Journal of International Information Management

Volume 5 | Issue 1

Article 8

1996

Enhancing menu-based executive information systems using a natural language-menu guide*

Sumali Conlon
University of Mississippi

Edward Gillenwater
University of Mississippi

Chi Hwang
California State Polytechnic University

Enhancing menu-based executive information systems using a natural language-menu guide*

Sumali Conlon
Edward Gillenwater
University of Mississippi

Chi Hwang
California State Polytechnic University

ABSTRACT

We argue that current Executive Information Systems (EIS), with simple menu-based user interfaces, support an inadequate range of information needs because a menu base which is too large or changes too rapidly becomes prohibitively difficult to use. We consider an alternative approach to increase executives' access to information bases by incorporating a series of component-specific Natural Language Interfaces (NLI) into the various components of the EIS (database, model base, etc.), and adding a "Natural Language-Menu Guide" (NLMG) to the traditional menu and graphics based executive information systems. The various NLI will allow users to get access to information bases without having to know information system structures or techniques. At the same time, the NLMG will help users to navigate more easily through large and changing menu bases. The larger, changing menu bases rendered usable by the NLMG can, in turn, offer more options, and options of a more timely nature. The use of a series of smaller, interrelated natural language processing systems, rather than one big NLP system, should also take fuller advantage of the limited nature of current NLP technology.

INTRODUCTION

As the downsizing and restructuring of the organization into a more teamwork-oriented horizontal framework continues, executives will need the ability to process a larger and more variable set of information than the traditional hierarchical organizational structure required.

* This research was supported by the University of Mississippi's School of Business Administration Summer Research Grant Program. Editorial assistance and conversations with John Conlon were also highly appreciated.

The increased span of control dictated by this more horizontal organizational structure will increase the need for more complicated and sophisticated information systems at the executive level (Robbins, 1993). Moreover, the increased variety of challenges faced by the typical executive will require the executive to handle problems which are unfamiliar, not only in terms of the underlying managerial issues, but also in terms of the computer skills needed to solve the problem.

This paper presents a conceptual framework for the integration of today's menu and graphics based Executive Information System (EIS) interfaces with a "Natural Language Menu Guide" (NLMG), and with an interrelated set of Natural Language Interfaces (NLI) incorporated into component systems (such as the model base or the data base). This integration may be able to provide the additional sophistication needed to enhance the information capturing ability of current Executive Information Systems, while staying within the boundaries of existing NLP technology. The increased operational flexibility offered by this new type of system will improve the decision-making capabilities of executives, therefore allowing them to extend their effective span of managerial control.

There are two key advantages of an NLI as compared to other types of command system. First, natural languages such as English are the languages in which people are used to communicating. Thus, people come to the system already knowing the language used by the interface. Second, natural languages are open ended. That is, it is possible to generate an infinite number of sentences or requests with even a very limited vocabulary. This is in contrast to menu bases, which are inherently finite in nature. These advantages, however, are countered by one major disadvantage of natural languages—natural languages are enormously complex, and so, extremely difficult to implement. Care must therefore be exercised to implement a practical NLI using the current, limited state of NLP technology.

An NL-menu interface which takes advantage of some of the open-ended nature of natural language will allow executives to obtain information not available through other input modes, by asking questions in human languages (English, for example) through ordinary typed requests. Commercially available systems already exist which allow for some of this open endedness. In addition, if the natural language interface and menu base are used interactively, they may be able to overcome many of the disadvantages of either system used in isolation.

The optimal NLP-menu based EIS as we envision it would involve several small NLI (such as Q&A, Intellect, and HAL) incorporated into various components of the EIS (database, model base, etc.), supplemented by an interactive NL-menu based system which would act as the major guide through the EIS. If this NL-based menu guide is sufficiently effective, then it should allow for a very complicated menu base, which would otherwise be too large or unwieldy to be used effectively. A large menu, then, supplemented by an effective NLMG, would permit the EIS to handle a much wider range of user needs efficiently. Conversely, the menu base should give the NLP system a fairly structured and restricted topic domain, and so, simplify the task of building the NLP system itself.

The remainder of this paper presents in Section 2 a description of current EIS structures and system interfaces. Section 3 assesses menu based interfaces. Section 4 proposes a structure to improve system flexibility by adding a series of Natural Language Interfaces (NLI) and a Natural Language Menu Guide (NLMG) to current EIS systems. Section 5 explains the Natural Language systems in detail, while Section 6 concludes.

CURRENT EIS STRUCTURE AND SYSTEM INTERFACES

Executive Information Systems (EIS) have been developed and installed in a number of organizations (Main, 1989; Millet & Mawhiney, 1992; Watson, Rainer, & Koh, 1991). These systems typically consist of PCs or workstations connected to larger systems such as mainframes or mini-computers, which contain organizational and external environment databases. As a result of these systems, executives have easier access to more information from larger and more complicated databases than their counterparts in organizations without an EIS.

