

# TAILORING LEXICAL CHOICE TO THE USER'S VOCABULARY IN MULTIMEDIA EXPLANATION GENERATION

Kathleen McKeown  
Jacques Robin  
Michael Tanenblatt

Department of Computer Science  
450 Computer Science Building  
Columbia University  
New York, N.Y. 10027

{kathy,robin,tanenbla}@cs.columbia.edu

## ABSTRACT

In this paper, we discuss the different strategies used in COMET (COordinated Multimedia Explanation Testbed) for selecting words with which the user is familiar. When pictures cannot be used to disambiguate a word or phrase, COMET has four strategies for avoiding unknown words. We give examples for each of these strategies and show how they are implemented in COMET.

## 1. Introduction

A language generation system should select words that its user knows. While this would seem to involve simply selecting a known word instead of an unknown word (as is done, for example, in [1]), in many cases it requires entirely rephrasing the rest of the sentence. For example, in our domain of equipment maintenance and repair, if the user does not know the word "polarity," a sentence like "Check the polarity." will be rephrased as "Make sure the plus on the battery lines up with the plus on the battery compartment." Even when alternative words can be used instead of an unknown word (e.g., a descriptive expression can be used instead of an object name), the alternative phrase may interact with other parts of the sentence which then need to be reworded as well.

In this paper, we discuss the different strategies used in COMET for selecting words with which the user is familiar. Since COMET integrates text and pictures in a single explanation<sup>1</sup>, unknown words are frequently disambiguated through accompanying pictures. For example, when the accompanying picture clearly shows the object and its location, COMET will use the most common object name even if the

user is unfamiliar with the name<sup>2</sup>. When pictures cannot be used to disambiguate a word or phrase, COMET has four strategies for avoiding unknown words:

1. Selecting an alternative word or phrase (e.g., generating "some number" instead of "arbitrary number")
2. Rephrasing by providing conceptual definitions (e.g., generating "Make sure the plus on the battery lines up with the plus on the battery compartment." instead of "Check the polarity")
3. Rephrasing by generating descriptive referring expressions (e.g., generating "the cable that runs to the KY57" instead of "the COMSEC cable")
4. Using past discourse to construct a referring expression (e.g., generating "Test the cable you just removed." instead of "Test the COMSEC cable." if the user had previously been instructed to remove this cable.)

In the following sections, we first provide an overview of lexical choice in COMET, showing how and where it occurs in the overall system. Each of the strategies is then described in turn, prefaced by a brief discussion of disambiguation of unknown terms through pictures. Finally, we compare our work with previous work in the area.

---

<sup>1</sup>See [2] for a system overview and [3, 4] for details on media coordination in COMET.

---

<sup>2</sup>This is similar to Appelt's [5] integration of language and physical actions for generating referring expressions.

