

Combining Alternatives in the Multimedia Presentation of Decision Support Information for Real-Time Control

G. Herzog, E. Andre, S. Baldes, T. Rist
DFKI GmbH, German Research Center for Artificial Intelligence
Saarbrücken, Stuhlsatzenhausweg 3, Germany,
{herzog, andre, baldes, rist}@dfki.de

Abstract

Multimedia technology is emerging as a key element in the area of Decision Support Systems (DSS) since well-designed multimedia presentations help the human decision maker to assimilate relevant information more easily. The use of multiple media, however, increases the complexity of the presentation design task. Especially when complex information structures have to be presented under time pressure 'ad hoc' solutions to presentation generation are getting more and more impractical, if not impossible to use. In this paper* we report on our approach to enhance a DSS for real-time traffic management with an advanced component for the automated generation of multimedia presentations. A common problem in this application class is the presentation of alternatives such as different explanations or predictions for a current traffic situation, or different sequences of control actions which may be initiated to resolve a problem. We describe a novel approach to provide aggregated information presentations rather than presenting alternatives just one after the other.

Keywords

Intelligent multimedia presentation, real-time decision support user interface, aggregated information presentation, automated multimedia presentation design

1 INTRODUCTION

Decision support systems (DSS) are interactive computer-based information systems that are designed to help human decision makers in utilising data and models in order to identify, structure, and solve semi-structured or unstructured problems and make choices among alternatives. Multimedia technology is emerging as a key element for

*The work described in this paper was partly supported by the project FLUIDS which is funded under the Telematics Application Programme by the European Commission, and partly by the AiA project funded by the German Ministry for Education and Research.

the adequate presentation of the complex information managed by a DSS since the ultimate goal is to effectively provide the human decision maker with the relevant information on the basis of available underlying data.

Especially in the area of real-time control applications there is a growing need to improve user-system interaction through multimedia-based decision support which integrates sophisticated problem solving capabilities with enhanced information presentation functionality. Potential application fields include for example: transport telematics for traffic control and traffic management, real-time control systems in industrial environments, monitoring and management of telecommunication networks as well as networks for power transmission and distribution, mission control and emergency management, and sophisticated decision support systems in the field of medical engineering.

The European project FLUIDS (Future Lines of User Interface Decision Support) aims at the design of a general environment for building intelligent interfaces to automated control systems that provide human operators with multimedia-enhanced real-time decision support. The integration of an advanced component for the automated generation of multimedia presentations constitutes a core element of the FLUIDS approach. In this paper, we report on the experience gained from adding this kind of multimedia functionality to concrete decision support applications in real-time traffic management. It turned out that one of the most challenging tasks is the adequate presentation of alternatives such as different explanations or varying predictions for a given traffic situation or several options for corrective control actions.

2 BACKGROUND

We are concerned with the development of an intelligent multimedia interface as backend to a decision support system which itself sits on top of a real-time traffic management system (cf. Figure 1).

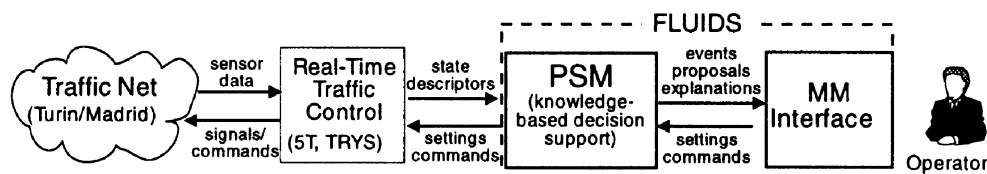


Figure 1 Components of an advanced traffic management system.

The FLUIDS approach is being tested on different real-time traffic management systems currently operating in the cities of Madrid and Turin. Both systems are connected with large networks of sensors delivering real-time data about the traffic state. Considering various types of problems, three distinct applications are under development. The TRYS system in Madrid aims at generating proposals for traffic control strategies according to actual traffic conditions. UTOPIA, the urban traffic control

component of the 5T system in Turin operates fully automated instead. In this context, FLUIDS is supposed to aid the traffic engineer in the diagnosis of system performance as well as the analysis of the causes of possible faults and to suggest possible traffic model improvements. The 5T system is an integrated control system for public and private traffic management with several subcomponents. A third FLUIDS application builds upon SIS, the 5T public transport management component, to provide operators with suggestions for suitable control actions to recover from service irregularities.

