LANGUAGE SYSTEMS, INC.
DESCRIPTION OF THE DBG SYSTEM AS USED FOR MUC-4

Christine A. Montgomery
Bonnie Glover Stalls
Robert E. Stumberger
Naicong Li
Robert S. Belvin
Alfredo Arnaiz
Susan B. Hirsh

Language Systems, Inc.
6269 Variel Avenue, Suite F
Woodland Hills, CA 91367
(818) 703-5034

Internet: chris@lsi.com

INTRODUCTION

LSI's Data Base Generation (DBG) system is a syntax-driven natural language processing system that integrates syntax and semantics to analyze message text. The goal of the DBG system is to perform full-scale lexical, syntactic, semantic, and discourse analyses of message text and produce a system-internal knowledge representation of the text that can serve as input to a downstream system or external data structure, such as the MUC-4 templates.

DBG's development has been based on analysis of large volumes of message traffic (thousands of Air Force and Army messages) in five domains. The DBG internal knowledge representation has been mapped to external data structures for purposes of data base update, expert system update, and fusion of message content with the content of other messages and other information sources. Although our research on natural language understanding systems goes back almost 20 years, the actual implementations for the individual components of the system are all quite recent, generally occurring within the last two to five years. The texts in the various domains range from formal written messages to transcribed radiotelephone conversations. The DBG system has been formally tested on previously unseen messages in three of the domains, with competitive tests against humans performing the same task in two domains. Recently, the system has been adapted to the Machine Aided Voice Translation (MAVT) project. In this application, the system takes a "live" voice input sentence, uses a speech recognizer to convert it to written text, processes the written text, and generates a written translation of the sentence in the target language. This written output is then input to a speech generator to produce a voice translation of the original utterance. The languages processed thus far by this version of the system are English and Spanish, with translation in both directions.

THE DBG SYSTEM MODULES

The DBG system consists of a series of modules that process message text in stages, and each major level of analysis is contained in a separate module. Processing is performed sequentially: the output of each module is a temporary data structure that serves as input to the succeeding module and is then available to all later modules. Each individual module contains domain-independent processing mechanisms and knowledge bases (rule sets). These knowledge bases allow the incorporation of domain-sensitive features. The modularity of the DBG system has allowed the individual components to be improved and in several cases completely redesigned without requiring changes in the underlying system architecture.

Figure 1 shows the functional flow of the DBG system. The processing modules are shown in boxes and the knowledge bases which apply at each processing stage are shown in ovals. The output of each processing stage