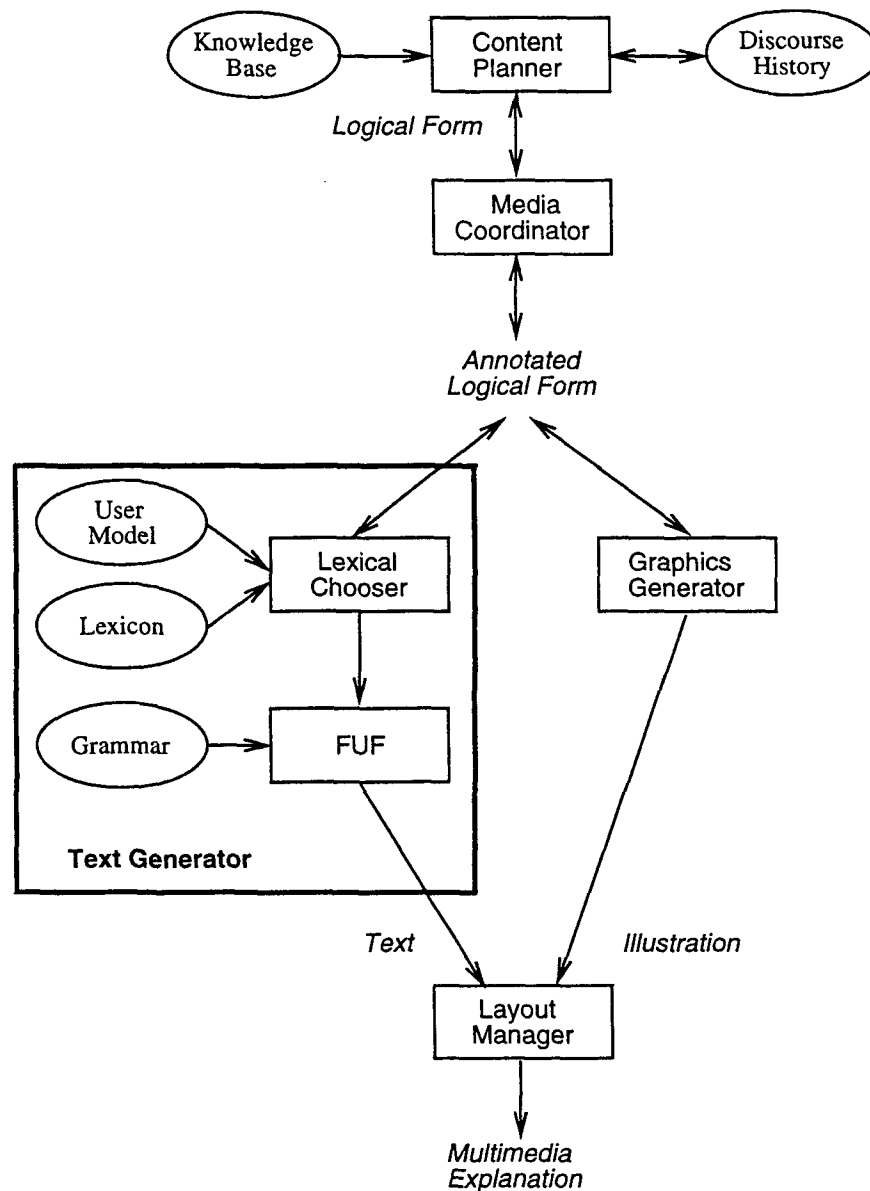


Figure 1: COMET System Architecture

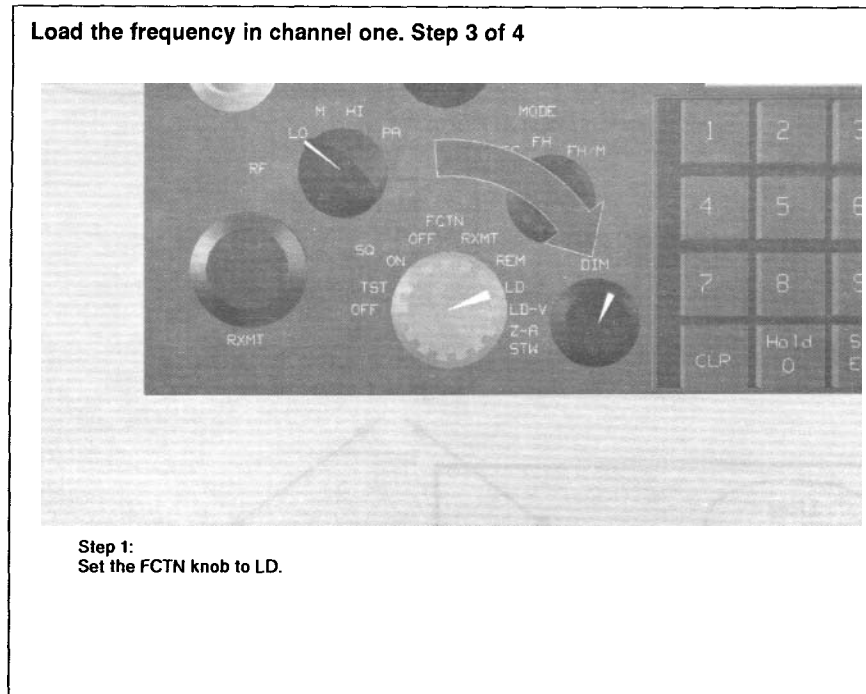
## 2. Lexical Choice and Architecture

COMET's architecture is shown in Figure 1. On receiving a request for an explanation via a menu interface, the **content planner** uses schemas [6] to determine which information should be included in the explanation from the underlying **knowledge sources**. The explanation content, represented as a hierarchy of logical forms (LFs) [7] is passed to the **media coordinator** [3, 8], which adds annotations indicating which portions are to be produced by the **text generator** and which by the **graphics generator** [9].

