PEGASUS: A Spoken Language Interface for On-Line Air Travel Planning[1]

Victor Zue, Stephanie Seneff, Joseph Polifroni, Michael Phillips
Christine Pao, David Goddeau, and James Glass, and Eric Brill[2]

Spoken Language Systems Group
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

This paper describes PEGASUS, a spoken language interface for on-line air travel planning that we have recently developed. PEGASUS leverages off our spoken language technology development in the ATIS domain, and enables users to book flights using the American Airlines EAASY SABRE system. The input query is transformed by the speech understanding system to a frame representation that captures its meaning. The tasks of the System Manager include transforming the semantic representation into an EAASY SABRE command, transmitting it to the application backend, formatting and interpreting the resulting information, and managing the dialogue. Preliminary evaluation results suggest that users can learn to make productive use of PEGASUS for travel planning, although much work remains to be done.

INTRODUCTION

Over the past few years, our group has participated, as a member of the ARPA Human Language Technology (HLT) research community, in the development of spoken language technology in the common domain called Air Travel Information Service, or ATIS [1]. ATIS permits users to verbally query for air travel information, such as flight schedules from one city to another, obtained from a small relational database excised from the Official Airline Guide. By requiring that all system developers use the same database, it has been possible to compare the performance of various spoken language systems based on their ability to extract the correct information from the database, using a set of prescribed training and test data, and a set of interpretation guidelines. Indeed, periodic common evaluations have occurred at regular intervals, and steady performance improvements have been observed for all systems [2, 3, 4].

While the ATIS task has been instrumental in the development of technologies that can understand spontaneously generated verbal queries in a limited domain, it does have some shortcomings. First, the current common evaluation focuses on the correctness of the information extracted from the database without any regard to the system's side of the interchange (e.g., clarification queries and helpful suggestions). Thus it has the effect of discouraging research on dialogue-based systems which, we believe, is a crucial aspect of human computer interactions. Second, ATIS makes use of a mock-up, static database containing flight and fare information for a small set of cities within the United States and Canada. It is not a realistic model of the databases actually being used by travel agents and travellers. In particular, operational flight information systems are much larger and more complex, and, most importantly, they contain information which is dynamic in nature.

The rapid technological progress that we are witnessing gives us hope that spoken language systems capable of performing real tasks will begin to emerge within the decade. To realize this potential, however, it is important that we begin to develop the technology using real databases, so that we can uncover limitations and gaps in our present research paradigm. To this end, we started in 1992 to investigate the feasibility of attaching a spoken language interface to an available on-line database. We selected the American Airlines EAASY SABRE system, which allows subscribers to obtain flight information and make flight reservations via a large dynamic database, accessed through their personal computers over the telephone line. This system currently has over 700,000 active subscribers, most of whom are travellers, not travel agents. We selected this database mainly because we believe we can leverage off of our existing ATIS system to build an appropriate user-friendly interface.

To communicate with EAASY SABRE in its normal mode of operation, users issue coded queries specifying restrictions such as source, date, and fare code. If the necessary pieces of information are omitted from the query, the system enters a tightly controlled menu protocol to fill them in. What we have attempted to accomplish is a replacement of this cumbersome interface with something that permits a more natural dialogue with the computer. Our system, called PEGASUS, acts as a mediator between

[1]This research was supported by ARPA under Contract N00014-89-J-1332, monitored through the Office of Naval Research.

[2]The authors are listed in reversed alphabetical order.

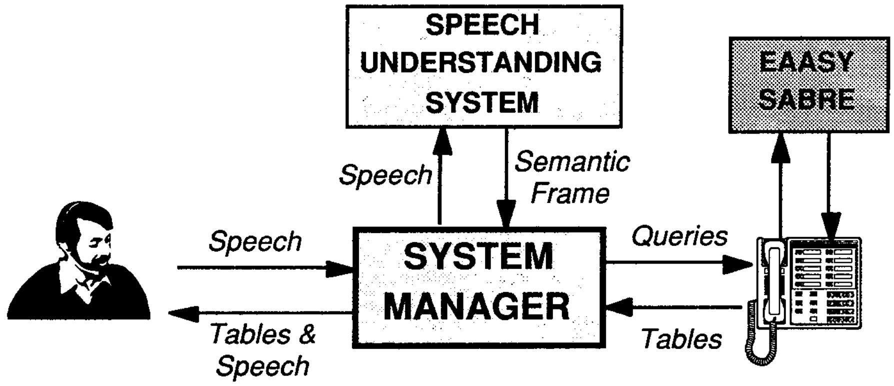

Figure 1: Schematic of the PEGASUS on-line travel planning system.

the user and the EAASY SABRE system, engaging in a spoken dialogue with the user and postprocessing tables delivered by EAASY SABRE for display on the terminal.