Though the above mentioned traffic management systems are build on elaborated models of the domain and tasks both systems lack of sophisticated explanation capabilities such as to aid users in understanding how and why the system reaches its conclusion, to convince users that conclusions drawn by the system are sound and reasonable, and also assist in debugging the knowledge and problem solving behaviour of the system. As a prerequisite to achieve these abilities, a knowledge-based module for problem solving (PSM) has been developed using the Knowledge Structure Manager environment (KSM, cf. (Cuenca, Hernández & Molina 1997)). This component includes qualitative models of the algorithmic processes of the underlying traffic control systems, and is able to provide qualitative explanations of proposed solutions for trouble shooting. As shown in Figure 1 the PSM component is also connected to the user interface. On request by the user, it provides three types of information: (1) the current situation, i.e. 'What happens?', (2) forecasts, i.e. 'What may happen?', and (3) potential control actions to be initiated for trouble shooting, i.e. 'What to do?'

A typical task of a traffic operator is to recognise the most critical network link, to identify the potential causes of an abnormal situation (e.g., by comparing all the estimated parameters with the nominal and historical parameters), and to select an applicable control action to solve the problem. For example, in response to the question 'What is happening on the network?' the system will present one or more areas where the difference between estimated delay of a bus line and the tolerable delay exceeds predefined thresholds (for example, an absolute threshold of 150 seconds). Concerning the follow-up question 'What to do?', the system will then inform the operator about possible control actions for solving the identified problem. Needless to say that it is the task of the user interface to present such information to the user in a way that effectively supports the operator in time-critical decision making.

The initial versions of the traffic management components within both systems, TRYS and 5T, are equipped with window-based interfaces. All these interfaces employ different media for the presentation of information; full text, short messages below sentence level, maps, and abstract diagrams, e.g., a horizontal bar with markers on it as an encoding of a bus route with stops. Our evaluation of the information presentations delivered by the interfaces, however, revealed a number of serious shortcomings:

- poor temporal output coordination, especially when distributed on different windows;

- no follow-up questions on presentations because of lacking semantic representations of system output;
- no means to condense presentations in order to reduce both redundancy and presentation time;
- little flexibility in the system's presentation behaviour because of a 'hardwired' mapping from data instances to presentation instances.

Further requirements for an improved system were obtained directly from potential system users. The interviewed users were experienced operators in the system control centres at Turin and Madrid. As expected, there was almost no need to generate a broad variety of different presentations to accommodate for different user profiles. Moreover, the operators indicated a strong preference for having only a limited number of presentation patterns with which they could easily get familiarised. For example, the operators preferred a small number of display frames with a fixed layout for graphics and text output, a small number of different graphic types (overview maps, network diagrams, line charts). On the other hand, there was a strong demand for improving the system's presentation capabilities by means of aggregation mechanisms. The less an operator had to browse through lists of textual messages and to switch between display frames in order to perform a supervision task or to decide among potential control actions, the better the system.

3 PRESENTATION TASKS AND PRESENTATION TYPES IN THE TRAFFIC MANAGEMENT APPLICATION

The task of presenting information is usually conceptualised as a mapping from given information units (domain concepts) to presentation instances (media objects or combinations of media objects). Following this view, we have to identify and classify the concepts relevant to the underlying domain, potential presentation instances, and the conditions under which a certain presentation instance should be chosen.

3.1 Domain concepts and their representation

As mentioned in section 2, the domain knowledge is modelled and represented within the KSM framework for the development and maintenance of large and complex knowledge-based applications. For the purpose of this paper we restrict ourselves to briefly introducing domain concepts which are referred to in other parts of the paper. These concepts are locations, vehicles, streets, routes, states, events and situations, and control actions.

Locations and trajectories of moving objects are conceptualised as particular positions or regions over a background frame. The background frame may be a geometric map of a town or neighbourhood so that all represented locations have denotations in the real world. However, a background frame may also be an abstract graph structure (e.g., providing topological information on routes). Domain objects are vehicles,

streets, routes, bus lines, traffic signs, etc.. Each represented object is internally accessible through a unique identifier, and may have a number of attributes assigned to it (e.g., a location, a "pretty name" or an icon for its graphical display). As some attributes of domain objects may change over time, object descriptions may vary from one instance in time to another. States and events are described by means of predicates that may hold for an object or some objects at a certain instance in time or over a certain time period. For example, a bus may be operable or broken, a bus line may be delayed, whereas a conjunction event may have been recognised or forecasted by the system. Situations are introduced to characterise relevant aspects of complex traffic situations. Situation descriptions may comprise a number of state and event descriptions. For example, the Lisp-style representation below captures the situation where a bus-line is delayed due to the delay of a bus (vehicle bus#5 has a delay of 17 minutes).

```
(current_situation (vehicle_state bus#5 delayed)
                  (vehicle_location bus#5 loc#188)
                  (vehicle_delay bus#5 17))
```

