LANGUAGE SYSTEMS, INC.
DESCRIPTION OF THE DBG SYSTEM AS USED FOR MUC-3

Christine A. Montgomery
Bonnie Glover Stalls
Robert S. Belvin
Robert E. Stumberger

Language Systems, Inc.

6269 Variel Avenue, Suite F
Woodland Hills, CA 91367
lsi001%lsi-la.uucp@ism.isc.com
1-818-703-5034

INTRODUCTION

LSI's Data Base Generation (DBG) approach to natural language understanding is characterized by three main features: First, the DBG system is comprehensive. It performs full-scale lexical, syntactic, semantic, and discourse analyses of the entire message text being processed and produces a complete knowledge representation of the text. Second, the DBG system is modular and flexible. The modular and transparent system architecture ensures easy extension, maintenance, and upgrading of the system. And, third, the DBG system is generic but at the same time domain-sensitive. It applies domain modeling to text interpretation, which enables the extension of the system to any number of new domains. In addition, it provides a powerful capability for handling unknown data in familiar domains. DBG's development has been based on analysis of large volumes of message traffic (thousands of Air Force and Army messages) in five domains, as described below. The system can currently process a large number of messages in each of these domains and has been formally tested on previously unseen messages in three of these, with competitive tests against humans performing the same task in two domains. The functional flow of the DBG system is shown in Figure 1 (actually Figure 1 of our site report [Language Systems Inc: MUC-3 Test Results and Analysis] in this proceedings).

THE DBG APPROACH TO NATURAL LANGUAGE PROCESSING

Foundational aspects of our approach include the use of frame hierarchies based on principles of mathematical logic for the knowledge representation; the incorporation of elements of discourse structure using insights on narrative structure from linguistics, anthropology, and sociology; and a multifunctional integrated unexpected inputs (UX) subsystem to deal with unknown input and that in addition grades the system on its performance. More recently, we have developed a bottom-up parser based on principles of government-binding (GB) theory, and a flexible mechanism for integrating and distributing lexical and semantic information. Several of these aspects have anticipated developments in the field of natural language understanding, whereas others, such as the (UX) subsystem, are, as far as we know, original with us. Other key aspects of DBG, for example, the incorporation of the sublanguage approach first defined by Zellig Harris in [5] and further developed by Naomi Sager and the NYU Linguistic String Project in [10], have made use of existing specialized natural language understanding techniques to solve the particular problems that we have faced, such as the challenge of building a generic system that could process messages from a variety of specialized military domains.

From a conceptual linguistic perspective, our system is principle-based. This is most obvious in the sentence processing mechanisms (versus mechanisms employed in processing larger units of language), wherein we rely heavily, though not exclusively on recent work in the Government and Binding grammatical framework ([2] and [3]). Underlying our system design is a conviction that there is a strong isomorphism between syntactic structure and semantic composition. The system attempts to take maximal advantage of this isomorphism to produce greater comprehension and efficiency in processing. An example of a parse-tree built using GB-based principles

is shown in Figure 2.

This isomorphism requires a strong linkage between the lexical/syntactic and semantic knowledge for words and phrases. We have created an external representation for words and concepts which encodes lexical, syntactic and semantic knowledge in a single structure. This representation allows an application developer to concisely express the knowledge required by the system during syntactic and semantic processing. The representation is read into the DBG system by a mechanism which formulates and distributes entries to the appropriate database (lexical/syntactic or semantic), linking corresponding lexical and semantic entries. During processing, the links between words and their conceptual representations allow the system to validate the semantic "correctness" of a word's attachment into the parse tree. An example of the translation from external to internal frame/lexicon entries is shown in Figure 3.

DBG SYSTEM ARCHITECTURE

The system architecture reflects the approach described above. From the outset, the DBG system was created to handle actual messages. As the core system was ported to new applications, with new domains and messages, enhanced capabilities were usually required. These capabilities were added to the core system, thus providing us with a steadily improving system with increased functionality and robustness. 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 four years. 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. The present major redesign of the parser, accomplished in the course of the MUC-3 effort, has involved the redistribution of processing tasks and re-integration of information shared among four of the main system modules (lexical, syntactic, semantic, and knowledge representation), however the basic system architecture has remained the same.

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. The system is organized such that the output data structure generated by each module serves as input to the succeeding module, and is then available to all later modules. The individual modules contain domain-independent processing mechanisms as well as rule sets that allow the incorporation of domain-sensitive features, which aid in processing and in many cases are essential for the correct interpretation of the message. The functional flow of the system is illustrated in Figure 1 (which is actually Figure 1 of our site report [Language Systems Inc: MUC-3 Test Results and Analysis] in this proceedings).