The **Lexical Chooser** is part of the text generator [7]. Typically, it selects a word or phrase for each semantic concept in the input LF (i.e., the semantic con-

straints on word choice). In terms of coverage, the implementation can select words for 148 different semantic concepts using 253 mapping rules, thus yielding on average slightly less than two alternative word choices per concept (there are many concepts which are mapped to a single word, while others have more than two alternatives). The lexicon contains 159 open class words.

In this paper, we show how the user model and past discourse (pragmatic constraints) also influence word choice. But these are not the only constraints on word choice. Syntactic form of the sentence and lexical constraints are other demonstrated [10, 11] influences on lexical choice. For example, once the verb has been chosen, syntactic constraints on its arguments (e.g., whether the object is a clause,



**Figure 2:** Accompanying Picture Clarifies Referent

an adj, or np) will influence what words are chosen to realize the semantic concept that fill these arguments. Conversely, if one of the verb roles can only be realized as a noun phrase, for example, and not as other syntactic categories, this restricts which verb is selected. Lexical constraints on word choice arise from the use of collocations [12]. For example, a verb like “stand” takes the preposition “on” for its location role, while the verb “turn” takes the preposition “onto.” Lexical choice is thus influenced by a wide variety of constraints which interact in many ways.

Since syntactic and lexical constraints are only available within the text generator, lexical choice is delayed until this point. Thus COMET waits until a variety of semantic, pragmatic, syntactic and lexical constraints are accumulated before selecting words. This means that COMET can use syntactic and lexical constraints on word choice in conjunction with semantic and graphical constraints provided as input, plus the new pragmatic constraints we present. Previous work addressing pragmatic constraints on word usage folded lexical choice into the content planner (e.g., [13], [1]). This was possible since the work focused primarily on lexical side effects of content determination (e.g., what property to include in a reference as opposed to what linguistic form to use for a property). Such approaches do not allow a system to take syntactic and lexical constraints on word choice into account.

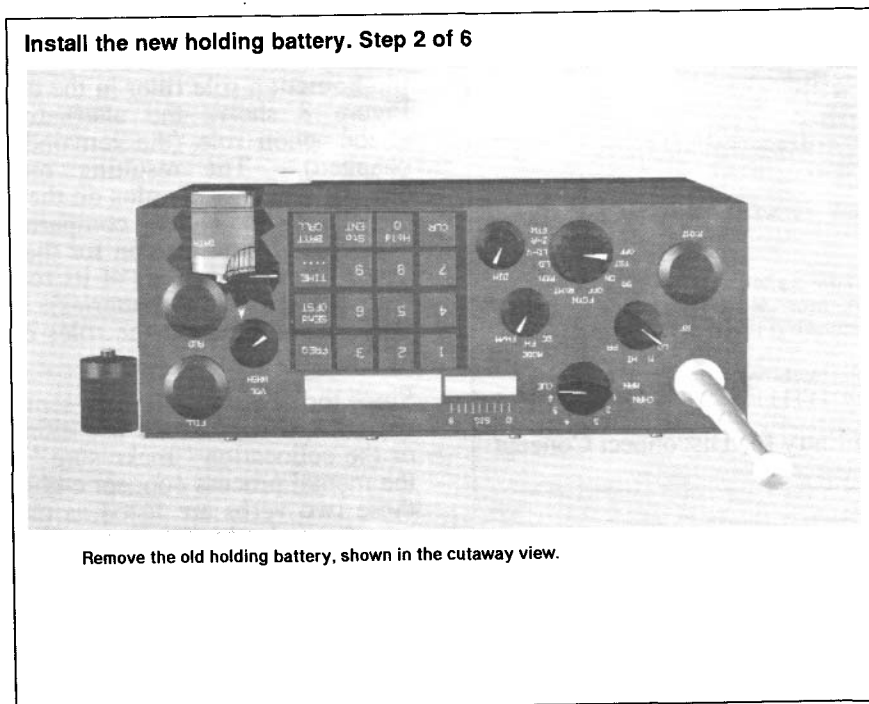
On receiving the hierarchy of logical forms, the Lexical Chooser determines the overall grammatical form of each sentence based on the semantic structure of the LFs (e.g., conditional sentences are generated for precondition-action structures) and selects the words