Explanations are event sequences whose outcome would be consistent with the current situation. For example, if a traffic problem has occurred, the operator may be interested in the events which caused the problem. In some cases, several plausible explanations may be found due to the system's incomplete knowledge of the real world. Predictions are possible future traffic situations. Starting from the current situation, they are computed by the problem solving module, e.g., through a traffic simulation process. In some cases, a high degree of uncertainty may lead to several potential situations of the same likelihood. Control actions are actions which may be initiated in order to resolve a traffic problem. For example, if a bus breaks down at a certain location, the diagnosis system may suggest either to send a replacement bus which continues the service, or if feasible, to make the passengers wait for the next bus of the same line.

In the following, we introduce a simplified notation for actions, action sequences and alternatives. Actions are characterised by an action type and a list of action parameters in the underlying domain representation. Furthermore, an action can be either primitive or a composition of other actions. Action terms are inductively defined over the set of primitive domain actions:

1. each primitive domain action is an action term;
2. if a_1, \dots, a_n are action terms, then the action sequence of the form $[a_1; \dots; a_n]$ denotes the temporally ordered sequence of the actions a_1, \dots, a_n and is also an action term;
3. if a_1, \dots, a_n are action terms, then the list of alternatives has the form $(\text{Alt } a_1, \dots, a_n)$ and is also an action term. In case of control actions, it refers to a list of several actions from which the operator has to choose exactly one.
4. Actions which are described by action terms may have a hierarchical structure including alternatives since each a_i in a sequence or a list of alternatives by itself is an action term.

3.2 Presentation types

For the traffic management domain, we have to define presentation types for accomplishing tasks such as presenting:

- objects, attributes and states of objects, object locations and trajectories;
- relevant aspects of complex traffic situations, such as events and involved objects;
- explanations, i.e. causes for the occurrence of an event or a problematic traffic situation;
- predictions how a certain traffic situation may evolve (e.g., within the next hour);
- sets of potential control actions from which the operator has to select one or more in order to avoid or resolve problems.

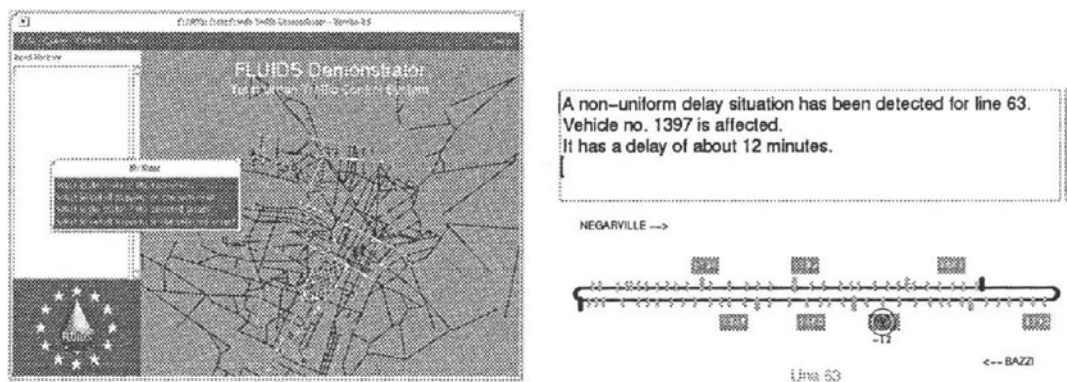


Figure 2 Graphical display types of the Fluids demonstrator: Street network and bus line diagram.

Two different sample displays are shown in Figure 2. In accordance with the user requirements study, the presentation media text, speech, 2D graphics and 2D animation are supported in the combinations listed in Table 1. In case of language (text or speech) predefined sentence patterns are used to encode descriptions for object states, events and actions. Because the operators preferred to see a kind of textual record, the use of the medium speech is supported only on demand and always in addition to text. Static graphics include several types of map displays, and special purpose diagrams such as bus line visualisations. Basic domain objects such as vehicles are graphically represented by icons. The set of icons comprises also conventionalised icons for the indication of some events (e.g. accident) and actions (e.g. driver exchange), and a few marker icons (e.g., blinking circles and arrows) which are used to draw the viewer's attention to a certain location on the display. For animations we distinguish between visualisations of moving objects on a map background, and the temporally coordinated annotation of a static display. That is, starting with a background display, the final static display is completed step by step with annotations before the operators eyes. In contrast to the usual form of animation, this type of an-

imation has the advantage that the last image frame can be viewed stand-alone as a static graphics which encodes all the relevant information that has been added during the preceding animation.