Currently, EIS are primarily menu- or graphics-based interface systems. Like decision support systems, EIS consists mainly of a database (with both internal and external information), model base (spreadsheets, statistical software, etc.), and system interface. The database is usually managed by database management system software (DBMS), which facilitates retrieval and manipulation of information in the database. The model base helps users to process data in various ways, such as through statistical analysis and/or optimization. Many EIS also provide communication capabilities which can access information from other locations. It would not be a significant departure from the basic EIS model to allow some of these components to incorporate component-specific natural language interfaces.

Most executives can handle only a limited range of software packages, and therefore consider the interface with the EIS to be the most important component of the system (Friend, 1986; Hogue & Watson, 1985; McLeod & Jones, 1986). Many EIS have improved the accessibility of the systems by employing hard input devices, such as a mouse, touch screen, or keyboard, and by incorporating soft presentation formats, such as icons, tables, and graphics (Watson, Rainer, & Koh, 1991). These systems are user-friendly and intuitive, but they are essentially menu bases and suffer from the drawbacks inherent in a menu-based interface. In the following section, we examine the advantages and disadvantages of menu-based interfaces.

ASSESSING MENU-BASED INTERFACES

The basic approach to using menus is easily understood by users, since selection among alternatives is one of the elementary processes in human problem-solving (Newell & Simon, 1972). Furthermore, menu bases do not require users to know complicated programming languages or have a detailed understanding of the entire information system structure. Menu-based systems, therefore, require little training. Since ease-of-use is the ultimate goal of interface design, menus have been the basic interface mode used in EIS (Karat, McDonald, & Anderson, 1986). Menu-based systems are also easy to implement because the limited set of options offers a manageable problem domain.

An EIS is designed to display information that is relevant to the critical success factors of an organization (Rockart & Delong, 1988; Watson, Rainer, & Koh, 1991). Often the focus on critical success factors in EIS design results in systems that deliver periodic and heavily aggregated information. However, an EIS should also allow executives to "drill down" into the data (Main, 1989; Morhen, Holstein, & Adams, 1990) to gain access to information at an increasing level of detail when necessary.

As executives navigate the menu base, the level of speed and accuracy attained by users in locating particular menu items depends strongly on the breadth (number of options per menu frame) and the depth (number of sequential frames) of the menu structure. For instance, Snowberry, Parkinson, and Sisson (1983) indicate that search speed and accuracy can be improved with increasing breadth and decreasing depth of the menu tree, while Landauer and Nachbar (1985) and Tullis (1985) also suggested advantages to broader as opposed to deeper menus. In accordance with Miller's (1956) "seven plus or minus two" theory of human short-term memory capacity, Miller (1981) argued that the menu with eight options in breadth and two levels in depth is the best design combination, and Kiger (1984) found that menus containing eight or nine selections promote the best user performance and preference. Thus, a menu base which is easy to use must offer a very limited range of choices. For example, a menu offering eight choices per level and two levels of depth will provide the user with at most a total of 64 options. The addition of a third level of depth would only increase the total number of options to 512, at the cost of a considerable increase in complexity.

As discussed above, an EIS should allow users to "drill down" for access to more detailed information. Therefore, current EIS, with "canned" screens or prearranged menu options, impose serious restrictions on the executive's access to flexible data sources. In general, the executive's information needs can be overwhelming, and these needs often cannot be met by a menu domain of a normal range. For example, there is an extremely large number of possible combinations of information that one could request from even a single table of a fairly simple database. Similarly, in model bases, users may want to have access to many types of statistical analysis, optimization programs, or simulation results. A menu base capable of meeting the vast array of user's needs must therefore be so large that it would become difficult for users to find their way through its numerous levels.

On the other hand, the semantic organization of menu items can also have a significant impact on information retrieval in menu-based systems. It has been found that search time can be decreased if large numbers of menu items can be grouped by their categorical relationship, and that a strong association between broad descriptor terms and specific target words improves the search accuracy (Barnard, Morton, Long, & Ottley, 1977; Snowberry, Parkinson, & Sisson, 1983).

However, a good semantic organization of the menu base may be difficult to maintain. In many EIS installations, the number of displays or screens increases over time as new options are added (Houdeshel & Watson, 1987; Watson, Rainer, & Koh, 1991) and changes in the systems usually require a substantial re-design and re-work effort (Millet & Mawhiney, 1992). In a rapidly changing business environment, it will be impossible to keep up with all of these redesigning needs. The structure of the menu base will therefore deteriorate over time. Thus, a complicated, rapidly changing menu base will inevitably suffer from design problems, and so, will be especially difficult to work with. In the next section we propose a modified menu base system, augmented with an NLMG, which will help the user to navigate through these large, complicated, rapidly changing, and poorly structured menu bases.