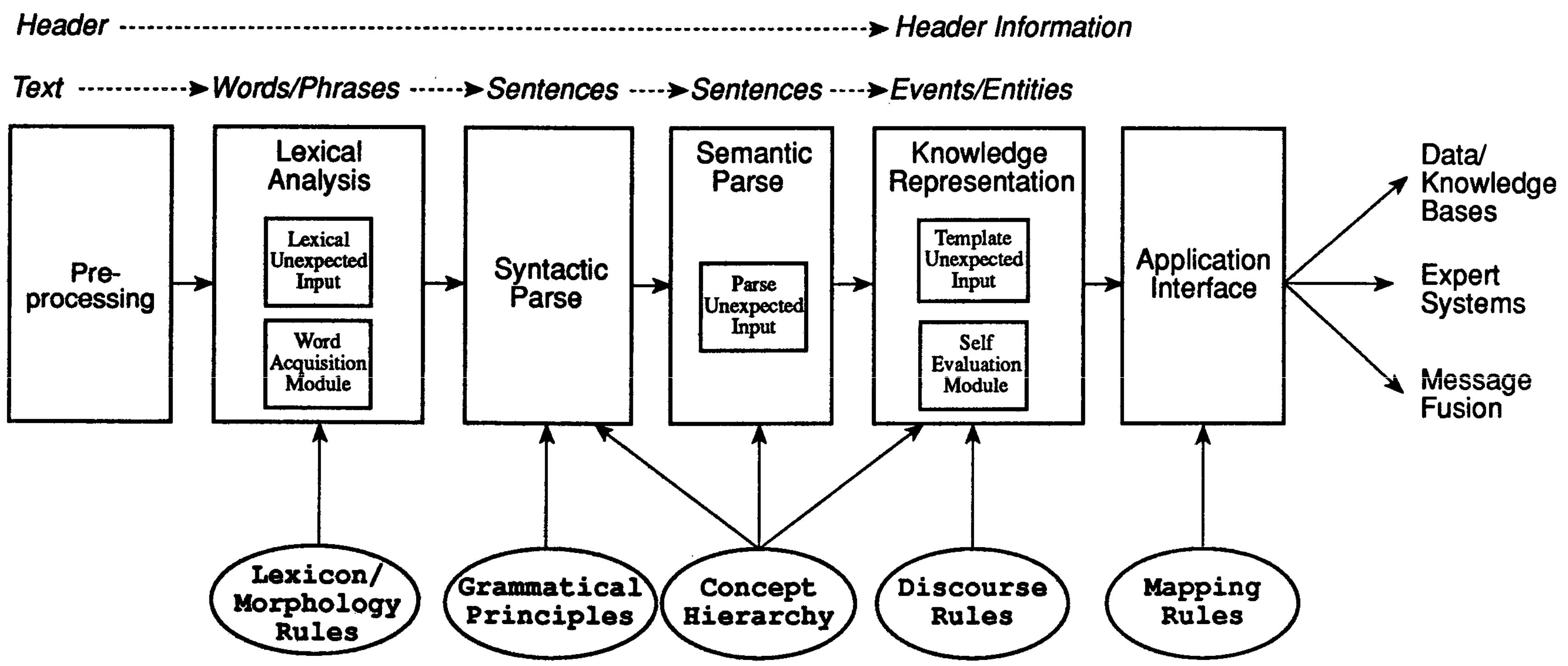

Figure 1. DBG System Diagram

Language Systems, Inc.

is indicated at the top of the particular box (e.g., words/phrases).

The basic components of the DBG system are the preprocessing module, the lexical analysis module, the syntactic parse module, the semantic parse module, and the knowledge representation module. These five modules form the core of the Expected Inputs Subsystem, which handles known, well-formed input text. The knowledge bases include the lexicon and morphological rules, the set of grammatical principles used to construct the syntactic parse trees, the concept hierarchy, the discourse rules, and the rules for mapping into external data structures (e.g., into the MUC-4 templates) for downstream applications. Items in the lexicon are linked to nodes in the concept hierarchy.

The DBG system also has the capability of handling unknown (i.e., new or erroneous) data. This is accomplished in the Unexpected Inputs Subsystem (UX). The UX subsystem consists of modules that are integrated into the Expected Inputs Subsystem. At the lexical level, the UX modules that can apply are the lexical unexpected inputs module (LUX) and the word acquisition module (WAM). At the syntactic parse level, partial parses are acceptable as input to the semantic parse module. At the knowledge representation level, the self-evaluation module (SEM), which records calls to the UX subsystem, scores the system on its performance. The functions of these various modules are described in the following section.

DBG SYSTEM PROCESSING STAGES

At the preprocessing stage, the message is first extracted from the message stream and the message text is segmented into distinct words and sentences. In the version of the system used for MUC-4, text to be processed is then identified by means of an event word list and a result word list. The event words are associated with the primary events of interest for the MUC-4 task, and result words are used to describe the results of those events. Sentences to be processed were selected using these lists. The selection of sentence with result words depends on the presence of certain event words in other sentences in the message. After the sentences are selected, the successive lexical, syntactic parse, and semantic parse modules then analyze the individual sentences.

In the DBG system, for each sentence the lexical definitions of the words and multi-word phrases are matched with items in the lexicon (or derived from Unexpected Input processing, as described below), yielding a lexical analysis for the sentence. In the course of MUC-4 development, the size of the DBG lexicon was expanded, to more than 15,000 (root-form) entries, with more than 14,000 inflected entries added after compilation, yielding a total system lexicon size of almost 30,000 entries. Each entry in the lexicon contains morphological information concerning any irregularities in form, morphosyntactic features pertaining to reference and agreement, subcategorization features, selectional restrictions, and links into the concept hierarchy. The output of this stage of processing is a set of words and phrases with their associated lexical features, which is then passed to the syntactic parser.