Current Situation	text	sentence by sentence enumeration of the occurred events
	speech	additional spoken telegraph-style descriptions (optional)
	static graphics	annotated maps (showing locations of involved objects)
	animation	no
Explanation	text	sentence by sentence enumeration of the potential causes
	speech	additional spoken telegraph-style descriptions (optional)
	static graphics	visualisation of diagnosis results: (a) static graphics: dynamic data such as object trajectories are shown by arrows;
	animation	(b) animation: dynamic scenes can be played back and forth with arbitrary speed
Prediction	text	sentence by sentence enumeration of the predicted events
	speech	additional spoken telegraph-style descriptions (optional)
	static graphics & animation	visualisation of simulation results (similar to explanation)
Control Actions	text	sentence by sentence enumeration of the proposed actions
	speech	additional spoken telegraph-style descriptions (optional)
	static graphics & animation	static graphics are modified by subsequent annotations on a map to show locations and trajectories of involved objects and locations where actions take place

Table 1 Presentation types used in FLUIDS to convey domain information.

3.3 Presentation planning

To map information units onto multimedia presentation instances, we rely on our framework for the representation and generation of multimedia presentations (cf. (André, Finkler, Graf, Rist, Schauder & Wahlster 1993, Rist, André & Müller 1997)). In this framework, we operationalise the generation of multimedia presentation by means of a goal-driven, top-down planning mechanism. The presentation planner receives as input a communicative goal (for instance, the user should be able to localise the malfunctioning vehicle on the network) and a set of generation parameters, such as target group, presentation objective, resource limitations, and target language. The task of the component is to select parts of a knowledge base and to transform them into a multimedia presentation structure. Whereas the root node of such a presentation structure corresponds to a more or less complex communicative goal, such as describing a prediction for a traffic situation, the leaf nodes are elementary generation or presentation acts, currently for text, graphics, and animations.

In order to cope with the dynamic nature of most multimedia presentations, the presentation planner has been combined with a temporal reasoner based on (Kautz & Ladkin 1991) whose task is to determine a preliminary presentation schedule. Since the temporal behaviour of presentation acts may be unpredictable at design time, the schedule will be refined at presentation runtime by adding new temporal constraints to the constraint network.

We use so-called presentation strategies to represent knowledge concerning how to decompose a given presentation task into subtasks or, in case of elementary subtasks, which media objects should be used to convey the subtasks. Presentation strategies consist of a *header*, a set of *applicability conditions*, a collection of *inferior acts*, a list of *qualitative* and *metric temporal constraints*, and a *start* and an *end interval*. The header of a strategy corresponds to a complex presentation act such as presenting a traffic situation. The applicability conditions specify when a strategy may be used and constrain the variables to be instantiated. The inferior acts provide a decomposition of the header into more elementary presentation acts. Qualitative temporal constraints are represented in an 'Allen-style' fashion which allows for the specification of thirteen temporal relationships between two named intervals: before, meets, overlaps, during, starts, finishes, equal and inverses of the first six relationships (cf. (Allen 1983)). Allen's representation also permits the expression of disjunctions, such as (A (before after) B), which means that A occurs before or after B. Metric constraints appear as difference (in)equalities on the endpoints of named intervals. They can constrain the duration of an interval (e.g., (10 <= Dur A2 <= 40)), the elapsed time between intervals (e.g., (4 < End A1 - Start A2 < 6)) and the endpoints of an interval (e.g., (Start A2 >= 6)).

The basic repertoire of presentation strategies for the traffic management application has been defined in a straightforward manner. For each of the information types listed in section 3.1 at least one presentation strategy has been defined. An example of a presentation strategy is shown below. It may be applied to inform the operator about a delay of a vehicle (e.g. a bus) via graphical and textual means.

```
(define-strategy
:HEADER
(A0 (INFORM-DELAY-DETAILED P A
      ?text-window ?graphic-window ?pos-1 ?pos-2 ?vehicle
      ?v-location ?v-delay ?delay-label))
:INFERIORS
((A1 (SHOW-VEHICLE P A ?graphic-window ?vehicle ?v-location ?pos-1))
 (A2 (VERBALIZE-VEHICLE P A ?text-window ?vehicle))
 (A3 (SHOW-RED-BLINKER P A ?graphic-window ?v-location ?pos-1))
 (A4 (VERBALIZE-VEHICLE-DELAY P A ?text-window ?minutes))
 (A5 (SHOW-LABEL P A ?graphic-window ?delay-label ?v-location ?pos-2)))
:QUALITATIVE ((A1 (e) A2) (A2 (s) A3) (A2 (m) A4) (A4 (m) A5))
:METRIC ((20 <= DURATION A3 <= 30)) :START A3 :FINISH A3)
```

