LANGUAGE SYSTEMS INC: DESCRIPTION OF THE DBG SYSTEM AS USED FOR MUC-5[1]

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

Language Systems, Inc.
6269 Variel Avenue, Suite F
Woodland Hills, CA 91367
(818) 703-5034
Internet: muc-5@lsi.com

INTRODUCTION

Language Systems, Inc. (LSI) believes that the best system for producing a complete and accurate automated analysis of natural language text is an in-depth text understanding system that employs linguistic as well as other analytical techniques to interpret the text. Our DBG (Data Base Generation) natural language processing system performs full-scale linguistic analysis of text in order to produce a system-internal text-level representation of the content of the text. This representation is composed of a set of entity and event frame structures, interrelated to reflect the organization and content of the text. This representation of the text can then be mapped into any data structure required by a downstream application, such as the templates specified for the MUC-5/Tipster applications. DBG has been designed as a single core system for handling texts of different types in different domains for a variety of applications. Application types for which DBG has provided the input include information extraction and database generation tasks such as MUC-5, message fusion (the combination of information derived from various kinds of sources, including text; see [1]), and the translation of text into another language using spoken input and output ([2]).

It is clear to us that our DBG system, while achieving MUC-5 results comparable to EME scores attained by Tipster contractors in the MUC-5 tests, is still far from achieving the level of performance that we believe is possible for it. LSI's official MUC-5 P&R score for English Microelectronics (EME) texts was 42.74, which represents a considerable improvement over our MUC-4 P&R score of 18.87 on TST3. As we continue to incorporate improved versions of the various components of our natural language processing system (Figure 1) and to exploit more fully the capabilities of existing components, we expect that our scores will continue to improve.

Founded upon research performed over the last twenty years, the DBG system has been under actual development for the last seven years. The basic architecture of the core system has remained the same over that time. Because the system is modular, the individual modules can be redesigned and updated without affecting the rest of the system. Most of the current modules have been redesigned or extended within the last three years.

[1]Support for MUC-5 final testing was provided by the Army Research Laboratory/SLCBR-SE-C, under Contract No. DAAA15-89-C-0004 (Subcontract No. 05-562-01 to Logicon, Inc.) Also, we gratefully acknowledge the assistance of Andrew Brislen and Michael Possedi from Sun Microsystems, and Carrie Du Bois from Quintus, during the final testing period.

