Artificial Intelligence 63 (1993) 387-427
Elsevier
ARTINT 1058

Plan-based integration of natural language and graphics generation

Wolfgang Wahlster, Elisabeth André, Wolfgang Finkler, Hans-Jürgen Profitlich and Thomas Rist

German Research Center for Artificial Intelligence (DFKI), Stuhlsatzenhausweg 3, D-66123 Saarbrücken 11, Germany

Abstract

W. Wahlster, E. André, W. Finkler, H.-J. Profitlich and T. Rist, Plan-based integration of natural language and graphics generation, Artificial Intelligence 63 (1993) 387-427.

Multimodal interfaces combining natural language and graphics take advantage of both the individual strength of each communication mode and the fact that several modes can be employed in parallel. The central claim of this paper is that the generation of a multimodal presentation can be considered as an incremental planning process that aims to achieve a given communicative goal. We describe the multimodal presentation system WIP which allows the generation of alternate presentations of the same content taking into account various contextual factors. We discuss how the plan-based approach to presentation design can be exploited so that graphics generation influences the production of text and vice versa. We show that well-known concepts from the area of natural language processing like speech acts, anaphora, and rhetorical relations take on an extended meaning in the context of multimodal communication. Finally, we discuss two detailed examples illustrating and reinforcing our theoretical claims.

1. Introduction

When explaining how to use a technical device, humans will often utilize a combination of language and graphics. It is a rare instruction manual that does not contain illustrations. Multimodal presentation systems combining natural language and graphics take advantage of both the individual strength of each communication mode and the fact that both modes can be employed in parallel. Allowing all of the modalities to refer to and depend upon each other is a key to the richness of multimodal communication.

In this paper, we describe the basic methods used in our attempt to integrate multiple AI components such as planning, knowledge representation, natural language generation, and graphics generation into a functioning prototype called WIP that plans and coordinates multimodal presentations in which all material is generated by the system. We will concentrate on the intercomponent interactions and synergies that arise from combining components.

A basic principle underlying the WIP model is that the various constituents of a multimodal presentation should be generated from a common representation of what is to be conveyed. This raises the question of how to decompose a given communicative goal into subgoals to be realized by the mode-specific generators, so that the modes[1] complement each other.

Correspondence to: W. Wahlster, German Research Center for Artificial Intelligence (DFKI) Stuhlsatzenhausweg 3, D-66123 Saarbrücken 11, Germany. E-mail: wahlster@dfki.uni-sb.de.

[1] Since one of the generation parameters of WIP is the specification of the output device, we use the term "medium" in the sense of a physical carrier of information. In contrast, the term "mode" is used throughout this paper to refer to the particular sign system. We are aware of the fact that other authors use these terms differently.

1.1. Major design goals of WIP

The major design goals of WIP are the generation of coordinated multimodal presentations from a common representation, the adaptation of these presentations to the intended target audience and situation, and the incrementality of all processes constituting the design and realization of the multimodal output.

1.1.1. Generating coordinated presentations

It is an important goal of this research not simply to merge the verbalization results of a natural language generator and the visualization results of a knowledge-based graphics design component, but to carefully coordinate natural language and graphics in such a way that they generate a multiplicative improvement in communication capabilities. Enforcing a consistent, harmonious and aesthetic integration of text and graphics is an essential subtask in automating the synthesis of multimodal presentations. To address this problem, we explored computational models of the cognitive decision process, coping with questions such as what should go into text, what should go into graphics, and which kinds of links between the verbal and non-verbal fragments are necessary.

In addition, WIP deals with page layout as a rhetorical force, influencing the intentional and attentional state of the reader. In summary, systems like WIP shift the metaphor of "computer as author" used in natural language generation to the broader view of "computer as desktop publisher" (cf. [14]).

1.1.2. Generating situated presentations

WIP is a highly adaptive interface since all of its output is generated on the fly and customized for the intended target audience and situation. The quest for adaptation is based on the fact that it is impossible to anticipate the needs and requirements of each potential user in an infinite number of presentation situations. Thus all presentation decisions are postponed until runtime. In contrast to hypermedia-based approaches to adaptive information presentation, WIP does not use any predesigned texts or graphics. That is, each presentation is designed from scratch by reasoning from first principles using common-sense presentation knowledge. Through its clear separation of content and form WIP goes well beyond hypermedia systems.

The concept of tailoring presentations to the user can be seen as an extended version of the view concept known from database technology. One step on the way to intelligent interfaces for computer-supported collaborative work (CSCW) is to use multimodal systems like WIP as presentation experts that map fragments of a shared knowledge-base onto a variety of presentations satisfying the information needs of the individual group members.

1.1.3. Incremental generation

An important design goal of WIP was that the incremental generation of a multimodal presentation should be supported. Incremental generation is the immediate realization of parts of a stepwise provided input. This means that most of the computations relevant to a text or picture element are performed not long before this element is output (see [66]). This is in contrast to non-incremental systems that rely heavily on pre-planning or lookahead and plan the whole multimodal presentation at once. While incremental generation is not always needed, we claim that for systems like WIP incrementality is essential:

On the one hand, WIP must be able to begin outputting words and graphical elements before the input is complete, when the information to be expressed arrives in a stream from the back-end system, as when reporting about simultaneous events (e.g., in a control panel situation). On the other hand, WIP should be prepared for cases when the presentation goal and the input to the generator are changed in the course of generation. Such a change might be due to new high priority goals in the back-end system or the addressee's reaction to the output generated so far. Whereas a non-incremental system is only able to react to unexpected events after the complete realization of a particular presentation plan, an incremental system is able to respond more promptly. It is obvious that in most situations, human presenters follow such an incremental processing strategy (cf. [37]).

Since, in an interactive setting, a multimodal presentation system should reply fast, incrementality is useful for the sake of decreasing response time, even if the entire input is available before generation.

Of course, WIP cannot be completely incremental in the sense that it converts an element in the input stream completely in a text or picture fragment before moving on to the next element of the input stream, since this would not allow for the necessary dependencies among choices.

1.2. The current prototype of WIP

The current prototype of WIP generates multimodal explanations and instructions on assembling, using, maintaining or repairing physical devices. WIP is currently able to generate simple German or English explanations on using an espresso machine, assembling a lawn-mower, or installing a modem, demonstrating our claim of language and application independence.

We view the design of multimodal presentations including text and graphics design as a subarea of general communication design. We approximate the fact that communication is always situated by introducing generation parameters in our model. The current system includes a choice between user stereotypes (e.g., novice, expert), target languages (German versus English), layout formats (e.g., hardcopy of instruction manual, screen display), and output modes (incremental output versus complete output only). The set of generation parameters is used to specify design constraints that must be satisfied by the final presentation. A diverse set of evaluation knowledge for text, graphics and layout is necessary to select a particular design that satisfies the design specifications stated as generation parameters. WIP provides computationally tractable evaluations of candidate designs at various levels of the incremental generation process.

In summary, WIP allows the generation of alternate presentations of the same content taking into account various contextual factors such as the user's degree of expertise and preferences for a particular output medium or mode.

One of the important insights we gained from building the WIP system is that it is actually possible to extend and adapt many of the fundamental concepts developed to date in AI and computational linguistics for the generation of natural language in such a way that they become useful for the generation of graphics and text-picture combinations as well. This means that an interesting methodological transfer from the area of natural language processing to a much broader computational model of multimodal communication seems possible. In particular, semantic and pragmatic concepts like coherence, speech acts, anaphora, and rhetorical relations take on an extended meaning in the context of text-picture combinations.

The rest of the paper is organized as follows: Section 2 provides a survey of related research and highlights the distinguishing features of the WIP approach. Sections 3 and 4 introduce the functionality and the architecture of the WIP system, respectively. In Section 5, we show that techniques for planning text and discourse can be generalized to plan the structure and content of multimodal communications. Section 6 introduces an RST-based presentation planner for communicating domain plans in multimodal documents. Section 7 provides a description of WIP's mode-specific generators. While in Section 8 the interplay between presentation planning, design and realization will be discussed and illustrated by means of examples, Section 9 concentrates on our model for the coordination of text and graphics generation. Finally, we discuss limitations of the current WIP system and give an outlook for our future research directions.

2. Related research

Over the past several years, a number of projects have entered the area between natural language processing and multimodal communication, often focusing on a single specific functionality, such as the use of pointing gestures parallel to verbal descriptions for referent identification (e.g., [13,36,43]). The automatic design of complete multimodal presentations has only recently received significant attention in artificial intelligence research. The most extensive discussion of active research in this field can be found in the proceedings of a series of workshops on intelligent multimedia interfaces (e.g., [6,40,60]).

We have been engaged in work in the area of multimodal communication for several years now, starting with the HAM-ANS (cf. [65]) and VITRA systems (cf. [1,27]), which automatically create natural language descriptions of pictures and image sequences shown on the screen. These projects resulted in a better understanding of how perception interacts with language production.

Since then, we have been investigating ways of integrating tactile pointing with natural language understanding and generation in the XTRA project (cf. [36,62]). WIP grew out of the results of our previous research into multimodal interaction, particularly in the VITRA and XTRA projects.

Various user interfaces to date combine natural language and graphics, but only a few of them (cf. [34,41,52,63]) generate both forms of presentation from a common representation and therefore can explicitly address the problem of media choice and coordination.

For example, Kerpedjiev has designed a system that transforms a dataset about a particular weather situation into a multimodal weather report consisting of a text illustrated by tables and weather maps with various icons and annotations (cf. [34]).

Whereas most systems combine text with informational graphics (e.g., maps, diagrams, charts), COMET [41] and WIP [2] generate text illustrated by 3D graphics of physical objects.

The work closest to our own is being carried out in the COMET project (cf. [18,19]). Both projects share a strong research interest in the coordination of text and graphics. COMET generates directions for the maintenance and repair of a portable radio using text coordinated with 3D graphics. In spite of many similarities, there are major differences between COMET and WIP, e.g., in the systems' processing strategies, representation languages, and architectures.

COMET uses a schema-based content planner while WIP uses an operator-based approach to planning. As was shown in [45], information concerning the effects of the individual parts of a schema is compiled out. If it turns out that a particular schema fails, the system may use a different schema, but it is impossible to extend or modify only one part of the schema. In contrast, an operator-based approach enables more local revisions by explicitly representing the effects of each section of the presentation. Another advantage of an operator-based approach is that mode information can be easily incorporated and propagated during the content selection process.[2] This method facilitates the coordination of the two processes as mode selection can take place during content selection and not only after as in COMET.

Another distinguishing feature of WIP's architecture is its function of supporting incrementality, thus insuring a more fine-grained division of work between the selected presentation modes.

In contrast to COMET, WIP allows for bidirectional communication between the presentation planner and the layout manager. While COMET's layout component is supposed to combine text and graphics fragments produced by mode-specific generators during one of the final processing steps, WIP's layout manager interacts with a presentation planner before text and graphics are generated so that layout considerations can influence the early stages of the planning process and constrain the mode-specific generators. In WIP, we view layout as an important carrier of meaning.

	Informational Graphics	**3D Graphics of Physical Objects**
Static Media	Maps, Charts, Diagrams Example Systems: SAGE, FNN	Rendered Pictures Example Systems: WIP, COMET
Dynamic Media	Hypermedia Presentations Example Systems: AlFresco, IDAS	Animation Example Systems: VITRA-SOCCER, AnimNL

Fig. 1. Combining text production with four types of graphics generation.

[2] This also applies to temporal information in the case of animated presentations.

The importance of the layout dimension is also stressed by recent work at ISI that involves the generation of formatted text exploiting the communicative function of headings, enumerations, and footnotes (cf. [30]).

Whereas the majority of work has concentrated on combining static media, the VITRA-Soccer project (cf. [27], for details of VITRA's animation component see [56]), the AnimNL project (cf. [10]) and recent extensions of COMET (cf. [17]) and WIP also deal with dynamic media, such as animation. Systems like AlFresco (cf. [59]) and IDAS (cf. [47]) demonstrate that natural language generation can be enhanced by integration with hypermedia systems. In such systems, the generated text may contain links to hypercards, and canned text or images can be combined with generated text for a hypermedia presentation.