The parser is a principle-based parser that uses grammatical principles from Government-Binding Theory to construct a parse tree for each sentence being processed. The parser combines a bottom-up, data-driven approach to attaching incoming words into the parse tree, with a top-down expectation that a complete tree will be built around a verbal projection (Cp-Ip-Vp). The parser mechanism works by projecting incoming words to maximal X-bar projections (three-level node-graphs), and then attempting to attach the projections into currently available "docking locations" on the existing tree, using syntactic and semantic checks to validate the attachment. The parse structure which is built up through these attachments is represented as an acyclic, directed graph. The mechanism itself can be thought of as a "window" which moves through the emerging parse-graph of the sentence, examining/attaching a pair of nodes at a time. The parser places theta-role information (similar to case frames) in properly attached verb-argument nodes.

The parse structure/graph for a sentence is then passed to the semantic parse module which traverses the graph to extract semantic elements and their relations based on the local graph structure, theta-role assignment, and semantic labels derived from the underlying concept hierarchy.

Due to the close integration of syntactic and semantic checking required by the parsers, a facility is also provided which reads integrated lexical/conceptual representations (human writable/readable) that are created by the lexicon developer and converts them into entries for the system-internal lexicon and concept hierarchy databases. This mechanism ensures that lexical entries containing syntactic data are properly linked to concept hierarchy entries containing semantic data.

At the knowledge representation stage, the sentential semantic parses of a message are searched for event and entity data elements having the appropriate category and relations to other elements to instantiate output frames. At this stage data elements from more than one sentence may be combined in the output knowledge representation, depending on the narrative structure of the messages in the particular domain. The knowledge representation is in the form of frame structures specifying the properties of events and entities and their relations to one another. In particular, the hierarchical organization of these frames enables the explicit representation of the relations of various events and entities to one another and degrees of certainty associated with those events. Information implicit to the message is provided by a mechanism of inheritance built into the concept hierarchy subsystem. The system in this way has the capability of incorporating into the knowledge representation generic and domain information not explicitly contained in the message, thus representing a deeper understanding of the message text.

Discourse rules also apply at this stage of processing. These rules relate primary events and results events to one another in the knowledge representation frames. We are also investigating the use of global variables to track entities in the text, in particular the notion of "deictic center" (Rapaport et al. 1989), which incorporates the notion of focus space (Sidner 1983), to track the objects of interest in discourse. Deictic center is the focus of attention in the reader/listener's mental model of the ongoing discourse, which consists of at least the WHO, WHERE, and WHEN information about the event being depicted in the discourse. It is the record that the reader/listener keeps, at any point of the discourse, concerning who or what is being talked about in the current discourse, as well as where and when an event or action is taking place.

There are linguistic devices/cues which serve to establish the deictic center (such as explicit mentioning of a discourse entity in a certain syntactic position and with a certain semantic role), to maintain the deictic center (such as the use of deictic verbs and adverbs, as well as pronominal forms), and to indicate the shift of the deictic center or the possibility of such shift (such as long referential distance, referential competition (Givon 1983), as well as the linguistic devices for establishing a new deictic center). Keeping track of the deictic center of the discourse will contribute to finding the correct referent for anaphoric expressions in the discourse, understanding the overall event-situation structures (Webber 1987) of the discourse, as well as, in the context of MUC, the task of event separation. In particular, the shift of WHERE, WHEN, and/or the perpetrator WHO, in combination with some other factors, are signs indicating that the discourse has taken up a new event. The deictic center approach is currently being implemented in our system.

Another important aspect of discourse modeling which is being incorporated into our system involves the notion of subjectivity/perspective and belief spaces (Banfield 1981, Rapaport 1986, Wiebe & Rapaport 1988) which have been applied to narrative discourse. In the context of MUC, such notions are important for referent tracking as well as event separation. The key notion here is that a discourse may present an event not only from the perspective of the speaker/reporter (which guides the construction of the discourse model of the listener/reader), but also from the the perspective of an entity mentioned in the discourse, such as another reporting source. Things that are considered to be already in the deictic center in one perspective may not yet be in the deictic center of another perspective. Therefore, although within one perspective, an event (or an entity participating in an event) which has already been introduced in the previous discourse would normally be referred to by a definite NP or a pronoun, it may be referred to by an indefinite NP if the perspective of discourse has shifted (to another reporting source, for example). Taking the issue of perspective into consideration would prevent the system from incorrectly inferring that this event (or entity) is a new one because of the use of the indefinite NP.