THE STRUCTURE OF A NATURAL LANGUAGE-MENU BASED EIS

If NLP technology had advanced to the point where computers could understand any natural language input, then executives could communicate their needs through ordinary English requests. Unfortunately, current NLP technology is too limited to allow a single NLP system to accommodate this sort of interaction over wide application domains.

What we envision is a compromise system which would attack this problem from two different angles simultaneously. On the one hand, we propose a system which builds on an unusually large menu base, but incorporates an NLP-based menu guide which would help users to more easily navigate through the enormous number of choices offered by this menu base. On the other hand, the system we propose would include a number of component-specific natural language interfaces, connected to specific components of the EIS such as the model base or the database. These later NLI would use technologies which are already commercially available, such as Q & A or Intellect (for databases), or HAL (for spreadsheets). The user would navigate through the menu base—with the help of the NLMG—until she found the specific data file or model she wished to work with, and would then enter the database management system, for example, to work with the relevant file with the help of the NLI connected with the DBMS. Since the proposed architecture would allow a larger menu base, it would also permit a breaking up of files and application programs into smaller, more user friendly components. This would therefore simplify the tasks of the component-specific NLI.

In summary, we envision a natural language-menu guide (NLMG) which would permit a larger menu base, while this larger menu base would, in turn, allow a large number of smaller, more user friendly components, ar d so, simplify the tasks assigned to component-specific NLIs. The NLMG would be used *interactively* with the menu base, so the weaknesses of each type of system would be partially compensated by the strengths of the other. Essentially, then, the NLMG would act as a sort of Information Retrieval system (Salton, 1989), facilitating the search through an extremely large menu base. The NLMG would also incorporate an NLI, analogous to the natural language query interpreter described by Samstag-Schnock and Meadow (1993). The structure of an EIS system, incorporating this interrelated set of natural language interfaces, is shown in Figures 1 and 2.

