Coordinating Text and Graphics in Explanation Generation*

Steven K. Feiner
Kathleen R. McKeown

Department of Computer Science
Columbia University
New York, New York 10027
Feiner@cs.columbia.edu
Mckeown@cs.columbia.edu

Abstract

To generate multimedia explanations, a system must be able to coordinate the use of different media in a single explanation. In this paper, we present the architecture that we have developed for COMET (*COordinated Multimedia Explanation Testbed*), a system that generates directions for equipment maintenance and repair, and we show how it addresses the coordination problem. COMET includes a single content planner that produces a common content description used by multiple media-specific generators, and a media coordinator that performs a fine-grained division of information among media. Bidirectional interaction between media-specific generators allows influence across media. We describe COMET's current capabilities and provide an overview of our plans for extending the system.

Introduction

A number of researchers have begun to explore the automated generation of multimedia explanations [Arens, Miller, and Sondheimer 88, Feiner 88, Neal and Shapiro 88, Roth, Mattis, and Mesnard 88, Wahlster et al. 89]. One common concern is how to coordinate the use of different media in a single explanation. How are the communicative goals that the explanation is to satisfy and the information needed to achieve those goals to be determined? How is explanation content to be divided among different media, such as pictures and text? Once divided, how can material in each medium be generated to complement that of the other media? In this paper, we describe an architecture for generating multimedia explanations that we have developed for COMET (*COordinated Multimedia Explanation Testbed*), a system that generates directions for equipment maintenance and repair. We use sample explanations produced by COMET to illustrate how its architecture provides some answers to these questions.

COMET's architecture includes a single content planner, a media coordinator, bidirectional links between the text and graphics generators, and a media layout component. The content planner determines communicative goals and information for an explanation in a media-independent fashion, producing explanation content in a common description language used by each media-specific generator [Elhadad et al. 89]. Using the same description language allows for more flexible interaction among media, making it possible for each generator to query and reference other generators. The media coordinator annotates the content description, noting which pieces should be conveyed through which media. Our coordinator is unique in its ability to make a fine-grained division between media. For example, COMET may generate a sentence accompanied by a picture that portrays just the modifiers of one of the sentence's referents, such as its location. The annotated content description will allow our media layout component to lay out text and pictures appropriately.

Bidirectional interaction between the media-specific generators makes it possible to address issues in how media can influence each other. For example, informal experiments that we performed when designing our current media coordinator showed that people strongly prefer sentence breaks that are correlated with picture breaks. This influence requires bidirectional interaction, since graphical constraints on picture size may sometimes force delimitation of sentences, while grammatical constraints on sentence construction may sometimes control picture size. Other influences that we are currently investigating include reference to pictures based on characteristics determined dynamically by the graphics generator (e.g., ''the highlighted dial'' vs. ''the red dial'') and coordination of style (e.g., whether the graphics generator designs a composite picture or sequence of pictures to represent a process can influence whether the text generator uses past or progressive tense).

In the following sections, we provide a system overview of COMET, discuss the production of explanation content in the common description language, describe our media coordinator, and preview our ongoing work on allowing the media to influence each other.

---

*This work is supported in part by the Defense Advanced Research Projects Agency under Contract N00039-84-C-0165, the Hewlett-Packard Company under its AI University Grants Program, the Office of Naval Research under Contract N00014-82-K-0256, the National Science Foundation under Grant IRT-84-51438, and the New York State Center for Advanced Technology under Contract NYSSTF-CAT(88)-5.

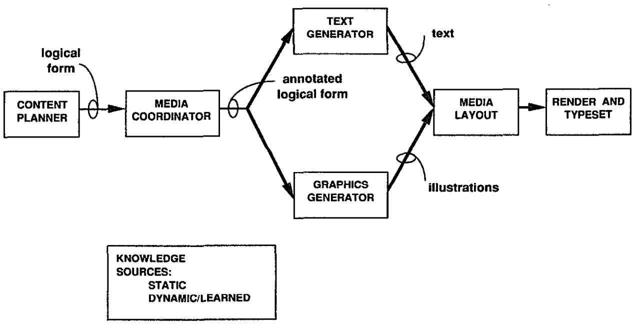


Figure 1: COMET system architecture.

