Application of Language Technology to Lotus Notes Based Messaging for Command and Control

Stephane Lajoie, Oliver Carr, Ahmad Hashemi-Sakhtsari and Michael Coleman
Information Technology Division
Defence Science and Technology Organisation
PO Box 1500, Salisbury, SA 5108, Australia
Phone: +618 8259 5651
Email: stephane_lajoie@hotmail.com
oliver.carr@dsto.defence.gov.au
ahmad.hashemi-sakhtsari@dsto.defence.gov.au
mike.coleman@dsto.defence.gov.au

Abstract

Speech can potentially provide a natural, efficient and flexible means of user interaction with computers in a messaging and communication environment. In presenting a spoken language data entry and database query prototype model, this report firstly discusses a technique for integrating a commercial large-vocabulary continuous speech recogniser with a Lotus Notes database user interface for data entry into Lotus Notes forms by speech. Secondly, the report brings together a number of software components, which include a commercial natural language processing product, in order to produce a query interface to access a Lotus Notes database. The military operators can query a database in plain English to obtain information and to monitor their internal workflow.

1. Introduction

Lotus Notes is used widely for messaging and communication throughout various headquarters in the Australian Defence Force. The Australian Deployable Joint Force Headquarters (DJFHQ) is one of these organisations. The Joint Logistics, Personnel and Manpower (J1/4) cell of the DJFHQ use a Lotus Notes database to log and distribute information regarding matters such as transportation, medical and legal events, and staffing levels. A user will enter information into the database by entering data into various fields. The fields contain various elements, for example formatted data tokens, attachments and formatted or unformatted text.

Australian Defence Industry contractors have implemented a database known as Command and Support System (CSS) database for use by the DJFHQ. The CSS database, as shown in figure 1, has five document types that the user creates, edits and sends. The documents are Log Entry, Action, Comment, Shift Handover and SITuation REPort (SITREP). Documents derived from these types can be sent to other cells within the DJFHQ and other external organisations. This database can be viewed by various parameters. For example, the view of the CSS log database shown in figure 1 is sorted by Event Date Time Group (DTG).

There are a number of advantages in applying language technology to produce an interface between humans and computers [Zue, 1994]. Speech is natural; humans speak before they can

read and write. Speech is efficient; generally 3 words or 10 phonemes are uttered per second, whereas typing speed rarely exceeds 60 words per minute. Finally, speech is flexible; it provides hands-free interaction. Therefore the goal of integrating language technology with applications is to increase reliability, performance and user satisfaction while at the same time decreasing stress and discomfort [Kloosterman, 1994].

In a survey on military applications of human-machine communication by voice, the most pervasive near-term application was identified as being voice data entry, which can range from entering numerical data to creating formatted military messages, to free-form report entry [Weinstein 1994]. Dragon NaturallySpeaking by Dragon Systems is a leading commercial speaker-dependant large-vocabulary continuous speech recogniser. This product was used to provide direct speech input to complete fields in the CSS database forms.

Although there has been recent advances in natural language processing, only systems in limited domain can be envisioned in the near term. A good example of this is question-answering domains such as database interfaces where the level of performance is regarded as acceptable [Bates, 1994]. English Wizard is a natural language processing software, produced by Linguistic Technology Corporation, which provides the capability of natural text query from a relational database by using natural language processing techniques.

As well as being able to enter data by speech, this paper presents a spoken language dialogue system that has been produced by integrating the two language technologies, Dragon NaturallySpeaking and English Wizard, with Lotus Notes databases. Since both Dragon NaturallySpeaking and English Wizard are Commercial-Off-The-Shelf (COTS) products, this paper also highlights some of the practical limitations in integrating the COTS language technology with military messaging systems.

2. Integration of Language Technology with Lotus Notes Databases

There were two aims in integrating speech recognition and natural language query capabilities with Lotus Notes databases. The first aim was to allow the user to dictate the contents of a field in a Lotus Notes database, or to select a field for data entry via a combination of keyboard, mouse and speech. Speech recognition technology can potentially provide users with means to enter data in a faster and more accurate way.

The second aim was to improve the user’s ability to query the Lotus Notes databases. The use of natural language processing allows users to obtain information from a Lotus Notes database by using queries constructed in plain English. The possibility of improving the search capabilities within Lotus Notes by allowing the user to ask questions that can not be ordinarily asked from Lotus Notes was also to be investigated. This would enable operators at the DJFHQ to monitor their internal workflow more efficiently.