Figure 1 summarizes the various types of graphical presentations that have been combined with generated text in recent research prototypes. In all these projects, the generation system is no longer only the author of a text, but also plays the role of a desktop publisher, a hypertext designer, a multimodal interface designer, or a commentator of animations.

Whereas the projects mentioned above focus on computational methods for the automatic synthesis of multimodal presentations, [7] concentrates on the analysis and representation of presentation knowledge.

3. A functional view of WIP

The task of the knowledge-based presentation system WIP is the context-sensitive generation of a variety of multimodal documents from an input including a presentation goal. The presentation goal is a formal representation of the communicative intent specified by the back-end application system.

The example of a presentation goal in Fig. 2 represents the system's assumption about the mutual belief (BMB) of the presenter P and the addressee A, that it is P's goal that A carries out a plan denoted by the constant FILL-IN-128. This is a concrete domain plan specified as part of WIP's application knowledge. In this case, the plan is a fully instantiated sequence of actions represented in the assertional part of the hybrid knowledge representation system RAT (Representation of Actions in Terminological logics, see Section 6.1). The terminological part of RAT is used to represent the ontology and abstract plans for a particular application domain (see Fig. 2).

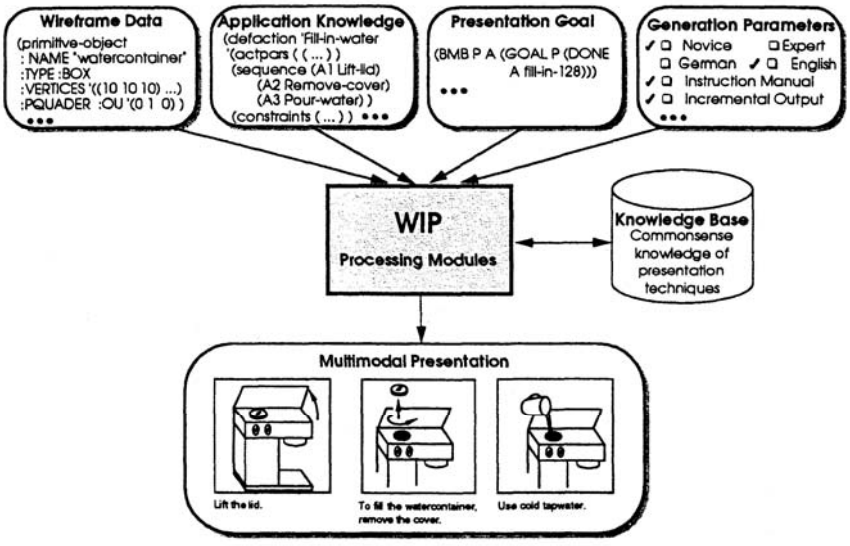

Fig. 2. WIP- a functional view.

In addition to this propositional representation, that includes the relevant information about the structure, function, behavior, and use of the technical device, WIP has access to an analogical representation of the geometry of the machine in the form of a wireframe model (see Fig. 2).

WIP is a transportable interface based on processing schemes, independent of any particular back-end system and so requires only a limited effort to adapt to a new application. Obviously, for a new

domain the application knowledge and the wireframe model must be transformed into WIP's representation schemes. In order to validate WIP's transportability we tested the system in three different application domains (espresso machine, lawn-mower, and modem). Starting from the original espresso-machine domain we did not have to change a single line of code in going to the two new domains. Only the declarative knowledge sources coded in RAT, the lexicon, and the geometric information are different. While for each domain the application knowledge and the wireframe model are fixed, the presentation goal and the generation parameters can be varied to tailor WIP's results for a particular communicative situation.

WIP is designed for interfacing with heterogeneous back-end systems such as expert systems, tutoring systems, intelligent control panels, on-line documentation, and help systems, which supply the presentation system with the necessary input. However, the current prototype has been tested with manually coded domain plans only. The presentation goal and the generation parameters have been set interactively in these test runs.

Note that the incremental output mode mentioned in Section 1.2 as one of the options for the generation of multimodal output, characterizes a likely application scenario for systems like WIP, since the intended use includes intelligent control panels and active help systems, where the timeliness and fluency of output is critical, e.g., when generating a warning. In such a situation, the presentation system must be able to start with an incremental output although it has not yet received all the information to be conveyed from the back-end system (cf. [22]). To adapt a generator to work incrementally usually complicates it, but WIP is designed right from the beginning with the incrementality of all processing stages in mind (see Section 1.1).

WIP can also be used in a stand-alone fashion, where an author specifies the necessary domain information. This leads to the long-term vision of an intelligent authoring system, that forces one to specify information only once in a formal way and then allows the generation of a possibly infinite variety of presentations of this information tailored to various audiences and media. In contrast to the current situation in technical writing and document preparation, this approach - similar to the view concept in database design - could ensure consistency across all derived presentations, since the underlying content is stored in only one place.

4. Structuring a multimodal presentation system

4.1. The need for an interleaved processing scheme

Most multimodal generation systems consist of three different kinds of processes: a content planning process, a mode selection process, and content realization processes. When designing an architecture for a multimodal presentation system, the question arises of how to organize these processes. Previous work on natural language generation has shown that content selection and content realization should not be treated independently of each other (see also [29,48]). A strictly sequential model in which data flow only from the "what to present" to the "how to present" part has proven inappropriate because the components responsible for selecting the contents would have to anticipate all decisions of the realization components. This problem is compounded if, as in WIP, content realization is done by separate components (currently a text and a graphics generator) of which the content planner has only limited knowledge.

It seems even inappropriate to sequentialize content planning and mode selection although mode selection is only a very rough decision about content realization. Selecting a mode of presentation depends to a large extent on the nature of the information to be conveyed. On the other hand, content planning is strongly influenced by previously selected mode combinations. For example, to graphically refer to a physical object, we need visual information that may be irrelevant to textual references.

A better solution is to interleave content planning, mode selection, and content realization. In the WIP system, we interleave content and mode selection using a uniform planning mechanism. In contrast to this, presentation planning and content realization are performed by separate components that access various knowledge sources. This modularization enables parallel processing, but makes interaction between the single components necessary.

Interactions are, however, only useful if the realization components are able to process information in an incremental manner. As soon as the content planner has decided which generator should encode a certain piece of information, this piece should be passed on to the respective generator. Conversely, the content planner should incorporate the results of the realization components as soon as possible.

4.2. The cascaded architecture of the WIP system

These considerations have led to the architecture shown in Fig. 3. The major components of the WIP system[3] are: a presentation planner that is responsible for determining the contents and selecting an appropriate mode combination, mode-specific generators (currently for text and graphics) and a layout manager (cf. [23]) that arranges the generated output in a document. Each generator consists of an incremental design and realization component which form a cascade. Thus the basic modularization is the same both for text and graphics generation, resulting in two parallel cascades.

The presentation planner and the mode-specific generators interact incrementally in a pipelined mode. In other words, text and graphics design and even the verbalization and visualization can start, before the presentation plan is completed. The text and graphics design components can be seen as micro-planners of the "what to say" and "what to show" parts of the mode-specific generators. For example, lexical choice is not carried out by the presentation planner on the macro-plan level, but by the text design component.

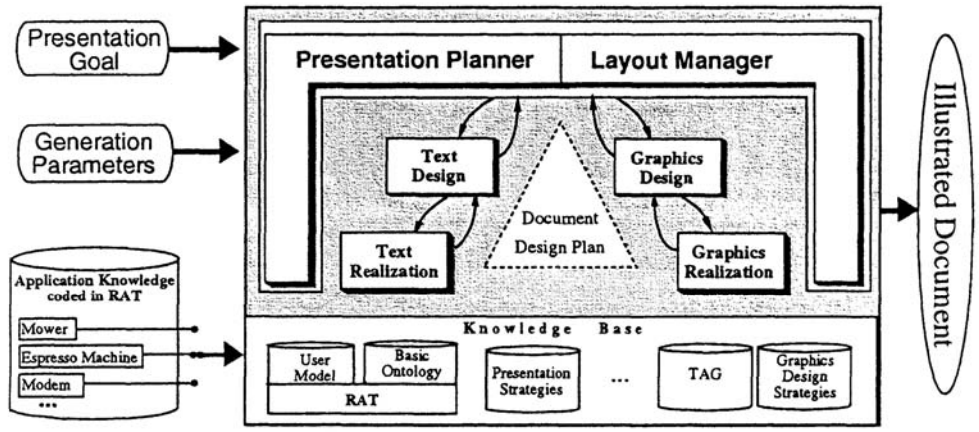

Fig. 3. The architecture of the WIP system.

There is no direct communication from a mode-specific realization module back to the presentation planner or layout manager, but all such communication is mediated by the corresponding design module. As soon as the presentation planner and the layout manager have made enough commitments to allow the mode-specific generators to start work, the text and/or graphics design components are activated. Then the control passes back and forth between the modules of the cascade, interleaving their execution.

To prevent disconcerting or incoherent output, the document design plan keeps the history of the design decisions on all levels of the incremental generation process. This means that decisions of the language generator may influence graphics generation and that graphical constraints may sometimes force decisions in the language production process.

The incremental processing mode with feedback and negotiation among the components supports self-monitoring and the anticipation of the addressee's interpretation (see [62]).

WIP's basic ontology and user model are represented in the terminological logic RAT (cf. [26] and Section 6.1). In addition, WIP's knowledge base includes declaratively coded presentation strategies (see Section 6.2.1), graphical design strategies (see Section 7.1) and a lexicalized Tree Adjoining Grammar (TAG, see [25] and Section 7.2).

[3] As the result of a 30-man-year effort the WIP prototype is fully implemented, comprising 5.5 MB of Common Lisp and CLOS source code.

5. Generalizing language generation to multimodal presentations

Since a lot of progress has been achieved in natural language generation, it is quite natural to wonder whether it is possible to generalize the underlying concepts and methods in such a way that they become useful in the broader context of multimodal presentations. Although new questions arise, e.g., how to optimally divide the work between the available presentation modes, a lot of tasks in multimodal generation bear much resemblance to problems occurring in natural language generation, in particular, the structuring of the presentation in a coherent manner and the establishment of cohesive links by appropriate cross-references.

5.1. The generation of multimodal documents as a goal-directed activity

Our approach is based on the assumption that not only the generation of text, but also the generation of multimodal documents can be considered as a goal-directed activity (cf. [4]). We presume that there is at least one act that is central to the goal of the whole document. This act is referred to as the *main act*. Acts supporting the main act are called *subsidiary acts*. This distinction between main and subsidiary acts essentially corresponds to the distinction between *global* and *subsidiary speech acts* in [57], *main speech acts* and *subordinate speech acts* in [61], and between *nucleus and satellites* in the Rhetorical Structure Theory (RST) proposed in [39]. Since main and subsidiary acts can, in turn, be composed of main and subsidiary acts, a hierarchical document structure results. While the root of the hierarchy generally corresponds to a complex communicative act such as describing a process, the leaves are elementary acts, i.e., speech acts (cf. [57]) or pictorial acts (cf. [35]).

5.2. An extended notion of coherence for multimodal documents

A number of textlinguists have characterized coherence in terms of semantic and pragmatic coherence relations that hold between the parts of the text (e.g., see [24,28]). Semantic relations, such as *Sequence*, directly correspond to the structure of the domain whereas pragmatic relations, such as *Motivation*, refer to the communicative function of document parts. Perhaps the most elaborated set of coherence relations is presented in RST (cf. [39]). Examples of RST relations are *Sequence, Motivation, Elaboration, Enablement, Interpretation,* and *Summary*. Text-picture researchers have investigated the role a particular picture plays in relation to accompanying text passages. E.g., Levin has found five primary functions (cf. [38]): *Decoration, Representation, Organization, Interpretation,* and *Transformation*. Hunter and colleagues distinguish between: *Embellish, Reinforce, Elaborate, Summarize,* and *Compare* (cf. [31]). An attempt at a transfer of the relations proposed by Hobbs to pictures and text-picture combinations has been made in [11]. Unfortunately, text-picture researchers only consider the communicative functions of whole pictures, i.e., they do not address the question of how a picture is organized. To get an informative description of the whole document structure, one has to consider relations between picture parts or between picture parts and text passages, too. E.g., a portion of a picture can serve as background for the rest of the picture or a text passage can elaborate on a particular section of a picture. We have analyzed several illustrated documents in order to find out which relations occur between textual and pictorial document parts (cf. [3]). In particular, we have examined the relations found by text-picture researchers (cf. [38]) and those proposed in RST (cf. [39]). To ensure that the user recognizes how document parts relate to others, a multimodal presentation system has to know which mode combination conveys a certain relationship most effectively.