The rest of the paper is organized as follows. We first describe the PEGASUS system, paying particular attention to the conversion of the parse tree to a semantic representation and the multiple roles played by the System Manager, mediating between the user and the EAASY SABRE back-end. We then discuss the dialogue management aspects of the system in some detail. This is followed by some preliminary evaluation results, using data collected from real users planning real trips. Finally, we summarize lessons learned and present our future plans.

SYSTEM DESCRIPTION

Overview

Figure 1 shows a block diagram of PEGASUS. The speech understanding component makes use of much of the original ATIS system to process user queries [5, 6]. The segment-based SUMMIT speech recognition component [7] produces a list of the top ten sentence hypotheses, which are then filtered by the probabilistic TINA natural language component [8]. The particular version of TINA employs a robust parsing strategy [9] that attempts to piece together parsable fragments, which is often necessary for spontaneous speech containing disfluencies. A semantic frame representation is used to encode the meaning efficiently. The entire speech understanding system, with a working vocabulary of approximately 1300 words, performs in near real-time using a high-end workstation with no additional hardware. As for its performance, our ATIS system achieved the second lowest error rate for both text and speech input in the last two annual ARPA spoken language systems common evaluations measuring the systems' ability to extract relevant information from the database [3, 4]. In the most recent evaluation, our system achieved an error rate of 12.5% and 14.2% on all answerable queries, when the transcription and the speech signal were provided as input, respectively.

```
INPUT: IS THERE A UNITED FLIGHT CONNECTING IN DENVER
FRAME:
  Clause: EXISTENTIAL
      Topic: FLIGHT
         Quant: INDEF
           Predicate: AIRLINE_NAME
                 Name: "United"
           Predicate: CONNECT
                 Predicate: IN
                       Topic: CITY_NAME
                           Name: "Denver"
```

Figure 2: An example of a semantic frame.

Meaning Representation

Through our previous experience in developing spoken language systems, we have learned that simplicity of form is an important principle in building effective meaning representations. Our view on the appropriate structural units of a semantic frame has evolved over time. Our present view is that all major constituents in a sentence can be classified as one of only three distinct categories, which we label as [`clause`], [`topic`], and [`predicate`]. Thus, verbs, adjectives, prepositions and modifier nouns are all considered to be predicates. Furthermore, grammatical constituents such as "subject" and "direct object" are not explicitly marked in the semantic frame. We have applied this new formalism successfully across several languages in our VOYAGER domain [10], and we are also using it in PEGASUS. An example semantic frame for the sentence, "Is there a United flight connecting in Denver," is shown in Figure 2.

System Manager

During the design phase of our project, we made a commitment not to alter the interface and protocols of EAASY SABRE. We see no benefit, nor do we feel competent, in making changes to a proven system used by many users. In fact, PEGASUS's interface to EAASY SABRE is identical to that of a user on a PC or a travel agent – EAASY SABRE cannot distinguish between a user speaking a natural utterance (such as "I want to go from Boston to San Francisco on October 21") or typing to the system in cryptic codes (such as `/schedule,BOS,SFO,21OCT,1`). Thus the first task of the System Manager is to transform the semantic frame into the appropriate EAASY SABRE command. When necessary, the System Manager engages in a clarification dialogue with the user until enough information is available to construct a *complete* EAASY SABRE command, thus preempting EAASY SABRE's original menu-based clarification protocol.

In response to a command, EAASY SABRE generally returns both formatted tables and additional options available by menu selection (such as "5: Show Seat Assignment" or "12: Show more flights"). The data stream (i.e., raw text) returned from EAASY SABRE must be parsed, filtered, and reformatted by the System Manager for display to the user. The tables and menu options extracted from the database are temporarily stored in a local cache. Menu options are selectively made available to the user, thus allowing the system, for example, to map a user's explicit request for seat assignment into the appropriate menu key. Tables can be postprocessed to apply constraints beyond those available through EAASY SABRE, such as "serving dinner," "nonstop," or "leaving in the afternoon."

In addition to providing the displays, the System Manager also provides a paraphrase of the relevant information. Some users have found this feature to be extremely beneficial to help detect the system's understanding errors. The user can also type natural language queries to PEGASUS, an appropriate action when speech recognition errors persist.