In processing, the message is first extracted from the message stream and the message text is segmented into distinct words and sentences. Successive lexical, syntactic, and semantic modules then analyze the individual sentences. In each sentence, the lexical definitions of the words and multi-word phrases are found in the lexicon (or derived from Unexpected Input processing, as described below), yielding a lexicalization for the sentence. The lexicalization is then passed to the Government Binding-based parser.

The parser mechanism works by projecting incoming words to maximal X-bar projections (three-level node-graphs), examining successive node pairs, performing various syntactic and semantic checks, and then attaching valid node pairs. The parse structure which is built up through these attachments is represented as an acyclic, directed graph. The mechanism 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 attaches theta-role information (similar to case frames) to properly attached verb-argument nodes.

The parse structure/graph for a sentence is then passed to the functional 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 semantic hierarchy.

At the final stage, the sentential semantic parses of a message are searched for 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 in particular domains and the relations of these events and entities to one another. In particular, the hierarchical organization of these frames enables the explicit representation of the relations of various event types to one another (i.e., domain events and meta-events,

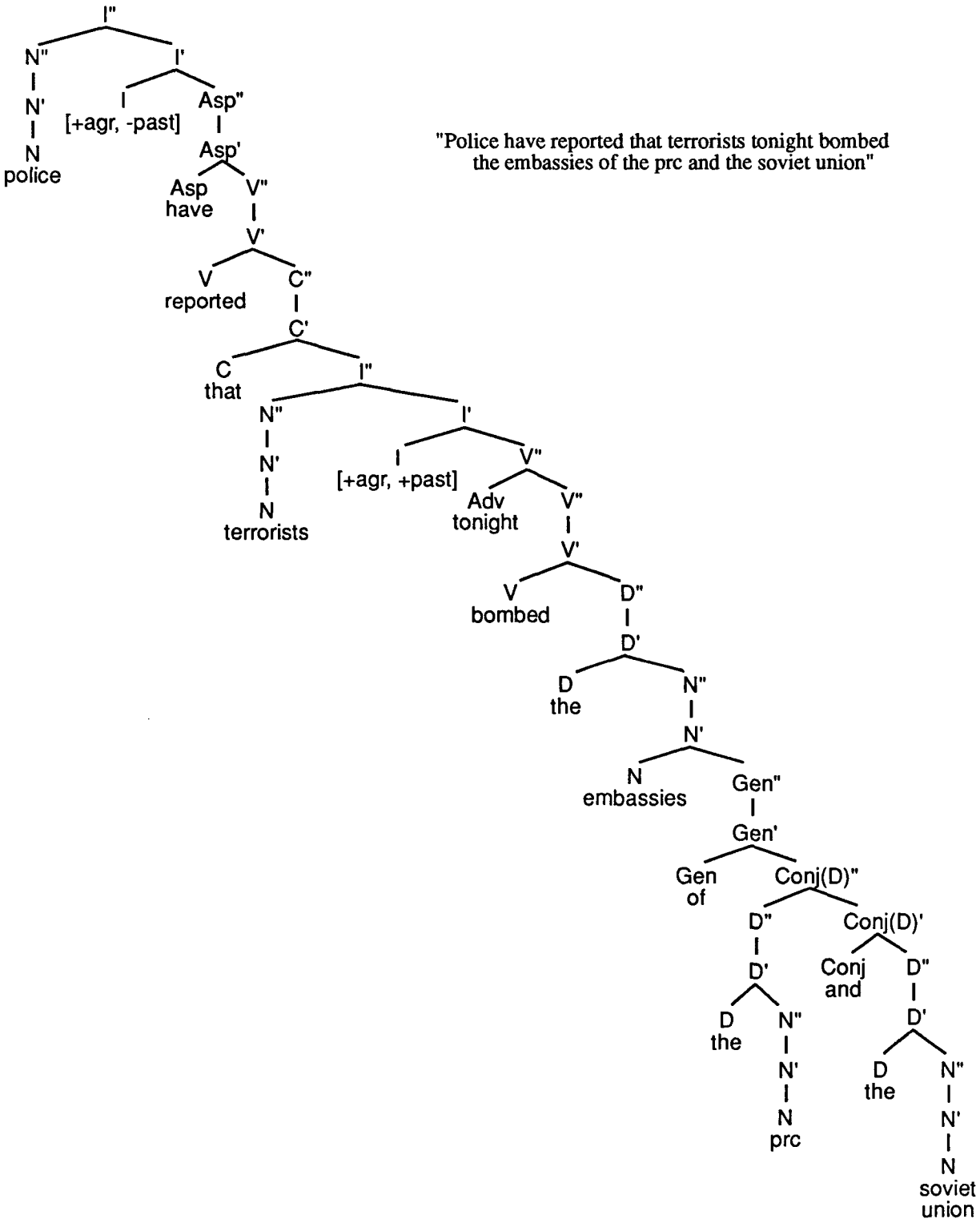

Figure 2: Parse Tree for Test Set 1 - Message 99 - Sentence 1

FLEX (Integrated Frame/Lexicon Definition)