6. Plans for communicating plans

A basic assumption behind the WIP model is that not only the generation of text and dialog contributions, but also the design of graphics and multi-modal presentations are planning tasks (cf. [5]). When explaining how a complex process functions, WIP generates and realizes plans for

communicating domain plans provided by the back-end system. The elements of the plans generated by WIP are communicative acts that verbalize and visualize the physical acts specified in a given domain plan.

6.1. Representing domain plans

As WIP is designed as a presentation system, our research is focused on the generation of presentation plans, not domain plans. Nevertheless, domain plans are an essential part of WIP's input and therefore must be made accessible to the presentation system. Moreover, for the design of presentations WIP must be able to perform certain reasoning tasks on domain plans - although domain plans are not generated by WIP, but by application systems. In order to have a well defined interface between the application system and WIP, we assume that domain plans are represented in RAT terms.

The RAT module (cf. [26]) is used both for the generation of text and graphics as the main source of knowledge about the domain. Besides the domain plans the entire information concerning the domain terminology is represented in RAT. In order to support the user modeling RAT provides partitioning mechanisms to reason about the potentially conflicting views of the world the user and the system may have.

The architecture of RAT was inspired by the need for a tool for the reasoning about concepts and instances of the domain as well as actions, plans, and relations between them. Terminological representation systems have proven to be adequate formalisms for the representation of ontologies in various applications [46]. However, besides their abilities of managing concept and instance descriptions, they do not provide any meaningful way of representing temporal or causal relationships. On the other hand various STRIPS-like systems have been developed that provide powerful tools to synthesize and retrieve plans (cf. [12]). The shortcomings of these systems, however, are their limited services concerning the reasoning about the objects in the domain and relations between plans. In order to merge the advantages of both types of systems, RAT was designed as an extension on top of the terminological logic KRIS [9] with close links between action and concept representation.

The presentation planner can make use of a number of reasoning services provided by RAT, e.g., temporal projection, plan subsumption, or the simulated execution of plans. For instance, suppose the domain plan is nonlinear, i.e., some subplans PI and P2 can be executed in any order and PI needs a longer explanation than P2 (because its explanation should contain an illustration, for instance). Now suppose that the layout manager informs the presentation planner that only a little space is left on the current document page. In this case the presentation planner would decide first to present P2 and then PI. In order to reason about the world state after the user's execution of P2 the presentation planner can make use of RAT's inference services, namely, the simulated plan execution. This is critical for the design of the illustration used for PI since the shown state of the world should include the effects of P2. In some cases it might be helpful to explain a sequence of several subactions on a more abstract level. RAT supports such an abstraction by finding a plan sequence which is composed of these sub-actions. In other cases the explanation of a later subplan can be shortened by referring to a subplan which has already been presented if RAT detects that they subsume each other.

Like in other state-based formalisms RAT actions are defined by the change they cause in the world state. We distinguish between *atomic actions*, which are non-decomposable and defined by a pre- and postcondition and *plan schemata*, which represent sequences of actions with possible constraints on the objects involved. In contrast to other STRIPS-like formalisms the pre-and postconditions of atomic actions are described by using a subset of the underlying terminological logic, namely, conjunctions of feature restrictions, agreements, and disagreements. By that the underlying terminological logic provides a limited form of a background theory and, as a consequence, predicates are not unrelated but ordered by the subsumption relation. In addition, a set of feature restrictions interpreted as *action parameters* is specified that play the role of "formal parameters" of the action.

Formally, an atomic action is a triplet ⟨*pars, pre, post*⟩ where pars is a conjunction of restrictions on feature atoms: $f_1 : C_1 \sqcap \ldots \sqcap f_n : C_n$, which is interpreted as a set of (typed) *action parameters; pre* is a conjunction of feature (or feature chain) restrictions (p : C), agreements ($p \doteq q$), and disagreements ($p \not\doteq q$), and is interpreted as the *precondition of the action*; *post* has the same form as *pre* and is

interpreted as the *postcondition* of the action.[4]

In order to illustrate the definition, let us consider the following two example actions:

```
put-cup-under-water-outlet =
  ⟨(agent : person ⊓ object : cup ⊓ machine : espresso-machine),
   (object.position ≐ agent.has-hand.inside-region),
   (object.position ≐ machine.has-water-outlet.under-region)⟩

turn-switch-to-espresso =
  ⟨(agent : person ⊓ machine : espresso-machine),
   (machine.has-switch.position : off-position ⊓
    machine.state : (off ⊓ ready)),
   (machine.has-switch.position : espresso-position ⊓
    machine.state : on)⟩
```

In plain words, the action put-cup-under-water-outlet has the action parameters agent, object, and machine, the precondition is that the cup is held by the agent's hand, and the postcondition is that the cup is located under the water outlet. Note that, e.g., agent.has-hand.inside-region is not a single, primitive feature, but the composition of the three features agent, has-hand, and inside-region, which are defined in the taxonomy. Similarly, the action turn-switch-to-espresso has two action parameters agent and machine, the precondition is that the switch is in the "off position and that the machine is off and ready, and the postcondition is that the switch is in the "espresso" position and the machine is running.

Atomic actions can be composed to form plan schemata, which are specified by a set of action parameters, a sequence of actions, and, in contrast to similar formalisms, equality constraints on the action parameters of the plan schema and the actions involved. Formally, a plan schema is a triplet ⟨*pars, seq, constr*⟩, where *pars* represents the action parameters of the plan schema in the same way as for atomic actions, *seq* is a sequence of pairs consisting of *labels* and *actions*, which may be either atomic actions or plan schemata, and *constr* is a conjunction of agreements expressing equality constraints on the action parameters. Consider as an example an excerpt of the plan schema for making espresso:

```
make-espresso =
  ⟨(agent : person ⊓ object1 : cup ⊓
    object2 : espresso-machine ⊓ ···),
   (...,
    A5: put-cup-under-water-outlet,
    A6: turn-switch-to-espresso,
    ...),
   (object2 ≐ A5.machine ⊓
    object2 ≐ A6.machine ⊓ ···)⟩
```

The precondition of an action must be satisfied by the current world state to allow the execution of the action. This is checked by mapping this problem into the underlying terminological logic and testing if the subsumption relation holds between the precondition and the current world state. The postconditions are asserted to be valid after the successful execution by interpreting their restrictions on the world state as assignments. Note that by allowing equations between feature chains in the postcondition we permit structural changes as opposed to simple changes in truth-values of atomic formulae, as in STRIPS-like systems.

RAT shows that the design of a plan representation system as an extension of a terminological logic can be successfully exploited to provide a variety of interesting and new reasoning services like plan subsumption, temporal projection of conditions, or the simulated execution of plans. In contrast to other approaches which combine terminological and temporal reasoning like CLASP [16] or T-REX [68] whose focus is on plan recognition, the RAT system additionally allows for detailed descriptions of states as pre-and postconditions. On the other hand, these systems currently provide a much richer language to combine actions to plans (regular expressions and temporal constraints, respectively).

[4] The formal notation follows [8].

6.2. Plan-based mode selection, content determination, and organization

As argued in Section 5, text-picture combinations follow similar structuring principles as text. In particular, a document is characterized by its intentional structure that is reflected by the presenter's intentions and by its rhetorical structure that is reflected by various coherence relations. Therefore, it was quite natural to extend methods for text planning in such a way that they become also useful for multimodal presentations.

6.2.1. Representing presentation knowledge

In order to generate multimodal presentations, we have defined a set of presentation strategies that can be selected and combined according to a particular presentation task. Such presentation strategies reflect general presentation knowledge or they embody more specific knowledge of how to present a certain subject.

To represent presentation strategies, we follow the approach proposed by Moore and Paris (cf. [42]) to operationalize RST for text planning. However, an additional slot for the presentation mode must be introduced. The strategies are represented by a name, a header, an effect, a set of applicability conditions and a specification of main and subsidiary acts. Whereas the header of a strategy is a complex communicative act (e.g., to enable an action), its effect refers to an intentional goal (e.g., the user knows a particular object).[5] After the successful execution of a strategy, the user model is updated by adding the effect to the knowledge base via RAT's TELL language. The applicability conditions specify when a strategy may be used, and constrain the variables to be instantiated. To evaluate an applicability condition, knowledge represented in RAT is accessed via the ASK language. Example requests are: finding all instances of a certain concept, finding role fillers, realizing object or domain action instances or finding all subactions of a domain plan. We would like to stress that some requests go beyond pure knowledge retrieval. For example, when describing a complex domain plan, a presenter often relies on presentation strategies which involve the depiction of intermediate world states after the execution of certain actions. Since the RAT representation of a complex domain plan does not comprise intermediate world states, they have to be inferred using RAT's inferential services (see Section 6.1).

The kernel of the presentation strategies is formed by main and subsidiary acts. For example, the strategies below can be used to show the orientation of an object in a picture and to ensure that it is identifiable. Whereas graphics must be used to carry out the main acts in these strategies, the mode for the subsidiary acts is still open.

```
(def-presentation-strategy
  :Header (Describe P A (Orientation ?orientation) G)
  :Effect (BMB P A (Has-Orientation ?x ?orientation))
  :Applicability-Conditions
    (Bel P (Has-Orientation ?x ?orientation))
  :Main-Acts
    (S-Depict P A (Orientation ?orientation) ?p-ori ?picture)
  :Subsidiary-Acts
    (Achieve P (BMB P A (Identifiable A ?x ?px ?picture))
             ?mode))

(def-presentation-strategy
  :Header (Background P A ?x ?px ?picture G)
  :Effect (BMB P A (Identifiable A ?x ?px ?picture))
  :Applicability-Conditions
    (AND (Bel P (Image-of ?x ?px ?picture))
         (Bel P (Perceptually-Accessible A ?x))
         (Bel P (Part-of ?x ?z)))
  :Main-Acts (S-Depict P A (Object ?z) ?pz ?picture)
  :Subsidiary-Acts
    (Achieve P (BMB P A (Identifiable A ?z ?pz ?picture))
             ?mode))
```

[5] In [42], this distinction between header and effect is not made because the effect of their strategies may be an intentional goal as well as a rhetorical relation.

Since there may be several strategies for achieving a certain goal, criteria for ranking the effectiveness, the side-effects, and costs of executing presentation strategies are needed.

To formulate selection criteria, we use meta-rules. For example, the metarule below suggests the use of graphics rather than text when presenting spatial information.

```
IF (IS-A ?current-attribute-value Spatial-Concept)
    THEN (Dobefore *graphics-strategies* *text-strategies*)
```

A basis for our meta-rules and presentation strategies form extended studies of relevant psychological literature and our own analyses of various illustrated documents. In particular, we identified seven information types (concrete, abstract, spatial, covariant, temporal, quantification, negation) with several subtypes and ten communicative functions (attract-attention, compare, elaborate, enable, elucidate, label, motivate, evidence, background, summarize) and examined which mode or mode combination conveys them best. For example, it is very difficult or even impossible to graphically depict quantifiers (such as *some* or a *few*) whereas graphics are in general the preferred modality for communicating visual attributes (concrete information), for more details see [5]. Although we focused on the nature of information and the communicative function of a document, there is no doubt that other criteria (e.g., user characteristics and resource limitations) are also important.

6.2.2. The presentation planning process

At the heart of the presentation system is a parallel top-down planner and a constraint-based layout manager. The presentation planner receives as input a high-level presentation goal (see Fig. 4). It then tries to find a presentation strategy whose effect or header match the presentation goal and generates a refinement-style plan in the form of a directed acyclic graph (DAG). The leaves of the planned DAG are specifications for elementary Integration of natural language and graphics generation acts of presentation. They are sent to the appropriate task queue (see Fig. 4). The text designer handles elementary speech acts, such as s-assert (generate a surface structure for an assertion) or s-request (generate a surface structure of a request), the graphics designer executes pictorial acts, such as s-depict (depict an object) or s-annotate (label an object). During the text and graphics generation processes, further refinements of individual presentation goals are possible.