DIALOGUE MANAGEMENT

Before discussing the dialogue management aspects of PEGASUS, we thought it would be instructive to examine the log of an actual round-trip booking, shown in Figure 3. In this example, one of the authors used PEGASUS to make a reservation in order to attend a workshop in the San Francisco area. Like a travel agent, PEGASUS needs to know the source, destination, and date before it can provide the flight information. The user utilized additional constraints to narrow down the choices before settling on a particular flight. It took two exchanges to arrive at the appropriate fare, and three more to book the return flight. The entire booking took nine exchanges, and lasted approximately 5 minutes. Note that a large fraction of the time is spent waiting for EAASY SABRE to respond.

The dialogue component of PEGASUS is significantly more complicated than that of the original ATIS system [11]. This is in large part due to the fact that it must monitor not only the user's dialogue state and the degree of completion of the booking, but also the state of the EAASY SABRE system. For instance, it must preprocess fare restrictions, warning the user of limits imposed on return dates for restricted fares, screening selected fares for possible constraint failure, and confirming availability on selected flights before attempting to issue bookings. Otherwise, the EAASY SABRE system would invoke a complex subdialogue which we wish to avoid.

The system keeps a record of the most recently displayed sets of flights and fares, as well as a ticket frame where slots (e.g., fare class) are periodically updated upon user specification. This information is also displayed to the user as a ticket. The system must consult all of these sources of information in addition to the user's queries in deciding its next move.

The dialogue is managed as a set of more than thirty distinct dialogue states. Any particular dialogue exchange consists of a response phase followed by an initiative phase. During the response phase, the system may have to consult the dialogue state to properly interpret the query. For instance, there are several different states that can provoke a "yes/no" response, and the system must also be prepared for the user not to comply, but instead to ask a completely independent question. While processing the user's query the system may need to update the dialogue state, and upon completing the response phase it consults the state to determine what if any initiative action is appropriate at this time.

There are several meta-level commands that have led to a more usable system. A request for help can be issued at any time, and the response it invokes is dependent upon the current state of the dialogue. For instance, at the very beginning of a dialogue, a request for help causes the system to provide an organized list of the cities it knows. There are also two meta-level commands that erase previous information. A "scratch that" command causes the system to erase the preceding query from context, restoring the discourse content to its former state. In the event that this query involved a booking request, the system must also issue a cancellation request to EAASY SABRE. The "scratch that" capability is particularly valuable to recover from damaging recognition errors. A more drastic "clear history" command allows the user to start completely afresh. We feel that the availability of these erasure commands allows the system to be more aggressive in taking action, without having to repeatedly ask for confirmation of user requests.

```
U: I WANT TO GO FROM BOSTON TO SAN FRANCISCO
P: What date will  you be travelling on?
U: OCTOBER TWENTIETH
P: Here are the flights from Boston to San Francisco on
   Wednesday October twentieth.
      FLIGHT    FROM  TO   DEPARTURE   ARRIVAL    NSTOPS
   1  DL   565  BOS   SFO   7:00 AM   11:55 AM     1
   2  US   31   BOS   SFO   7:00 AM   12:03 PM     1
   3  NW   181  BOS   SFO   7:30 AM   10:58 AM     0
   4  UA   31   BOS   SFO   8:10 AM   11:23 AM     0
   ...
   You can ask for more flights.
U: I'D LIKE A NONSTOP UNITED FLIGHT
P: Here are the non-stop United flights from Boston to
   San Francisco on Wednesday October twentieth.
      FLIGHT    FROM  TO   DEPARTURE  ARRIVAL    NSTOPS
   1  UA   31   BOS   SFO   8:10 AM   11:23 AM     0
   2  UA   59   BOS   SFO   9:05 AM   12:34 PM     0
   3  UA   93   BOS   SFO   5:10 PM    8:23 PM     0
   ...
U: BOOK THE THIRD FLIGHT PLEASE
P: The following fares are available.  Which fare
   class would you like?
    ONE WAY  FARE CLASS      MIN-STAY MAX-STAY RTRN RQ'D
   1 $ 214  THRIFT DISCOUNT   YES       30        YES
   2 $ 234  COACH DISCOUNT    YES       30        YES
   3 $ 294  THRIFT DISCOUNT   YES       NO        YES
   4 $ 304  COACH DISCOUNT    YES       30        YES
   ...
U: COACH DISCOUNT
P: The following fares are available.  Which fare class
   would you like?
    ONE WAY  FARE CLASS      MIN-STAY MAX-STAY RTRN RQ'D
   1 $ 234  COACH DISCOUNT    YES       30        YES
   2 $ 304  COACH DISCOUNT    YES       30        YES
   3 $ 336  COACH DISCOUNT    31        NO        YES
   ...
U: THE CHEAPEST ONE
P: I'll book United flight 93 from Boston to San
   Francisco on Wednesday October twentieth for you...
   Now I'll help you with the return flight.
   What date will you be returning on?
U: ON SUNDAY
P: Here are the United flights from San Francisco to
   Boston on Sunday October twenty-fourth.
      FLIGHT    FROM  TO   DEPARTURE  ARRIVAL    NSTOPS
   1  UA   92   SFO   BOS   7:45 AM    4:06 PM     0
   2  UA   20   SFO   BOS   1:30 PM    9:54 PM     0
   3  UA   90   SFO   BOS  10:10 PM    6:32 AM     0
   ...
U: SHOW ME NONSTOP FLIGHTS LEAVING IN THE MORNING
P: Here is United flight 92 from San Francisco to
   Boston on Sunday October twenty-fourth.
      FLIGHT    FROM  TO   DEPARTURE  ARRIVAL    NSTOPS
   1  UA   92   SFO   BOS   7:45 AM    4:06 PM     0
U: I'LL TAKE THIS FLIGHT
P: ... Your reservation has been entered.
   Give me a moment to confirm your reservation.
   Here's your ticket summary.  You saved a total of
   894 dollars  over regular coach fare...
   Please wait while I retrieve your itinerary...
    STEPHANIE SENEFF                       Price:   488.00
 ---------------------------------------------------------
    Date      Flight    Class  From    To      Leave    Arrive
 1  OCT20     UA  93     Q     BOS     SFO     510P     823P
 2  OCT24     UA  92     Q     SFO     BOS     745A     406P
 ---------------------------------------------------------
   Can I help you with something else?
```