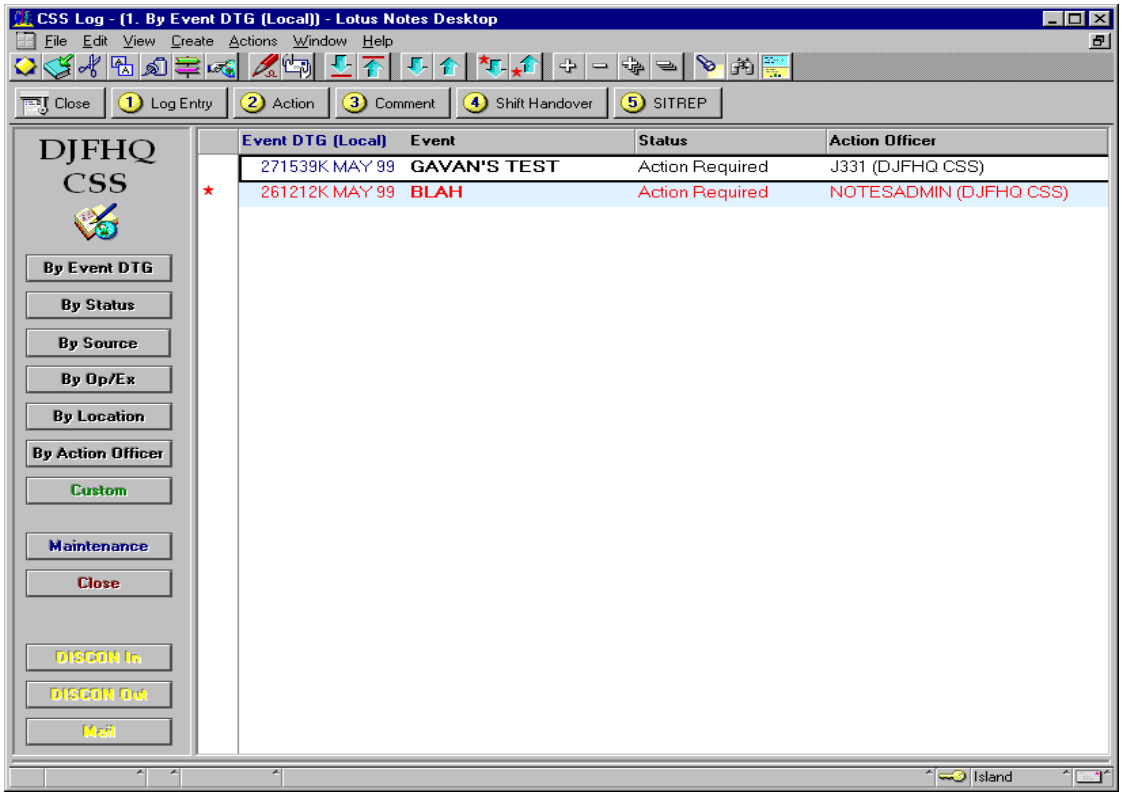

Figure 1. Main View of Command Support System Database

2.1 Integrating Dragon NaturallySpeaking Speech Recogniser with a Lotus Notes Database for Data Entry

To achieve the first aim, Dragon Systems provide several possible methods of integrating Dragon NaturallySpeaking with an application.

The quickest and easiest method of integrating Dragon NaturallySpeaking with an application is by using the scripting commands [Dragon Systems, 1997]. The scripting command controls Lotus Notes by simulating keyboard, mouse and Dynamic Data Exchange (DDE) events that are generated when the recogniser hears a word or a phrase that is predefined and has a simple syntax. In order to use the scripting commands, knowledge of how to navigate an application via keyboard and mouse events is required. Figure 2 below shows how the scripting commands simulate what the user can do with the keyboard and mouse.

The scripts can be made application and user specific. A default set of scripts can be given to a new user of Dragon NaturallySpeaking. The scripting language has functions which use Dynamic Data Exchange (DDE) calls to start and close applications and switch between windows in the Windows environment. The use of scripting language does not require the source code of an application in order to integrate speech with that application.

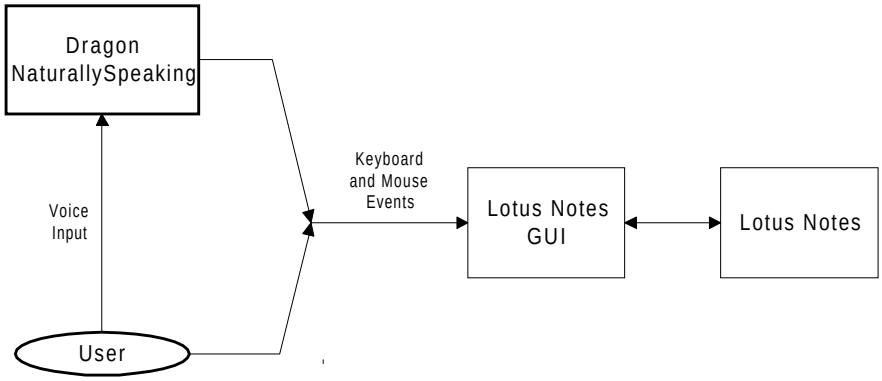

Figure 2. A block diagram showing how Dragon scripting commands are sent to a Lotus Notes database to simulate keyboard and mouse events.