At this stage of the project, two improvements over the original interfaces of the traffic management systems (TRYS and 5T) have been achieved. It is now possible

to ensure a proper temporal coordination between presentation acts, only by specifying temporal relationships between the inferior acts in the strategies. Furthermore, there is now a clear separation between the representation of domain knowledge and presentation knowledge which facilitates the modification and fine tuning of presentation types. However, the basic repertoire of presentation strategies defined so far did not help to avoid redundancies when presenting event and action sequences with overlapping subparts. This problem occurs when alternatives have to be presented, e.g., in situations in which the system comes up with different explanations or predictions for a certain situation, or with different sequences of control actions for problem solving. In case of the FLUIDS system, usually a single explanation and a single prediction is delivered but for control actions the set of alternatives does frequently contain 2–3 instances. In order to further improve the system's presentation abilities, the aggregation task has to be addressed, too. In the following section, we concentrate on control actions and sketch how our approach handles aggregation tasks.

4 AGGREGATED PRESENTATION OF CONTROL ACTIONS

To illustrate the problem, let's consider the following scenario: The system has informed the operator that a bus, say bus#11, broke down at location loc#347 and is now no longer able to continue its service for the corresponding bus line. After the operator has asked for advice on what to do, the diagnosis subsystem suggests two alternative action sequences which may be initiated to fix the problem.

The first solution is to send a repair car and a replacement bus to the location where the broken bus#11 is standing. Then the drivers are exchanged and the passengers will be transferred to the replacement vehicle. Finally the broken bus will be towed-away with the repair car. The first action of the second solution coincides with the first action of the first alternative. That is a repair car is moved to the location of bus#11. However, instead of using a replacement bus, the system suggests to wait for the arrival of the next bus of the same line. Then the passengers have to change to bus#12 and the broken bus will be towed away. Using a Lisp-style notation, the output of the diagnosis component is as follows:

```
(Alt [ (move repair-car#5 loc#347);
      (move bus#15 loc#347);
      (exchange-drivers bus#11 bus#15 loc#347);
      (transfer-passengers bus#11 bus#15 loc#347);
      (tow-away bus#11 repair-car#5 loc#347) ],
 [ (move repair-car#5 loc#347);
   (wait-for-next-bus-of-line bus#12 loc#347);
   (transfer-passengers bus#11 bus#12 loc#347);
   (tow-away bus#11 repair-car#5 loc#347) ] )
```

4.1 Subsequent presentation of all alternatives

A straightforward way of presenting potential control actions is to produce first a kind of advance organiser which introduces the alternatives and second to describe all alternatives in detail. If we apply this strategy on the previous example, we get the presentation structure shown in Figure 3.

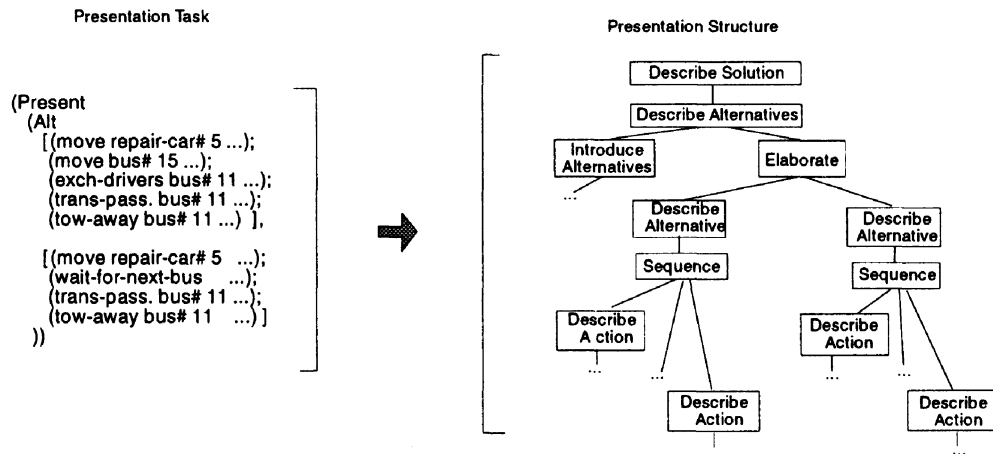


Figure 3 Presentation task and corresponding presentation structure.

While it is easy to define a presentation strategy for this case, the resulting presentations are often long-winded and thus are not suitable for the support of decision-making under time pressure. This is especially crucial when speech and animation get involved in the descriptions of subactions since the total presentation time is determined by the sum of the time needed for each single description. Furthermore, such presentations make it very difficult for the decision maker to recognize similarities and differences between alternatives.