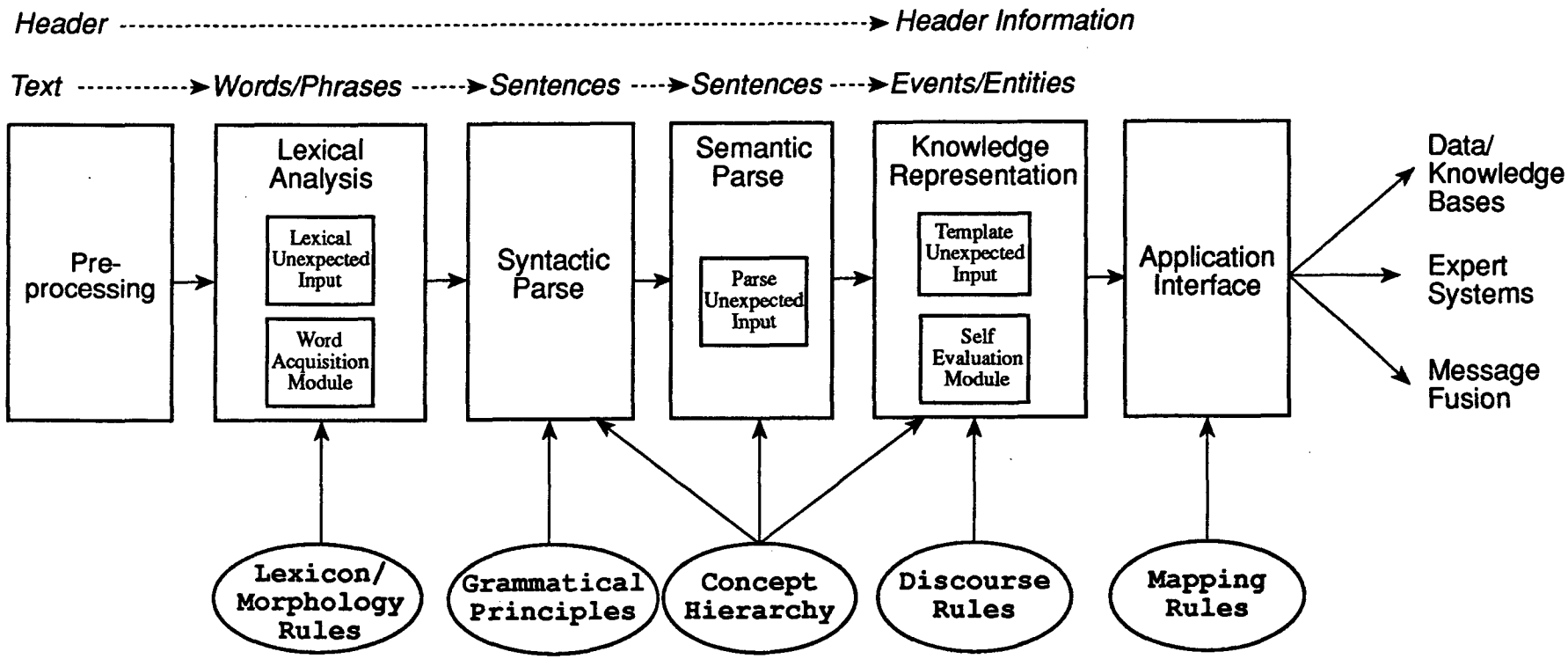

Figure 1: The DBG Message Understanding System

INNOVATIVE ASPECTS OF THE SYSTEM

Among the innovative aspects of the DBG system as used for MUC-5 are a flexible frame-based concept hierarchy that is accessible at every stage of processing; a principle-based syntactic parser which enables the identification of sentence-level event-entity relations very early in processing; an integrated ability to handle incomplete information and structures; and the frame-based text-level representation mentioned above which allows for intersentential reference resolution and for the explicit representation of the event-entity relations and implicit content of the text. These relations were then easily mapped into EME templates.

LSI's main area of development for MUC-5 was the extension of the knowledge contained in the concept hierarchy and the use of this knowledge at every stage of processing. Although the frame-based hierarchy was a component of the DBG system as early as MUCK-2, it was not exploited significantly until MUC-5. This, more than any other factor, was responsible for the increase in our performance over MUC-4. Each item in the lexicon is associated with a concept frame, as illustrated in Figure 2. The frames or concept nodes bear "isa" (set membership) relations to other concepts in the hierarchy. During lexical analysis, the lexical items in the text are linked to concepts in the hierarchy. These concept links then provide the framework for producing the set of instantiated concept frames and links (the frame-based text-level representation) which is the final output of internal DBG processing. The frame-based concept hierarchy also allows for semantic checking at any point in processing and provides a mechanism for the inheritance of features and other information to direct descendants in the hierarchy.

A related innovative DBG component is the Word Acquisition Module (WAM), which uses morphological analysis to provide grammatical category information for words for which no lexical entry exists. Based on the category assignment, a lexical entry and an "isa" link to a concept frame are automatically generated for each unknown item to allow complete processing of all sentences containing unknown words.

The DBG syntactic parser as implemented for MUC-5 has a number of innovative aspects. The parser is to a great extent language-independent; it produces structures which reflect the partial isomorphism holding between syntactic and semantic structures in order to increase accuracy of processing. While the goal is to produce complete parses, the parser is also robust enough to produce usable partial parses in the absence of a complete parse.

The design of the LSI parser is based on the Government-Binding theory of syntax. It is essentially a head-driven parser, and includes both bottom-up and expectation-based aspects. Argument structures associated with lexical items are projected into the syntax. Syntactic structure is determined from both item-specific lexical requirements as well as general requirements on syntactic structures (e.g., all sentences in languages like English require a subject). The use of empty categories and syntactic chains, combined with knowledge of event types contained in the concept hierarchy, enables the parser to associate thematic roles with entities expressed in noun phrases in a variety of construction types correctly and in a relatively straightforward manner. Constructions which are usually assumed to include empty categories (i.e., phonetically null syntactic elements) include passives, embedded infinitival sentences, questions, and relative clauses. The rationale for this assumption is that in constructions of this type, words (usually verbs) which typically require either an external argument (an argument in the specifier position), or an internal argument (an argument in complement position) appear with no appropriate argument in one of these positions. Since syntactic structures are characterized as "projections" of lexical items, these positions are assumed to be latently present, and linked to a phonetically realized argument in some other position via coindexing. The phonetically realized argument and the empty category thus form a syntactic "chain". By using these chains, we can associate the usual thematic roles assigned to certain positions with a given overt noun phrase, even if the overt phrase is not in the usual position. So adherence to certain grammatical principles in conjunction with a well-engineered event knowledge base has enabled us to get something "for free", as it were, in the MUC-5 task.

Several of the innovative aspects of the DBG frame-based text-level knowledge representation, which we will here call the LSI templates (as distinct from the MUC-5 application-oriented EME templates) were not used extensively for MUC-5. These include the ability to combine contextual information – e.g.,

lithography

```
%%   "lithography" is a process following layering, where the
%%   pattern of a circuit is rendered onto the layered surface.
%%   It is done by exposing the surface to certain types of
%%   light/radiation.
%
%*lithography*
%    isa: microelectronics_process
%    end_product: *device*
%    process_granularity: device_structure_granularity
%    process_equipment: microelectronics_equipment

'*lithography*'(
    isa(microelectronics_process),
    slotorder(name, definiteness, quantity, type, description,
              process_product, end_product, process_material,
              process_actions, process_conditions,
              process_equipment, process_granularity, device,
              'fp name', head_fp_id, display(no)),
    muc5_vct(m5_litho),
    muc5_type('UNKNOWN'),
    related_equipment(lithography_system),
    type(specifier('*lithography*')),
    noun(lithography),
    adj(lithographic)).
```

silicon_oxide

```
%silicon_oxide
%    isa: insulator_film
%    comment: An insulator film resulting from the exposure of
%    the silicon surface in oxidation, chemical vapor
%    deposition, or sputtering.

silicon_oxide(
    isa(insulator_film),
    chemical_symbol('SiO'),
    muc5_type('SILICON_OXIDE'),
    noun('SiO', mstem(no_plural)),
    noun('silicon oxide')).
```

