A Dialogue Environment for Accessing Public Administration Data: the TAMIC-P System

Clara Bagnasco[1], **Amedeo Cappelli**[2] and **Bernardo Magnini**[3]

Abstract. The purpose of this paper is to show that with a combination of state of art technologies it is possible to realize effective and powerful dialogue systems for Information-Seeking applications. TAMIC-P, a prototype natural language based interface will be described. The system is designed to support Public Administration desk operators during an information session with a citizen. Three aspects work together to determine the system's usability: (i) a strong multimodal approach, allowing full interchangeability between natural language and direct manipulation; (ii) a robust dialogue component, allowing complex sequences of turns; (iii) the adaptive integration of different information sources, such as personal data and related norms.

1 INTRODUCTION

This paper follows a tradition in the development of natural language interfaces to databases (NLIDB) which includes, among many others, experimental prototypes (TEAM [9], IRUS [3], JANUS [17], SHOPTALK [6]) and commercial products (INTELLECT [10], PARLANCE [4], English Wizard).

In recent years, the diffusion of *Direct Manipulation* (DM) techniques has slowed down the use of NLIDBs. Indeed, the selection of visible objects is more efficient through pointing than by means of a linguistic description. However, also direct manipulation suffers from some drawbacks [15]: only a limited numbers of actions can be performed at a given time, and objects on which actions are applied must be unambiguously displayed on the screen. However, in the Human Computer Interaction field, the direct manipulation in itself has been criticized. Analysis of the intrinsic drawbacks of direct manipulation induced Gentner and Nielsen [8] to recognize that linguistic interaction is a key modality with which to overcome those limitations.

At the same time, in the AI community *multimodality* has been proposed as a solution to overcome the limitations of a single modality [12]. In particular, Cohen and Sullivan [5] proposed coupling NL and DM interactions. On the one hand, direct manipulation mitigates the opacity of language (see [1] for a review of NLIDBs), enabling users to learn which objects and actions are available in the system. On the other hand, natural language helps reduce the DM limitations by allowing users to specify targets and actions by means of descriptions that may refer to objects not displayed on the screen.

According to the multimodal paradigm we focused on three aspects: (i) a strong multimodal approach, allowing full interchangeability between natural language and direct manipulation; (ii) a robust dialogue component, allowing complex sequences of turns; (iii) the adaptive integration of different information sources, such as personal data and related norms.

The paper first (section 2) introduces the TAMIC-P scenario, as well as its main functionalities. Section 3 describes the system architecture. Section 4 presents the procedures underlying the dialogue processing. Section 5 reports the application building experience. Two evaluation experiments are reported in section 6 to highlight the application benefits. Finally, section 7 sets out the basic software and hardware system requirements.

2 THE TAMIC-P SCENARIO

TAMIC-P (Transparent Access to Multiple Information for the Citizen - Pensions) addresses the problem of the information flow from the Public Administration (PA) to citizens. A feasibility study [2] conducted at INPS[4] pointed out three main requirements in the relationship between PA and citizens:

- The communication between PA and citizens is going to be more *transparent* than in the past. This affects the way information has to be accessed and presented. Citizens have to follow what the operator asks to the system; furthermore data must be shown in a transparent way.
- The role of human operators is still crucial, but their work could be significantly improved with the use of more suitable tools for accessing data. In the current situation, each PA sector has its own access procedures, which can not be used by operators of other sectors. This situation dramatically limits the mobility of employees, who tend to be highly specialized but unable to be of help outside their area. This also contrasts with the current tendency to multi-function desks, where at the same point the citizen will receive a wide range of information. To sum up, the requirement is for flexible tools able to adapt themselves to different situations and with high usability level.
- In the perspective of multi-functional desks, one of the more crucial needs will be not only the integration of data from different sources but also the capability to access and manage data adaptively, considering their usefulness in the appropriate context, as for example the capability to automatically provide help and explanations at the right moment.