4.2 Factoring out common subactions

Obviously presentation time can be saved if it is possible to restructure the presentation in such a way that descriptions of common subactions only appear once in the presentation. The two sequences of the example have the subactions (*move repair-car#2 to loc#347*) and (*tow-off repair-car#2 bus#11 from loc#347*) in common. Our approach to factor out such common parts is to reformulate the given presentation task into a new task with a less redundant structure. Figure 4 illustrates the intended reformulation. In essence, we go through the list of control actions in order to figure out whether there are pairs of common actions. If such pairs exist, the given presentation task is reformulated into a new task which can be accomplished more efficiently than the original task. The rationale behind this approach is the assumption, that we can use similar presentations for similar action instances. However, it is not

always advisable to perform all possible transformation because the resulting structure may become even more difficult to present as the original list of alternatives. In the FLUIDS system, we restrict ourselves just to factor out common start, middle or end subsequences and avoid structures with nested branchings.

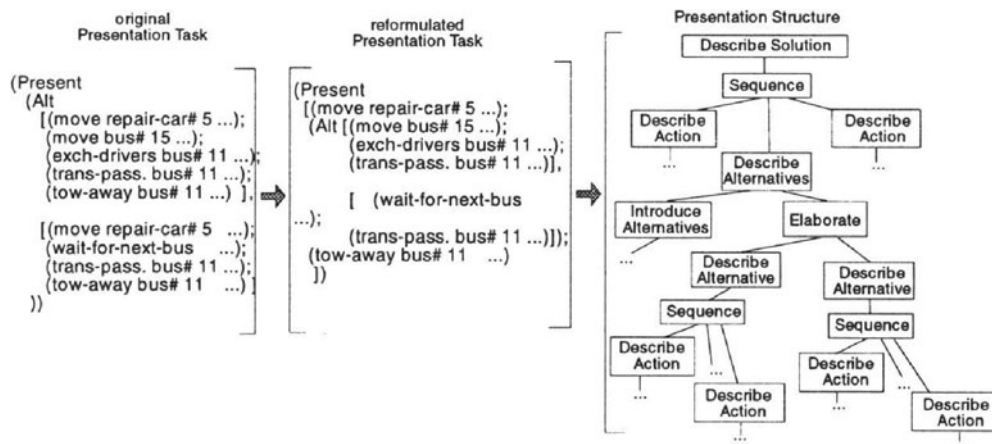


Figure 4 Presentation structure for the reformulated presentation task.

For the combined presentation of the two alternative control actions we deploy the graphical display shown in Figure 5. It is used to convey the trajectories of the involved vehicles. While both action sequences comprise the same trajectory for vehicle r-5, the trajectories of b-15 and b-12 represent alternatives.

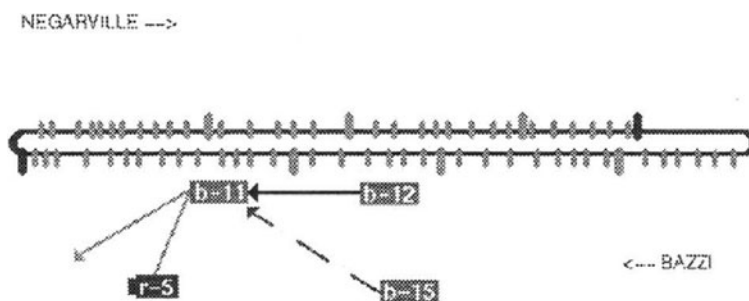


Figure 5 Combined graphical display of two alternative control actions.

4.3 Factoring out common aspects of actions

In some cases, the only difference between two alternatives is only due to different bindings of some action parameters. That is, two actions a and b are of the same type,

but at least one action parameter has a different binding. Consider for example the situation in which the operator should send a repair car to a certain location but may have the choose between a red and a blue car. The presentation of this alternative may be shortened by factoring out the common aspects of nearly similar actions, e.g. by saying `move the red or blue repair car to loc...'. This can be achieved by means of a further reformulation strategy which would merge

```
(Present (Alt [ ... (move repair-car#1 loc#347) ...],
             [ ... (move repair-car#2 loc#347) ...]))
```

into (Present ... (move (Alt repair-car#1 repair-car#2) loc#347) ...).