Figure 2: Sample EME Lexicon Entries

information derived from the header of a military message – with information extracted from the text; the capability of interpreting degree-of-belief information and relating it to meta-levels of event structure; and the incorporation of text grammar expectations as to the form and location of information within the text. Although these aspects of DBG are important in processing written and even transcribed voice messages in the Air Force and Army messages to which the system has been applied, the MUC-5 application does not depend heavily on outside information, and is not concerned with evaluation of the information received, and is characterized by more variability in information structure across texts.

The relational organization of the DBG event and entity templates, however, was extremely useful in MUC-5 processing. Because it resembles the object-oriented organizational structure of the EME templates, the generation of EME template relations was relatively straightforward. Also, the thematic roles of the entity templates in relation to the events is explicit in the LSI templates and so can be easily associated with the role-specific slots (e.g., manufacturer, distributor) in the EME templates. In addition, the ability to establish co-reference at the text level in the LSI templates prevents overgeneration of EME templates and facilitates the correct template linkage.

In the next section, the basic processing is described and illustrated by an example sentence.

SYSTEM MODULES AND PROCESSING STAGES

Because the DBG system modules and the processing of text have been previously described in detail ([3], [4]), we will here present only a brief summary of processing and then show a short example sentence to illustrate the innovative aspects of our system discussed in the previous section.

The basic components of the DBG system are shown in the system diagram in Figure 1. They are the preprocessing module, the lexical analysis module, the syntactic parse module, the semantic parse module, and the knowledge representation module. An additional module, the application interface, maps the extracted knowledge into appropriate data structures according to the requirements of a given downstream application. Processing is sequential–the output of each module is a data structure that serves as input to the succeeding module and is then available to all later modules. Each module contains a processing mechanism and a knowledge base that includes general as well as domain-sensitive knowledge. The respective knowledge bases, indicated in ovals, are 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. As described in the previous section, at the stage of lexical analysis, the lexical items in the text are linked to nodes in the concept hierarchy, which enables information derived from the concept hierarchy to be used at any point in processing. Sample lexical items are shown in Figure 2.

In addition to the basic components, the DBG system has an Unexpected Inputs (UX) Subsystem that handles new or erroneous data and evaluates and records system performance. This subsystem consists of modules that are integrated into the system modules; they are shown in the system diagram as small boxes inside the larger modules to which they apply. The two UX modules of the DBG system that were used for MUC-5 are the Lexical Unexpected Inputs (LUX) module and the Word Acquisition Module (WAM), both of which apply at the Lexical Analysis stage. The LUX module corrects errors by attempting partial matches between unmatched words in the text and items in the lexicon, using rules based on certain error hypotheses. As mentioned previously, new or unidentified words are passed on to WAM, wherein word class information is assigned based on morphological analysis.

To illustrate the processing, the lexical analysis, syntactic parse, and semantic parse for an example sentence are shown in Figure 3, and the LSI internal templates for the same sentence, are shown in Figure 4. The sentence used is the first sentence of the example text 2606871, i.e., "Hampshire Instruments has sold an x-ray lithography system to AT&T Bell Laboratories."

The syntactic parse is created by projecting lexical items into elementary phrasal trees, which are then linked according to subcategorization and selectional requirements, as well as general principles of grammar.

TEXT OF EXAMPLE SENTENCE

Hampshire Instruments has sold an x-ray lithography system to AT&T Bell Laboratories

LEXICAL ANALYSIS OF EXAMPLE SENTENCE

```
1     lxi(noun,'Hampshire Instruments','Hampshire Instruments',[],[],[],[],
          [],[],[],[hampshire_instruments_inc])
2     lxi(aux,has,has,[perf],[p(3),n(s)],[],[],[xp('-agr','+past')],
          ['+agr','-past'],[],[has])
      lxi(third_pres,has,have,[],[],[],[],[strict(np)],['+agr','-past'],[],
          [have])
3     lxi(past,sold,sell,[],[],[],[],[strict(np,bi_np(to),np_prpt)],
          ['+agr','+past'],[],[sell])
      lxi(pastpart,sold,sell,[],[],[],[],[strict(np,bi_np(to),np_prpt)],
          ['-agr','+past'],[],[sell])
4     lxi(det,an,an,[],[],[],[],[strict(qp,ap,np)],[],[],[an])
5     lxi(noun,'x-ray','x-ray',[],[],[],[],[],[],[],[x_ray])
6     lxi(noun,'lithography system','lithography system',[],[],[],[],[],[],
          [],[lithography_system])
7     lxi(prep,to,to,[],[],[],[],[strict(argp)],[],[],[to])
      lxi(to,to,to,[],[],[],[],[],[],[],[to])
8     lxi(noun,'AT&T Bell Laboratories','AT&T Bell
          Laboratories',[],[],[],[],[],[],[],[a_t_and_t_bell_labs])
```