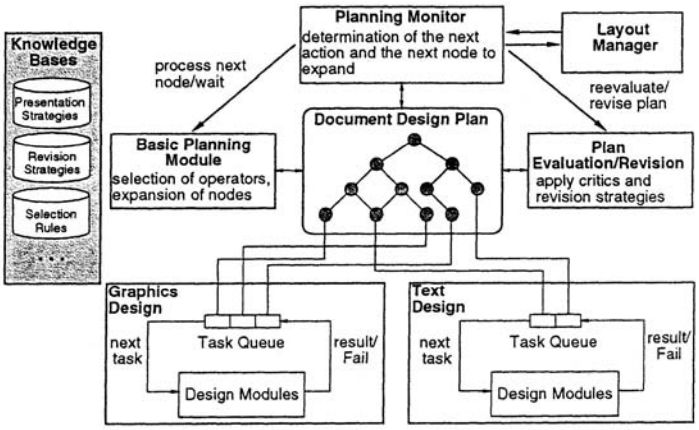

Fig. 4. The presentation planner of WIP.

Since the presentation planner has no direct access to knowledge concerning mode-specific realization, it cannot consider this information when building up a candidate document structure. Consequently, it may happen that the results provided by the generators deviate to a certain extent from the initial document plan. Such deviations are reflected in the DAG by *output sharing, structure sharing,* and *structure adding*. Output sharing occurs when parts of the generated output are reused for different purposes, e.g., as part of a labeling relation and as part of a background relation (see also Section 8). Structure sharing is similar to output sharing. It occurs when not only parts of the output, but also a more complex part of the DAG are shared. For example, let us assume the presentation

planner decides to show an action and its result by means of two pictures. To orientate the user, it is planned to show background objects in both pictures. If the graphics designer is able to convey the requested information in a single picture, the background for the actions has, however, to be included only once. Consequently, the structure of the document can be simplified by factoring out the background branch. Whereas structure sharing leads to simplifications of the initial document plan, structure adding results in a more complex plan. It occurs, e.g., if the graphics generator is expected to integrate information in a single picture, but is only able to convey the information by generating several pictures.

Restructuring methods are applied when the results of the generators do not correspond to the initial document plan. However, it may also happen that the generators are not able to accomplish a task. In such situations, restructuring methods do not lead to a result. Instead, the planner will have to revise its initial proposal by choosing another presentation strategy or by instantiating variables differently. To ensure consistency of the document, all changes have to be propagated to other parts of the document.

Information must flow not only between the content planner and the generators, but also from one generator to the other. Let us suppose the text generator has generated a referring expression for an object shown in a picture. If the picture is changed due to graphical constraints, it might happen that the referring expression no longer fits. Thus, the planner will have to create a new object description and pass this description on to the text generator, which will have to replace the initial referring expression by a new one. As shown in Fig. 4, the leaves of the document plan are connected to entries in the task queues of the mode-specific generators. Thus, the document design plan serves not only as an interface between the planner and the generators, but also enables a two-way exchange of information between the generators.

Furthermore, the need for propagating data during presentation planning arises when dealing with dependencies between presentation strategies. For example, a decision about mode selection often depends on earlier decisions. Assume the system decides to compare two objects by describing the different values of a common attribute. At this time, the only restriction is that both descriptions should be realized in the same mode. Once the system has decided on the mode for the attribute value of the first object, the result of this decision must be made available for describing the value of the second object. This problem can be handled by passing mode information during the planning process both from top to bottom and from bottom to top. Mode information is propagated via the header of a strategy. Depending on whether the main acts of a strategy are to be realized in text, graphics, or both modes, the values T(ext), G(raphics), or M(ixed) are assigned. The mode remains unspecified until mode decisions are made for the main acts of a strategy. By deferring mode decisions for as long as possible, the planner is able to continue planning without making selections that are too specific.

Due to the distributed processing scheme of WIP, there is no guarantee that the results of the individual components will always be available at a given time. In some situations, it might happen that the planner is not able to expand a node because it is still waiting for a generator to supply results. To avoid processing delays, WIP's presentation planner does not always expand nodes in a depth-first fashion, but selects the nodes to be expanded in a flexible way, considering heuristics, such as the number of assumptions to be made. To allow for alternating revision and expansion processes, WIP's presentation planner is controlled by a plan monitor that determines the next action and the next nodes to be expanded.

7. Mode-specific content realization

7.1. The graphics generator

In illustrated instructions for technical equipment, graphics are used in order to accomplish presentation tasks, such as depicting a domain object in a certain state, showing an object's location, or visualizing the course of an action. As a starting point, we operationalized certain 2D and 3D illustration techniques frequently used by human illustrators. Inspired by the compositional approach to computational semantics of natural language, our formalization is based on a compositional semantics of pictures. A picture is regarded as a composition of a picture frame and a set of images located within this frame. Each image is treated as an object that is characterized by a restricted 2D

region and a set of attributes including visual properties, such as shape, color/gray pattern. In accordance with the underlying source from which an image is derived (cf. Fig. 5), we can distinguish between several basic image types:

- images that result from mapping a 3D model of an object or an object configuration onto a plane 2D region;
- images of 2D concepts such as point, line, arrow, rectangle, etc., which are often used in 3D illustrations as metagraphical objects. These images are considered as instantiations of generic 2D concepts;
- images that are created by typesetting character strings or symbols.

To produce graphics including different image types, we have developed a graphics realization component, comparable to an object-oriented graphics editor (cf. [51]). The operators handled by this component fall into three classes:

(1) operators for creating and manipulating wireframe models of 3D objects; examples include: adding an object to a configuration, spatially separating object parts in order to construct exploded views, and cutting away object faces to make opaque parts visible;
(2) operators which constrain projection parameters and map wireframe models onto images, e.g., it is possible to map models onto schematic line drawings or to produce more realistic looking depictions using rendering techniques;

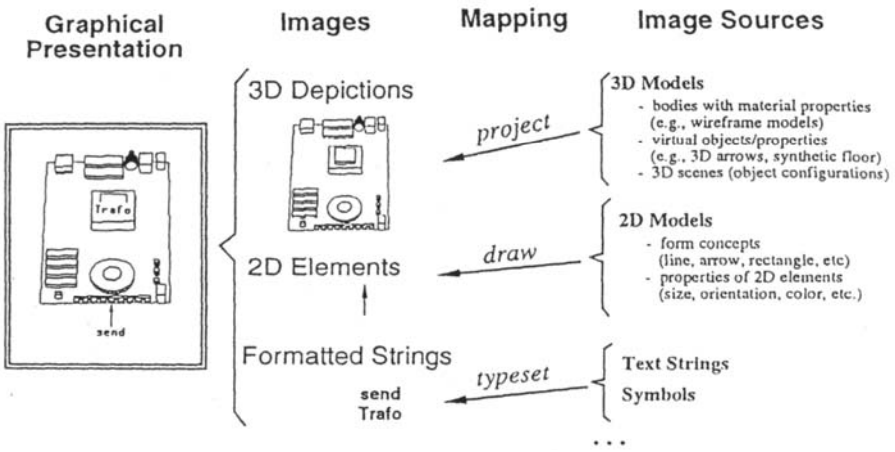

Fig. 5. Graphics as compositions of images from different sources.

(3) operators that are defined on the picture level, e.g., annotating an object image with a text label, or scaling/framing/coloring picture parts.

The annotated modem shown in the left part of Fig. 5 can be created with the realization component through the following sequence of operators: take a wireframe model of the modem circuit board, choose a viewing specification so that the whole object is in the view-volume and the top part of the circuit board is visible, take a schematic perspective projection as mapping function, apply the projection and paste the resulting image into a picture frame, then make an arrow-annotation to relate the formatted string "send" to the image of the modem's LED indicating the send mode, finally annotate the transformer image with the string "Trafo" by writing it onto the image.

In the WIP system, operator sequences, as in the example above, are generated by the graphics design component (cf. [50,51]) starting with presentation tasks forwarded by the presentation planner. A basic idea which underlies our representation of design knowledge, is that we do not directly relate presentation tasks to graphical presentations. Instead we associate presentation tasks with a set of constraints. These constraints place restrictions on image sources (e.g., 3D models), mappings and images in a picture. Thus, they eventually constrain the set of graphical presentations in a way that a

presentation task can be achieved. This enables us to cover a variety of plausible designs for one and the same presentation task with a single set of constraints. Among others, this has the advantage that the graphics designer can flexibly carry out several presentation tasks with a single graphics - provided the graphics satisfies all constraints associated with the presentation tasks. Such flexibility is particularly needed, if the graphics generator receives input from the presentation planner in a piecemeal fashion, as is the case in the WIP system. Recognizing whether or not new information can be incorporated into an already designed picture is done by checking whether the picture already meets the new constraints or whether it can be modified in such a way that the new constraints can be met.

While the presentation strategies introduced in Section 6.2.1 serve to decompose communicative goals into elementary presentation tasks, we use *graphical design strategies* in order to relate elementary tasks to constraints on the graphics to be generated. Some of these constraints are directly related to operators which are to be executed by the realization component, others lead to the application of further design strategies. For example, a graphical design strategy to depict a physical object in a picture embodies the following constraints: there must be a wireframe model of the object that is to serve as an image source. If the object is to be shown with further objects, there must be a viewing specification such that the object is visible. The resulting image must be included in a picture, and the image must not be obstructed by any other picture elements.

Using graphical design strategies, graphics design is in principle a goal-driven planning process, i.e., presentation tasks are related to constraints and after several refinement steps a sequence of instructions for the realization component is obtained. However, it does not seem feasible to strictly separate a graphics design and a realization phase as some realization operators have side effects which are difficult (i.e., computationally expensive) to anticipate in advance. For example, minor changes in a 3D configuration may dramatically affect the visibility of objects and the discriminability of object images. A solution to this problem is to interleave graphics design with graphics realization and to allow for feedback. During the design process we have to check whether constraints have already been satisfied and if they are still being met even after further realization operators have been applied. Therefore, the realization component provides not only *achievement* operators which produce effects on models, mappings or pictures, but also provides *evaluation* operators[6] (e.g., in checking if an object as part of an object configuration is visible from a given viewing specification, or in checking if a picture part can be discriminated from other picture components). Evaluation operators are also useful in coping with phenomena that cannot be properly described by our compositional semantics. For example, using reverse video is a means for highlighting an item. However, applying this technique too frequently in a single picture will weaken the intended effect.

To enable the graphics designer to flexibly combine design strategies, the strategies should only contain a minimal set of graphical constraints associated with a single presentation task. For example, the task of depicting an object does not prescribe how to encode particular object attributes (such as shape, size or surface structure); therefore, a graphical design strategy should not include any corresponding constraints. In some situations the set of constraints placed on a picture may be augmented by further presentation tasks. In situations where several choices remain, the graphics designer uses heuristics to make the necessary decisions, e.g., to choose among a set of possible view directions (cf. [49]). Heuristics are also needed in finding a priority order if several design strategies apply, or if more than one realization operator can be used to satisfy a constraint. For example, to establish a labeling relation between an image and a text string in a picture the graphics designer's repertoire of annotation techniques currently covers 33 variations of three basic annotation techniques (writing on an object image, along an object image and annotating with an arrow). Which annotation technique applies depends on syntactic criteria (e.g., formatting restrictions) as well as semantic criteria. For example, in order to avoid confusion, the same annotation technique should be used for all instances of the same basic concept. The current WIP prototype relies on about 50 rules in order to select an appropriate annotation technique (cf. [69]). This shows that WIP does not use planning operators or schemata on all levels of the design process, but exploits expert design knowledge for routine subtasks like annotation, grid determination and font selection.

---

[6] Achievement operators and evaluation operators are comparable to style methods and style evaluators in the IBIS system (cf. [58]).

7.2. The text generator

As for the graphics generator the design of the text generator was strongly influenced by the quest for incremental processing. Thus the form and size of basic processing units, data flow, and the interaction between the components of the text generator were determined by this incremental processing scheme.

This section focuses on principles of the inner working of the generator, especially on the interrelations between the levels of generation resulting from dependencies among choices (see Fig. 6).