Figure 3: An example of an actual verbal booking dialogue using PEGASUS. Due to space limitations, irrelevant parts of the system's responses have been omitted. U=user, P=PEGASUS.

EVALUATION

PEGASUS first came into being in January 1993. Since then, we have been actively improving and extending its capabilities. Thus the system is in a constant state of flux – deficiencies are corrected as new capabilities are introduced. Nevertheless, it is fully functional in the sense that members of our group have been able to use it to make actual travel arrangements since last spring, using naturally spoken English. Even though it is definitely premature to accurately assess the usefulness of the system, we have recently begun to formally monitor its performance longitudinally by keeping a time-stamped log file of all transactions. In this section, we will present some very preliminary results on the system's performance since early fall, 1993. The results are obtained from ten bookings made by eight members of our group in order to satisfy their real travel needs. All of them represented round-trip bookings from one city to another. In some cases, the time for travel was important whereas in others, the cheapest airfare was desired. Seven of the ten bookings were successfully completed. Statistics on some of the objective measures for the successfully completed bookings are shown in Table 1.

Averaged across all six subjects who completed the bookings successfully, it took almost 25 queries and more than 13 minutes for the subjects to complete a booking. It is interesting, however, to compare the statistics of the three experienced users[2] with the other three, who were using the system for the first time. Compared to the naive users, the experienced users completed the bookings with considerably less effort – using less than one-third of the number of queries and taking one-fourth the amount of time. The variations in their performance are also considerably less. In general, one can expect the system's performance on totally naive subjects to degrade. On the other hand, the results give us hope that experienced travellers can learn to put PEGASUS to productive use, once they become familiar with its capabilities.

We also examined the log files for the three unsuccessful bookings in order to identify the system's shortcomings. In one case, the user successfully completed the forward leg of a trip, but the system booked an erroneous return leg, causing him to start over. He cleared the discourse history, but did not explicitly cancel the booking on EAASY SABRE. Thus, even though the user successfully booked the flights he wanted, EAASY SABRE was unable to reconcile the double booking on the forward leg. In the second case, the user initially selected a fare that was incompatible with his travel plans. He did not successfully cancel his initial reservation or clear the discourse history. The system continued to enforce the restrictions on the previous fare, even though he at-

[2]Two of them are developers of PEGASUS, and the remaining one has used the system extensively. All three were familiar with the capabilities of the system.