Of course such reformulations make only sense if there is a presentation strategy which is able to handle the encoding of alternative parameter bindings. In the example presented above we have a slightly different case concerning the subaction *transfer-passengers* which occur in both alternatives. The only difference on the propositional level lies in the binding of the second parameter which is bound to bus#15 in the first sequence, and to bus#12 in the second alternative. However, in this case the action context determines which of the two bindings must be chosen. If an aggregation strategy is applied, we have to ensure that this dependency is reflected on the surface level, too. Instead of just saying “transfer passengers from the broken bus (bus#11) to the substitute bus (bus#15) or the next bus in line (bus#12)”, we would mark the dependency by adding “respectively”. Unfortunately, it can be quite difficult to determine whether or not an alternative for a parameter binding depends on a previous decision. In the transfer-passenger example, it may suffice to trace back the occurrence of the corresponding parameters and to figure out that the two bindings (replacement bus bus#15 and next bus of line bus#15) were introduced in alternative preceding subsequences. In the general case, however, deeper reasoning on the domain knowledge will be required in order to avoid useless factoring.

4.4 Embedding the approach into presentation planning

This approach has been included into our presentation planning environment by augmenting the repertoire of presentation strategies by task-reformulation strategies. The header of such task-reformulation strategies represents the initial task while the body refers to its reformulation. The constraint slot of the strategies is used to specify conditions under which a reformulation should be performed. For example, a constraint for factoring out a certain subaction is that it must occur in two alternative sequences. Further constraints have to be formulated in order to avoid too many reformulations. For example, we avoid reformulations which lead to nested branching structures as they often become quite difficult to present. Whenever the planner encounters a new presentation task, it first tries to reformulate the task by using task-reformulation strategies before decomposing it by applying presentation strategies. Note that in case a reformulated task cannot be solved eventually, the planner will launch a backtracking process that withdraws the reformulation decision.

5 RELATED WORK

There are enormous efforts of the software industry to provide multimedia functionality with their DSS products. For example, many database vendors aid decision makers within a business context in accessing and presenting the information provided by an enterprise decision support systems. In this application area capabilities for information presentation range from simple tabular to advanced multidetail reports with all types of graphs and charts. Such systems incorporate dedicated generation modules such as table formatters or chart drawing components. Promising experiences in enhancing DSS with multimedia components have also been reported from research activities in the area of medical decision making. However, current DSS yet do not take advantage of more recent methods for the automated design of multimedia presentations (cf. (Feiner & McKeown 1991), (Maybury 1991), (Stock 1991), (André et al. 1993), (Arens, Hovy & van Mulken 1993), and (Roth & Hefley 1993) for an overview). Vice versa, real-time decision support is only rarely chosen as an application domain for automated presentation generation, like for example in (Sutcliffe & Faraday 1994). This may be one of the reasons why the aggregation problem has not been addressed very detailed so far in this research community.

With the application data on the one side, the generated presentation parts on the other side, and the presentation generator in between, there are three different approaches to information aggregation which aggregate either over (1) domain data, (2) media objects, or (3) intermediate presentation structures. Following the first approach means to introduce additional concepts in the representation of the domain and the definition of presentation strategies for these additional concepts. The problem with this approach is that it blurs the borderline between domain modelling and specification of presentation knowledge. In our project consortium the engineers responsible for modelling the domain didn't feel comfortable with the idea of defining new domain concepts "just" to improve the systems presentation abilities. They were in favour of keeping the modularization of tasks and responsibilities as it was in the initial systems. Approaches that relate to the second alternative can be found in the area of text summarisation (e.g. (Spärck-Jones, Endres-Niggemeyer, Hobbs, Liddy & Paris 1993)). In this community, a number of techniques have been developed in order to derive a summary from a source text. Such an approach seemed to inefficient for our application as we would have to generate first a complete presentation as input for a subsequent aggregation process. Approaches that fall under the third alternative have in common that they try to perform aggregations on representation formats that are used in the generation process. These formats can be media-independent presentation acts, presentation acts to be conveyed in a certain medium, or media-specific structures of presentation units, such as preverbal messages during text generation. Usually, an aggregation module is added between the content planner and the text generator (for example, see (Dalianis & Hovy 1993), (Shaw 1983)).

Our approach aims at aggregations at the level of presentation acts, too. However, we apply restructuring strategies at an early stage during presentation planning. This approach enables us to consider dependencies between content structuring and

aggregation which are more crucial in the FLUIDS application than dependencies between aggregation and realization since we rely on prestored text patterns and schema-based graphics instead of fully-fledged media design as for example the graphics design approach proposed in (Casner 1991).