Another parameter relevant to event separation, which has already been implemented in our system concerns certain verbal acts such as confirmation. Some speech act verbs such as "confirm" or "repeat" usually indicate that the event/entity mentioned in the embedded clause has already been introduced in the previous discourse, even though it is referred to by an indefinite NP.

A key feature of the system that increases its flexibility and provides a built-in means of extending the system to new material is the Unexpected Inputs (UX) subsystem. The UX subsystem, which is a fully integrated part of the DBG system, automatically handles new or erroneous material at all levels, including lexical, syntactic, and semantic/discourse unexpected input. At the same time, it tallies the number of times it is invoked, the number of error hypotheses utilized, and the type and degree of deviance of the data it processes in order to provide the user with a measure of its performance and a check on the system output.

The UX subsystem accomplishes its task by intelligently relaxing the well-formedness constraints on textual data that the system normally requires and by providing tools for adding new words to the system. At the lexical level, the Lexical Unexpected input module (LUX) corrects errors by allowing partial matches between words in the text and the lexical entries stored in the lexicon. These partial matches are based on a set of error hypotheses relating to typographical and Baudot code transmission errors. New or unidentified material is passed to the on-line Word Acquisition Module (WAM1) for preliminary classification by the user by means of menu selection; alternatively, the system can operate in an autonomous mode, wherein a word class is assigned based on the system's morphological analysis of the word. The new words can also be stored for later incorporation into the system by means of a second, more extensive mode of the Word Acquisition Module (WAM2), which operates off-line to allow periodic lexicon update by the System Administrator.

The processing of unknown syntactic material is fully integrated into the syntactic parser. This module constructs parse fragments using the same grammatical principles as the normal syntactic parser but allowing output of other than complete sentences. The semantic rules can then operate on these parse fragments, as well as on complete parses, to extract meaningful data. The function of the Parse Unexpected Input (PUX) module is to record whether the syntactic and semantic parses obtained for an individual sentence are partial or complete, which contributes to the evaluation of how successfully the system has processed the message.

At the discourse level, partial as well as complete semantic parses can be searched to instantiate the frames, or templates as they are termed in the DBG system, of the internal knowledge representation. The Template Unexpected Inputs Module (TUX) allows certain conditions to be relaxed in applying the rules used for searching the semantic parses, in order to try to fill the internal knowledge representation templates more completely. TUX also records for evaluation purposes the rules relaxed and any semantic cooccurrence anomalies detected. For the MUC-4 task, the PUX and TUX modules were not in operation. The scoring against correctly filled-in MUC-4 templates constituted the evaluation. For other applications, however, a system-generated evaluation of how well the various modules of the system performed is extremely useful. For this purpose, the Self-Evaluation Module (SEM) rates the overall UX processing by the UX Subsystem by combining reports for the other UX modules and numerically rating the accuracy of processing performed by them.

DBG runs on all Sun workstations (including Sun3, Sun4 and Sun386i models) under the SunOS (UNIX) operating system using Quintus Prolog.

FORMAL TESTING OF THE DBG SYSTEM AND EXTENSION TO NEW DOMAINS

We have conducted formal tests of the DBG system on previously unseen messages from two domains, Space Event and Long Range Air. In these tests, the system's performance was measured in comparison both to ideal output and to humans performing essentially the same task as the system -- extracting information from message text and generating application- oriented output templates(*) containing that information. We then collected and evaluated the test data, including the output frames, SEM scores, and the processing time, and analyzed and categorized the system errors. For both domains, the mean percentage scores for correctly filled output vector (an application oriented output structure similar to the MUC templates) slots were above 90%.