Since the scripting language can only send keyboard, mouse and DDE events and is context insensitive, i.e. it can only differentiate between windows that have unique titles. Although a window can be brought into focus, fields from a form in that window can not be. For example, a word can be mistakenly uttered into a number field, but since the speech recogniser is unable to distinguish which field is in focus, the word recognised will be displayed in the number field. Context is required in error recovery since re-training erroneous utterances can automatically be initiated taking into account context.

Not withstanding that the context insensitivity was a problem, it was regarded that building a prototype for data entry using scripted commands would allow user requirements to be elicited. Once the requirements are obtained, a robust and context sensitive technique, such as use of ActiveX components can be employed. ActiveX creates and responds to custom events within an application.

Dragon NaturallySpeaking ActiveX components can be integrated into any Integrated Development Environment (IDE) that supports ActiveX components. These ActiveX components have been tested with the Delphi 4, Visual Basic 5 and Visual J++ 6 IDE's. Figure 3 shows how the ActiveX components can be integrated with Dragon NaturallySpeaking speech recogniser. Data flow between an application and Dragon NaturallySpeaking through ActiveX components is seen in Figure 3. These components allow full integration of Dragon NaturallySpeaking with an application [Dragon Systems, 1998]. The available functions provided by the ActiveX components include:

- Switching between users and creating new users
- Calling microphone setup program and initiating general training
- Executing Scripting Commands
- Generating custom events for each utterance depending on the current context of the application
- Creating commands for control and navigation within an application
- Dictation into free text fields
- Text to speech synthesis

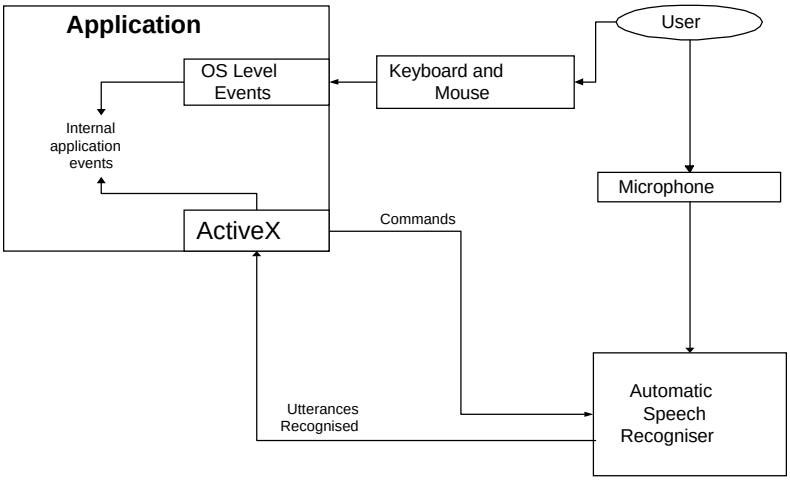

Figure 3. Data flow between a speech recogniser and an application through ActiveX components.

2.2 *Integrating English Wizard with a Lotus Notes Database for Data Query*

As stated earlier, the second aim was to improve the users' ability to query the Lotus Notes databases so that they can obtain information and monitor the workflow within their messaging environment. The issues to be explored were improving the resolution of the data to be searched, reducing the system response time, and allowing for a broad range of questions to be asked.

Since Lotus Notes does not allow for an Open DataBase Connection (ODBC) to a database, the strategy adopted to achieve this aim was to create a mirror image of the Lotus Notes database in a relational database and apply a querying interface to the relational database. There were two reasons for transferring data into the relational database; firstly, it provided a more efficient data searching capability, and secondly the meta-data from the relational database was used to construct most of the syntactic and semantics rules used in the natural language processing within English Wizard.

English Wizard is a natural language processing software that enables a user to query a database by entering a question as an English expression, either by voice or keyboard, rather than writing queries in Structured Query Language (SQL). English Wizard incorporates a dictionary [Linguistic Technology, 1997a and 1997b] that allows relationships and definitions to be constructed, such as:

- Relationships between tables in the database, as defined in a Map, through key-word labeling
- Definitions of group values, root words and formulas and functions
- Definition of synonyms, e.g. 'author' is equivalent to 'written by'.