The first component that is activated during natural language generation in WIP is the *text design* component. As soon as the presentation planner decides, in its mode selection process, that a particular element should be presented as part of a text, the element is handed over as input to this component. The main task of the text design component is the organization of the input elements into clauses. This comprises for example the determination of the order in which the given input elements can be realized in the text, the control of the use of anaphora to obtain a coherent text, and lexical choice. The resulting preverbal message is input to the *text realization* component in a piecemeal fashion where grammatical encoding, linearization, and inflection take place.

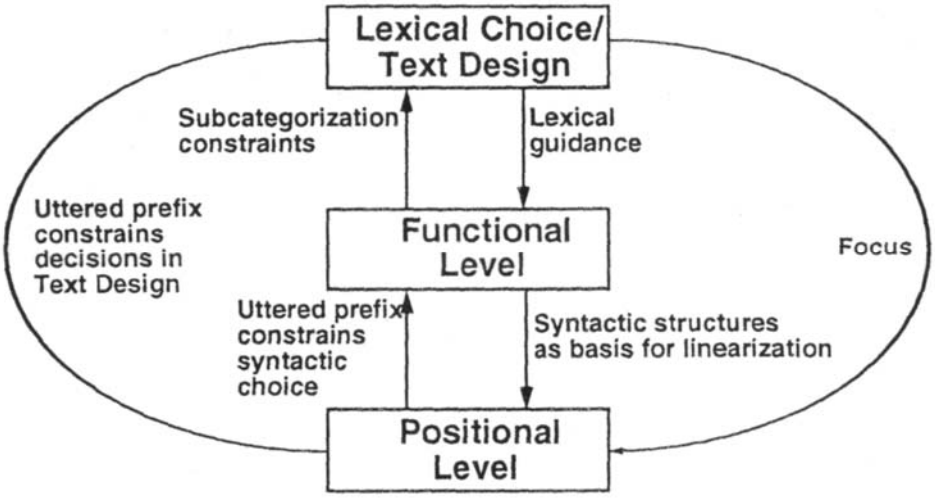

Fig. 6. Design of a system with incremental output.

In accordance with the requirement of lexical guidance of the generation process (cf. [33,44]) the process of lexical choice for an input element is made within the text design component before the element is handed over to the text realization component. The text realization component consists of a functional and a positional level (see Fig. 6). Lexemes in the input to the text realization component direct the choice of syntactic structures. To facilitate this selection process, WIP uses a lexicalized grammar where each syntactic rule is associated with at least one lexeme serving as head element in the represented phrase. These anchors of the grammar in the lexicon can be utilized to select the elementary structures for grammatical encoding (cf. [25]). A second dependency between text design and text realization results from the subcategorization constraints of the previously chosen lexemes. They provide a syntactic context for the further lexical choice. In order to be able to report this additional syntactic information to the text design component, the cascaded architecture of the text generator allows for feedback between the two components.

The granularity of the processing units is especially important in the text realization component that is conceived as a distributed parallel model, because the simultaneous activity on existing parts of the syntactic structure supports the incremental processing of these parts (cf. [21,55]). These structures must be small enough to avoid redundancy and to allow the specification of input in a piecemeal fashion. They must be large enough to be operated on relatively independently from other structures. We use a lexicalized TAG (LTAG, cf. [53]) for the syntactic level of description. Its extended domain of locality (cf. [32]) and the flexible expansion of partial structures by substitution and adjunction (cf. [15]) make it a good candidate for incremental syntactic generation (cf. [20,54]).

The separation of knowledge concerning dominance and linear precedence relations (see Fig. 6) is a result of the assumption that the chronological order in which syntactic segments are attached does not correspond to the linear order of the resulting utterance. This separation results in another bidirectional dependency between processing levels: On the one hand, the syntactic structures at the functional level are the data to be linearized at the positional level. On the other hand, in a system with incremental output it is no longer guaranteed that a correct position can be found for each syntactic structure that can be integrated at the functional level (cf. [22]). For example, it is always possible to realize a modifying adjective as an attribute in an NP at the functional level. This results in phrases such as "the big switch". If however, the noun was already uttered, then for example the realization of "big" in a relative clause as in "the switch ... that is big" should be preferred. In this case, knowledge at the positional level orders the selection of structures at the functional level.[7]

Furthermore, dependencies exist between decisions in the text design component and the positional level of the text realization component: the interpretation of semantic and pragmatic criteria by the text design component may influence the selection of linearization rules. Conversely, the prefix, which has already been uttered, may constrain the realization of further input elements, directed by the text design component. An example of this dependency is depicted in the following situation: suppose WIP has already output the fragment "Then the modem sends you". If the text design component decides to reduce the NP "the return code" to "it", the pronom-inalization has to be rejected by the positional level. There are two options: performing a repair like "... sends it to you" or ignoring the demand for pronominalization as in "... the return code".

8. Interleaving presentation planning, design, and realization

In this section, the planning process and the interplay of the planner and the generation components for text and graphics are discussed in more detail. Let us assume the presentation planner intends to describe a sequence of two actions PUT-1 and TURN-1. Figure 7 shows the DAG that has been produced by the presentation planner. The presentation goal

(DESCRIBE P A (SEQUENCE PUT-1 TURN-1) T)

has been decomposed into two subgoals: (REQUEST P A PUT-1 T) and (REQUEST P A TURN-1 T). After the refinement of (REQUEST P A PUT-1 T), five acts[8] have been posted as new subgoals: a complex communicative act (ENABLE) which has to be further expanded, an elementary speech act (S-REQUEST) which is passed onto the text designer and four referential acts (ACTIVATE) for filling the semantic case roles associated with the action PUT-1.

As mentioned above, the planner passes a certain piece of information onto the respective generator as soon as it has decided which component should encode it. In the example, (S-REQUEST PA...) is sent to the text designer although the semantic case roles have not yet been filled at that stage. The text designer attempts to generate input for the TAG generator which starts processing this input, but is not able to produce any output before a content word of the utterance has been determined. While the text generator is still working on the realization of the actual request, the presentation planner already expands the ENABLE act. Since it assumes that the user is not able to localize the water outlet, it decides to introduce it by annotating it in a picture that includes the water outlet and the espresso machine as background. As a first presentation task, the graphics designer receives

(S-DEPICT P A (OBJECT WATEROUTLET-1)
(? PX) (? PICTURE)).

The graphics design component has to map this presentation task onto a sequence of operators to be executed by the graphics realization component.

Note that the graphics designer receives the presentation tasks in a piecemeal fashion. As a

[7] Note, that this construction is only possible if in the meantime nothing was uttered after the noun. For more details about synchronization and effects of incremental output on incremental natural language generation, see [22].

[8] In Fig. 7, MA stands for main act, SA for subsidiary act (cf. Section 5.1).

consequence, the graphics generator must be able to process new input depending on what has been generated before. Among other things, this includes recognizing whether new information can be incorporated into previously designed pictures or not (cf. Section 7.1). In our example, the graphics designer receives the task of depicting the espresso machine as background while processing the first presentation task. To accomplish the new presentation task, the same graphical design strategy as before may be applied. However, the graphics generator has to check after each step whether previously satisfied constraints are still being fulfilled; e.g., it might happen that objects which were previously visible are obstructed by objects that are added at a later stage. In the example, the perspective has been constrained in such a way that both the entire espresso machine and the water outlet are visible.

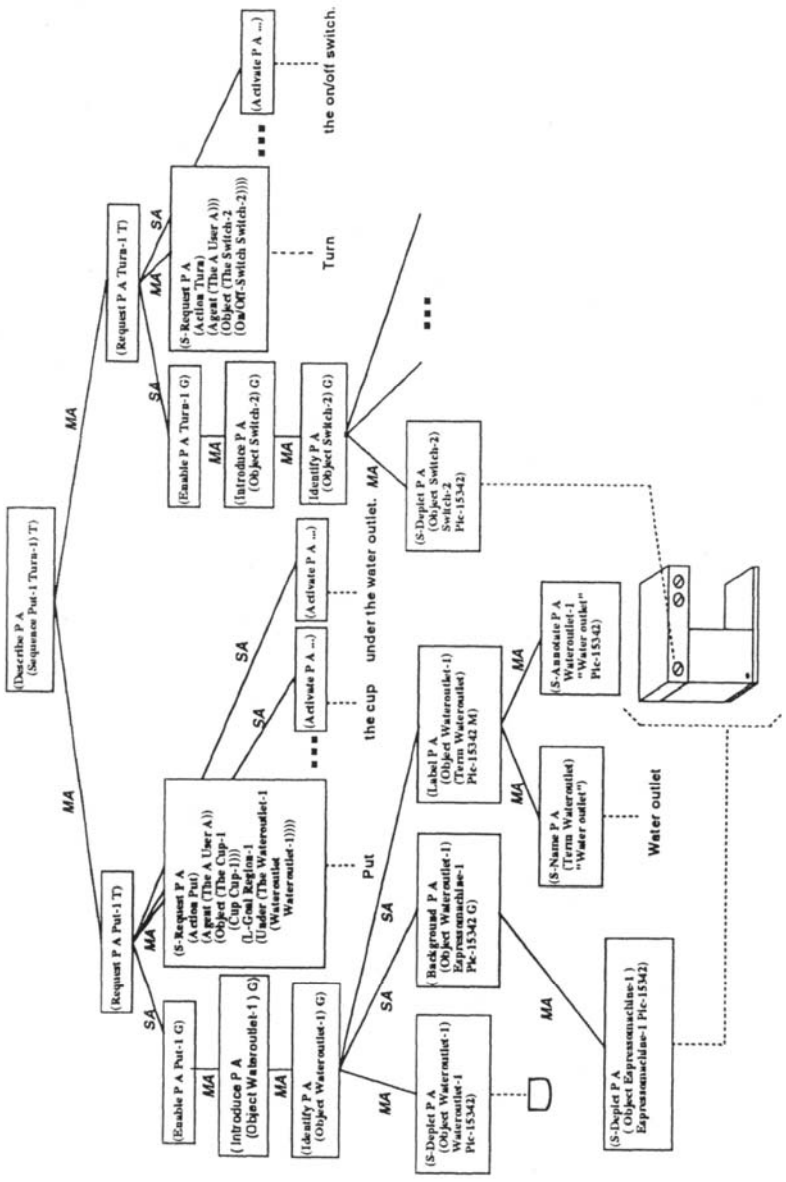

Fig. 7. DAG representation of the planned multimodal document.

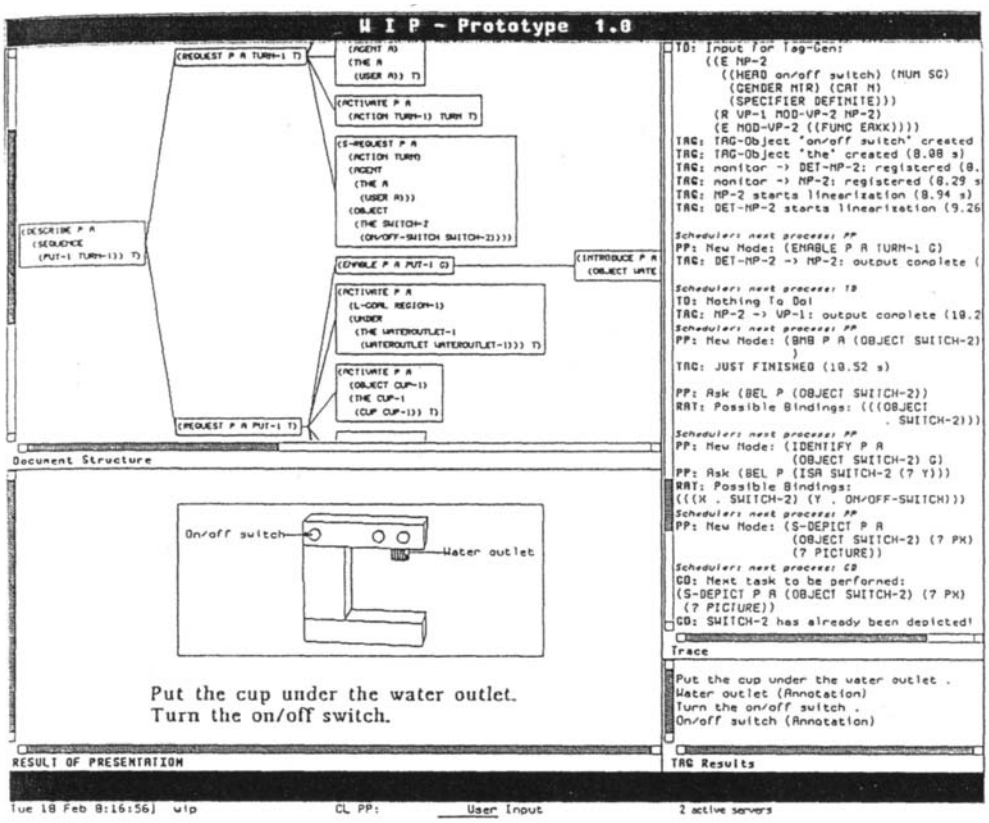

Fig. 8 Generating a multimodal presentation

While the graphics generator is still concerned with depicting the espresso machine as background, the TAG generator produces a natural language expression for the label which later has to be pasted as part of an arrow annotation into the picture.