SYNTACTIC PARSE OF EXAMPLE SENTENCE

```
'Cmax1':
Cmax(Cbar(C,
          Imax(Nmax(Nbar(N(['Hampshire Instruments']:noun))),
               Ibar(I(+agr,
                      -past),
                    Aspmax(Aspbar(Aux([has]:aux),
                                  Vmax(Vbar(Vbar(V([sold]:pastpart),
                                                 Dmax(Dbar(D([an]:det),
                                                           Nmax(Nbar(N(['x-ray']:noun),
                                                                     Nmax(Nbar(N(['lithography system']:noun))))))),
                                            Pmax(Pbar(P([to]:prep),
                                                      Nmax(Nbar(N(['AT&T Bell Laboratories']:noun))))))))))).
```

SEMANTIC PARSE OF EXAMPLE SENTENCE

```
fp1:
  'MAINPRED'('1.0') = 'INDEX'('1.1')
  'SUBJECT/AGENT'('1.1') = 'INDEX'('1.2')
  'NOUN/HAMPSHIRE_INSTRUMENTS_INC'('1.2') = 'Hampshire Instruments'
  'TENSE'('1.1') = 'PRESENT PERFECT'
  'VOICE'('1.1') = 'ACTIVE'
  'PREDICATE/SELL'('1.1') = sell
  'OBJECT/PATIENT'('1.1') = 'INDEX'('1.3')
  'DETERMINER'('1.3') = an
  'NOUN QUALIFIER/X_RAY'('1.3') = 'x-ray'
  'NOUN/LITHOGRAPHY_SYSTEM'('1.3') = 'lithography system'
  'PREPOSITIONAL PHRASE/RECIPIENT'('1.1') = 'INDEX'('1.4')
  'PREPOSITION'('1.4') = to
  'PREP OBJECT'('1.4') = 'INDEX'('1.5')
  'NOUN/A_T_AND_T_BELL_LABS'('1.5') = 'AT&T Bell Laboratories'
```

Figure 3: DBG Analysis of Sample Sentence

Among the principles which are most important in the current version of the parser are the projection principle, which constrains which positions in the tree can be assigned thematic roles, the extended projection principle, which attempts to link every clause to a subject, trace theory and the theta-criterion, which ensure that every argument position has a one-to-one correspondence with a theta-role, and X-bar theory, which limits branching to binary structures following the X-bar schema. Structural characteristics of the tree are then matched with the thematic specifications of the lexical items heading the phrases in the tree.

In the semantic parse, shown in Figure 3, the thematic roles are explicitly labeled and related by indices to the verb 'sell'. The AGENT is 'Hampshire Instruments', the PATIENT is 'an x-ray lithography system', and the RECIPIENT is 'AT&T Bell Laboratories'. Other information, including the tense and voice of the verb, is also given. Before a noun phrase can be assigned a given thematic role, it has to qualify both syntactically and by meeting semantic categorial requirements. This is established through checking the the link into the concept hierarchy of the head noun of the noun phrase and matching it with the selectional information for the verb in the lexicon. All of this is done at the sentence level. Information from the semantic parses of the text is then used to generate and instantiate the LSI templates.

The frame-based text-level data structures (LSI templates) for a text are generated on the basis of the semantic parse of that text and the frame information associated with the lexical items in the semantic parse.

There are three major steps in LSI template generation: 1) the first pass, in which templates for specified events and the related entity templates are generated; 2) the second pass, in which entity templates are generated for MUC-5 relevant words that are not treated in the first pass; and 3) template linking, in which co-reference relations are determined.

An event template includes a set of empty slots which represent the various thematic roles associated with the event. The processing goal is to fill the thematic role slots of each of the event templates with a reference to an entity template. The slots for event and entity templates are pre-defined in our concept hierarchy. For MUC-5, the following event-related thematic roles were handled: agent, patient, experiencer, recipient, beneficiary, source, and location. For example, while attempting to fill the slots of a manufacture event, as in "Sony manufactured a new DRAM", "Sony" will trigger an agent template, and "a new DRAM" will trigger a patient template. The entity template has a pointer to the semantic parse node which triggered it. The slots of an entity template are determined by the entity type. For MUC-5, all the entity templates have at least the following slots: *name, type, quantity, definiteness.* Different types of entities have different additional slots representing the relevant attributes for the given type of entity. For example, company entities will also have the slots for *location* and *nationality*; equipment entities have slots for *model, manufacturer*, and *wafer size*; and so on. A few attributes such as *location* and *granularity* are themselves represented by templates.

In the rule set for filling template slots, a rule is associated with a given slot. The rules make use of the indexed relational structure of the semantic parse, as well as the frame information associated with the relevant lexical item. For example, to fill the agent slot of an event template, a semantic parse node which is co-indexed with the event verb, and which is labeled "AGENT" is required. Similarly for patient, recipient, and others.

During the second pass, all the important MUC-5 words (specific processes, company names, equipment, devices) which were not handled in the first pass are processed, and each triggers an entity template. The filling of templates is carried out in the same way as in the first pass.