and phrases realizing semantic concepts of the LF. It passes a specification of the sentence’s grammatical form and open-class words to the general purpose surface sentence generator FUF [14, 15, 16]. The Lexical Chooser uses a rewriting system itself implemented on top of FUF. Its lexicon consists of a base of rules, where each rule rewrites a given set of semantic features into a corresponding set of lexical and syntactic features. Thus, each lexicon entry associates a semantic concept with words that can be used to realize it. Additional constraints from the user model, past discourse, and the underlying knowledge base determine which of the alternative words or phrases should be selected.<sup>3</sup> The user model indicates both the reading level of the current user<sup>4</sup>, any individual words that COMET knows the user does not understand, and any wording preferences (e.g., the user knows abbreviations, the user is familiar with military terminology). We make no claims about which of these forms of user models is easier to acquire, but simply show how to use them when available.

If none of the alternative wordings for a given semantic concept of the LF are known to the user and the

<sup>3</sup>When these constraints come from knowledge sources external to FUF, the Lexical Chooser uses FUF extensions to access such knowledge through the use of coroutines [17].

<sup>4</sup>We currently use two levels for a poor and good reader. At the beginning of the session, the reading level is either preset or COMET can ask the user.



**Figure 3:** Use of Cross References: Remove the holding battery, shown in the cutaway view

accompanying illustration cannot disambiguate these words, COMET reinvokes the content planner to replan portions of the sentence content or to include additional semantic information. Thus, COMET's architecture interleaves lexical choice and content planning in order to account for a wide variety of interacting constraints on word choice.

### 3. Multimedia Disambiguation

An accompanying picture often makes clear what the referent of a referring expression is. If the user is unfamiliar with a term, the accompanying picture might define it. For example, Figure 2 shows one step of an explanation generated by COMET for loading frequency into the radio. The text refers to a "FCTN knob" and the accompanying picture clearly singles out the knob on the front panel of the radio [4]. COMET can also generate an explicit reference to the illustration itself (called a *cross reference*). For example, the cross reference shown in Figure 3 is generated if the user does not understand the term "holding battery". In this case, the Lexical Chooser, on determining that "holding battery" is an unfamiliar term, reinvokes the content planner which finds that no accompanying illustration is currently planned and invokes graphics to generate an accompanying illustration that depicts the holding battery and its location. For full details on cross referencing in COMET see [18].

### 4. Selecting a Familiar Word/phrase

Whenever possible, COMET simply selects a familiar word over an unknown word from the list of

alternatives in the lexicon. Figure 4 shows some paired sentences that COMET generates which illustrate alternative wordings. The first italicized phrase is generated if the user's vocabulary level is above a certain reading level or if a word is *not* explicitly listed in the user model as unknown. Since the lexicon maintains a simple association between the semantic concept and alternative phrasings, COMET selects the first alternative which the user model indicates is familiar to the user. For example, Figure 5 shows that for any concept under the concept **c-disconnect** in the knowledge base taxonomy, COMET will use the word "disconnect" if the user's vocabulary level is high and the word "remove" otherwise. COMET also checks whether the user knows abbreviations and if so, will use a referring expression such as "FCTN knob" as shown in Figure 2. If not, COMET uses the full name ("function knob"). If COMET has no information about the user, it generates the abbreviation and relies on the accompanying illustration to clarify the referent.

1. Screw the new manpack antenna onto the RT and tighten until the manpack antenna is *snug/tight*.
2. *Disconnect/Remove* the COMSEC cable from the KY57 audio connector.
3. This will cause the display to show *an arbitrary/some* number.

**Figure 4:** COMET-Generated Word Substitutions

```

(; semantic key
 ((concept #(under c-disconnect)))
 ; realization
 ((process
  ((cat verb-group) ; will be a verb
   (alt
    (
     ; if level high select "disconnect"
     ((CONTROL (OK-Lex-UM 'c-disconnect high))
      (lex "disconnect")))
    ; else select "remove"
    ((lex "remove"))))))))

```

Figure 5: Lexicon Entry for Disconnect Concept

## 5. Rephrasing through Replanning

Selecting an alternative wording for a semantic concept is not always possible since none of the alternatives may be known by the user. Instead, COMET can describe concepts at a more detailed semantic level of abstraction by retrieving additional definitional information from the knowledge base and it can create referring descriptions when object names are not known, by retrieving object attributes.