Figure 1. Natural Language-Menu Based Executive Information System Architecture

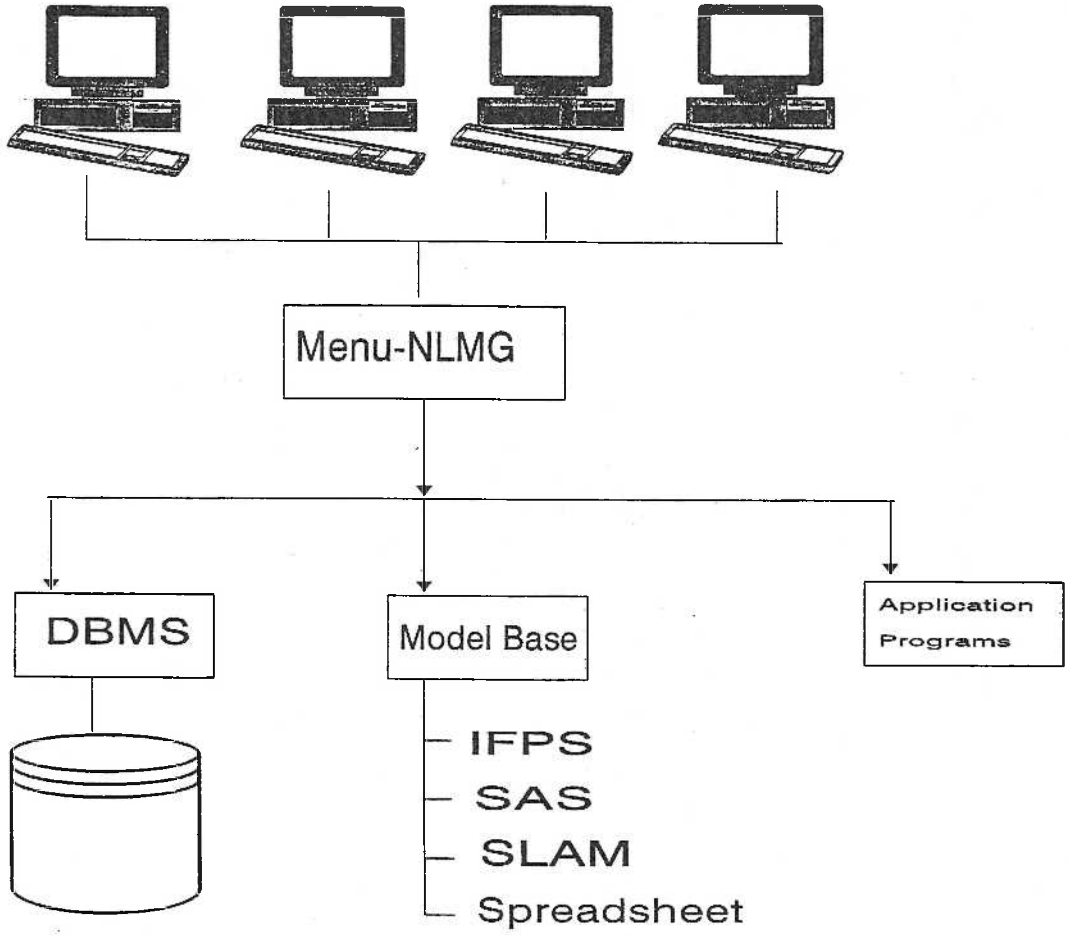
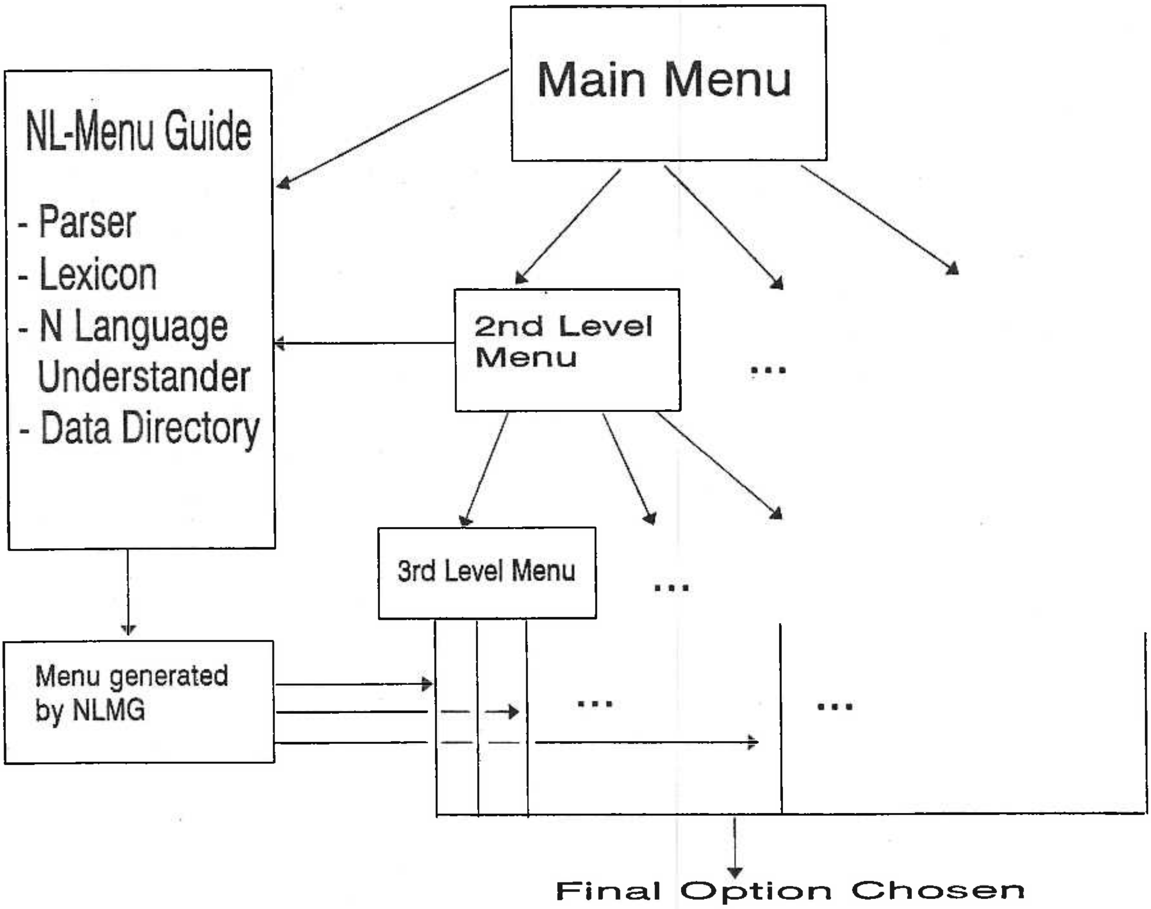


Figure 2. The Menu-Base/Natural Language Menu Guide Interaction



This system consists of:

1. **Database:** This component consists of files containing the specific information, possessed by the organization, which a user may want to work with. This may be information about the organization itself, or about the outside environment. These files are handled using database management system (DBMS) software. The DBMS may or may not incorporate a component-specific NLI.
2. **Model base:** This component consists of specific models which a user may want to work with. These models are handled using model base system software, such as IFPS, SAS, SLAM, and spreadsheets. The model base system also may or may not incorporate a component-specific NLI.
3. **Other options:** These might include application programs, communication systems (e-mail, etc.), and so on.
4. **Menu base:** This part would allow the user to select from among the various options in items 1, 2, and 3 above. If the user chooses a specific database file, for example, then the system would automatically call up the DBMS and cause the DBMS to retrieve the requested file. Note that, if the system incorporates a large number of detailed database and model base options, etc., then the resulting menu base could become very large. The menu base may incorporate a mouse, graphics, etc.
5. **Natural language-menu guide:** This part would process natural language requests, and select the items on the menu which would most closely correspond to the system's understanding of these requests. Each menu in the menu base could offer, as one of its options, access to the NLMG. If the user selects the NLMG, she would then simply type in her request using an ordinary natural language such as English.