Following these requirements a prototype work environment for the PA desk operator has been realized; the prototype is currently

[1] Quinary S.p.A., Milano, Italy, email: bag@quinary.it
[2] CNR-ILC, Istituto di Linguistica Computazionale, Pisa, Italy, email: amedeo@ilc.pi.cnr.it
[3] ITC-irst, Istituto per la Ricerca Scientifica e Tecnologica, 38040, Povo, TN, Italy, email: magnini@irst.itc.it

[4] INPS (Istituto Nazionale per la Previdenza Sociale), is the biggest Italian organization for Social Security, serving about 19 M Italian working people and 17 M of retired people, with more than 2000 local agencies in every part of Italy.

installed at the INPS provincial agency in Trento for experimentation. Three main functionalities, all accessible by natural language, are fully integrated in the system:

- access to archives containing personal information on citizens;
- access to a technical dictionary providing general information on domain terms and concepts;
- access to a textual database containing information on laws and norms in the social insurance area.

The scenario is that of an attended desk at INPS, where citizens receive information concerning their *contributions*, *social insurances* and *pensions*. The goal of TAMIC-P is to help the operator to access all available knowledge on a given topic; however the system does not possess knowledge to guide the user to the solution of a problem (i.e. it is not an expert system).

The design of TAMIC-P was conceived as an instance of a multimodal interaction style along the lines proposed by Norman and Draper [15], Cohen [6] and Gentner and Nielsen [8]. The modalities involved are the usual Windows-Icons-Menus-Pointing style and natural language. They are fully interchangeable, i.e. the same information can be reached either by natural language or by direct manipulation: the degree of efficacy of each modality depends both on the task to be solved and on the user's expertise.

The graphical interface is based on the idea of consulting information cards. To access a card through natural language it is sufficient to enter a short description of the card content in the character input field of the interface.

In what follows, some of the main features of the TAMIC-P system are presented.

- *Linguistic Flexibility.* Descriptions in natural language are flexible both with respect to the syntactic structure, (e.g., either *contributions after 1980 voluntary* or *voluntary contributions after 1980*) and the used terms (e.g., either *voluntary payments from 1980* or *voluntary contributions*). The language may be telegraphic and synthetic, as in *Cig contributions 1970-1975* (where *Cig* stands for lay-off fund), but even redundant or partially inappropriate queries are accepted (e.g.: *contributions to the lay-off fund administration from year 1970 to year 1975*).
- *Synthetic and Analytical Views.* Due to the complexity of the available information, two views are enabled: a synthetic view, with the minimum needed information, used to review data lists (for example a list of contribution periods), and an analytical view, used to inspect details, for example concerning a single contribution period.
- *Information Integration.* The citizen's personal data are always displayed with links both to technical explanations and to the related norms. Thanks to this mechanism, after an input request like *contributions for University degree redemption* three kinds of information may be obtained on the specific subject of the contributions for University degree redemption.
- *Hypertextual Navigation.* Each card is a hypertext, where information is presented by means of active objects, which can be inspected for further details. By means of the hypertextual navigation modality, it is possible to move from one subject to another, even without detailed knowledge on the underlying domain. At any moment, the operator can move from the hypertextual navigation to the natural language modality, in a complete transparent way.
- *Contextualization.* Natural language queries are interpreted by the system in the context of the preceding queries. This allows follow-up sequences, which prove to be quite natural and effective during the consultation of complex data. For example the sequence:
  $>$ *contributions*
  $>$ *deemed*
  $>$ *for Lay-off pay fund*
  gradually restricts the output data by using the power of the contextualization mechanism (see section 4 for more details).
- *Correction of Typing Errors.* The most common typing errors (e.g. inversions of adjacent keys on the keyboard or omissions of a character in a word) are automatically corrected. Only in the case of very complex errors does the system ask the user to choose among possible corrections.
- *Analysis of Abbreviations.* Because of the frequent use of abbreviations, in particular by specialized operators, a highly effective abbreviations recognizer has been devised. For example, a query like *contr. serv. mil.*, is correctly interpreted by the system as *contributions for military service*, automatically eliminating other less plausible interpretations, such as *contract related to military service*, or *control of military service*.
- *Personalized Presentations.* The TAMIC-P system allows different presentations of the same data. In order to obtain a particular presentation, it is sufficient to perform a query in natural language by specifying the ordering criterion, as in: *contributions by type*.