System Organization and Domain

COMET, shown below in Fig. 1, features a single *content planner* which uses text plans, or *schemas*, to determine which information from the underlying *static knowledge sources*, encoded in LOOM [MacGregor & Brill 89], should be included in the explanation. The content planner produces the full content for the explanation, represented as a hierarchy of logical forms (LFs) [Allen 87], which are passed to the *media coordinator*. The media coordinator annotates the LFs to indicate which portions are to be produced in each medium.

COMET currently includes text and graphics generators. The *text generator* and *graphics generator* each process the same LFs, producing fragments of text and graphics that are keyed to the LFs they instantiate. Text and graphics fragments are passed to the *media layout* component, which lays them out on the screen.

Much of our work on COMET has been done in a maintenance and repair domain for the US Army AN/PRC-119 portable radio receiver-transmitter [DOA 86]. An underlying expert system determines which problems the radio is experiencing, which components are suspect, and which tests would be most useful in identifying the causes. The generation facilities create multimedia explanations of how to test and fix the radio. A user interacts with COMET using a menu interface through which the expert system can be invoked and requests for explanations made.

Figure 2, which we will refer to later, shows a simple example of the text and 3D graphics that COMET generates to describe how to clear the radio's display.

A Common Content Description for Multiple Media Generators

In COMET, explanation content is produced by a single content planner that does not take into account which media will be used for presentation. The content planner outputs a hierarchy of LFs that represent the content for the entire explanation. Content is later divided among the media by annotating the LFs. As a result, the system maintains a single description of the content to be generated, which is annotated and accepted as input by both the text generator (FUF [Elhadad 88]) and the graphics generator (IBIS [Seligmann and Feiner 89]). Thus, both FUF and IBIS share a common description of what is to be communicated. Just as both generators accept input in the same formalism, they may both annotate the description as they carry out its directives. This design

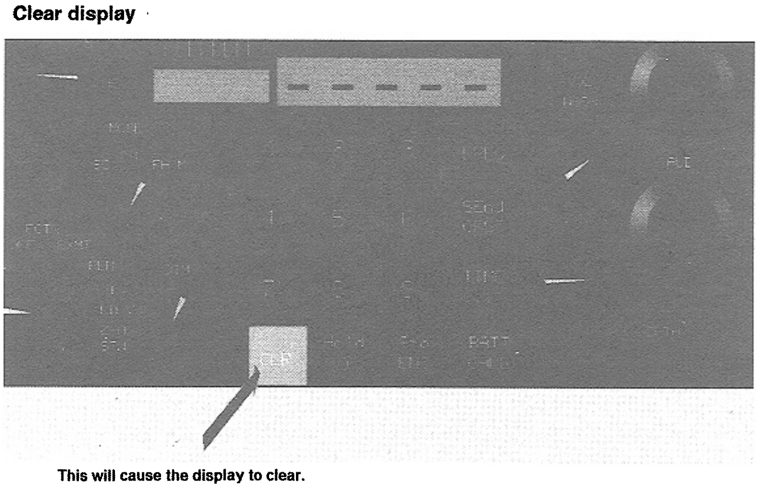


Figure 2: Text and graphics produced by COMET to explain how to clear the AN/PRC-119 display.

has several ramifications for the system:

Single content planner

COMET contains only one component dedicated to determining the communicative goals and subgoals needed to produce an explanation. COMET's content planner uses a schema-based approach that was originally used for text generation [McKeown 85, Paris 87], and which has proved successful for multimedia explanations as well. Keeping the content planner media-independent means that it only has to determine what information must be communicated to the user, without worrying about how. If it did select information with a specific medium in mind, it would have to carry out the media coordinator's task simultaneously.

Separation of goals from resources

The specification of content must be made at a high enough level that it is appropriate as input for all generators. We have found that by expressing content as communicative goals and information needed to achieve those goals, each generator can select the resources it has at hand for achieving the goals. In text, this means the selection of specific syntactic or lexical resources (e.g., passive voice to indicate focus), whereas in graphics, it means the selection of a conjunction of visual resources (e.g., to highlight an object, IBIS may change its color, outline it, and center it).

Text and graphics can influence each other

Since both FUF and IBIS receive the same annotated content description as input, they know which goals are to be expressed in text, which in graphics, and which in both. Even when a media-specific generator does not realize a piece of information, it knows that information is to be conveyed to the user and thus, it can use this information to influence its presentation.