Thus, a user of the EIS could either search through the menu in the usual way or use typed natural language requests to facilitate the search process. Suppose, for example, that a user types in a request such as "I would like to know who the biggest customer in 1993 was." This request might be processed as follows: First, the past tense formulation "was" might lead the NLMG to search for database files (just as a formulation such as "what would happen if" might suggest a model base file). Then, if three different database files have information about "customers," the NLMG might create a menu which contains these three files (see Figure 2). The user might then use a help command to request fuller descriptions of each of these three files. When the user then selects one of these three files, the system would call up the DBMS which is capable of handling this file, and cause the DBMS to retrieve the file. At this point, the user would have to use the commands of the DBMS to extract the relevant information from the file. If the DBMS incorporates a component-specific NLI, as, for example, does Q & A, then the user will be able to use natural language commands to query the system. Otherwise, the user will have to know how to work with the DBMS in question.

Note also that since the NLMG allows a larger menu base, it becomes possible to build a great deal of redundancy into the system. For example, different menu items could offer different views of the database, which display various sets of attributes and records obtained by joining various tables, under various constraints. Large numbers of such menu items could be prepared by system designers in anticipation of user needs. Feedback from users could also help to guide the system designers' efforts. Since these views would normally be smaller than the original unrestricted tables, they would be easier to use, and so, give users better access to the relevant information. This would allow a user who is unfamiliar with database techniques and structures to nevertheless have a number of options in terms of how they may view the information.

It may also be worthwhile to record related information in different files at different levels of aggregation. Thus, summary data may be offered closer to the "surface" of the menu base, more easily accessible to the user, with more detailed information, arranged in various ways, available several layers down in the menu base, to be retrieved with the help of the NLMG when needed. Though this redundancy and splitting up of related files would increase the size of the menu base and disturb its semantic organization, access to the NLMG would nevertheless render the menu base usable.

The next section briefly describes existing natural language interface systems, NLP technology, and the structure of the NLMG.

NATURAL LANGUAGE PROCESSING SYSTEMS AND THE NLMG

One way to understand the significance of NLP technology is to compare natural languages to programming languages on the one hand, and to menu bases on the other. Like programming languages, and unlike menu bases, natural languages are essentially open ended. That is, they can draw on even a limited vocabulary to generate a set of statements which is essentially infinite. On the other hand, natural languages, like menu bases, but unlike programming languages, do not require users to have specialized training. Of course, natural languages, unlike programming languages or menu bases, have an incredibly complicated syntax which is extremely difficult for NLP systems to duplicate or even approximate.

Natural language interfaces (NLI) are designed to take advantage of the open ended quality of natural language in restricted subject domains which do not overly tax the capabilities of current NLP technology. These systems have received increasing attention in recent years (Barros & DeRoeck, 1994; Bates et al., 1^{o}94; Capindale & Crawford, 1990; Davidson & Kaplan, 1983; Holt, 1988; Perraut & Grosz, 1986; Samstag-Schnock & Meadow, 1993; Vassiliou, Jarke, Stohr, Turner, & White, 1983; Wu & Dilts, 1992). In practice, a typical NLI receives input text in natural language form (such as English) and translates these sentences into computer languages which software systems can understand. In requests directed to database systems, for example, the NLI may convert English sentences into a language such as Structured Query Language (SQL), which is widely used by most relational database management systems. Thus, a sentence like "Which dealer sold the most cars last year?' can be translated into an SQL command as:

```
SELECT DEALER_NAME FROM SALE_INF
     WHERE AMOUNT_SALE=
          (SELECT MAX (AMOUNT_SALE) FROM SALE_INF
               WHERE SALE_INF.YEAR = 1993
                    AND PRODUCT_NAME = 'CAR')
```

Today, many NLI systems have been proposed and developed. These systems have been found to be efficient and powerful means for question-answering information-retrieving tasks performed by casual users (Capindale & Crawford, 1990; Kelly, 1983; Turner, Jarke, Stohr, Vassiliou, & White, 1984). For instance, Conlon, Reithel, Aiken, and Shirani (forthcoming), proposed an NLI to facilitate information retrieval in a group decision support system (GDSS), and Wu and Dilts (1992) developed an NLI to help production managers to get access to heterogenous database in a computer integrated manufacturing (CIM) environment. NLIs include HAL, Intellect, Clout, and Q & A, among several others.