After all the templates are filled, co-reference links among the entity templates are established. The rules used for MUC-5 were extremely simple and applied only to definite NPs, using the precedence of the entity templates and compatibility of semantic features to determine co-reference. In the next version of this component, a focus list will track each of the entities in the discourse to facilitate reference resolution.

In the templates for the example sentence, shown in Figure 4, all the relevant entities are handled during the first pass. "Sell" is identified as a critical EME domain verb and an event template is generated for it, with a set of theta role slots (agent, patient, recipient) to be filled. Template rules identified "Hampshire Instrument" as the agent, since it has the AGENT theta role label in the semantic parse (Figure 3). An

LSI EXAMPLE SENTENCE TEMPLATES

EVENT report [0]

class:	meta
application:	muc5
domain:	microelectronics
reference number:	999999999
document number:	9999999
source:	NONE
event:	[0.1]
process:	[0.2]

Event sell [0.1]

agent:	[0.1.1]
recipient:	[0.1.2]
patient:	[0.1.3]
fp name:	fp6
tmp_parent:	[0]
frame_ref:	sell

Entity Agent [0.1.1]

name:	Hampshire Instruments
quantity:	1
definiteness:	definite
fp name:	fp2
head_fp_id:	fp3
tmp_parent:	[0.1]
frame_ref:	hampshire_instruments_inc

Entity Recipient [0.1.2]

name:	AT&T Bell Laboratories
quantity:	1
definiteness:	definite
fp name:	fp13
head_fp_id:	fp14
tmp_parent:	[0.1]
frame_ref:	a_t_and_t_bell_labs

Equipment Patient [0.1.3]

name:	lithography system
type:	x-ray
quantity:	1
definiteness:	indefinite
fp name:	fp7
tmp_parent:	[0.1]
frame_ref:	lithography_system

Process x_ray [0.2]

name:	x-ray
definiteness:	indefinite
quantity:	1
fp name:	fp9
head_fp_id:	fp9
tmp_parent:	[0]
frame_ref:	x_ray

Figure 4: LSI Templates for Sample Sentence

entity template is generated for it with slots to be filled. The template rules identified "x-ray lithography system" as the patient, since it has the PATIENT theta role label in the semantic parse. The PATIENT theta role label was derived from the fact that this noun phrase was identified as the direct object of a verb whose internal argument is PATIENT. An entity template is also generated for it, with slots to be filled. "AT&T Bell Laboratories" is determined as the RECIPIENT since it is the object of the preposition "to" (indirect object marker), and its semantic features qualify it as a recipient for the given verb. An entity template is generated for it, also with slots to be filled. The fills for the slots or attributes are derived by rules which utilize syntactic and semantic information about the noun phrase constituents.

Co-reference resolution occurs next. "Lithography system" cannot be co-referent with "Hampshire Instruments" since they belong to different semantic classes, as we know from the frames; "AT&T Bell Laboratories" cannot co-refer with "Hampshire Instruments" since they refer to different specific entities. (If the third entity were "the company," however, instead of "AT&T Bell Laboratories", it would be considered as a possible co-referent of "Hampshire Instruments").

WALKTHROUGH TEXT

The sample walkthrough text shows both strengths and weaknesses of our approach and of the DBG system as it has been implemented so far. As the last row of scores in Figure 5 shows, DBG scored quite well on this text, with a 79.25% P&R score. On closer examination, however, the performance of some of the individual modules is somewhat disappointing. For the walkthrough sample, the semantic parse of the first sentence is shown in Figure 6 and the DBG templates are shown in Figure 7. The LSI EME output templates are shown in Figure 8.

In the LSI template generation for the walkthrough text, during the first pass two critical EME domain verbs are identified: "use" and "sell", Event templates are generated for them with a set of theta role slots (agent, patient, etc.) to be filled. For "use", for example, "the stepper" was identified as the agent of the predicate, since it has a co-indexed AGENT theta role label in the semantic parse. An entity template is generated for it with slots to be filled. Template rules for determining the patient identified "excimer laser" as the patient, since it has the PATIENT theta role label in the semantic parse. An entity template is also generated for it. To determine whether an attribute template should be generated for granularity, the sentence in which "excimer laser" occurred is searched for words indicating units of granularity such as "micron","nm", and if the search is successful, the previous co-indexed word indicating size is selected to fill the *gran_size* slot.

MUC-5 relevant entities not identified during the first pass are handled during the second pass. These are: "Nikon Corp", "NSR-1755EX8A", "a new stepper", "64-Mbit DRAMS", "a light source", "the company", "latest stepper", "Nikon", "the excimer laser", "stepper", "system".

An entity template is produced for each of these entities, with a set of attribute slots to be filled for the entity. The *size* slot of "DRAMs" is derived from the co-indexed size unit word and the previous numeral ("64-Mbit"), and the *granularity* slot of "latest stepper" is filled with "0.5 micron" by the above rules. In a more recently updated version of DBG, the *module* slot for equipment can be filled. So for "excimer laser stepper", the *module* slot for "stepper" has a pointer to the excimer laser template, based on the fact that a) "excimer laser" modifies "stepper", and b) the two have a part-whole relationship in the concept hierarchy.