Representation
and
Translation to Frame and Lexicon Databases

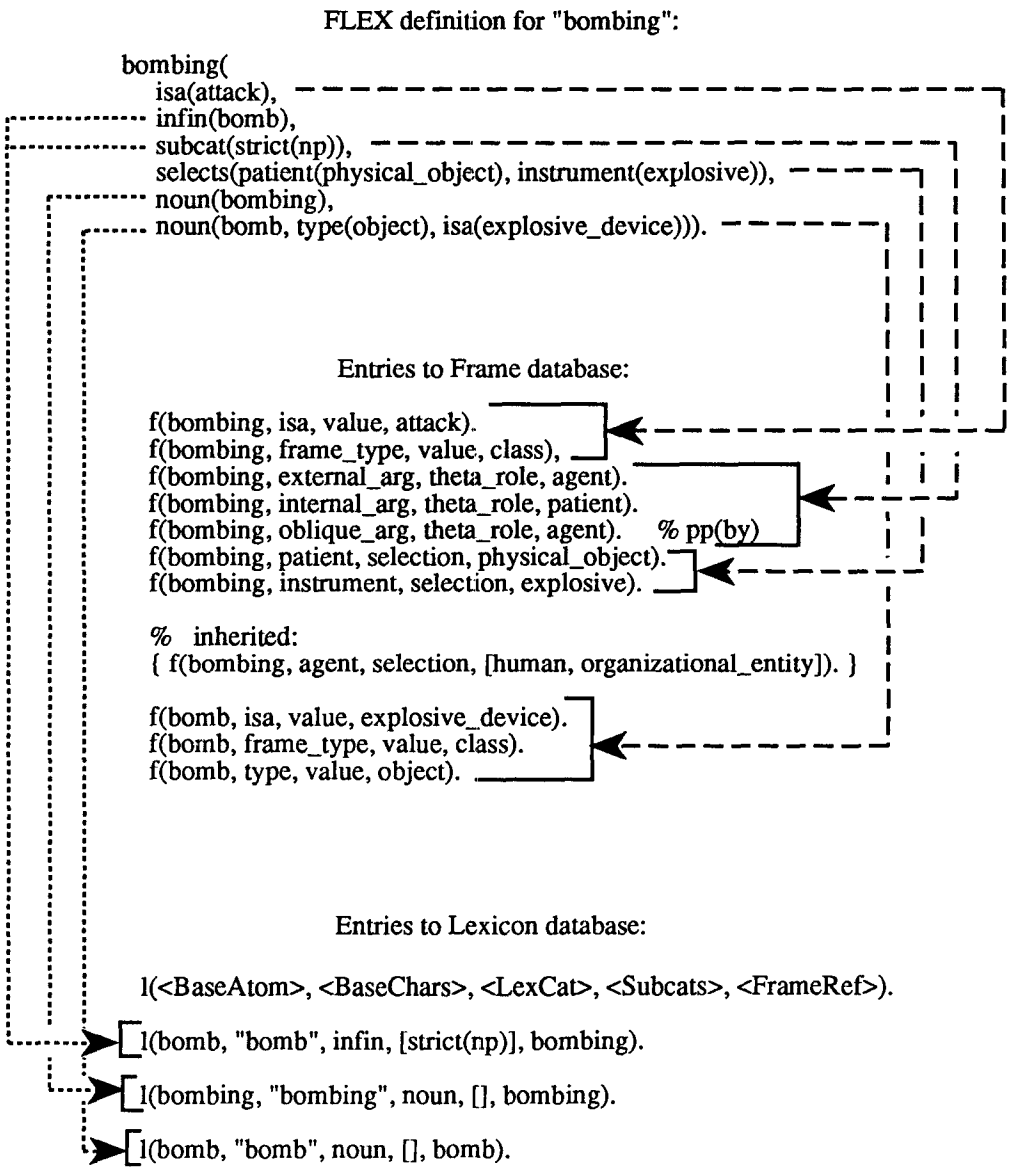

Figure 3: Sample FLEX Entry

described in [6]). Domain information is stored in a frame subsystem and information implicit to the message is provided by a mechanism of inheritance built into the frame subsystem. The system thus has the capability of incorporating domain information not explicitly contained in the message, thus representing a deeper understanding of the message text.