The system we propose in this paper would incorporate a Natural Language Menu Guide (NLMG) in addition to the various component-specific NLI. This NLMG, like the traditional NLI, would be designed to take advantage of the users' familiarity with natural languages. However, it would not have to take as full advantage of the open-ended nature of natural language. This is because the output of this system would essentially be only a choice of items on the menu. That is, the NLMG would take as input the users' typed natural language requests, and submit as its semantic output a choice of menu items. The system therefore acts like a type of information retrieval (Salton, 1989). The "meaning" of the users' requests, as interpreted by the system, would therefore be essentially this choice of items. This restricted nature of meaning attached to natural language requests would therefore significantly simplify the construction of the NLMG.

The most elementary possible NLMG would be a simple keyword search system. More sophisticated systems might be able to extract additional information from a given request. Below we will briefly describe the potential structure and operation of a relatively sophisticated system. Such a system would generally incorporate the following components:

1. The *data directory,* which would present a list of keywords and menu items associated with those keywords.
2. The *lexicon,* which contains information such as spelling, word's part of speech, syntactic behavior, and semantic roles.
3. The *parser,* which analyzes sentences and converts them into parse trees that are easier for the computer to process.
4. The *language understanding system,* which analyzes the parse tree to find the meanings of the corresponding sentences.

The system would then be structured roughly as follows. First, the system designer would attach a number of keywords (and key phrases, such as "customer name" and "employee salary") to each item in the menu base. These keywords would then be collected into the data directory, which would indicate which menu items correspond to each keyword.

The lexicon would contain, among other things, a network of lexical-semantic relations which would connect words to one another, and, in particular, to the keywords in the data directory. Lexical-semantic relations are relations between words which can be used by a system to draw various kinds of inferences. For example, two of the most important lexical-semantic relations are synonyms (company SYN firm) and taxonomic relations (sales representative ISA employee). Other examples of lexical-semantic relations, some of which will be used in the discussion below, include:

Chicago	ISA	city
city	ISA	location
customer	SYN	buyer
purchase	NOMVERB	buy

Note here that from the first two lexical-semantic relations the system can infer the additional lexical-semantic relation Chicago ISA location. The fourth lexical-semantic indicates that the noun "purchase" describes the action of the verb "buy." These lexical-semantic relations would allow the system to connect words in a user request to keywords in the data directory (see below). For more on lexical-semantic relations, see Fox (1980), Conlon, Evens, Ahlswede and Strutz (1993), and Conlon, Riethel, Aiken, and Shirani (1994).

To facilitate the extraction of keywords (and key phrases) from a user request, a sophisticated NLMG would use a parser to determine the phrase structure of the request. Such a parser would determine which collections of words constitute noun phrases, verb phrases, etc., which nouns are modified by which adjectives or prepositional phrases, and so on. The parsed sentence would then serve as the input into the language understanding system which would use lexical-semantic relations and other information to match the request to the most relevant keywords in the data directory.

To consider a somewhat detailed example of how a relatively sophisticated NLMG could process a user request, suppose that a user submits a request such as "I would like to know how many computers customers in Chicago bought."

First, since the sentence has a simple past tense structure ("bought"), as opposed to a conditional structure ("what would happen if . . ."), the system could determine that the request is probably directed towards the database rather then the model base.

Next, a parser could determine the phrase structure of this sentence. This phrase structure would indicate that the expression "customers in Chicago" is a noun phrase, and that "in Chicago" is a prepositional phrase modifying the noun "customers." The lexical-semantic relation, Chicago ISA location, would then suggest that the location of the customers is an issue, and so the database file of interest should not only have information about customers, but should also have information about customer location.

In addition, the parse would indicate that the noun phrase "customers in Chicago" constitutes the subject of the verb "bought." The language understanding system would then use the network of lexical-semantic relations (in particular, purchase NOMVERB buy) and other information to determine that if "customers in Chicago" is the subject of bought," then "customer purchases" are an issue. A sophisticated parser/language understanding system might also be able to determine that the "purchases" are purchases of "computers." However, even a system incapable of determining this would be able to identify "computer" as a potentially important keyword.

In any case, the above would indicate that the database file of interest should contain information on "customer location," "customer purchases," and "computers." The data directory would then indicate which database options have information on customer location, which have information on customer purchases, and which have some sort of information on computers. If, say, there are four options, each of which has information of all three types, then the system would generate a menu containing these four options. The user could then use help keys to access brief descriptions, prepared by the system designer, for each of these four options. When the user finally selects an option to work with, the system would automatically load the DBMS, and have the DBMS call up the relevant files into memory.