Text and graphics generators can communicate with each other

Since both generators understand the same formalism, they can decide to provide more information to each other about the resources they have selected to achieve a goal, simply by annotating the content description. For example, if IBIS has decided to highlight a knob by changing its color to red, it might note that decision in the description, and FUF could ultimately generate the reference "the red knob", instead of "the highlighted knob". Communication requires bidirectional interaction and is discussed further below.

Single mechanism for adding annotations

Since different system tasks (e.g., dividing information between text and graphics, and communication between text and graphics generators) are achieved by adding annotations, the same mechanism can be used to make the annotations throughout the system. COMET uses FUF for this task. This simplifies the system and provides more possibilities for bidirectional interactions between components, as discussed below.

```
((cat lf)
 (directive-act substeps)
 (substeps
  [((process-type action)
    (process-concept c-push)
    (mood non-finite)
    (speech-act directive)
    (roles
     ((medium
       ((object-concept c-button-clr)
        (roles
         ((location ((object-concept c-location)))
          (size     ((object-concept c-size)))))
        (quantification ((definite yes)
                         (countable yes)
                         (ref-obj singular)
                         (ref-set singular)))
        (ref-mode description))))))])
 (effects
  [((process-type action)
    (process-concept c-clear)
    (mood non-finite)
    (speech-act assertive)
    (roles
     ((agent
       ((object-concept c-display)
        (roles
         ((location ((object-concept c-location)))
          (size     ((object-concept c-size)))))
        (quantification ((definite yes)
                         (countable yes)
                         (ref-obj singular)
                         (ref-set singular)))
        (ref-mode description)))))))]))
```

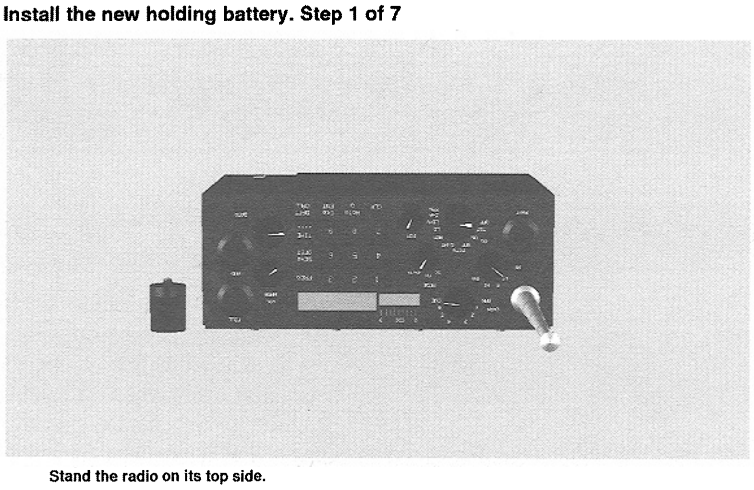
Figure 3: Content planner output (LF 1): Press the CLR button. This will cause the display to clear.

Examples

Clearing the display

To see how these points relate to COMET, consider how it generates the response shown in Fig. 2. The content planner selects one of its schemas, the *process schema* [Paris 87], and produces content by traversing the schema, which is represented as a graph, producing an LF (or piece of an LF) for each arc it takes. For this simple example, it produces one simple LF.

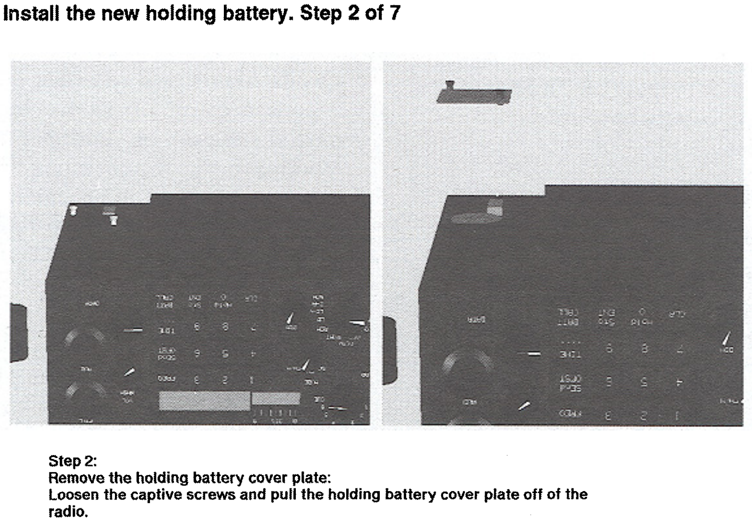
Figure 3 shows the LF produced by the content planner for this example. It contains several communicative goals. The main goal is to describe an action (`c-push`) and its role (`medium`). Subgoals include referencing an object (e.g., `c-button-clr`, the clear button) and conveying its `location`, `size`, and `quantification`. IBIS and FUF use different resources to achieve these goals. For example, FUF selects a lexical item, the verb "press", to describe the action. "Press" can be used instead of other verbs, because of the characteristics of the medium, `c-button-clr`. If the medium were a slider, a verb such as "push" or "move" would be required. In contrast, IBIS uses a meta-object, an arrow, to depict the action of pushing. To refer to the clear button, FUF uses a