3 SYSTEM ARCHITECTURE

The TAMIC-P system is highly heterogeneous both with respect to the kind of knowledge it is based on and with respect to the data store to which it makes access[5]

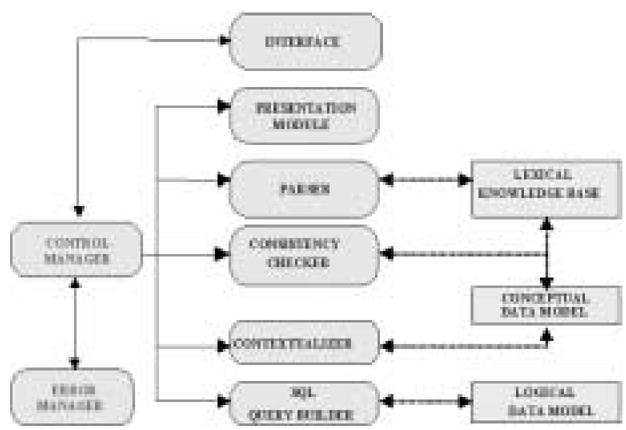

Figure 1. TAMIC-P main modules (oval boxes) and knowledge bases (rectangular boxes).

The system architecture is based on four layers:

- The *graphical interface layer*, which defines all the allowed interactions between the user and the application.
- The *linguistic layer*, which carries out the basic steps of natural language interpretation: tokenization and morphological analysis, parsing, semantic analysis (resulting in a logical form), contextualization.

[5] The INPS domain is characterized by several archives distributed nationwide. Most of them are old hierarchical databases and are now migrating to relational DB2. Many data access procedures access a common subset of data to provide different "presentations", either by accessing the same data in different databases, or by accessing the same database in different environments (e.g. CICS, IMS, AS400, etc.).

- The *data access layer*, which retrieves data from the INPS archives and records them in a temporary local database, where integration and homogenization operation are performed. Data access includes several modules which: (i) transform the logical form into an SQL query to a local database; (ii) compile and execute a set of middleware messages which gather data from the physical archives; (iii) integrate the data gathered into the local database and execute the SQL query.
- The *presentation layer*, which aggregates the retrieved data and displays them in the most convenient output format.

Each layer is in turn built around a number of more specialized modules (see Figure 1) whose interaction is driven by a Control Manager that coordinates the execution of complex tasks. Each module can thus be considered an agent that activates itself in response to a message sent by the Control Manager. Then, each module communicates the computation results to the Control Manager which, on the base of the received output, can fire a subsequent interpretation phase or transfer the control to an Error Manager module.

The system makes use of different information sources. The domain terminology is represented in a *Lexical Knowledge Base*, a WordNet-like conceptual dictionary [13] whose synsets are defined by their synonyms, pointers to direct hypernyms and hyponyms, an explanatory gloss, a logical form (used to build the semantic representation of the NL query), and pointers to legal textual documents.

The domain knowledge is represented in a *Conceptual Data Model* composed by a limited set of frames organized into an ISA hierarchy. Frames are independent both from the organization of the data archives and from the particular words used when speaking about them.

Finally, the *Logical Data Model* describes the structure and format of the data stored in the user archives.

Given these knowledge levels, a crucial problem is providing a consistent mapping among them. This is done in a declarative way, defining the data structure transformations needed to share information among modules at different levels. As an example, for the mapping between the logical and the conceptual model, quite complex transformations are necessary, such as the transformation of a flat record into a hierarchy, the possibility to aggregate data, and the transformation of a relationship represented with external keywords into a relationship among frames.