During the template linking phase, all of the templates referring to Nikon Corp ("Nikon Corp", "the company", "Nikon", "the company") are linked correctly. This is done by searching through the entities mentioned in the previous discourse and checking for semantic compatibility. The "NSR-1755EX8A" template did not get properly linked because appositives were not handled in this version of the system; however "a new stepper" and "the stepper" are linked together correctly. Moreover, "the latest stepper" did not get linked to the previous stepper templates, which is correct since it they do not co-refer. On the other hand, "(the excimer laser) stepper" in the third sentence of the text, was incorrectly linked to "the latest stepper"

Official Scores for MUC-5 EME Final Test

71	60	31	15	.7582	.7923	34	58	42.74	89

Unofficial Scores for EME Walkthrough Message (Run on system used for Final Test)

34	28	12	0	.3667	.3793	72	88	79.25	100
ERR	UND	OVG	SUB	MinE	MaxE	REC	PRE	P&R	TF

Figure 5: Summary of LSI MUC-5 Scores

```
fp1:
  'DESCRIPTION'('1.0') = 'INDEX'('1.1')
  'PREPOSITION'('1.1') = in
  'PREP OBJECT'('1.1') = 'INDEX'('1.2')
  'DETERMINER'('1.2') = the
  'PARTIAL PARSE'('1.1') = 'INDEX'('1.3')
  'NOUN/*SECOND*'('1.3') = second
  'PARTIAL PARSE'('1.3') = 'INDEX'('1.4')
  'NOUN/GENERAL_LOCATION'('1.4') = quarter
  'GEN PHRASE'('1.4') = 'INDEX'('1.5')
  'GEN OBJECT'('1.5') = 'INDEX'('1.6')
  'NUM'('1.6') = '1991'
  'NOUN/NIKON_CORP'('1.6') = 'Nikon Corp.'
  'QUANTIFIER PHRASE'('1.6') = 'INDEX'('1.7')
  'NUM'('1.7') = '7731'
  'NOUN/*ENTITY*'('1.7') = plan
  'NUMBER'('1.7') = 'PLURAL'
  'DESCRIPTION'('1.4') = 'INDEX'('1.8')
  'PREPOSITION'('1.8') = to
  'PREP OBJECT'('1.8') = 'INDEX'('1.9')
  'NOUN/COMMERCIAL_FACILITY'('1.9') = market
  'PARTIAL PARSE'('1.8') = 'INDEX'('1.10')
  'DETERMINER'('1.10') = the
  'NOUN/NSR_1755EX8A'('1.10') = 'NSR-1755EX8A'
  'PARTIAL PARSE'('1.10') = 'INDEX'('1.11')
  'DETERMINER'('1.11') = a
  'ADJECTIVE MODIFIER/*THING*'('1.11') = new
  'NOUN/*STEPPER*'('1.11') = stepper
  'SMALL CLAUSE'('1.11') = 'INDEX'('1.12')
  'TENSE'('1.12') = 'PAST'
  'VOICE'('1.12') = 'PASSIVE'
  'PREDICATE/INTEND'('1.12') = intend
  'PREPOSITIONAL PHRASE'('1.12') = 'INDEX'('1.13')
  'PREPOSITION'('1.13') = for
  'PREP OBJECT'('1.13') = 'INDEX'('1.14')
  'NOUN/*EVENT*'('1.14') = use
  'PREPOSITIONAL PHRASE'('1.14') = 'INDEX'('1.15')
  'PREPOSITION'('1.15') = in
  'PREP OBJECT'('1.15') = 'INDEX'('1.16')
  'DETERMINER'('1.16') = the
  'NOUN/ACTION'('1.16') = production
  'GEN PHRASE'('1.16') = 'INDEX'('1.17')
  'GEN OBJECT'('1.17') = 'INDEX'('1.18')
  'NUM'('1.18') = '64'
  'NOUN QUALIFIER/*MEGABIT*'('1.18') = mbit
  'NOUN/DYNAMIC_RAM'('1.18') = 'DRAM'
  'NUMBER'('1.18') = 'PLURAL'
```

Figure 6: Semantic Parse for Sentence of Walkthrough Text

Event report [1]

class:	meta
application:	muc5
domain:	microelectronics
reference number:	000132038
document number:	2789568
date:	191090
source:	Comline Electronics
event:	[1.1]
	[1.2]
entity:	[1.3]
	[1.8]
	[1.10]
equipment:	[1.4]
	[1.5]
	[1.7]
	[1.9]
	[1.11]
	[1.12]
	[1.13]
device:	[1.6]

Event use [1.1]

agent:	[1.1.1]
patient:	[1.1.2]
fp name:	fp54
tmp_parent:	[1]
frame_ref:	use

Equipment Agent [1.1.1]

name:	stepper
process_granularity:	[1.1.1.1]
quantity:	1
definiteness:	definite
fp name:	fp48
prev_tmp:	[1.5]
tmp_parent:	[1.1]
frame_ref:	*stepper*