Install the new holding battery. Step 1 of 7

Stand the radio on its top side.

(a)



Install the new holding battery. Step 2 of 7

Step 2:
Remove the holding battery cover plate:
Loosen the captive screws and pull the holding battery cover plate off of the radio.

(b)

Figure 4: Part of an explanation produced by COMET for installing the holding battery.

definite noun phrase, whereas IBIS highlights the object in the picture.

To portray its location, IBIS uses a combination of techniques: it selects a camera position that locates the button panel (of which the button is part) centrally in the picture; it crops the picture so that additional, surrounding context is included; it highlights the button; and it subdues the surrounding objects, since it determines that the text on the button cannot be made any brighter than the surrounding text. If FUF were to convey location, it would use a prepositional phrase. Here, COMET's media coordinator has determined that location information appears in graphics only, while the causal relation between the action and the display clearing is realized only in text. In general, COMET performs a mapping from communicative goals to text and graphics resources, using media-specific knowledge about the resources available to achieve the goals. A discussion of communicative goals and the associated media-specific resources that can achieve them can be found in [Elhadad et al. 89].

```
((cat lf)
  (directive-act goal)
  (goal
   ((distinct
     ((car
       ((process-type action)
        (process-concept c-remove)
        ...
        (roles
         ((medium
           ((object-concept c-hb-cover-plate)
            ....))))))))))
  (substeps
   ((distinct
     ((car
       ((process-type action)
        (process-concept c-loosen)
        ...
        (roles
         ((medium
           ((object-concept c-captive-screw)
            ...))))))
      (cdr
       ((car
         ((process-type action)
          (process-concept c-pull)
          ...
          (roles
           ((medium
             ((object-concept c-hb-cover-plate)
               ...))
            (from-loc
             ((object-concept c-rt)
              ....))))))))
    ...)))))
```

Figure 5: Content planner output (LF 2): Step 2: Remove the holding battery cover plate: Loosen. . . .

Installing the holding battery

COMET produces seven LFs as part of an explanation of how to install the "holding battery" (which provides power for the memory when the main battery has been removed). The first LF, which corresponds to the generated explanation shown in Fig. 4(a) is a simple LF. The second, which corresponds to Fig. 4(b) is a complex LF. The complex LF consists of one goal (to remove the holding battery cover plate) and two complex substeps. The remainder of the explanation, which is not shown, contains additional simple and complex LFs.

This example illustrates how information in the LF that is *not* realized by a medium can influence that medium's generator. The second LF of this explanation, shown in outline form in Fig. 5, contains one goal and two substeps that carry out that goal. As can be seen in Fig. 4, the media coordinator determines that the goal is to be generated in text ("Remove the holding battery cover plate:") and that the substeps are to be shown in both media. Although IBIS is to depict just the substeps of the LF, it receives the entire annotated LF as input. Since it receives the full LF, and not just the pieces earmarked for graphics, IBIS knows that the actions to be depicted are steps that achieve a higher-level goal. Although the higher-level goal is not actually realized in graphics, IBIS uses this information to create a composite picture. If IBIS were to receive only the substeps, it would have no way of knowing that in the explanation as a whole these actions are described in relation to the goal, and it would produce two separate pictures, just as it does for each simple LF, such as that in part (a) of the figure. Thus, information that is being conveyed in the explanation as a whole, but not in graphics, is used to influence how graphics depicts other information.

location information	graphics only
physical attributes	graphics only
simple actions	text and graphics
compound actions	text and graphics
conditionals	text for connectives, text and graphics for actions
abstract actions	text only

Figure 6: Division of information.