While the above assumes that the system incorporates a fairly sophisticated parser, even a simpler system would be able to determine that "computers," "customers," "Chicago," and "bought" are potentially important keywords, and that the words "Chicago" and "bought" suggest the additional keywords "location" and "purchase."

Also note that parsers would be important for other reasons as well. For example, prepositional phrase constructions may identify parts of a request as referring to subject ("files on <topic>"), or time period ("sold in <year>"). In addition, a sophisticated parser can help the system to identify logical connectors such as "and," "or," and "not." For more on these issues, see Biswas, Bezdek, Marques, and Subramanian (1987) and Samstag-Schnock and Meadow (1993).

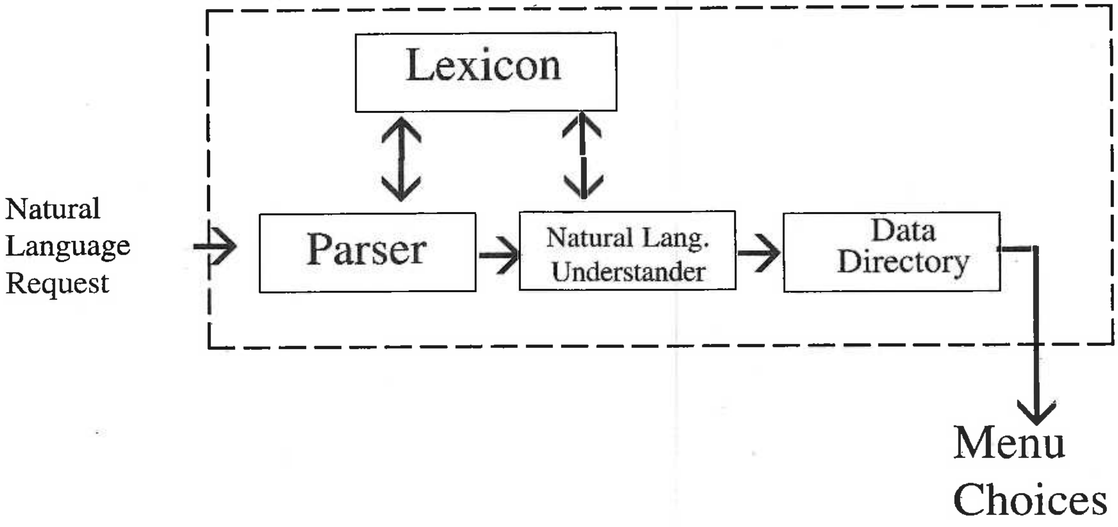
An NLMG incorporating a lexicon, parser, language understanding system, and data directory is illustrated in Figure 3.

CONCLUSIONS

In this paper we have described an architecture which would allow a sophisticated EIS to take advantage of the potential synergism between menu bases and existing NLP technology. The architecture builds on an unusually large menu base, combined with a natural language-menu guide which would help the user to navigate through this menu base.

System flexibility for EISs can be improved if users are allowed to interact with the system freely. Current EISs provide user-friendliness and rapid access to information systems. These features can be augmented by a system with an NLMG which would give users access to a wider range of choices more easily. Such systems would provide more information, but still be easy to use. Executives would therefore be able to take advantage of a wider range of information systems available in the organization, which would increase the effectiveness of the decision-making process.

Figure 3. The Structure of the Natural Language-Menu Guide (NLMG)



REFERENCES

Barnard, P. J., Morton, J., Long, J. B. & Ottley, E. A. (1977, April). Planning menus for display: Some effects of their structure on user performance. *IEEE Publication 150,* 130-133.

Barros, F. A. & DeRoeck, A. (1994, October). Resolving anaphora in a portable natural language front end to databases. *Proceedings of the Fourth Conference on Applied Natural Language Processing,* 119-124.

Bates, M., Bobrow, R., Ingria, R. & Stallard, D. (1994, October). The delphi natural language understanding system. *Proceedings of the Fourth Conference on Applied Natural Language Processing,* 132-137.

Biswas, G., Bezdek, J. C., Marques, M. & Subramanian, V. (1987, March). Knowledge-assisted document retrieval: I. The natural language interface. *Journal of American Society for Information Science,* 83-96.

Capindale, R. A. & Crawford, R. (1990). Using a natural language interface with causal users. *Int. J. Man-Machine Studies, 341-362.*

Conlon, S., Evens, M., Ahlswede, T. & Strutz, R. (1993). Developing a large lexical database for information retrieval, parsing, and text generation. *Information Processing and Management, 28*(4), 415-431.

Conlon, S., Reithel, B., Aiken, A., & Shirani, A. (1994). A natural language processing based group decision support system. *Decision Support Systems, 12*(3), 181-188.