6 CONCLUSION

In this paper we have reported on our work to equip an existing real-time traffic management application with a component for the automated design of multimedia presentations. In particular, we sketched how our framework for plan-based presentation design was adapted and augmented to suit this application. From the view point of research on real-time decision support systems, this work may be of interest because it enables us to replace ad hoc solutions for the handling of crucial presentation issues by a principled approach for the intention-based coherent structuring of presentations and the temporal coordination of media items. On the other hand, real-time decision support appeared as a promising, but challenging application area for research on automated multimedia generation. To ensure that presentations are both short and easily to follow for time-pressured controllers, the generation of aggregated information presentations is an important issue which has to be addressed. In our proposed solution a presentation planner attempts to reduce the number of propositions to be communicated by factoring out information units such as common actions of alternative action sequences. The approach helped to significantly improve the presentation abilities of the traffic management system in comparison to the original interface.

However, there is still much room for further improvements. First of all, it is important to extend the set of multimedia presentation types for condensed information presentations. While in the case of text valuable inspirations can be found in the literature, pioneering work is still required when it comes to graphics and animation. Currently, we are experimenting with graphical forms for the presentation of alternatives. For example, alternative object movements may be visualised through colour coding, or more dynamically, by alternating superimpositions of arrows for the alternative trajectories. In the current implementation, we are quite restrictive when factoring out common information units. We do not perform reformulations which would produce more complex branching structures. This restriction increases the chance that a suitable presentation can be generated for a reformulated task. On the other hand, there may still be unnecessary redundancy in generated presentations of alternatives. Another issue concerns the generality of our task reformulation strategies to aggregation. They essentially merge separated items (i.e. action sequences) in case that they share common parts (i.e. subactions). This approach was reasonable since in the FLUIDS context we had to start from a given domain representation, namely the one being used in the diagnosis system. One could certainly imagine a diagnosis system which delivers a graph-like structure instead of a list of alternatives. In this case, part of the aggregation task would be to split the graph structure into reasonable units which can be presented together.

REFERENCES

- Allen, J. F. (1983), 'Maintaining knowledge about temporal intervals', *Communications of the ACM* **26**(11), 832–843.
- André, E., Finkler, W., Graf, W., Rist, T., Schauder, A. & Wahlster, W. (1993), WIP: The automatic synthesis of multimodal presentations, in M. T. Maybury, ed., 'Intelligent Multimedia Interfaces', AAAI Press, Menlo Park, CA, pp. 75–93.
- Arens, Y., Hovy, E. & van Mulken, S. (1993), Structure and rules in automated multimedia presentation planning, in 'Proc. of the 13th IJCAI', Chambéry, France, pp. 1253–1259.
- Casner, S. M. (1991), 'Task-analytic approach to the automated design of graphic presentations', *ACM Transactions on Graphics* **10**(2), 111–151.
- Cuena, J., Hernández, J. & Molina, M. (1997), An exercise of knowledge-oriented design: Architecture for real-time decision support systems, in S. G. Tzafestas, ed., 'Knowledge Based Systems. Advanced Concepts, Techniques and Applications', World Scientific, Singapore.
- Dalianis, H. & Hovy, E. (1993), Aggregation in natural language generation, in 'Proc. of the Fourth European Workshop on Natural Language Generation', Pisa, Italy.
- Feiner, S. K. & McKeown, K. R. (1991), 'Automating the generation of coordinated multimedia explanations', *Computer* **24**(10), 33–41.
- Kautz, H. A. & Ladkin, P. B. (1991), Integrating metric and qualitative temporal reasoning, in 'Proc. of AAAI-91', Anaheim, CA, pp. 241–246.
- Maybury, M. T. (1991), Planning multimedia explanations using communicative acts, in 'Proc. of AAAI-91', Anaheim, CA, pp. 61–66.
- Rist, T., André, E. & Müller, J. (1997), Adding animated presentation agents to the interface, in 'Proc. of UII '97: Int. Conf. on Intelligent User Interfaces', Orlando, FL, pp. 79–86.
- Roth, S. F. & Hefley, W. (1993), Intelligent multimedia presentation systems: Research and principles, in M. T. Maybury, ed., 'Intelligent Multimedia Interfaces', AAAI Press, Menlo Park, CA, pp. 13–58.
- Shaw, J. (1983), Conciseness through aggregation in text generation, in 'Proc. of the 33th ACL', Cambridge, MA, pp. 329–331.
- Spärck-Jones, K., Endres-Niggemeyer, B., Hobbs, J., Liddy, E. & Paris, C., eds (1993), *Dagstuhl Seminar on Summarizing Text for Intelligent Communication*, Dagstuhl, Germany.
- Stock, O. (1991), Natural language and exploration of an information space: The ALFresco interactive system, in 'Proc. of the 12th IJCAI', Sidney, Australia, pp. 972–978.
- Sutcliffe, A. & Faraday, P. (1994), Designing presentation in multimedia interfaces, in 'Proc. of CHI-94', Boston, MA, pp. 92–98.