Media Coordinator

The media coordinator receives as input the hierarchy of LFs produced by the content planner and determines which information should be realized in text and which in graphics. Our media coordinator does a fine-grained analysis, unlike other multiple media generators (e.g., [Roth, Mattis, and Mesnard 88]), and can decide whether a portion of an LF should be realized in either or both media. Based on informal experiments and on work reported on in the literature, we distinguish between six different types of information that can appear in an LF, and have categorized each type as to whether it is more appropriately presented in text or graphics, as shown in Fig. 6 [Lombardi 89].

Our experiments involved hand-coding displays of text/graphics explanations for situations taken from the radio repair domain. We used a number of methods for mapping media to different kinds of information, ranging from the use of text only, graphics only, and both text and graphics for all information, to several variations on the results shown in Fig. 6. Among the results, we found that subjects preferred that certain information appear in one mode only and not redundantly in both (e.g., location information in graphics, and conditionals in text). Furthermore, we found that there was a strong preference for tight coordination between text and graphics. For example, readers much preferred sentence breaks that coincided with picture breaks.

The media coordinator is implemented using our functional unification formalism (see the following section), and has a grammar that maps information types to media. This grammar is unified with the input LFs and results in portions of the LF being tagged with the attribute value pairs `(media-text yes)` and `(media-graphics yes)`, with a value of `no` used when the information is not to be presented in a given medium. The media coordinator also annotates the LFs with indications of the type

```
((cat lf)
 (directive-act substeps)
 (substeps
  [((process-type action)
    (process-concept c-push)
    (mood non-finite)
    (speech-act directive)
    (function ((type substeps)
               (media-text yes)
               (media-graphics no)))
    (roles
     ((medium
       ((object-concept c-button-clr)
        (roles
         ((location ((object-concept c-location)
                     (media-graphics yes)
                     (media-text no)))
          (size     ((object-concept c-size)
                     (media-graphics yes)
                     (media-text no)))))
        (quantification ((definite yes)
                         (countable yes)
                         (ref-obj singular)
                         (ref-set singular)))
        (ref-mode description)
        (cat role)))))
    (cat lf)
    (media-graphics yes)
    (media-text yes)))
 (effects
  [((process-type action)
    (process-concept c-clear)
    (mood non-finite)
    (function ((type effects)
               (media-text yes)
               (media-graphics no)))
    (speech-act assertive)
    (roles
     ((agent
       ((object-concept c-display)
        (roles
         ((location ((object-concept c-location)
                     (media-graphics yes)
                     (media-text no)))
          (size     ((object-concept c-size)
                     (media-graphics yes)
                     (media-text no)))))
        (quantification ((definite yes)
                         (countable yes)
                         (ref-obj singular)
                         (ref-set singular)))
        (ref-mode description)
        (cat role)))))
    (cat lf)
    (media-text yes)
    (media-graphics yes))]))
```

Figure 7: Media coordinator output
(LF 1 with annotations).

of information (e.g., simple action vs. compound action), as this information is useful to the graphics generator in determining the style of the generated pictures. Portions of the resulting annotated output for the first LF are shown below in Fig. 7, with the annotations that have been added for the media generators in boldface.

The explanation shown in Fig. 4 illustrates how COMET can produce a fine-grained division of information between text and graphics. In both of the segments (a) and (b), location information is portrayed in the picture only (as dictated by annotations such as those shown in Fig. 7), while the entire action is realized in both text and graphics. In contrast, the overview in part (b) is realized in text only.

Bidirectional Interaction between Components

We have been able to achieve a good level of coordination between text and graphics through a common content description and the media coordinator. The use of a common description language allows each media generator to be aware of the goals and information the other is realizing and to let this knowledge influence its own realization of goals. The media coordinator performs a fine-grained division of information between media, allowing for a tightly integrated explanation. There are certain types of coordination between media, however, that can only be provided by incorporating interacting constraints between text and graphics. Coordination of sentence breaks with picture breaks, references to accompanying pictures (e.g., "the knob in the lower left hand corner of the picture"), and coordination of pictorial and textual style are all examples that require bidirectional interaction between text and graphics components.