SAMPLE DBG PROCESSING FOR MUC-4 MESSAGE TST2-MUC4-0048

In this section, output data structures for the lexical analysis, syntactic parse, semantic parse, knowledge representation modules, and the MUC-4 templates are given to illustrate some of the problems and some of the successes that the DBG system had in processing messages for MUC-4. We will track one sentence through processing, Sentence 13, "a 15-year-old niece of Merino's was injured." In general, the DBG system was successful in extracting information from the message and instantiating the internal DBG knowledge representation for the sentences that it processed. For a variety of reasons, primarily having to do with event merging, the system did not properly pass on and synthesize the information for the MUC-4 templates. It generated too many templates that diffused the appropriate information. This result reflects the state of implementation of the system at the time of testing, rather than the capability of the system as it currently operates and in the long term.

(*) It is important to note that the term "template" in the DBG system is a label for the generic message level semantic and pragmatic representational units, not an application-oriented structure like the MUC templates. It is the glass box output or internal representational output, as opposed to the MUC templates, which are black box outputs mapped to the external representation required by a given application.

In the preprocessing stage, the sentence was selected for processing and identified as possibly containing information pertaining to the results of a critical event on the basis of the word "injured," which is part of the result word list described above. The output of the lexical analysis module is shown in Figure 2. The results of the lexical matching and morphological processing can be seen here. Displayed are the lexical category, (e.g., det, adj, noun, aux, etc.) of the matched item, or its morphological category if morphological processing applied (e.g., past, pastpart); the stem form (which in many instances is the same as the text item); the parse features, if any (e.g., persname, cont(inuous), passive) that are used during syntactic parsing; the subcategorization features (e.g., strict(argp) for 'of', which means that it must be followed by an argument of some kind); tense and agreement features for auxiliaries (e.g., '+agr', '-past'); and links into the concept hierarchy (e.g., family-member, for 'niece'). Other features in the lexicon, such as the selectional restrictions are not displayed in the data structure, but can be checked.

```
1    lxi(det,a,a,[],[],[],[],[strict(qp,ap,np)],[],[],[art])
2    lxi(adj,'15-year-old','15-year-old',[],[],[],[],[strict(np)],[],[],['15-year-old'])
3    lxi(noun,niece,niece,[],[],[],[],[opt(genp)],[],[],[family_member])
4    lxi(of,of,of,[],[],[],[],[strict(argp)],[],[],[of])
5    lxi(noun,merino,merino,[persname],[],[],[],[],[],[],[surname])
6    lxi(noun,'''s',''''s',[],[],[],[],[],[],[],['*thing*'])
7    lxi(aux,was,was,[cont],[],[],[],[xp('-agr','-past')],['+agr','+past'],[],[was])
     lxi(aux,was,was,[passive],[],[],[],[xp('-agr','+past')],['+agr','+past'],[],[was])
     lxi(past,was,be,[],[],[],[],[strict(pred(adj,np),pp)],['+agr','+past'],[],[be])
8    lxi(adj,injured,injured,[],[],[],[],[opt(np,infp)],[],[],[injured])
     lxi(past,injured,injure,[],[],[],[],[strict(np)],['+agr','+past'],[],[injure])
     lxi(pastpart,injured,injure,[],[],[],[],[strict(np)],['-agr','+past'],[],[injure])
```

Figure 2. Lexicalization for Sentence 13 of Message TST2-MUC4-0048

A representation of the syntactic parse for Sentence 13 is shown in Figure 3. In this parse, the verb is determined to be passive, as is shown by the creation of a voice node (Vcemax). The subject of the sentence is identified as the object of the verb, which is evidenced in the '+3' index on the Dmax node dominating the subject of the sentence as well as the trace (i.e., *empty*) constituent under the Dmax node that serves as the object of the verb 'injured.'

```
Cmax(Cbar(C,
       Imax(Dmax+3(Dbar(D([a]:det),
                    Amax(Abar(A(['15-year-old']:adj),
                          Nmax(Nbar(N([niece]:noun),
                                Genmax(Genbar(Gen([of]:of),
                                        Nmax(Nbar(N([merino]:noun_name),
                                              Nmax(Nbar(N(['''s']:noun))))))))))))),
          Ibar(I(+agr,
                 +past),
             Vcemax(Vcebar(Aux([was]:aux),
                       Vmax(Vbar(V([injured]:pastpart),
                             Dmax+3(Dbar(D(*empty*))))))))))).
```

