32)
                      >> (RECEDE VW1 TRUCK1 14 32)
                      T:  (RECEDE VW1 TRUCK1 14 32)
             G:  (MOVE VW1 1 32)
             T:  (MOVE VW1 1 32)
                G:  (MOVE TRUCK1 1 32)
                T:  (MOVE TRUCK1 29 34)
                T:  (MOVE TRUCK1 1 23)
                   G:  (BESIDE VW1 TRUCK1 2 22)
                   >> (BESIDE VW1 TRUCK1 9 14)
                   T:  (BESIDE VW1 TRUCK1 9 14)
                      >> (OVERTAKE VW1 TRUCK1 (1 10) (15 23))
                      T:  (OVERTAKE VW1 TRUCK1 (1 10) (15 23))
       G:  (IN-FRONT-OF VW1 X3? 1 23)
       >> (IN-FRONT-OF VW1 BUILDING1 4 15)
       T:  (IN-FRONT-OF VW1 BUILDING1 4 15)

FOUND:

        (OVERTAKE VW1 TRUCK1 (4 10)
        (IN-FRONT-OF VW1 BUILDING1 4 15)
```