Equipment Patient [1.1.2]

name:	excimer laser
process_granularity:	[1.1.2.1]
quantity:	1
definiteness:	indefinite
fp name:	fp55
next_tmp:	[1.11]
tmp_parent:	[1.1]
frame_ref:	excimer_laser

Attribute Granularity [1.1.1.1]

gran_size:	248
gran_unit:	nm
fp name:	fp57
tmp_parent:	[1.1.1]

Attribute Granularity [1.1.2.1]

gran_type:	resolution
gran_size:	0.45
gran_unit:	micron
fp name:	fp77
tmp_parent:	[1.1.2]

Event sell [1.2]

agent:	[1.2.1]
fp name:	fp130
tmp_parent:	[1]
frame_ref:	sell

Entity Agent [1.2.1]

name:	company
quantity:	1
definiteness:	definite
fp name:	fp122
head_fp_id:	fp124
prev_tmp:	[1.10]
tmp_parent:	[1.2]
frame_ref:	general_company

Entity nikon_corp [1.3]

name:	Nikon Corp.
quantity:	1
definiteness:	definite
fp name:	fp12
head_fp_id:	fp12
next_tmp:	[1.8]
tmp_parent:	[1]
frame_ref:	nikon_corp

Equipment nsr_1755ex8a [1.4]

name:	NSR-1755EX8A
manufacturer_name:	nikon_corp
model:	NSR-1755EX8A
quantity:	1
definiteness:	definite
fp name:	fp23
tmp_parent:	[1]
frame_ref:	nsr_1755ex8a

Equipment *stepper* [1.5]

name:	stepper
quantity:	1
definiteness:	indefinite
fp name:	fp27
next_tmp:	[1.1.1]
tmp_parent:	[1]
frame_ref:	*stepper*

Figure 7a: LSI Internal Templates for Walkthrough Text

Entity dynamic_ram [1.6]

name:	DRAM
size:	64
size_unit:	mbit
speed_unit:	second
quantity:	PLURAL
definiteness:	definite
fp name:	fp45
head_fp_id:	fp45
tmp_parent:	[1]
frame_ref:	dynamic_ram

Equipment light_source [1.7]

name:	light source
quantity:	1
definiteness:	indefinite
fp name:	fp64
tmp_parent:	[1]
frame_ref:	light_source

Entity general_company [1.8]

name:	company
quantity:	1
definiteness:	definite
fp name:	fp92
head_fp_id:	fp92
prev_tmp:	[1.3]
next_tmp:	[1.10]
tmp_parent:	[1]
frame_ref:	general_company

Equipment *stepper* [1.9]

name:	stepper
process_granularity:	[1.9.1]
quantity:	1
definiteness:	indefinite
fp name:	fp96
next_tmp:	[1.12]
tmp_parent:	[1]
frame_ref:	*stepper*

Attribute Granularity [1.9.1]

gran_type:	resolution
gran_size:	0.5
gran_unit:	micron
fp name:	fp87
tmp_parent:	[1.9]

Entity nikon_corp [1.10]

name:	Nikon
quantity:	1
definiteness:	definite
fp name:	fp101
head_fp_id:	fp101
prev_tmp:	[1.8]
next_tmp:	[1.2.1]
tmp_parent:	[1]
frame_ref:	nikon_corp

Equipment excimer_laser [1.11]

name:	excimer laser
quantity:	1
definiteness:	definite
fp name:	fp108
prev_tmp:	[1.1.2]
tmp_parent:	[1]
frame_ref:	excimer_laser

Equipment *stepper* [1.12]

name:	stepper
quantity:	1
definiteness:	definite
fp name:	fp109
prev_tmp:	[1.9]
tmp_parent:	[1]
frame_ref:	*stepper*

Equipment general_equipment [1.13]

name:	system
quantity:	PLURAL
definiteness:	indefinite
fp name:	fp133
head_fp_id:	fp133
tmp_parent:	[1]
frame_ref:	general_equipment

Figure 7b: LSI Internal Templates for Walkthrough Message

```
<TEMPLATE-2789568-1> :=
    DOC NR: 2789568
    DOC DATE: 191090
    DOCUMENT SOURCE: "Comline Electronics"
    CONTENT: <MICROELECTRONICS_CAPABILITY-2789568-1>
             <MICROELECTRONICS_CAPABILITY-2789568-2>
    DATE TEMPLATE COMPLETED: 060893
<MICROELECTRONICS_CAPABILITY-2789568-1> :=
    PROCESS: <LITHOGRAPHY-2789568-1>
    MANUFACTURER: <ENTITY-2789568-1>
    DISTRIBUTOR: <ENTITY-2789568-1>
<MICROELECTRONICS_CAPABILITY-2789568-2> :=
    PROCESS: <LITHOGRAPHY-2789568-2>
<ENTITY-2789568-1> :=
    NAME: Nikon CORP
    TYPE: COMPANY
<LITHOGRAPHY-2789568-1> :=
    TYPE: LASER
    EQUIPMENT: <EQUIPMENT-2789568-1>
<LITHOGRAPHY-2789568-2> :=
    TYPE: UNKNOWN
    EQUIPMENT: <EQUIPMENT-2789568-2>
<DEVICE-2789568-1> :=
    FUNCTION: DRAM
    SIZE: ( 64 MBITS )
<EQUIPMENT-2789568-1> :=
    NAME_OR_MODEL: "NSR-1755EX8A"
    MANUFACTURER: <ENTITY-2789568-1>
    EQUIPMENT_TYPE: STEPPER
    STATUS: IN_USE
<EQUIPMENT-2789568-2> :=
    EQUIPMENT_TYPE: STEPPER
    STATUS: IN_USE
<EQUIPMENT-2789568-3> :=
    EQUIPMENT_TYPE: RADIATION_SOURCE
    STATUS: IN_USE
<EQUIPMENT-2789568-4> :=
    EQUIPMENT_TYPE: RADIATION_SOURCE
    STATUS: IN_USE
```