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.

Unknown syntactic material is processed by the Parsing Unexpected Inputs processor (PUX). This module constructs parse fragments using the same syntactic grammar rules as the normal syntactic parser but allowing output of other than complete sentences. The semantic rules can then operate on these parse fragments to extract meaningful data. At the discourse level, the Template Unexpected Input module (TUX) searches for expected information missing in the final output knowledge representation from among leftover or unused semantic information. Since such information can include unidentified strings -- e.g., the name of a new terrorist group in the MUC-3 domain -- TUX provides a means for recognizing unknown proper names and specifying their function in a text. Finally, the Self-Evaluation Module (SEM) rates the overall processing by the UX Subsystem by combining reports for the other UX modules and numerically rating the accuracy of processing performed by them.

Due to the close integration of syntactic and semantic checking required by the parser, a facility is also provided which reads integrated frame/lexicon representations (human writable/readable) and converts them into entries for the system-internal lexicon and frame databases. This mechanism ensures that lexical entries containing syntactic data are properly linked to frame entries containing semantic data. As mentioned before, an example of the translation from external to internal frame/lexicon entries is shown in Figure 3.

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-3 templates) slots were above 90%. The results of these tests appear in [7] for the Space Event domain and in [11] for Long Range Air.

(*) 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-3 templates. It is the glass box output or internal representational output, as opposed to the MUC-3 templates, which are black box outputs mapped to the external representation required by a given application.

The five domains to which the DBG system has so far been applied are subdomains of the Air Force, Army, and Navy. Although these domains have in common the fact that they are military, the event and object types of the individual domains exhibit considerable variation and the message structures are of correspondingly varying degrees of predictability. Ranged along an axis of predictability, the domains are:

<-- More predictable ---------- Less predictable -->

(1)	(2)	(3)
Space Event	AirLand Battle Management	Ft. Riley/voice
Long Range Air	MUC-2/Navy	

The three degrees of predictability shown here correspond to characteristics of the domain. The most predictable type, the Space Event domain, has the most limited set of event and object types, i.e., the launch, orbit, and deorbit of certain satellites. Long Range Air is similarly structured, although somewhat less limited, with a certain number of airplane types that can engage in several types of events (e.g., flying, refueling, performing various missions, taking off, landing). Less predictable are the main battle events that take place on land and sea; these are described in the messages of the AirLand Battle Management corpus and the MUC-2 naval corpus. Least predictable are the ongoing events at the company and platoon levels, especially when these are described in terms of spoken, rather than written message traffic. The Ft. Riley voice corpus, which consists of lower echelon voice communications collected during the four days of a training exercise, exemplifies this degree of predictability.

The medium of the message is also important. The messages from four of the domains are written. The headers of the written messages are typically formatted and provide information about the message source, distribution and type; however, the main body of the message is free text, in some cases containing tabular data as well. The message text typically contains at least three different kinds of discourse--title sentences (comparable to the telegraphic style of newspaper headlines), reports of events in the domain, and analysis of those events. We have described the role and different properties of these various kinds of discourse in [8].

In the fifth domain, the message corpus consists of transcribed Army radiotelephone dialogs from field-training exercises. This voice data is highly unpredictable and complex to analyze. The message equivalent in this corpus is the dialog, defined as continuous conversation between the same speaker/hearer(s), usually on a particular topic. A major task in processing this corpus is to locate relevant dialogs and synthesize the information within them.

Message structure also differs in these domains according to whether it is event-driven or topic-driven, or both. Event-driven messages (Space Event and Long Range Air) are structured narratives of specific event types; generally speaking, no message is sent unless a particular event of that type (e.g., a satellite launch) occurs. The meaningful discourse unit for this type is the paragraph. Topic-driven (AirLand Battle Management and MUC-2) are usually periodic status reports that have labeled or numbered portions of text with predefined general topics (e.g., current location of forward elements); a context is established but the text is less predictable. The main meaningful discourse unit is the sentence; sentences within the same paragraph may or may not be related. Conversation, in addition to its other distinctive properties, combines the two other types. The Ft. Riley corpus of transcribed voice data contains both event-driven (i.e., information about the battle as it unfolds) and topic-driven (e.g., periodic spot reports) types of information, as well as a great deal of less predictable conversational material. (This corpus is described in [1], [4], and [9]).