Figure 3. Syntactic Parse for Sentence 13 of Message TST2-MUC4-0048

The semantic parse is shown in Figure 4. In this data structure, the data elements are labeled syntactically and semantically, and indexing displays the constituent structure of the sentence. Also, the tense and voice of the sentence are identified.

```
fp297:
’MAINPRED’(’9.0’) =                       ’INDEX’(’9.1’)
’SUBJECT’(’9.1’) =                        ’INDEX’(’9.2’)
’DETERMINER’(’9.2’) =                     a
’ADJECTIVE MODIFIER’(’9.2’) =             ’15-year-old’
’NOUN/FAMILY_MEMBER’(’9.2’) =             niece
’GEN PHRASE’(’9.2’) =                     ’INDEX’(’9.3’)
’GEN OBJECT’(’9.3’) =                     ’INDEX’(’9.4’)
’NOUN QUALIFIER/PERSON’(’9.4’) =          merino
’NOUN/*THING*’(’9.4’) =                   ’’’s’
’TENSE’(’9.1’) =                          ’PAST’
’VOICE’(’9.1’) =                          ’PASSIVE’
’RESULT’(’9.1’) =                         injure
```

Figure 4. Functional (Semantic) Parse for Sentence 13 of Message TST2-MUC4-0048

Portions of the internal knowledge representation (DBG templates) of the second bombing are shown in Figure 5. According to the discourse rules, the attack with explosives in Sentence 11 of the message caused an "explode" template to be generated as the "event_parent" of an "attack" template in the knowledge representation, and both of these have the same agent ("guerrillas"), patient ("Merino’s home), and location (San Salvador). This results in the generation of a bombing MUC-4 template the physical target of which is Merino’s home (see Figure 5). The result template "injure" is not properly tied in to the explode/attack event, and so does not appear in the MUC-4 template.

```
Event             explode  [1.8]

e_quant:          1
agent:            [1.9.1]
patient:          [1.9.2]
location:         [1.9.3]
loc_qualifier:    in
completion:       PAST
definiteness:     indefinite
event_parent:     [1.8]
event_child:      [1.9]


Event             attack  [1.9]

e_quant:          1
agent:            [1.9.1]
patient:          [1.9.2]
location:         [1.9.3]
loc_qualifier:    in
completion:       PAST
definiteness:     indefinite
event_parent:     [1.8]
```

Entity	Agent [1.9.1]
class:	human
type:	HUMAN
position:	guerrilla
position_text:	guerrillas
quantifier:	PLURAL
definiteness:	indefinite

Entity	Patient [1.9.2]
type:	RESIDENCE
subtype:	*THING*
description:	home
description_text:	merino 's home
quantifier:	1
definiteness:	indefinite

Entity	Location [1.9.3]
city:	san salvador
country:	el salvador
type:	CITY
description:	san salvador
description_text:	san salvador

Result	injure [1.10]
e_quant:	1
patient:	[1.10.1]
completion:	PAST
definiteness:	indefinite

Entity	Patient [1.10.1]
type:	FAMILY_MEMBER
subtype:	PERSON
description:	niece
description_text:	15-year-old niece of merino's
qualifier:	15-year-old
quantifier:	1
definiteness:	indefinite

Figure 5. DBG Templates (Internal Knowledge Representation) of a Portion of Message TST2-MUC4-0048