### 5.1. Retrieving alternative concept definitions

Sometimes the original text uses a word or phrase that abstracts the details of a concept to allow generation of a very concise expression. If unfamiliar with the word or phrase, the user will be unable to infer the specifics needed to perform the task. Alternative wordings require choosing a less abstract level of semantic decomposition at which to describe the concept. In these cases, COMET's lexical chooser reinvokes the content planner to retrieve a finer grained definition of the concept from the knowledge base.

For example, this strategy is used for rephrasing the request "Check the polarity" which COMET issues when providing instructions for installing a new holding battery. More detailed semantics of checking the polarity are stored as different tokens of the concept **c-polarity** in the knowledge base.<sup>5</sup> For example, in Figure 6 polarity is represented as the equivalence between the two plusses on two batteries<sup>6</sup>. Now, if the plan calls for checking polarity, it can be represented in terms of a checking action on the equivalence of these two plusses (i.e., that they line up). If the user

<sup>5</sup>The more detailed definition is stored with **c-polarity** and not with **c-check** since in our domain checking is carried out on many different objects, while few actions are carried out on polarity.

<sup>6</sup>The equative relations has two roles, *identified* and *identifier*. Since they are included here, the equative relation (i.e., that the two plusses "line up") is inferred to hold.

is unfamiliar with the word "polarity," an alternate decomposition will be retrieved and replace the **phenomenon** role filler in the original LF (Figure 7). Figure 8 shows the alternative LF with a new phenomenon role (the remainder of the LF is unchanged). The resulting rephrased sentence is "Make sure that the plus on the battery lines up with the plus on the battery compartment." "Lines up" is selected in the lexicon for the equivalence relation based on the semantics of its roles (i.e., that they are both plusses on the batteries). Here semantic selectional restrictions on the roles control lexical choice of the verb.

Since the object of the new sentence is an embedded sentence, COMET can use either the verb "check" or the collocation "make sure" as the verb realizing the mental process concept **c-check**. Note that, while these two verbs are listed as alternatives in the lexicon for **c-check**, "make sure" cannot be used in the original sentence due to a syntactic constraint: its object cannot be an NP as one cannot say "Make sure the polarity." This is an example of interaction between syntactic and pragmatic constraints. Since syntax does not constrain the choice of verb in the modified sentence, COMET arbitrarily selects "make sure".

The lexicon entry containing these two verbs is shown below in Figure 9. Note that the entry is indexed by the semantic concept **c-check**. There are two alternative verbs, only one of which is compatible with a clause as phenomenon role (ultimately the object). When the phenomenon is an NP, both verbs are valid and one is randomly selected.

```

; Instance definitions for polarity
(tellm (polarity polarity-1)
      (polarity polarity-2))

; More detail for one instance: polarity is
; represented as two plusses which should
; be equivalent. The roles of the equative
; relation are identified and identifier
(:about polarity-2
 (identified plus-1)
 (identifier plus-2))

; one is located on the battery
(:about plus-1 (on-loc battery-1))

; one is located on the battery compartment
(:about plus-2 (on-loc bc-1)))

```

Figure 6: Knowledge base tokens for polarity

```

((Concept C-Check) ; "check"
 (Process-Type Mental)
 (Roles
  ((Phenomenon
   ((Concept C-Polarity)))))) ; "the polarity"

```

**Figure 7: Logical Form for Original Sentence**

```

((Concept C-Check) ; "make sure that"
 (Process-Type Mental)
 (Roles
  ((Phenomenon
   ((Concept C-Polarity)
    (Process-Type Equative) ; "lines up with"
   (Roles
    ((Identified
     ((Concept C-Plus) ; "the plus"
      (Roles
       ((On-Loc ; "on the battery"
        ((Concept C-Battery))))))
    (Identifier
     ((Concept C-Plus) ; "the plus"
      (Roles
       ((On-Loc
        ; "on the battery compartment"
        ((Concept C-BC))))))))))))))