Figure 8 shows a screen copy of a session with the WIP prototype. The snapshot of the system trace was taken immediately after the second request was verbalized and the on/off switch was annotated in the picture. In the upper part of the trace pane, one can see the input specification for a noun phrase that the text designer has sent to the TAG generator. Note that the specification for other parts of the sentence have already been sent and processed earlier.

In the third last line of the trace pane, the graphics designer selects

(S-DEPICT P A (OBJECT SWITCH-2) (? PX) (? PICTURE))

from the task queue. Since it finds out that the switch has already been depicted, no further picture generation is necessary (see the last line in the trace pane). The presentation planner registers this by linking the corresponding parts with each other in the DAG (cf. Fig. 7) that forms a part of the document design plan.

The above example is a kind of visual anaphora. As for a linguistic anaphora, such as (1), the antecedent of the anaphora is part of an object that was previously mentioned in the discourse.

The machine is running. The on/off switch was turned on. (1)

In the case discussed here, a projection of SWITCH-2 has already been displayed as part of the background provided for the picture of the water outlet (see Fig. 7). The multimodal document design plan plays the role of a discourse model in traditional natural language systems. It helps to determine whether or not an anaphoric reference is possible. In the example presented above, the metagraphical arrow generated by WIP's annotation component is the equivalent of a pronoun since it focuses attention on a specific part of the visual antecedent. Mixed anaphoric reference generation is also

supported by WIP's architecture. In a sequence like (2), the antecedent of the anaphora "the on/off switch" is a visual object stored in the document design plan and focused by the cross-modal reference in the sentence preceding the anaphora.

Fig. 3 provides a survey. The on/off switch ... (2)

Note that the final result shown in the lower left pane was produced incrementally. The incrementality of the overall generation process that was initiated by expanding the presentation goal

(DESCRIBE P A (SEQUENCE (PUT-1 TURN-1)) T)

(see Fig. 8) is illustrated in the TAG Results pane. The generation component first verbalizes the PUT-1 request and forwards the label "Water outlet" to the graphics designer. The second request is then verbalized and the corresponding label "On/off switch" is produced and inserted in the previously generated picture. In addition to this incrementality across generation components, there is another level of incrementality in the individual generation components.

9. Coordinating text and graphics generation

In a multimodal presentation, cross-modal expressions establish referential relationships of representations in one modality to representations in another modality. The use of cross-modal deictic assertions such as (3) is essential for the efficient coordination of text and graphics in illustrated documents (see Fig. 9).

The on/off switch is located in the upper left part of the picture. (3)

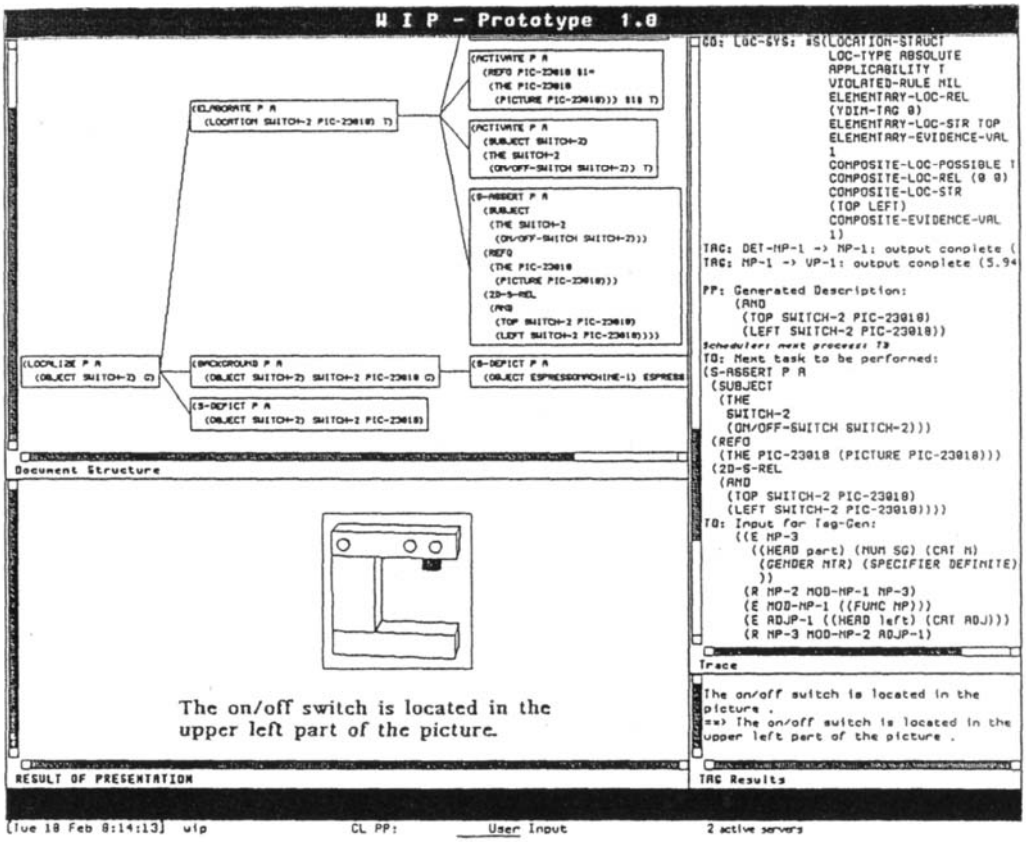

Fig. 9 Incremental generation fo a cross-modal reference.

Given the presentation goal

(BMP P A (LOCATION SWITCH-2 (? LOCATION))),

the presentation planner designs the text-picture combination in the bottom left pane of Fig. 9 communicating the relevant information about the spatial position of the on/off switch.
In this example, WIP uses a spatial description to refer to an object shown in a synthetic picture of the espresso machine. Note that the multimodal referential act can only be successful if the addressee is able to identify the intended knob of the real espresso machine. It is clear that the depiction of the switch cannot be used as an on/off switch, but only the physical object identified as the result of a multi-level reference resolution (see Fig. 10). The cross-modal assertion in the text refers to a pictorial element that visualizes an instance of a concept represented by a RAT term as part of WIP's application knowledge. An additional coreferentiality relation exists between the individual constant SWITCH-2 in the ABox of RAT and an object in the wireframe model of the machine providing a description of the geometry of that knob. Finally, the depiction of the knob generated by WIP's graphics design component in turn refers to the corresponding switch of the real machine.

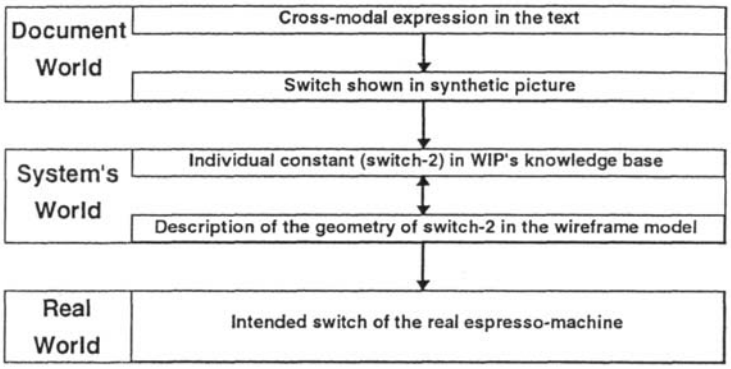

Fig. 10. WIP's multi-level reference process.

The generation of cross-modal expressions highlights the tight interaction between various components of WIP and the cross dependencies among decisions of the mode-specific generators. In our example the text design component, that is activated by the presentation planner after a first draft of the picture has been completed by the graphics designer, calls the graphics component once again to ask for a localization of a pictorial element.

The top left pane in Fig. 9, labelled "Document Structure", shows a fragment of the DAG produced by the presentation planner. Note that the LOCALIZE act is decomposed into three acts. The main act specifies the task for the graphics designer to depict SWITCH-2 in a picture. One subsidiary act tries to provide background information for the generated depiction by showing other salient parts of the machine as the visual context of the switch. The other subsidiary act is supposed to generate text that elaborates on the picture. Further refinements using presentation strategies for textual elaboration finally lead to the cross-modal expression discussed above. Although the mode flag is set to TEXT for this elaboration (coded as T in the corresponding node of the presentation plan, see Fig. 9), the evaluation facilities of the graphics generator are used to compute a spatial relation describing the absolute localization of the switch in the picture.

The most important steps in the design process leading to the cross-modal assertion (3) are shown in the top right pane of Fig. 9 that displays a partial trace of the interaction between the major components of the presentation system. After the presentation planner (PP in the trace) has established a new node in the DAG that contains an unbound variable representing a description of the location of the switch in the picture, the graphics designer (GD in the trace) calls its localization component to determine the value of that variable.

One of the basic ideas behind this component is that absolute localizations like "in the upper left part of the picture" can be derived from relative spatial predicates like LEFT-OF(X,Y) and ON-TOP-OF(X,Y) through the use of virtual reference objects induced by the page layout. This means that

objects depicted in a figure can be spatially related to the center, the corners, the borderline and even to the caption of that figure.

In the example shown, the rectangular picture region, in which the image of the espresso machine is displayed by the graphics component, is used as a frame of reference for the spatial description encoding the position of SWITCH-2's depiction (see the bottom left pane of Fig. 9). The relative location of the on/off switch is described by the conjunction of the literals LEFT-OF(SWITCH-2, CENTER(PIC-23018)) and ON-TOP-OF(SWITCH-2,CENTER(PIC-23018)) that use the center of the figure as a reference object. In WIP, the center of a picture is approximated by a virtual rectangle in the middle with one third of the horizontal and vertical extension of the whole figure (for more details see [67]).

These relative localizations are then transformed into absolute ones through deleting the second argument. The presentation planner forwards the result of the localization process to the text design (TD) component for lexical choice (see top left pane of Fig. 9).

The generation of cross-modal expressions can involve various levels of recursion. One subtlety not illustrated by the example above is the use of different frames of reference for spatial relations in a single cross-modal expression. Suppose that in addition to the picture discussed in the previous example, another figure is placed on the same page. Then the generic localization methods of WIP will generate another relative description like RIGHT-OF(PIC-23018,CENTER(PAGE-1)) leading to a recursive spatial reference such as "in the upper left part of the figure on the right".

Since the layout constraints specified in WIP's input together with revisions of the presentation planner force the layout manager to backtrack from time to time during the incremental design of a multimodal presentation, it may turn out that a figure has to be repositioned and thus parts of the cross-modal expression have to be revised. For example, "the figure on the right" may become "the figure on the top".

Another level of recursion in the localization process is introduced by dealing with groups of objects. In this case, a group can serve at the same time as a frame of reference for one of its elements and as a perceptual unit that itself must be localized using other reference objects in the figure (cf. [64]). For example, the generation of a localization for the group of two switches on the right part of the machine in Fig. 9 leads to a cross-modal expression like "The left button on the right part of the picture is the selector switch" (see [67] for further details).