Measurements	All Users		Experienced		Naive	
	mean	s.d.	mean	s.d.	mean	s.d.
Number of of Queries Used	24.5	13.4	10.0	3.2	34.3	5.9
Time to Completion (s.)	814	501	309	87	1311	326

Table 1: Objective measures of PEGASUS's performance for the seven bookings that were completed successfully.

tempted to rebook with an unrestricted fare. In the third case, the discount fare selected for the forward leg was not available on the return flight. Both the second and third users eventually gave up in frustration.

Since mid January, we have begun to save the speech waveform, in addition to the log-file. We were thus able to also measure the system's speech recognition performance. The word and sentence recognition error rates for these bookings were found to be 10.6% and 28.6%, respectively.

DISCUSSION AND FUTURE PLANS

This paper describes our recent effort in developing a spoken language interface to an on-line, dynamic airline reservation system. By leveraging off our ATIS development effort and paying particular attention to dialogue management, we were able to produce a working interface that enables users to make real flight bookings using spoken language.

PEGASUS is the outcome of a new research strategy that we have adopted, one that strives to develop language-based technologies within the context of *real* application back-ends, rather than relying on mock-ups, however realistic they might be. We believe that this strategy will force us to confront some of the critical technical issues that may otherwise elude our attention, such as dialogue modelling and new word detection/learning. We also believe that the time is ripe for us to begin demonstrating the usefulness of these technologies. Working on real applications thus has the potential benefit of shortening the interval between technology demonstration and its ultimate use. Besides, real applications that can help people solve problems *will* be used by real users, thus providing us with a rich and continuing source of useful data.

While we are encouraged by our initial success with PEGASUS, much work remains to be done. One of the major deficiencies of the system is its inability to gracefully coerce the user back on track when his/her request cannot be satisfied. A common problem arises when the cheapest fare that the user specified is not available on the selected return flight. The user is faced with the multiple choices of modifying his/her choice for the flight, date, or fare. Rather than leaving the user to explore all these dimensions freely and run the risk of confusion, a more productive solution may be for the system to take control of the dialogue by offering explicit choices. Of course, the user should still be free to diverge from the computer's goal whenever he/she so chooses.

Until very recently, the system's knowledge has been limited to fewer than sixty major cities in North America, Europe, and Japan. We have just expanded PEGASUS's knowledge base to more than 220 major cities worldwide. Nevertheless, it is still a very small set considering that EAASY SABRE contains flight information for nearly two thousand cities worldwide. Rather than making all the cities, airports, and airlines available with equal probability at all times, we will explore ways to constrain the search while maintaining full flexibility. One possibility is to allow a user to customize the system to suit their needs. Thus, for example, a user could specify the cities and airlines that they care about, in much the same way they presently specify their frequent flyer number, seating preferences, and credit card information for billing. The system will need to be supplemented with tools that will enable users to interactively and incrementally add appropriate information. In addition, the system could also automatically adjust language probabilities based on the user's dialogue history.

At the moment, the system can only book a single seat under the name of the user currently logged onto EAASY SABRE. In the future, we would like to add the capability of changing the name on the ticket, or booking multiple tickets for the user and accompanying family members, for example.

The present implementation of PEGASUS assumes that information is provided to the user both visually and aurally. This assumption obviously affects significantly the nature of the responses generated by PEGASUS. For example, the system will currently say, "Here are the flights from Boston to San Francisco on October 20," and proceed to display them. We believe that there will be many occasions in which a user may be communicating with the system by telephone. In such a case, the information must be presented in a different manner (e.g., "There are seventeen direct flights from Boston to San Francisco on October 20.") The resulting human-computer dialogue will be quite different from that in our current implementation. We intend to pursue such a "display-less" implementation in the future, eventually leading to the development of telephone-based applications.

State	Condition	Action	New State
Flight booked	Has first leg	Conclude booking	Initialize new transaction
	Round trip required?	Show return flight	—
	Default	—	Return leg?

Table 2: Example entry from our dialogue control table.

Our experience in designing PEGASUS has led us to the realization that considerable care must go into providing mechanisms to easily manage and maintain dialogue coherence. While our dialogue states are a convenient representation, the current mechanism for controlling them is becoming unwieldy, and therefore needs to be reorganized prior to adding some of the enhancements mentioned here.

Through our experience in developing a preliminary version of PEGASUS, we discovered that the capability to specify the dialogue flow explicitly at some high level is necessary, in order to be able to understand and manage the dialogue effectively. To that end, we recently redesigned the PEGASUS control strategy, so that dialogue moves conditioned on prior states can be conveniently specified in tabular form.