```

**Figure 8: Logical Form of Rephrased Sentence**

```

( ; semantic key
 ((concept #(under c-check))

 ; realization
 (cat verb-group) ; will be a verb
 (alt
  ( ; if phenomenon realized by NP
   ((roles
    ((phenomenon ((cat #((under np)))))))

   ; then always choose "to check"
   (lex "check"))

 ; if phenomenon realized by clause
 ((roles
  ((phenomenon ((cat #((under clause))))))

  ; then randomly pick "to check" or
  ; "to make sure"
  (lex ((Ralt ("check" "make sure"))))))))

```

**Figure 9: Lexicon Entry for Check Concept**

## 5.2. Generating New Referential Descriptions

If the user does not know an object name, the content planner is reinvoked to generate object attributes to build a referential description. Although our selection algorithm is not as sophisticated as others [19, 5, 13] because we do not use a detailed model of user beliefs, we address a new issue: the interaction between the new description and other parts of the original sentence which may require rephrasing. Two types of object attributes are used in a referring expression in COMET: object subpart relations and

spatial relations to other objects in the accompanying illustration. COMET selects the relations that uniquely identify the object.

For example, suppose COMET's Lexical Chooser is provided with the LF for sentence 1, Figure 10, but the user does not know the term "COMSEC." Instead of generating sentence 1, COMET generates sentence 2. To do this, COMET first selects a unique relation between the cable and a known object. In this case, it selects the **connects** spatial relation between the Radio Transmitter (RT) and the KY57, since this cable is the only one that connects the radio and the KY57. Selecting this relation for the description and substituting it for "the COMSEC cable" would result in sentence 3, Fig. 10. However, COMET notes the redundant references to the audio connector and removes one from the cable modifier by selecting the verb "runs to" instead which only requires one role in the generated sentence. This would result in the sentence 4, Fig. 10. In this sentence, the attachment of the prepositional phrase "from the KY57 audio connector" is ambiguous. COMET detects this ambiguity when it removes the first from-location; since the two from-locations would have occurred side by side and both previous verbs of the sentence take it as a modifier, the generator must clarify that it is the from-location of the earlier verb "disconnect" and not "run to." To remove ambiguity, COMET surrounds the modifier of the cable by commas in sentence 2, Fig. 10<sup>7</sup>.

### *Descriptions Generated by COMET:*

1. "Disconnect the COMSEC cable from the KY57 audio connector."
2. "Disconnect the cable, which runs to the RT, from the KY57 audio connector."

### *Descriptions Avoided by COMET:*

3. "Disconnect the cable that connects the RT to the KY57 audio connector from the KY57 audio connector."
4. "Disconnect the cable that runs to the RT from the KY57 audio connector."

**Figure 10: Generated Object Description**

<sup>7</sup>Another possible way to avoid ambiguity would be to generate two sentences such as "Find the cable that runs from the RT to the KY57 audio connector. Disconnect the cable from the audio connector."

## 6. Using Past Discourse

For subsequent reference, the presence of a discursive context allows for a wider variety of strategies to get around gaps in the user's vocabulary. COMET takes advantage of this fact by maintaining a discourse history. The content planner records all descriptions into the discourse history, creating one record for the description as a whole and a separate record for each of its roles. The entry for the description has four fields:

- The name of the concept.
- The description used in the reference.
- The action in which the referring description plays a role.
- The list of roles that the description fills in that action (e.g., "COMSEC cable" is the **medium** of the action "disconnect").

For each subsequent reference, the concept name is used as the access key and the three other fields are updated; they thus always contain the information on the *last* reference. By looking up information in the discourse history, the content planner is able to construct object descriptions in terms of the last action it was involved in.

### *Sentences generated if the user knows "COMSEC"*

1. "Disconnect the COMSEC cable from the KY57 audio-connector."
2. "Plug in the handset to the KY57 audio-connector."
3. "Test the COMSEC cable."

### *Sentences generated if not:*

4. "Disconnect the cable, which runs to the RT, from the KY57 audio connector."
5. "Plug in the handset to the KY57 audio connector."
6. "Test the cable that you just disconnected."

**Figure 11:** Use of Previous Discourse

As an example, consider the explanations COMET generates when instructing the user how to diagnose loss of side tone. When the user has no vocabulary gaps, COMET generates sentences 1-3, Figure 11. When the user is unfamiliar with the term "COMSEC," sentences 4-6 are generated instead. Here COMET uses past discourse to produce a descriptive

reference for the second reference to the COMSEC cable.