4 DIALOGUE CAPABILITIES

TAMIC-P exploits a number of dialogue capabilities that traditional natural language interfaces to databases do not possess. The dialogue manager is based on a *Semantic Dialogue Tree* (SDT) [11] which allows a robust linguistic interaction and a close correspondence with graphic objects. A key concept when building SDTs is that there is no one-to-one relationship between turns and tree nodes. The basic assumption is that both tree nodes and edges must explicitly represent the semantic relations denoted in a turn, taking advantage of domain knowledge. There are two main advantages with respect to dialogue processing: (i) semantic inconsistencies are easily detected, even in case of a well-formed turn that does not fit with any previous discourse segment; (ii) the contextualization procedure is more robust, because all the segments of each dialog turn are available as a context for subsequent turns.

Figure 2 shows a SDT for a simple dialogue. A turn is decomposed into its meaning components, which are then available for successive contextualizations. For instance, turn 1, *contributions as employee from 1990* is decomposed into nodes 1a-1c. At turn 4, the contextualization procedure starts from the last attached node (i.e. 3b) until a consistent merge is reached at node 1a.

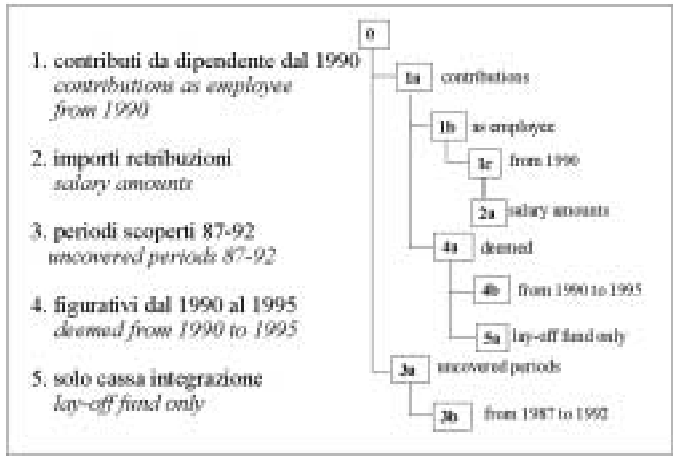

Figure 2. "Semantic dialogue tree" for a sample interaction. A simple line stands for the subsumption relation, while a double line stands for the property-value relation.

Associated with each node in a SDT is a Logical Form (LF). A new node is attached to the tree by comparing its LF with the ones of the predecessors: the LF of the current user query will be attached to the first SDT node providing consistency. The resulting dialogue tree exploits the semantic relations among the dialogue turns rather than their temporal proximity (i.e. temporal adjacent turns may have low semantic affinity, and vice-versa).

The procedure which builds the SDT consists of three steps: (i) first, a logical form is built for a single turn, whose clauses are grouped according to their functional role within the user request; (ii) then, the overall dialogue tree is searched for the positions where the input logical form can be best inserted; (iii) finally, the logical form is enriched with the inherited information, converted into a dialogue subtree and attached to the SDT.

The SDT structure is also used to provide a semantic interpretation for direct manipulation operations on graphical objects. For example, when the user double-clicks on a row representing a particular contribution, the system opens a new card showing a detailed description of the selected object. This is interpreted as an instruction to add a new node to the SDT with a "member-of" (i.e. a class instance) relation to the previous node.