Coordinating sentence breaks with picture breaks

Consider the task of coordinating sentence breaks with picture breaks. IBIS uses a variety of constraints to determine picture size and composition, including how much information can easily fit into a single picture, the size of the objects being represented, and the position of the objects and their relationship to each other. Some of these constraints cannot be overridden. For example, if too many objects are depicted in a single picture, individual objects may be rendered too small to be clearly visible.

This situation suggests that constraints from graphics should be used to determine sentence size and thereby achieve coordination between picture and sentence breaks. However, there are also grammatical constraints on sentence size that cannot be overridden without creating ungrammatical, or at the least, very awkward text. Verbs each take a required set of inherent roles. For example, "put" takes a medium and to-location. Thus, "John put." and "John put the book." are both ungrammatical. Once a verb is selected for a sentence, this can in turn constrain minimal picture size; the LF portion containing information for all required verb roles should not be split across two pictures. Therefore, we need two-way interaction between text and graphics.

Our proposed solution is to treat the interaction as two separate tasks, each of which will run independently and annotate its own copy of the LF when a decision is made. The text generator will produce text as usual, but once a verb is selected for a sentence, the text generator will annotate its copy of the LF by noting the roles that must be included to make a complete sentence. At the same time, the graphics generator will produce pictures as usual, creating a hierarchical picture representation incorporating pieces of the LF. This representation indicates where picture breaks are planned. The graphics generator will annotate its LF with pointers into the picture hierarchy, indicating these tentative picture breaks.

When there is a choice among different possible sentence structures, the text generator will use the graphics generator's annotations to make a choice. The text generator can read the graphics generator's annotations by using unification to merge the graphics generator's annotated LF with its own, or can examine the relevant portions of the LF. Similarly, *when there is a choice* among different possible picture breaks, the graphics generator can use the text generator's annotations on minimal sentence size when making the decision. When there are real conflicts between the two components, either one component will generate less than satisfactory output or coordination of sentence breaks with picture breaks must be sacrificed.

Coordination through unification

While there are clearly many difficult problems in coordinating the two tasks, our use of FUF for annotating the LF allows for some level of bidirectional interaction quite naturally through unification. We use FUF in our system for the media coordination task, for the selection of words, for the generation of syntactic structure (and linearization to a string of words), and for the mapping from communicative goals to graphics resources. Each of these components has its own ''grammar'' that is unified with the LF to enrich it with the information it needs. For example, the lexical chooser's ''grammar'' is a Functional Unification Lexicon, which contains domain concepts as keys and associated attribute-value pairs that enrich the input LF with selected words, their syntactic category, and any syntactic features of the selected words. The result is a cascaded series of FUF ''grammars,'' each handling a separate task.

Currently, the unifier is called separately for each grammar, as we are still developing the system. We plan to change this, eventually calling the unifier once for the combined series of grammars, thus allowing complete interaction through unification among the different types of constraints. In this scenario, a decision made at a later stage in processing can propagate back to undo an earlier decision. For example, selection of syntactic form can propagate back to the lexical chooser to influence verb choice. Similarly, selection of a verb can propagate back to the grammar that maps from goals to graphics resources, to influence the resource selected.

There are many problems that must be addressed for this approach to work. We are currently considering whether and how to control the timing of decision making. Note that a decision about where to make a picture break, for example, should only affect sentence size when there are no reasonable alternatives for picture division. Unresolved issues include at what point decisions can be retracted, when a generator's decisions should influence other generators, and what role the media coordinator should play in mediating between the generators.

Conclusions and Future Work

We have focused on three features of COMET's architecture that allow the dynamic generation of integrated multimedia explanations: a common content description, the fine-grained assignment of information to media, and bidirectional interaction among components. The use of an annotated common content description allows each media-specific generator to be aware of all information to be communicated in the explanation, and to use that information to influence the realization of segments for which it is responsible. Our media coordinator allows for small portions of the same LF to be realized in different media. For example, location modifiers of an object may be expressed in graphics only, while the remainder of the LF is expressed in both text and graphics. Similarly, conditionals may be expressed in text only, while the conjoined actions may be expressed in both text and graphics. Finally, our proposed approach for accomplishing bidirectional interaction between components will make it possible for the text and graphics generators to communicate with each other. This will allow decisions made by each generator to influence the other.