As in the previous examples, the gap is detected when the Lexical Chooser checks the user model. Since there is no alternative phrase for "COMSEC" in the lexicon, COMET calls the content planner to replan the reference. Since it is not the first reference to the cable, COMET uses the discourse history to plan a modifying description. A reference to the cable is discovered in the history (its entry is shown in Figure 12) and the action in this entry is selected as the modifier to build a referring expression.<sup>8</sup> The role of the cable was **medium** and thus, COMET can generate the modifier as a relative clause. The LF for this referring expression is shown in Figure 13. This LF is sent back to the lexical chooser, which selects the words for the concepts within it, and continues with generation where it left off. On third and fourth reference to the same concept, COMET uses its anaphoric reference facility to generate either a bare head (e.g., "cable") or a pronoun (e.g., "it").

```
(; The concept name:
((Concept C-Comsec-Cable))

; The initial generated description:
; included where connected to and from.
((Concept C-Cable)
 (Roles ((To-Loc ((Concept C-RT)))
         (From-Loc ((Concept C-KY57))))))

; The role it plays in the action:
((Roles Medium))

; The action itself: `disconnect the cable`.
((Process-Type Material)
 (Concept C-Disconnect)
  .... ; Rest of action description
  .... ; in discourse history
 ))    ; but not shown here
```

**Figure 12:** Entry for COMSEC Cable in the Discourse History

## 7. Conclusions and Related Work

COMET performs several lexical choice tasks. It can choose between alternative words or phrases for any part of speech. When generating a request to perform an action, it chooses a level of detail in the concept description appropriate to the user. When generating both initial and subsequent referring expressions, it selects a set of distinguishing properties of the referent and chooses words to express the selected

<sup>8</sup>There is a limit to how far back COMET looks in the discourse to construct a new referring expression: the discourse history is cleared after each menu request for a new explanation.

```

((Concept C-Cable)
 (Roles
  ((Latest-Participation
   ((Process-Type Material)
    (Concept C-Disconnect)
    (Roles
     ((Agent ((Concept C-User)))
      (Medium
       ((Concept {^5 Concept}))))))))))

```

**Figure 13:** "the cable you just disconnected"

properties. Finally, for subsequent references, COMET can use previous discourse to avoid unknown words.

COMET is thus using constraints from the user model, the accompanying illustration, and past discourse in addition to traditional constraints from semantics, syntax, and other word choices. Although other generation systems take into account some of these constraints, COMET is the first attempt to integrate such a variety of constraints and lexical choice strategies in a single system. In addition, because COMET is a multimedia system, it can use the accompanying illustrations advantageously for disambiguation.

WIP [20] can also generate cross references but does not rely on a user model for either cross reference generation or lexical choice. EPICURE [19], KAMP [5], and FN [13] tailor references based on situation, but they do not constrain this choice based on the user's lexical knowledge. EPICURE uses the user's domain knowledge, KAMP mutual beliefs about the domain, and FN the user's domain knowledge in conjunction with rules on implicatures. They focus on the selection of appropriate properties to distinguish an object in generating references but do not choose between alternative wordings for the selected properties. None of these systems reword action descriptions or use past discourse to avoid terms the user does not know. While Bateman and Paris' system [21] uses different dialects depending on which class of users it is addressing through register mappings, in COMET different terms can be mixed and matched depending on the individual user model.

### Acknowledgements

Research on language generation in COMET has been supported in part by Defense Advanced Research Projects Agency Contract N00039-84-C-0165, National Science Foundation Grants IRT-84-51438 and GER-90-2406, New York State Center for Advanced Technology Contracts NYSSTF-CAT(90)-053, (91)-053, and (92)-053, and Office of Naval Research Contracts N00014-82-K-0256 and N00014-89-J-1782. COMET's development is an ongoing group effort and has benefited from the contributions of Michael Elhadad (FUF), Doree Seligmann (graphics generator), Andrea

Danyluk (diagnostic rule base), Yumiko Fukumoto (media coordinator), Jong Lim (static knowledge base and content planner), Christine Lombardi (media coordinator), Jacques Robin (lexical chooser), James Shaw (anaphoric reference facility), Michael Tanenblatt (knowledge base, content planner), Michelle Baker, Cliff Beshers, David Fox, Laura Gabbe, Frank Smadja, and Tony Weida.