5 APPLICATION BUILDING

The TAMIC-P environment has been defined and implemented by two EU supported projects: a one-year long feasibility study and a two year long implementation and experimentation project. The second project resulted in the implementation of two prototypes, both integrated in the user working environment, one handling dialogues in Italian and the other handling dialogue in German. The total effort of the implementation project was of 305 man months, with a total cost of 1,350,000 Euro. The effort needed to build the Italian prototype (to which this article refers) was about 92 man months, partly devoted to the implementation of a reusable infrastructure. Due to the nature of the project, combining Natural Language Processing with Database Integration issues, a multidisciplinary team had to be set up. The team was formed by seven persons: two working on Natural

Language Processing, one on Human Computer Interaction, three on Database Integration and one on User Modeling. Other people joined the team for short periods in the course of the project.

A key factor enabling both co-operation among the four groups mentioned above and software integration was the early definition of common representation structures and information exchange protocols. This is also an enabling factor for the system maintenance and for its portability to different domains.

Physical changes in the database structures can be handled by upgrading the logical model mentioned in section 3, without any impact on NL understanding modules. In a similar way porting to another language (INPS may have to deliver information to foreign citizens) will involve no upgrade of the data access component. More significant changes in the domain, that could result from changes to the norms governing pension administration, would require modifications to the logical model and to the data model and to the mapping between them. However, even if these changes imply careful updating of the system knowledge bases, which could be helped by maintenance tools monitoring the overall coherence of the models, no major upgrade is foreseen on the software routines. The basic communication and representation infrastructure could be preserved even when considering porting to a complete different domain, thus significantly lowering the implementation effort.

It is worth mentioning that, with respect to the original work plan, we had to re-plan the Database Integration task. It proved harder then expected, both because of security problems, which obliged us to access some databases by means of existing data procedures, and because of a lack of documentation. Much information about the behavior of data access procedures and about the form and content of their output had to be discovered by on-field experimentations, with the help of domain experts, each possessing a chunk of the whole knowledge. This was a further indication of the need to develop a unifying and transparent dialogue environment to facilitate data access.

6 SYSTEM EVALUATION

In this section we report the results of two evaluation phases: the first, carried out at the end of the TAMIC-P design, was based on a *Wizard-of-Oz* simulation; the second is an on-field evaluation carried out during the last phase of the system development.

6.1 Wizard-of-Oz evaluation

TAMIC-P was simulated by the Wizard-of-Oz technique [7], in which a human, i.e. the wizard, plays the role of the system and acts behind the compuer interface. The main hypothesis of this study was that TAMIC-P improves competence sharing, that is, it should support familiar and unfamiliar tasks with the same efficacy. In general, it was expected that unfamiliar tasks would be performed significantly better by TAMIC-P than by the INPS system; whereas familiar tasks would be performed just slightly better by INPS than by TAMIC-P . A comparative evaluation contrasting TAMIC-P with the system currently in use at INPS was run. To test TAMIC-P quality as interface, two tasks differing along the familiarity continuum were used. *Familiar tasks* addressed problems that employees normally encounter at the workplace; *unfamiliar tasks* addressed problems that employees encounter only exceptionally. The experimental manipulations gave rise to four conditions under which each participant was tested.

As expected (see Figure 3), the increase in complexity that derives from performing an unfamiliar task was detrimental only when the

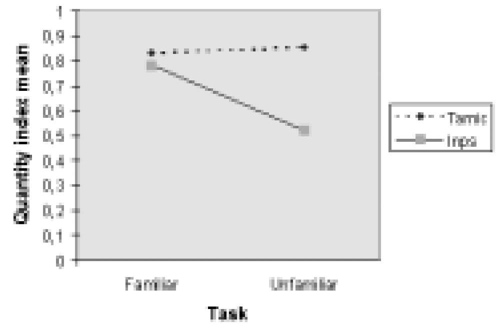

Figure 3. Average quantity index as a function of task and system. F=familiar; U=unfamiliar. Error bars show Mean Standard Errors.

user interacted with the INPS system. Indeed, TAMIC-P supported familiar and unfamiliar tasks with the same robustness.