Figure 8: LSI MUC-5 Output Templates for Walkthrough Text

in the second sentence, because the co-reference rules used for MUC-5 were too simple to distinguish between two entities in different sentences having the same semantic features. The two occurrences of "excimer laser" were linked together correctly, however, the entity "light source" did not get properly linked in, since our analysis was not complete.

Reference resolution is now performed using a discourse focus list. To determine whether a definite noun phrase refers to an entity which has already been mentioned in the discourse, it is compared with the most recent entity in the focus list, semantic feature checking is performed as before.

Also, appositives such as "NSR-1755EX8A, a new stepper" and cases like "X as Y" are now handled during the first pass. Thus "light source" can now be linked to "excimer laser" because it occurs in an "as" prepositional phrase and the two entities are of the same type.

Following reference resolution, the mapping from the LSI internal templates to the MUC-5 output templates is relatively straightforward.

Answers to the specific questions posed about the MUC-5 templates generated from the walkthrough text are given below.

```
(1) What information triggers the instantiation of each
    of the two LITHOGRAPHY objects?
```

"NSR-1755EX8A" triggered the first lithography object since it is a defined as a lithography system in our concept hierarchy and there is a rule stating that equipment can trigger related process objects.

In the second sentence, "the stepper" triggered the second lithography object by the rule given above. This is wrong because "the stepper" is co-referent with "NSR-1755EX8A". The two templates were not linked properly because "the stepper" gets linked to "a new stepper", and DBG was unaware that "a new stepper" is "NSR-1755EX8A" since appositives were not handled in that version of the system. This is corrected in a more recent version of the system, where "the company's latest stepper" triggers the second lithography object.

```
(2) What information indicates the role of the Nikon
    Corp. for each Microelectronics Capability?
```

The concept hierarchy contains the information that Nikon is the manufacturer of "NSR-1755EX8A", so the manufacturer slot of "NSR-1755EX8A" is filled with a pointer to the "Nikon" object. The MANUFACTURER role for the second Microelectronics Capability was not filled.

```
(3) Explain how your system captured the GRANULARITY
    information for ''The company's latest stepper.''
```

We got "0.5 micron" for "The company's latest stepper" (which is correct) by pattern matching and by accident. When we were filling out the granularity slot for "The company's latest stepper", "0.5 micron" was the only granularity attribute available in the sentence, since "0.45 micron" was already applied to "the stepper" in the same sentence.

```
(4) How does your system determine EQUIPMENT_TYPE for
    "the new stepper"? and for "the company's latest
    stepper"?
```

This knowledge is specified in our concept hierarchy (see above).

(5) How does your system determine the STATUS of each equipment object?

Via a default rule, which fills otherwise unspecified "status" slots with "IN USE".

(6) Why is the DEVICE object only instantiated for LITHOGRAPHY-1?

The DEVICE object was not linked to LITHOGRAPHY-1 in our output.

MUC-5 EXPERIENCE AND SUMMARY

The innovative aspect most responsible for our success this year was exploitation of the concept hierarchy, especially the links from lexical items to concept nodes in the hierarchy, which facilitated generation of the internal LSI templates. Recall scores steadily improved during addition of slot-filling rules without significantly increasing overgeneration and compromising precision. The preliminary addition of event semantics did not degrade system performance, but the complete system is not yet able to take full advantage of it. Improvements in this aspect of the DBG system should push scores significantly higher.

REFERENCES

[1] Montgomery, C. A., Hirschberg, M. A., De Cesare, A. G. Natural Language Research Aids Battlefield Coherence. Signal 45:9, May, 1991.

[2] Montgomery, C. A., Stalls, B. G., Stumberger, R. E., Li, N., Walter, S., Belvin, Robert S., and A. Arnaiz. Machine-Aided Voice Translation. Proceedings of the IEEE Conference on Dual-Use Technologies, May, 1993, pp. 96-101.

[3] Montgomery, C.A., Stalls, B.G., Stumberger, R.E., Li, N., Belvin, R.S., Arnaiz, A., and Hirsh, S.B. 1992. Description of the DBG System as Used for MUC-4. Proceedings of the Fourth Message Understanding Conference (MUC-4), pp. 197-206, sponsored by Defense Advanced Research Projects Agency (DARPA) Software and Technology Systems Office. San Mateo, CA: Morgan Kaufmann Publishers, Inc.

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