Davidson, J. & Kaplan, S. (1983) Natural language access to databases: Interpreting update requests. *American Journal of Computational Linguistics, 9*(2), 57-68.

Fox, E. A. (1980). Lexical relations: Enhancing effectiveness of information retrieval systems. *ACM SIGIR Forum*, 5-36.

Friend, D. (1986, April). Executive information system: Successes, failures, insights, and misconceptions. *Transaction from the Sixth Annual Conference on Decision Support Systems*, J. Fedorowicz (Ed.), Washington, DC, pp. 35-40.

Hogue, J. T. & Watson, H. J. (1985, April). An examination of decision maker's utilization of decision support system output. *Information & Management, 8*(4), 205-212.

Holt, J. (1988). *Cases and applications in Lotus 1-2-3 with HAL*, 2nd ed., Irwin.

Houdeshel, G. & Watson, H. J. (1987, March). The management information and decision support (MIDS) system at Lockheed-Georgia. *MIS Quarterly, 1*(1), 127-140.

Karat, J. McDonald, J. E. & Anderson, M. (1986). A comparison of menu selection techniques: Touch panel, mouse and keyboard. *International Journal of Man-Machine Studies, 25*, 73-88.

Kelly, J. F. (1883). An empirical methodology for writing user-friendly natural-language computer applications. In*Proceedings CHI183 Human Factors in Computing Systems (Boston, Mass., Dec. 12-15)*. New York: ACM. pp. 193-196.

Kiger, J. I. (1984). The depth/breadth trade-off in the design of menu-drive user interface. *International Journal of Man-Machine Studies, 20,* 201-213.

Landauer, T. K. & Nachbar, D. W. (1985). Selection from alphabetical and numeric menu trees using a touch screen: Breadth, depth and width. *CHI '85 Conference Proceedings,* San Francisco, CA. pp. 73-78.

Main, J. (1989, March 13). At last, software CEOs can use. *Fortune, 119*(6), 77-83.

McLeod, R. & Jones, J. W. (1986). Making executive information systems more effective. *Business Horizons.*

Miller, D. P. (1981). The depth/breadth tradeoff in hierarchical computer menus. *Proceedings of the 25th Annual Meeting of the Human Factors Society,* pp. 296-299.

Miller, G. A. (1956). The magic number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review, 63,* 81-97.

Miller, I. & Mawhiney, C. H. (1992). Executive information systems: A critical perspective. *Information & Management, 23,* 83-92.

Morhen, L., Holstein, W. K. & Adams, R. B. (1990, December). EIS: It can work in the public sector. *MIS Quarterly, 14*(4), 435-448.

Newell, A. & Simon, H. A. (1972). *Human problem solving.* Hillsdale, NJ: Prentice Hall.

Perraut, R. & Grosz, B. (1986). Natural language interface. *Annual Review of Computer Science, 1,* 47-82.

Robbins, S. P. (1993). *Organizational behavior: Concepts, controversies, and applications.* Englewood Cliffs, NJ: Prentice Hall. pp. 506, 530.

Rockart, J. F. & Delong, D. W. (1988). *Executive support systems: The emergence of TOP management computer use.* Homewood, IL: Dow Jones-Irwin.

Salton, G. (1989). *Automatic text processing.* Reading MA: Addison-Wesley.

Samstag-Schnock, U. & Meadow, C. T. (1993). PBS: An economical natural language query interpreter. *Journal of the American Society for Information Science, 44*(5), 265-272.

Snowberry, K., Parkinson, S. R., & Sisson, N. (1983). Computer menu displays. *Erognomics, 26,* 699-712.

Tullis, T. S. (1985). Designing a menu interface to an operating system. *CHI '85 Conference Proceedings,* San Fransisco, CA, pp. 79-84.

Turner, J. A., Jarke, M., Stohr, E. A., Vassiliou, Y. & White, N. (1984). Using restricted natural language for data retrieval: A plan for field evaluation. In *Human Factors and Interactive Computer Systems*, Y. Vassiliou (Ed.). Norwood, NJ: Ablex, pp. 163-190.

Vassiliou, Y., Jarke, M., Stohr, E. A., Turner, J. A., & White, N. (1983, December). Natural language for database queries: A laboratory study. *MIS Quarterly, 7*(4), 47-61.

Watson, H. J., Rainer Jr., R. K. & Koh, C. E. (1991, March). Executive information systems: A framework for developm ent and a survey of current practice. *MIS quarterly, 15*(1), 13-30.

Wu., W. & Dilts, D. M. (1992, November/December). Integrating diverse CIM data bases: The role of natural language interface. *IEEE Transactions on Systems, Man and Cybernetics, 22*(6), 1331-1347.