6.2 On-field evaluation

A user-based evaluation was carried out to evince the most interaction difficulties encountered by a novice user performing an ecological task interacting with the system. The subjects of the experiment were 10 INPS operators. Each session included 20' of training, 30' for task execution and 10' for filling in a questionnaire. The usability problems encountered during the interaction were summarized according to the following taxonomy:

Attribution. Each problem was attributed to one of the two interaction agents. If it derived from improper use of the system or from a misunderstanding of its functionality, it was attributed to the user. If it derived from a design error or from a deficit in linguistic coverage it was attributed to TAMIC-P . In the latter case, the user's input was theoretically correct but for different reasons the prototype failed to process it.

Category. This refers to the type of problems encountered. Four main categories were adopted:

- *Natural Language*: problems related to linguistic errors; this was further distinguished into five sub-categories: a) lexical errors (in this case, the system showed the following error message: "word not recognized"); b) syntax errors: (the error message was "incorrect utterance"); c) contextual misinterpretation: (the error message was "inconsistent request"); d) data-base access: the system correctly parsed the request and showed the appropriate card, but the data displayed were not in conformity with the data-base content; e) uncertain problems: the parser failed, but it was not possible to understand the origin of the problem.
- *Task model*: problems due to a mismatch between the user's task model and the task model implemented in the system; we further distinguished between difficulties due to information form (the way in which information is displayed) and problems due to information content (what is actually displayed).
- *Interface*: problems related to direct manipulation, ranging from menu selection to shortcut commands; two sub-categories emerged: problems due to feedback (system answers) and difficulties in navigation (user operations).
- *Missing functionality*: functions requested by users but not implemented in the prototype; two sub-categories emerged: computational capabilities (some functions were not implemented) and

CATEGORY OF ERRORS	№	%
NATURAL LANGUAGE	**56**	**62.9%**
Lexical errors	18	20.2%
Syntax errors	8	9%
Contextual misinterpretation	6	6.7%
Data-base access	12	13.5%
Uncertain problems	12	13.5%
TASK MODEL	**12**	**13.5%**
Information form	2	2.3%
Information content	10	11.2%
INTERFACE	**11**	**12.4%**
Feedback	3	3.4%
Navigation	8	9%
MISSING FUNCTIONALITY	**10**	**11.2%**
Computational capability	9	10.1%
Hardware	1	1.1%
TOTAL	**89**	**100%**

Figure 4. Errors due to the system in five scoring categories.

hardware limitations (difficulties due to the specific hardware under evaluation).

Figure 4 reports the results of the evaluation. Note that NL is the most frequent category. Problems are mainly due to lexical errors (in some case easily solvable by expanding the lexicon, in other cases due to failure in multiwords recognition), followed by data-base access (probably due to unpredictable changes in the data-base), syntax errors (mainly due to difficulties in understanding dates), and contextual misinterpretation.

As for subjective judgments, user opinions on TAMIC-P were highly positive. Participants found the prototype easy to use and useful. They especially appreciated the possibility of accessing different data-bases from a single interface.

7 HARDWARE AND SOFTWARE ENVIRONMENT

This section schematically reports the software and the hardware requirements of the TAMIC-P application.

Used Software packages.

- Allegro Common Lisp (ACL) for UNIX and for Windows (both by Franz Inc.). This constitutes the main programming environment for the system.
- TCL/TK and TIX. TCL and TK are two packages for building graphical interfaces. TIX is an extension of TK. Both packages are Public Domain available.
- Allegro ODBC (Open Database Connectivity Interface), for UNIX and Windows. This is used to search and to update the TAMIC-P local database.
- DataDirect ODBC Pack by INTERSOLV (UNIX); Microsoft ODBC drivers (Windows). This is required by Allegro ODBC.
- Microsoft Access 97 is the local database used in the prototype.
- Microsoft Visual C++ 5.0 is used to implement the PDL routine, and to interface the middleware libraries.
- IBM/CICS Client v2.0.4 and IBM/Personal communication v4.11 with HLLAPI environment are necessary to interface the access procedures to the INPS archives.