We are pursuing a number of different research directions, in addition to our work on bidirectional interaction. Our media coordinator is currently more of a dictator than a coordinator. We are interested in developing strategies for those situations in which the media generators determine that the assignments made by the media coordinator are unsatisfactory. In these cases, the generators could provide feedback to the coordinator, which could in turn modify its plan. We are also interested in situations where context influences the selection of media. Finally, although a single media-independent content planner has definite advantages, there are situations in which it, too, should accept feedback from the generators and modify the content specification.

Acknowledgements

The development of COMET is an ongoing group effort and has benefited from the contributions of Michelle Baker (plan execution component), Cliff Beshers (menu interface), Andrea Danyluk (learned rule base), Michael Elhadad (FUF), Laura Gabbe (static knowledge base and content planner), David Fox (text formatting component for media layout), Jong Lim (static knowledge base and content planner), Jacques Robin (lexical chooser), Doree Seligmann (IBIS), Tony Weida (static knowledge base), Matt Kamerman (user model), and Christine Lombardi and Yumiko Fukumoto (media coordinator).

References

[Allen 87] Allen, J.
Natural Language Understanding.
Benjamin Cummings Publishing Company, Inc., Menlo Park, CA, 1987.

[Arens, Miller, and Sondheimer 88]
Arens, Y., Miller, L., and Sondheimer, N.
Presentation Planning Using an Integrated Knowledge Base.
In *Proc. ACM SIGCHI Workshop on Architectures for Intelligent Interfaces*, pages 93-107. Monterey, March 29-April 1, 1988.

[DOA 86] Department of the Army.
TM 11-5820-890-20-1 Technical Manual: Unit Maintenance for Radio Sets AN/PRC-119, . . .
Headquarters, Department of the Army, June, 1986.

[Elhadad 88] Elhadad, M.
The FUF Functional Unifier: User's Manual.
Technical Report, Columbia University, June, 1988.

[Elhadad et al. 89] Elhadad, M., Seligmann, D., Feiner, S., and McKeown, K.
A Common Intention Description Language for Interactive Multi-media Systems.
In *A New Generation of Intelligent Interfaces: Proceedings of IJCAI89 Workshop on Intelligent Interfaces*, pages 46-52. Detroit, MI, August 22, 1989.

[Feiner 88] Feiner, S.
An Architecture for Knowledge-Based Graphical Interfaces.
In *Proc. ACM SIGCHI Workshop on Architectures for Intelligent Interfaces*, pages 129-140. Monterey, April, 1988.

[Lombardi 89] Lombardi, C.
Experiments for determining the assignment of information to media in COMET.
1989.
Columbia University, New York, NY.

[MacGregor & Brill 89]
MacGregor, Robert and David Brill.
LOOM Reference Manual.
Technical Report, USC-ISI, Marina del Rey, CA, 1989.

[McKeown 85] McKeown, K.R.
Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.
Cambridge University Press, Cambridge, England, 1985.

[Neal and Shapiro 88]
Neal, J. and Shapiro, S.
Intelligent Multi-Media Interface Technology.
In *Proc. ACM SIGCHI Workshop on Architectures for Intelligent Interfaces*, pages 69-91. Monterey, March 29-April 1, 1988.

[Paris 87] Paris, C.L.
The Use of Explicit User models in Text Generation: Tailoring to a User's Level of Expertise.
PhD thesis, Columbia University, 1987.

[Roth, Mattis, and Mesnard 88]
Roth, S., Mattis, J., and Mesnard, X.
Graphics and Natural Language as Components of Automatic Explanation.
In *Proc. ACM SIGCHI Workshop on Architectures for Intelligent Interfaces*, pages 109-128. Monterey, March 29-April 1, 1988.

[Seligmann and Feiner 89]
Seligmann, D.D., and Feiner, S.
Specifying Composite Illustrations with Communicative Goals.
In *Proc. UIST89 (ACM SIGGRAPH Symposium on User Interface Software and Technology)*, pages 1-9. Williamsburg, VA, November 13-15, 1989.

[Wahlster et al. 89]
Wahlster, W., Andre, E., Hecking, M., and T. Rist.
WIP: Knowledge-based Presentation of Information.
Technical Report WIP-1, German Research Center for Artificial Intelligence, Saarbruecken, FRG, May, 1989.