The MUC-4 output template for the second bombing in Message TST2-MUC4-0048 is shown in Figure 6. Because the event merging processes for the MUC-4 domain were not yet implemented at the time of testing, we generated seven templates for Message TST-MUC4-0048, whereas only two were required. Because the events were not merged properly, some of the information that we extracted correctly was output to other MUC-4 templates than the two bombing templates that were the closest match to the correct templates for the scoring program. For Message TST-MUC4-0048, we did in fact generate templates for the two bombings reported in the message. Unfortunately, the name and description of the human target ended up in spurious attack templates. Also, some information that came through in the internal knowledge representation did not appear in the MUC-4 templates because it was not properly linked to other information in the knowledge representation. A case in point is the injury of Merino's 15-year-old niece, which is represented in templates [1.10] and [1.10.1], but is not linked to other template information and did not appear in any MUC-4 template. The Template Unexpected Input (TUX) Module described above has the capability of using means other than the normal links to determine the role of unused data. Unfortunately, the TUX Module was not yet integrated into the version of the DBG system that processed the MUC-4 test messages. On the other hand, the physical target, "Merino's home", was filled in correctly.

```
0. MESSAGE: ID                          TST2-MUC4-0048
1. MESSAGE: TEMPLATE                    6
2. INCIDENT: DATE                       - 19 APR 89
3. INCIDENT: LOCATION                   EL SALVADOR: SAN SALVADOR (CITY)
4. INCIDENT: TYPE                       BOMBING
5. INCIDENT: STAGE OF EXECUTION         ACCOMPLISHED
6. INCIDENT: INSTRUMENT ID              -
7. INCIDENT: INSTRUMENT TYPE            BOMB
8. PERP: INCIDENT CATEGORY              STATE-SPONSORED VIOLENCE
9. PERP: INDIVIDUAL ID                  "GUERRILLAS"
10. PERP: ORGANIZATION ID               -
11. PERP: ORGANIZATION CONFIDENCE       REPORTED AS FACT
12. PHYS TGT: ID                        "MERINO 'S HOME"
13. PHYS TGT: TYPE                      FACILITY: "MERINO 'S HOME"
14. PHYS TGT: NUMBER                    1: "MERINO 'S HOME"
15. PHYS TGT: FOREIGN NATION            -
16. PHYS TGT: EFFECT OF INCIDENT        SOME DAMAGE: "MERINO 'S HOME"
17. PHYS TGT: TOTAL NUMBER              -
18. HUM TGT: NAME                       -
19. HUM TGT: DESCRIPTION                -
20. HUM TGT: TYPE                       -
21. HUM TGT: NUMBER                     -
22. HUM TGT: FOREIGN NATION             -
23. HUM TGT: EFFECT OF INCIDENT         -
24. HUM TGT: TOTAL NUMBER               -
```

Figure 6. MUC-4 Output Templates for Message TST2-MUC4-0048

We believe that in the DBG system we have the capability to solve the problems that we encountered in the MUC-4 test and to perform efficiently a deeper level of text analysis than that required for MUC-4. LSI's goal remains the construction of a true text understanding system. Our continuing long-range research goals in the areas of syntactic parser development, lexical/semantic development, and discourse processing have contributed, and will continue to contribute, to the success of this effort.

REFERENCES

[1] Banfield, Ann. Unspeakable Sentences: Narration and Representation in the Language of Fiction. Boston: Routledge & Kegan Paul. 1982.

[2] Givon, Talmy (ed.) Topic Continuity in Discourse. Amsterdam: Benjamins. 1979.

[3] Montgomery, C.A., Glover Stalls, B., Belvin. R.S., and R.E. Stumberger "Language Systems, Inc.: Description of the DBG System as Used for MUC-3" in 'Proceedings, Third Message Understanding Conference (MUC-3)' pp. 171-177, Defense Advanced Research Projects Agency (DARPA), Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1991.

[4] Rapaport, William J. "Logical Foundations for Belief Representation," Cognitive Science 10: 371-422. 1986.

[5] Rapaport, William J., Erwin M. Segal, Stuart C. Shapiro, David, A. Zubin, Gail A. Bruder, Judith F. Duchan, David M. Mark. "Cognitive and Computer Systems For Understanding Narrative Text." Technical Report 89-07, Department of Computer Science, SUNY Buffalo. 1989.

[6] Sidner, Candace L. "Focusing in the Comprehension of Definite Anaphora," in M. Brady & R. Berwick (eds.) Computational Models of Discourse. Cambridge, MA: MIT Press, 267-330. 1983.

[7] Webber, Bonnie L. "The Interpretation of Tense in Discourse," Proc. 25th Annual Meeting of the Association for Computational Linguistics (Stanford Univ.). Morristown, NJ: Association for Computational Linguistics, 147-154. 1987.

[8] Wiebe, Janyce M. & William J. Rapaport. "A Computational Theory of Perspective an Reference in Narrative," Proc. 26th Annual Meeting of the Association for Computational Linguistics (SUNY Buffalo). Morristown, NJ: Association for Computational Linguistics, 133-138. 1988.