In more recent work, we are attempting to exploit more fully the notion of text grammar or discourse modeling at all processing levels as well as in extending the system to new domains, such as the terrorist incident messages of the MUC-3 corpus. A text grammar, following van Dijk [12], is a semantic and pragmatic model of discourse. It specifies how domain information is expressed within the discourse at the phrase, sentence, paragraph, message, and even intermessage levels. An example of a text grammar rule at the sentence level is the following content rule for title sentences in launch messages from the Space Event corpus (elements in parentheses are optional):

Sentence[title] --> Object (Booster) (Action) Launchsite (Time) Date

This information may be actualized in the title sentence of a message in a variety of lexicosyntactic configurations. Knowledge and expectations concerning the kinds of information being processed at a certain

point in the message can be crucial in efficiently processing and accurately representing that information and in filling in gaps where there is new or erroneous information that is not understandable by other means (in the case of DBG, by the UX subsystem processing). In the DBG system, it can direct template(*) generation and filling and interpret unexpected input, as well as tracking possible antecedents for anaphoric references.

Currently, each application of DBG contains rules using slightly different strategies to generate and fill templates, depending on the various properties, as described above, of the domain, the domain sublanguage, and the message type. We envision comprehensive high-level domain-sensitive text grammar rules, selected from a generic set of options, that would direct template-generation and filling and could be used to extend the system to an entirely new domain. Because of the major redesign of the core system modules which is in the process of being implemented and tested, we have not yet incorporated this more global model of text grammar into our system. However, the flexibility and modularity of the DBG system makes such an approach feasible, and MUC-3 provides a fertile ground for further development of DBG and for testing a more comprehensive text grammar approach to message processing.

REFERENCES

[1] Belvin, R.S., Holmback, H.K., Montgomery, C.A., Stalls, B.G., and Stumberger, R.E., "Message Fusion: Final Report", (CDRL A003, CLIN 0002, Contract No. DAAA15-85-C-0115, Ballistic Research Laboratory), Logicon Inc., Operating Systems Division, OSD-W-R88-06, 1989.

[2] Berwick, Robert C., "Principle-Based Parsing", Technical Report 972, MIT Artificial Intelligence Laboratory, MIT, Cambridge, MA, 1987. Burge et al [1989]

[3] Chomsky, Noam A., "Lectures on Government and Binding", Foris, Dordrecht, 1981.

[4] Glover, B. C., and C. A. Montgomery, "Message Fusion: The Maneuver Sublanguage", in 'Proceedings of the 1986 Conference on Intelligent Systems and Machines', pp. 193-198, Oakland University, Rochester, MI, 1986.

[5] Harris, Z. S., "Mathematical Structures of Language", Wiley (Interscience), New York, 1968.

[6] Montgomery, C.A., "Distinguishing Fact from Opinion and Events from Meta-Events", in 'Proceedings of the Conference on Applied Natural Language Processing', pp. 55-61, Santa Monica, CA, 1983.

[7] Montgomery, C.A., Burge, J., Holmback, H., Kuhns, J.L., Stalls, B.G. (Glover), Stumberger, R, and R. L. Russel Jr., "The DBG Message Understanding System", in 'Proceedings of the Annual AI Systems in Government Conference', pp. 258-265, Computer Society of the IEEE, Washington, D.C., 1989.

[8] Montgomery, C. A., and B. C. Glover, "A Sublanguage for Reporting and Analysis of Space Events," in 'Analyzing Language in Restricted Domains: Sublanguage Description and Processing', R. Grishman and R. Kittredge (eds.), pp. 129-161, Lawrence Erlbaum Associates, Hillside, NJ, 1986.

[9] Montgomery, C. A. and B. C. Glover, "A Text Grammar for Automated Analysis of C3I Messages", in 'Proceedings of the 1984 Conference on Intelligent Systems and Machines', pp. 347-352, Oakland University, Rochester, MI, 1984.

[10] Sager, N., "Natural Language Information Processing: A Computer Grammar of English and Its Applications", Addison-Wesley, Reading, MA, 1981.

[11] Stalls, B.G., Stumberger, R. and Montgomery, C. A., "Long Range Air (LRA) Data Base Generator (DBG) Final Technical Report", RADC-TR-89-366, Rome Air Development Center, Griffiss Air Force Base, Rome, NY, 1990.

[12] van Dijk, T. A., "Some Aspects of Text Grammar", Mouton, The Hague, 1972.