System Requirements. Windows 95/NT. Intel Pentium or higher (Pentium II recommended). 32 MB RAM minimum (64+ recommended). 20 MB free space on the hard disk. TAMIC-P is available also in UNIX environment in off-line operations.

Accessible Users Archives. UNEX (contributions), ARCA (personal data), AZIENDE (Firms).

8 CONCLUSIONS

TAMIC-P, a multimodal dialog environment designed to help PA operators, has been presented. Evaluation of the prototype in real working situations has demonstrated its flexibility and adaptability to the aims and skills of different types of user. The use of NL technologies yielded significant advantages in making the interaction more easy and natural, in facilitating navigation through complex environments (data bases, conceptual data modeling, text retrieval), in aggregating heterogeneous data in a flexible way, and, finally, in increasing the transparency of the man-machine communication. From the PA point of view, many benefits can be obtained, in particular the reduction both of the training period and of the technical knowledge required to use the system. Finally, a more effective and clear interaction increases the degree of participation by the citizen to the social affairs.

REFERENCES

[1] I. Androstopulos, G.D. Ritchie and P. Thanisch, Natural language interfaces to databases - an introduction, *Natural Language Engineering*, 1(1), 1995.

[2] C. Bagnasco, P. Bresciani, B. Magnini and C. Strapparava, Natural Language Interpretation for Public Administration Database Querying in the TAMIC Demonstrator, in R.P. van de Riet, J.F.M Burg and A.J. van der Vos (eds.), *Applications of Natural Language to Information Systems*, IOS Press, Amsterdam, Netherlands, 1996.

[3] M. Bates, M.G. Moser and D. Stallard, The IRUS transportable natural language database interface, in L. Kerschberg (ed.), *Expert Database Systems* Benjamin/Cummings, 1986.

[4] M. Bates, M. G. Moser and D. Stallard, BBN Parlance Interface Software: System Overview, 1989.

[5] P.R. Cohen and J.W. Sullivan, Synergestic use of direct manipulation and natural language, in *Proc. of CHI'89*, 227-233, ACM Press, New York, 1989.

[6] P.R. Cohen, The role of natural lanaguage in a multimodal interface, in *Proc. of UIST'92*, 143-149, ACM Press, New York, 1992.

[7] N.M. Fraser and G.N. Gilbert, Simulating speech systems, *Computer, Speech and Languages*, 5, 81-99, 1991.

[8] D. Gentner and J. Nielsen, The anti-Mac interface, *Communication of the ACM*, 39(8), 70-82, 1996.

[9] B.J. Grosz, D.E. Appelt, P.A. Martin and F.C.N. Pereira, TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces, *Artificial Intelligence*, 32, 173-243, 1987.

[10] L.R. Harris, Experience with INTELLECT: Artificial Intelligence Technology Transfer, *The AI Magazine*, 5(2), 43-50, 1984.

[11] B. Magnini and M. Huber, Direct Manipulation of Information-Seeking Dialogues, ITC-irst Technical Report, 1999.

[12] M.T. Maybury, *Intelligent Multimedia Information Retrieval*, AAAI Press and MIT Press, 1997.

[13] G.A. Miller, WordNet: An on-line lexical database, *International Journal of Lexicography (special issue)*, 3(4), 235-312, 1990.

[14] J. Nielsen, *Usability Engineering*, Academic Press, San Diego, California, 1993.

[15] D.A. Norman and S.W. Draper, *User Centered System Design: New Perspectives on Human-Computer Interaction*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1986.

[16] S. Oviatt, P.R. Cohen and M. Wang, Toward interface design for human language technology: Modality and structure as determinants of linguistic complexity, *Speech Communication*, 15, 283-300, 1994.

[17] P. Resnik, Access to Multiple Underlying Systems in JANUS, BBN report 7142, Bolt Beranek and Newman Inc., Cambridge, Mass, 1989.