As illustrated by this example such verbal descriptions can get quite long-winded. Therefore WIP's presentation strategies include alternate methods to establish cross-modal referential relations. As mentioned in Section 7.1, the graphics generator supports various labeling techniques for placing text strings in a figure so that they annotate the parts of a composite object in an illustration. The generation of labels as a part of the graphics design is an example where in comparison to the previous discussions concerning the localization component, the dependency between graphics generation and text generation is reversed. In this case the text generator is activated during the graphics design process in order to produce a string that can be used for labeling a picture element. Note that one has to ensure that the same description is used for referring to the object in the text, as it would lead to an incoherent text-picture combination, if a switch that is labelled "on/off switch" in a picture is referred to as "starting switch" in the corresponding text. This means that for the generation of multi-modal presentations the document design plan plays the same role as the discourse model for verbal communication, namely allowing the presentation planner to ensure the consistent use of referential expressions across modes.

Suppose that in our example, the text generator is asked to find a lexical realization for the concept EM-SELECTOR-SWITCH and comes up with the description "selector switch for coffee and steam". When trying to annotate the switch with this text string, the graphics generator finds out that none of the available annotation techniques apply. Placing the string close to the corresponding depiction causes ambiguities. The string also cannot be placed onto the projection of the object without occluding other parts of the picture. For the same reason, annotations with arrows fail. Therefore, the text generator is asked to produce a shorter formulation. Unfortunately, it is not able to do so without reducing the contents. Thus, the presentation planner is informed that the required task cannot be accomplished. The presentation planner then tries to reduce the contents by omitting attributes or by selecting more general concepts from the subsumption hierarchy encoded in terms of the terminological logic. Given that EM-SELECTOR-SWITCH is a compound description which inherits information from the concepts SWITCH and EM-SELECTOR, the planner has to decide which

component of the contents specification should be reduced. As the concept SWITCH contains less discriminating information than the concept EM-SELECTOR and the concept SWITCH is at least partially inferable from the picture, the planner first tries to reduce the component SWITCH by replacing it by PHYSICAL-OBJECT. Thus, the text generator has to find a sufficiently short definite description containing the components PHYSICAL-OBJECT and EM-SELECTOR. Since this fails, the planner has to propose another reduction. It now tries to reduce the component EM-SELECTOR by omitting the coffee/steam mode. The text generator then tries to construct a NP combining the concepts SWITCH and SELECTOR. This time it succeeds and the annotation string can be put into place.

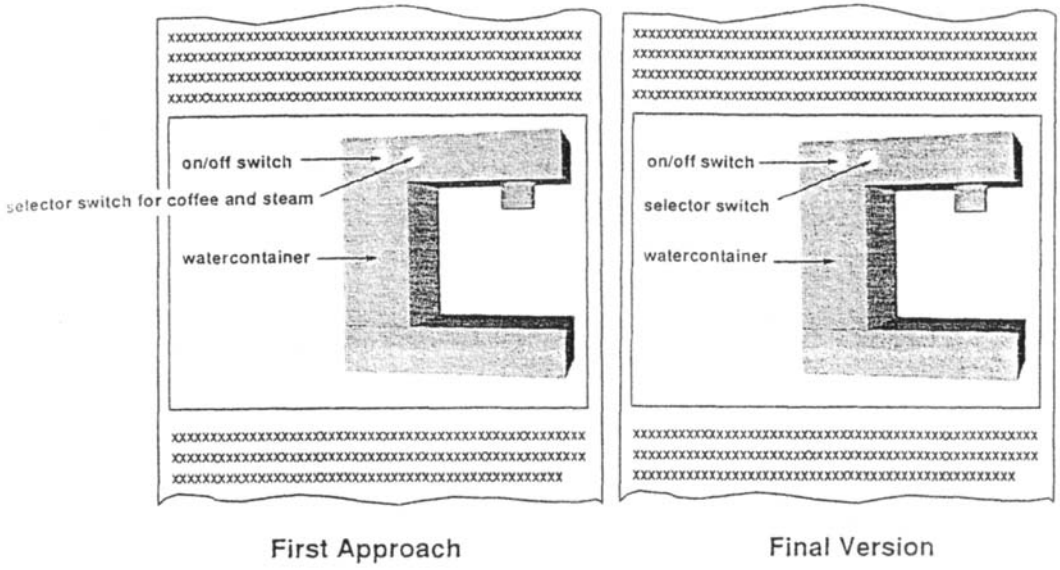

Fig. 11. Annotating a rendered picture.

Figure 11 is a hardcopy produced by WIP showing the rendered espresso machine after the required annotations have been carried out.

No serial architecture with a total ordering of the components for text and graphics generation would be adequate in this case. On the one hand, the text strings have to be produced by the TAG generator before they are put into place by the graphics generator. On the other hand, graphical knowledge is necessary to determine how long a text string may be. Since determining the maximal admissible length of a text string is no local decision, but depends, among others, on the position of other picture elements, the processes for text and graphics generation cannot be sequentialized.

10. Future research

It is obvious that the current WIP system has serious shortcomings with respect to the interactive aspects of multimodal presentations. In our future research, much more attention will be placed on the following problems:

Interactive multimodal presentations

WIP's most significant current limitation is that it does not support user interaction during the multimodal presentation. An interactive user may want to interrupt the presentation before it is completed for one of the following reasons:

- he is dissatisfied with the current style of presentation,
- he has a question about the presentation generated so far.

Since WIP's output is generated incrementally, much of the machinery is already in place to accommodate such interruptions. However, the presentation planner has to be extended so that it

allows for the necessary reactive planning. Clearly, the next step is to allow the user to change the generation parameters during the presentation, e.g., by demanding the system to change the level of detail or the speed of the current presentation. Probably the greatest opportunity lies in the generalization of methods, which generate cooperative responses to follow-up questions in natural language dialog systems, to the broader domain of multimodal communication.

Planning multimodal presentation acts

Another important deficiency of the current WIP system is that it merely generates coordinated language and graphics according to a particular presentation goal, rather than planning when and how to present this material to a particular user. We expect more efficient presentations from an augmented version of WIP, in which an animated character called PPP (Personalized Plan-based Presenter) will play the role of a presenter, showing, commenting and explaining the generated material. This means that the system should be able to plan presentations as well as presentation acts and their temporal coordination. For example, PPP could point to a particular section of an illustrated explanation and, at the same time, produce an utterance highlighting the importance of a particular instruction step.

Monitoring the effectiveness of a presentation

A further limitation of the current version of WIP is that it has no means to check whether the user really has understood the presentation and has followed the instructions correctly. In a follow-up project to WIP, we plan to provide the presentation system with an indirect feedback on the user's physical behavior after he has received the instructions, by evaluating the state changes caused by his actions. A simple method to obtain such a feedback, without relying on a sophisticated vision system, is to use a data bus to physically connect the technical device, which is to be serviced by the user, with the presentation system. The presentation system could, based on such a connection, keep track of the relevant behavior of the user, monitor the effectiveness of the presentation and continuously adapt its presentations to the current situation. Our main interest here is the close integration of presentation planning and plan monitoring, in order to improve the effectiveness of the generated multimodal presentations.

11. Conclusions

The central claim of this paper is that the generation of a multimodal presentation can be considered as an incremental planning process that aims to achieve a given communicative goal.

We have shown how techniques for planning text and discourse can be generalized to allow the structure and content of multimodal communications to be planned as well. When explaining how a complex process functions, WIP generates and realizes plans for communicating domain plans provided by the back-end system. While the root of the hierarchical plan structure for a particular presentation corresponds to a complex communicative act such as describing a process, the leaves are elementary acts that verbalize and visualize the physical acts specified in a given domain plan.

A key observation is that it is possible to use a slightly extended version of RST to describe important semantic and pragmatic coherence relations not only between text fragments, but also picture elements, pictures, and text or sequences of text-picture combinations. We have explored the question of how the presentation planner can decide what should go into text, what should go into graphics, and how to link verbal and non-verbal fragments by cross-modal references. We have formalized the knowledge needed for the planning of coordinated multimedia presentations, thereby introducing new concepts like presentation strategies, design strategies, and meta-rules for mode selection.

Since one of the design principles behind WIP is that the theoretical basis of all components should be sound enough to allow scale-up, we have combined and extended only formalisms that have reached a certain level of maturity, in particular terminological logics, RST-based planning, constraint processing techniques, and tree adjoining grammars with feature unification.

One of the surprises from our research is that it is actually possible to extend and adapt many of the fundamental concepts developed to date in AI and computational linguistics for the generation of natural language in such a way that they become useful for the generation of graphics and text-picture combinations as well. In particular, we have shown that well-known concepts from the area of natural

language processing like speech acts, anaphora, and rhetorical relations take on an extended meaning in the context of multimodal communication.

The experience we gained from the design and implementation of the WIP prototype provides a good starting point for a deeper understanding of the interdependencies of language and graphics in coordinated multimodal communication.

Acknowledgements

The WIP project is supported by the German Ministry of Research and Technology under grant ITW8901 8. The development of WIP has been a group effort and has benefited from the contributions of our collaborators Winfried Graf, Karin Harbusch, Jochen Heinsohn, Anne Kilger, and Bernhard Nebel as well as our students Jochen Bedersdorfer, Andreas Butz, Bernd Herrmann, Antonio Krüger, Daniel Kudenko, Peter Poller, Thomas Schiffmann, Georg Schneider, Frank Schneiderlochner, Christoph Schom-mer, Dudung Soetopo, Martin Weiler, and Detlev Zimmermann. We would like to thank the anonymous referees for helpful comments on an earlier draft of this paper.

References

[1] E. André, G. Bosch, G. Herzog and T. Rist, Characterizing trajectories of moving objects using natural language path descriptions, in: *Proceedings Seventh European Conference on Artificial Intelligence, Vol. 2*, Brighton, England (1986) 1-8.

[2] E. André, W. Finkler, W. Graf, T. Rist, A. Schauder and W. Wahlster, WIP: the automatic synthesis of multimodal presentations, in: M. Maybury, ed., *Intelligent Multimedia Interfaces* (AAAI Press, Cambridge, MA, 1993).

[3] E. André and T. Rist, Synthesizing illustrated documents: a plan-based approach, in: *Proceedings InfoJapan*, Tokyo, Japan (1990) 163-170.

[4] E. André and T. Rist, Towards a plan-based synthesis of illustrated documents, in: *Proceedings Ninth European Conference on Artificial Intelligence*, Stockholm, Sweden (1990) 25-30.

[5] E. André and T. Rist, The design of illustrated documents as a planning task, in: M. Maybury, ed., *Intelligent Multimedia Interfaces* (AAAI Press, Cambridge, MA, 1993).

[6] Y. Arens, S.K. Feiner, J. Hollan and B. Neches, A new generation of intelligent interfaces, Workshop IJCAI-89, Detroit, MI (1989).

[7] Y. Arens, E.H. Hovy and M. Vossers, The knowledge underlying multimedia presentations, in: M. Maybury, ed., *Intelligent Multimedia Interfaces* (AAAI Press, Cambridge, MA, 1993).

[8] F. Baader, H.-J. Burckert, J. Heinsohn, B. Hollunder, J. Müller, B. Nebel, W. Nutt and H.-J. Profitlich, Terminological knowledge representation: a proposal for a terminological logic, Tech. Memo TM-90-04 DFKI Saarbrücken, Germany (1990).

[9] F. Baader and B. Hollunder, KRIS: Knowledge representation and inference system, *SIGART Bull.* 2 (3) (1991) 8-14.

[10] N. Badler, B. Webber, J. Kalita and J. Esakov, Animation from instructions, in: N. Badler, B. Barsky and D. Zeltzer, eds., *Making Them Move: Mechanics, Control and Animation of Articulated Figures* (Morgan Kaufmann, San Mateo, CA, 1991) 51-93.

[11] S. Bandyopadhyay, Towards an understanding of coherence in multimodal discourse, Tech. Memo TM-90-01, DFKI, Saarbrücken, Germany (1990).

[12] D. Chapman, Planning for conjunctive goals, *Artif. Intell.* 32 (3) (1987) 333-377.

[13] P.R. Cohen, J.W. Sullivan, M. Dalyrymple, R.A. Gargan, D.B. Moran, J.O. Schlossberg, F.C.N. Perreira and S.W. Tyler, Synergistic use of direct manipulation and natural language, in: *Proceedings CHI-89*, Austin, TX (1989) 227-233.