### REFERENCES

1. Swartout, W.R., "XPLAIN: a system for creating and explaining expert consulting systems", *Artificial Intelligence*, Vol. 21, No. 3, 1983, pp. 285-325.
2. Feiner, S. and K.R. McKeown, "Generating Coordinated Multimedia Explanations", *Proceedings of the IEEE Conference on AI Applications*, Santa Barbara, CA., March 1990.
3. Feiner, S. and K.R. McKeown, "Coordinating Text and Graphics in Explanation Generation", *Proceedings of the National Conference on Artificial Intelligence*, Boston, Mass., August 1990.
4. Feiner, S. and McKeown, K.R., "Automating the Generation of Coordinated Multimedia Explanations", *IEEE Computer*, Vol. 24, No. 10, October 1991, pp. 33-41.
5. Appelt, D.E., *Planning English Sentences*, Cambridge University Press, Cambridge, England, 1985.
6. McKeown, K.R., *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*, Cambridge University Press, Cambridge, England, 1985.
7. McKeown, K.R., Elhadad, M., Fukumoto, Y., Lim, J., Lombardi, C., Robin, J., and Smadja, F., "Language Generation in COMET", in *Current Research in Language Generation*, Mellish, C., Dale, R., and Zock, M., eds., Academic Press, London, 1990.
8. Elhadad, M., Seligmann, D., Feiner, S., and McKeown, K., "A Common Intention Description Language for Interactive Multi-media Systems", *A New Generation of Intelligent Interfaces: Proceedings of IJCAI89 Workshop on Intelligent Interfaces*, Detroit, MI, August 22 1989, pp. 46-52.
9. Seligmann, D.D., and Feiner, S., "Specifying Composite Illustrations with Communicative Goals", *Proc. ACM Symposium on User Interface Software and Technology*, Williamsburg, VA, November 13-15 1989, pp. 1-9.
10. McDonald, D.D., "On the place of words in the generation process", in *Natural Language Genera-*



- tion in Artificial Intelligence and Computational Linguistics*, Paris, C., Swartout, W. and Mann. W.C., eds., Kluwer Academic Publishers, 1991.
11. Danlos, L., *The Linguistic Basis of Text Generation*, Cambridge University Press, Cambridge, England, 1987.
  12. Smadja, F. and K.R. McKeown, "Automatically Extracting and Representing Collocations for Language Generation", *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, Pittsburgh, Pa., June 1990, pp. 252-9.
  13. Reiter, E.B., *Generating appropriate natural language object description*, PhD dissertation, Center for research in computing technology, Harvard University, 1990.
  14. Elhadad, M., "The FUF Functional Unifier: User's Manual", Tech. report, Columbia University, 1988.
  15. Elhadad, M., "Types in Functional Unification Grammars", *Proceedings of the 28th meeting of the Association for Computational Linguistics*, Pittsburgh, Pa, June 1990.
  16. Elhadad, M., *Using argumentation to control lexical choice: a unification-based implementation*, PhD dissertation, Computer Science Department, Columbia University, 1993.
  17. Elhadad, M. and Robin, J., "Controlling Content Realization with Functional Unification Grammars", in *Aspects of Automated Natural Language Generation*, Dale, R. and Hovy, H. and Roesner, D. and Stock, O., ed., Springer Verlag, 1992, pp. 89-104.
  18. McKeown, K. R., Feiner, S.K., Robin, J., Seligmann, D., and Tanenblatt, M., "Generating Cross References for Multimedia Explanations", *Proceedings of AAAI-92*, AAAI, July 1992.
  19. Dale, R., *Generating Referring Expressions*, ACL-MIT Press Series in Natural Language Processing, Cambridge, Ma., 1992.
  20. Wahlster, W., Andre, E., Hecking, M., and T. Rist, "WIP: Knowledge-based Presentation of Information", Tech. report WIP-1, German Research Center for Artificial Intelligence, May 1989.
  21. Bateman, J.A. and Paris, C.L., "Phrasing a text in terms the user can understand", *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Detroit, MI, 1989, pp. 1511-1517.