An example entry from our newest implementation is shown in Table 2. This entry states that when the user has just completed a successful booking, the system should examine the conditions in the order presented and take the appropriate action when they are met, setting the dialogue state to the new value, if appropriate. Thus, in our example, once a flight has been booked, the first thing the system does is check to see if there is a first-leg flight associated with the current one (i.e., "Has-first-leg?"). If so, the system performs the actions associated with concluding a booking (e.g., summarizing the flight information) and resets the dialogue state to anticipate a completely new exchange. If the first condition is not met, the system proceeds in the same manner through the others in the order given.

Ultimately, we would like a dialogue framework that is *domain independent.* We have begun to define a *dialogue-description language* in which different types of user interactions can be represented. The terminal nodes of the grammar would be associated with user query classes. User interactions expected within a particular domain would be described in this meta language, and that description would be used by the system to direct the human machine interaction.

There has been some theoretical work on the structure of human-human dialogue [12], but this has not yet led to effective insights for building human-machine interactive systems. We believe it should be possible to define a hierarchy of dialogue types: for example, the air travel dialogue is an instance of a more general transaction dialogue in which the user acquires information about the choices available, commits to a purchase, perhaps authorizes payment, and verifies the entire transaction. It should be possible to compile a domain-specific dialogue model from a general transaction dialogue framework and a description of the particular sub-domain.

ACKNOWLEDGEMENTS

Part of the work described in this paper is conducted in collaboration with American Airlines SABRE Travel Information Network. We also benefited from the contributions made by two past members of our group: David Goodine and Lynette Hirschman.

REFERENCES

[1] Price, P., "Evaluation of Spoken Language Systems: the ATIS Domain," *Proc. DARPA Speech and Natural Language Workshop*, 91-95, Hidden Valley, PA, 1990.

[2] Pallett, D., Dahlgren N., Fiscus, J., Fisher, W., Garafolo, J., and Tjaden, B., "February 1992 ATIS Benchmark Test Results," *Proc. DARPA Speech and Natural Language Workshop*, 15–27, Harriman, NY, 1992.

[3] Pallett, D., Fiscus, J., Fisher, W., and Garafolo, J., "Benchmark Tests for the DARPA Spoken Language Program," *Proc. DARPA Speech and Natural Language Workshop*, 7–18, Princeton, NJ, 1993.

[4] Pallett, D., Fiscus, J., Fisher, W., and Garafolo, J., Lund, B., and Pryzbocki, M., "1993 Benchmark Tests for the ARPA Spoken Language Program," *These Proceesings.*

[5] Seneff, S., Glass, J., Goddeau, D., Goodine, D., Hirschman, L., Leung, H., Phillips, M., Polifroni, J., and Zue, V., "Development and Preliminary Evaluation of the MIT ATIS System," *Proc. DARPA Speech and Natural Language Workshop*, 88–93, Pacific Grove, CA., 1991.

[6] Zue, V., Glass, J., Goddeau, D., Goodine, D., Hirschman, L., Leung, H., Phillips, M., Polifroni, J., and Seneff, S., "The MIT ATIS System: February 1992 Progress Report," *Proc. DARPA Speech and Natural Language Workshop*, 84–88, Harriman, NY, 1992.

[7] Zue, V., Glass, J., Phillips, M., and Seneff, S., "The MIT SUMMIT Speech Rrecognition System: A Progress Report," *Proc. DARPA Speech and Natural Language Workshop*, 179–189, Philadelphia, PA, 1989.

[8] Seneff, S., "TINA: A Natural Language System for Spoken Language Applications," *Computational Linguistics*, Vol. 18, No. 1, 61–86, 1992.

[9] Seneff, S., "Robust Parsing for Spoken Language Systems," *Proc. ICASSP*, 189–192, San Francisco, CA, 1992.

[10] Glass, J., Goodine, D., Phillips, M., Sakai, S., Seneff, S., and Zue, V. "A Bilingual VOYAGER System," *Proc. Eurospeech*, 2063–2066, Berlin, Germany, 1993.

[11] Seneff, S., Hirschman, L., and Zue, V., "Interactive Problem Solving and Dialogue in the ATIS System," *Proc. DARPA Speech and Natural Language Workshop*, 354–359, Pacific Grove, CA, 1991.

[12] Grosz, B., and Sidner, C. "Plans for Discourse," *Intentions in Communication*, MIT Press, Cambridge, Massachusetts, 1990.