[14] R. Dale, Visible language: multimodal constraints in information presentation, in: R. Dale, E.H. Hovy, D. Rosner and O. Stock, eds., *Aspects of Automated Natural Language Generation*, Lecture Notes in Artificial Intelligence 587 (Springer, New York, 1992) 281-283; also in: *Proceedings Sixth International Workshop on Natural Language Generation*, Trento, Italy (1992).

[15] K. De Smedt and G. Kempen, Incremental sentence production, self-correction and coordination, in: G. Kempen, ed., *Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics*, NATO ASI Series E 135 (Martinus Nijhoff, Dordrecht, Netherlands, 1987) 365-376 (Chapter 23).

[16] P.T. Devanbu and D.J. Litman, Plan-based terminological reasoning, in: J.F. Allen, R.E. Fikes and E. Sandewall, eds., *Proceedings Second International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, MA (1991) 128-138.

[17] S.K. Feiner, D.J. Litman, K.R. McKeown and R.J. Passonneau, Towards coordinated temporal multimedia presentations, in: M. Maybury, ed., *Intelligent Multimedia Interfaces* (AAAI Press, Cambridge, MA, 1993).

[18] S.K. Feiner and K.R. McKeown, Coordinating text and graphics in explanation generation, in: *Proceedings AAAI-90*, Boston, MA (1990) 442-449.

[19] S.K. Feiner and K.R. McKeown, Automating the generation of coordinated multimedia explanations, *IEEE Computer* 24 (10) (1991) 33-41.

[20] W. Finkler, Incremental natural language generation with TAGs in the WIP-project, in: W. Wahlster and K. Harbusch, eds., *First International Workshop on Tree Adjoining Grammars*, Dagstuhl, Germany (1990) 64-70.

[21] W. Finkler and G. Neumann, POPEL-HOW - a distributed parallel model for incremental natural language production with feedback, in: *Proceedings IJCAI-89*, Detroit, MI (1989) 1518-1523.

[22] W. Finkler and A. Schauder, Effects of incremental output on incremental natural language generation, in: B. Neumann, ed., *Proceedings Tenth European Conference on Artificial Intelligence*, Vienna, Austria (1992) 505-507.

[23] W.Graf, Constrained-based graphical layout of multimodal presentations, in: *Proceedings Advanced Visual Interfaces (AVI)* Workshop, Rome, Italy (1992).

[24] J.E. Grimes, *The Thread of Discourse* (Mouton/de Gruyter, The Hague, Netherlands, 1975).

[25] K. Harbusch, W. Finkler and A. Schauder, Incremental syntax generation with tree adjoining grammars, in: W. Brauer and D. Hernandez, eds., *Proceedings Fourth International GI Congress on Knowledge-Based Systems*, Munich, Germany (1991) 363-374.

[26] J. Heinsohn, D. Kudenko, B. Nebel and H.-J. Profitlich, RAT - representation of actions using terminological logics, Research Report, DFKI, Saarbrücken, Germany (1992).

[27] G. Herzog, C.-K. Sung, E. André, W. Enkelmann, H.-H. Nagel, T. Rist, W. Wahlster and G. Zimmermann, Incremental natural language description of dynamic imagery, in: W. Brauer and C. Freksa, eds., *Proceedings Third International GI Congress*, Munich, Germany (1989) 153-162.

[28] J.R. Hobbs, Why is a discourse coherent?, Tech. Report 176, SRI, Menlo Park, CA (1978).

[29] E.H. Hovy, *Generating Natural Language under Pragmatic Constraints* (Lawrence Erlbaum, Hillsdale, NJ, 1988).

[30] E.H. Hovy and Y. Arens, Automatic generation of formatted text, in: *Proceedings AAAI-9I*, Anaheim, CA (1991) 92-94.

[31] B. Hunter, A. Crismore and P.D. Pearson, Visual displays in basal readers and social studies textbooks, in: D.M. Willows and H.A. Houghton, eds., *The Psychology of Illustration 2, Basic Research* (Springer, New York, 1987) 116-135.

[32] A.K. Joshi, How much context-sensitivity is necessary for characterization structural descriptions - tree adjoining grammar, in: D. Dowty, L. Karttunen and A. Zwicky, eds., *Natural Language Processing - Theoretical, Computational and Psychological Perspective* (Cambridge University Press, New York, 1985).

[33] G. Kempen and E. Hoenkamp, An incremental procedural grammar for sentence formulation, Cogn. Sci. 2 (11) (1987) 201-258.

[34] S. Kerpedjiev, Automatic generation of multimodal weather reports from datasets, in: *Proceedings Third Conference on Applied Natural Language Processing (ANLP-92)*, Trento, Italy (1992) 48-55.

[35] S. Kjorup, Pictorial speech acts, *Erkenntnis* 12 (1978) 55-71.

[36] A. Kobsa, J. Allgayer, C. Reddig, N. Reithinger, D. Schmauks, K. Harbusch and W. Wahlster, Combining deictic gestures and natural language for referent identification, in: *Proceedings Eleventh COLING*, Bonn, Germany (1986) 356-361.

[37] W.J.M. Levelt, *Speaking: From Intention to Articulation* (MIT Press, Cambridge, MA, 1989).
[38] J.R. Levin, G.J. Anglin and R.N. Carney, On empirically validating functions of pictures in prose, in: D.M. Willows and H.A. Houghton, eds., *The Psychology of Illustration* 1 (Springer, New York, 1987) 51-85.
[39] W.C. Mann and S.A. Thompson, Rhetorical structure theory: description and construction of text structures, in: G. Kempen, ed., *Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics*, NATO ASI Series E 135 (Martinus Nijhoff, Dordrecht, Netherlands, 1987) 85-95.
[40] M. Maybury, ed., *Intelligent Multimedia Interfaces* (AAAI Press, Cambridge, MA, 1993); Workshop Notes from AAAI-91, Anaheim, CA (1991).
[41] K.R. McKeown and S.K. Feiner, Interactive multimedia explanation for equipment maintenance and repair, in: *Proceedings DARPA Speech and Language Workshop* (1990) 42-47.
[42] J.D. Moore and C.L. Paris, Planning text for advisory dialogues, in: *Proceedings 27th Annual Meeting of the Association for Computational Linguistics*, Vancouver, BC (1989).
[43] J.G. Neal and S.C. Shapiro, Intelligent multi-media interface technology, in: J.W. Sullivan and S.W. Tyler, eds., *Intelligent User Interfaces* (Addison-Wesley, Reading, MA, 1991) 11-43.
[44] G. Neumann and W. Finkler, A head-driven approach to incremental and parallel generation of syntactic structures, in: *Proceedings Thirteenth COLING*, Helsinki, Finland (1990) 288-293.
[45] C.L. Paris, Generation and explanation: building an explanation facility for the explainable expert systems framework, in: C.L. Paris, W.R. Swartout and W.C. Mann, eds., *Natural Language Generation in Artificial Intelligence and Computational Linguistics* (Kluwer, Boston, MA, 1991) 49-82.
[46] P.F. Patel-Schneider, B. Owsnicki-Klewe, A. Kobsa, N. Guarino, R. MacGregor, W.S. Mark, D. McGuinness, B. Nebel, A. Schmiedel and J. Yen, Term subsumption languages in knowledge representation, *AI Mag.* 11 (2) (1990) 16-23.
[47] E. Reiter, C. Mellish and J. Levine, Automatic generation of on-line documentation in the IDAS project, in: *Proceedings Third Conference on Applied Natural Language Processing (ANLP-92)*, Trento, Italy (1992) 64-71.
[48] N. Reithinger, A parallel and incremental natural language generation system, in: C.L. Paris, W.R. Swartout and W.C. Mann, eds., *Natural Language Generation in Artificial Intelligence and Computational Linguistics* (Kluwer, Boston, MA, 1991) 179-200.
[49] T. Rist and E. André, Wissensbasierte Perspektivenwahl fur die automatische Erzeugung von 3D-Objektdarstellungen, in: K. Kansy and P. WiBkirchen, eds., *Graphik und KI*, Informatik Fachberichte 239 (Springer, Berlin, 1991) 48-57.
[50] T. Rist and E. André, From presentation tasks to pictures: towards an approach to automatic graphics design, in: B. Neumann, ed., *Proceedings Tenth European Conference on Artificial Intelligence*, Vienna, Austria (1992) 764-768.
[51] T. Rist and E. André, Incorporating graphics design and realization into the multimodal presentation system WIP, in: *Proceedings Advanced Visual Interfaces (AVI) Workshop*, Rome, Italy (1992).
[52] S. Roth, J. Mattis and X. Mesnard, Graphics and natural language as components of automatic explanation, in: J.W. Sullivan and S.W. Tyler, eds., *Intelligent User Interfaces* (Addison-Wesley, Reading, MA, 1991) 207-239.
[53] Y. Schabes, A. Abeille and A.K. Joshi, Parsing strategies with lexicalized grammars: application to tree adjoining grammar, in: *Proceedings Twelfth COLING*, Budapest, Hungary (1988).
[54] A. Schauder, Inkrementelle syntaktische Generierung natürlicher Sprache mit Tree Adjoining Grammars, Master's Thesis, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, Germany (1990).
[55] A. Schauder, Incremental syntactic generation of natural language with tree adjoining grammars, Document D-92-21, DFKI, Saarbrücken, Germany (1992).
[56] J. Schirra, A contribution to reference semantics of spatial prepositions: the visualization problem and its solution in VITRA, in: C. Zelinsky-Wibbelt, ed., *The Semantics of Prepositions - From Mental Processing to Natural Language Processing* (Mouton/de Gruyter, Berlin, 1992).
[57] J.R. Searle, Speech Acts: *An Essay in the Philosophy of Language* (Cambridge University Press, Cambridge, England, 1969).

[58] D.D. Seligmann and S.K. Feiner, Automated generation of intent-based 3D illustrations, *Comput. Graph.* 25 (4) (1991) 123-132.
[59] O. Stock, Natural language and exploration of an information space: the AlFresco interactive system, in: *Proceedings IJCAI-91*, Sydney, Australia (1991) 972-978.
[60] J.W. Sullivan and S.W. Tyler, eds., *Intelligent User Interfaces* (Addison-Wesley, Reading, MA, 1991).
[61] T.A. van Dijk, *Textwissenschaft* (DTV, Munich, Germany, 1980).
[62] W. Wahlster, User and discourse models for multimodal communication, in: J.W. Sullivan and S.W. Tyler, eds., *Intelligent User Interfaces* (Addison-Wesley, Reading, MA, 1991) 45-67.
[63] W. Wahlster, E. André, S. Bandyopadhyay, W. Graf and T. Rist, WIP: the coordinated generation of multimodal presentations from a common representation, in: A. Ortony, J. Slack and O. Stock, eds., *Communication from an Artificial Intelligence Perspective. Theoretical and Applied Issues* (Springer, Heidelberg, 1992) 121-144.
[64] W. Wahlster, A. Jameson and W. Hoeppner, Glancing, referring and explaining in the dialogue system HAM-RPM, *Amer. J. Comput. Ling.*, Microfiche 77 (1978) 53-67.
[65] W. Wahlster, H. Marburger, A. Jameson and S. Busemann, Over-answering yes-no questions: extended responses in a NL interface to a vision system, in: *Proceedings IJCAI-83*, Karlsruhe, Germany (1983) 643-646.
[66] N. Ward, A flexible, parallel model of natural language generation, Ph.D. Thesis, Report No. UCB/CSD 91/629, Computer Science Division (EECS), University of California, Berkeley, CA, (1991)
[67] P. Wazinski, Generating spatial descriptions for cross-modal references, in: *Third Conference on Applied Natural Language Processing {ANLP-92)*, Trento, Italy (1992) 56-63.
[68] R. Weida and D.J. Litman, Terminological reasoning with constraint networks and an application to plan recognition, in: B. Nebel, W. Swartout and C. Rich, eds., *Principles of Knowledge Representation and Reasoning: Proceedings Third International Conference* (Morgan Kaufmann, San Mateo, CA, 1992) 282-293.
[69] D. Zimmermann, Anna: Ein wissensbasiertes System zur automatischen Annotation von Graphiken, Document, DFKI, Saarbrücken, Germany (1993).