The application developer provides some syntactic information such joins missed by the automatic construction, but the developer provides mainly semantic information.

English Wizard can not translate every query put to it. The SQL English Wizard produces must be executable in the information domain of the relational database. Therefore a user query must result in obtaining either a collection of fields within the database or some quantitative functions (such as a total or an average value) on a collection of fields. In summary, the set of queries English Wizard can answer is limited by the structure of the database it is connected to. The structure of the InterBase database that was created from the Lotus Notes database is shown in Figure 4. Here the relationship defined by the word 'wrote' between the tables called Master Table and Author Table associates document identification values and the author of the document.

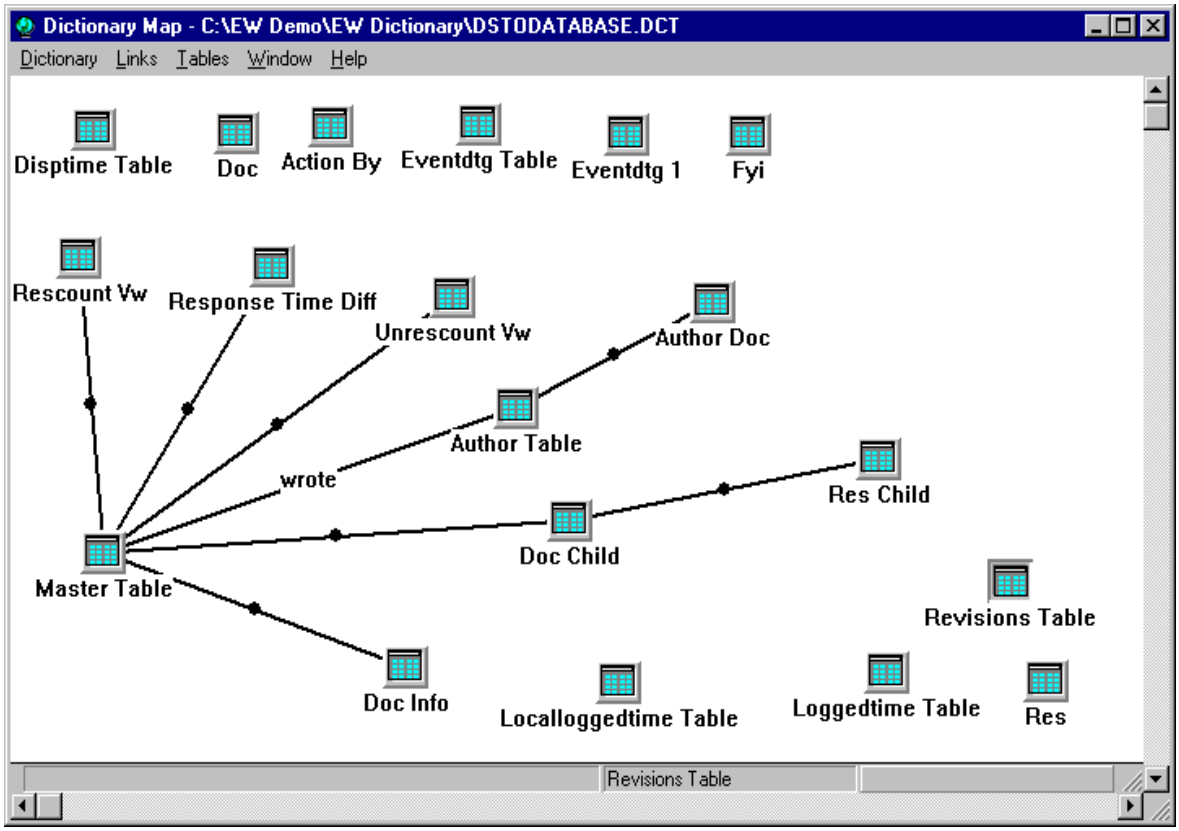

Figure 4. English Wizard Map.

English Wizard, with its dictionary, was connected to a relational database (InterBase by Borland) through an Open Database Connection (ODBC). English Wizard converts an English question as text to a SQL query through the ODBC using a combination of meta-data information from the relational database and the question.

English Wizard executes the SQL it has constructed and the records are retrieved from the InterBase database and are displayed in Lotus Notes Query Program or as a Lotus Notes view. A Lotus Notes database view is a list of documents that meet some criteria. Figure 5 illustrates the components of the Lotus Notes Query Program.

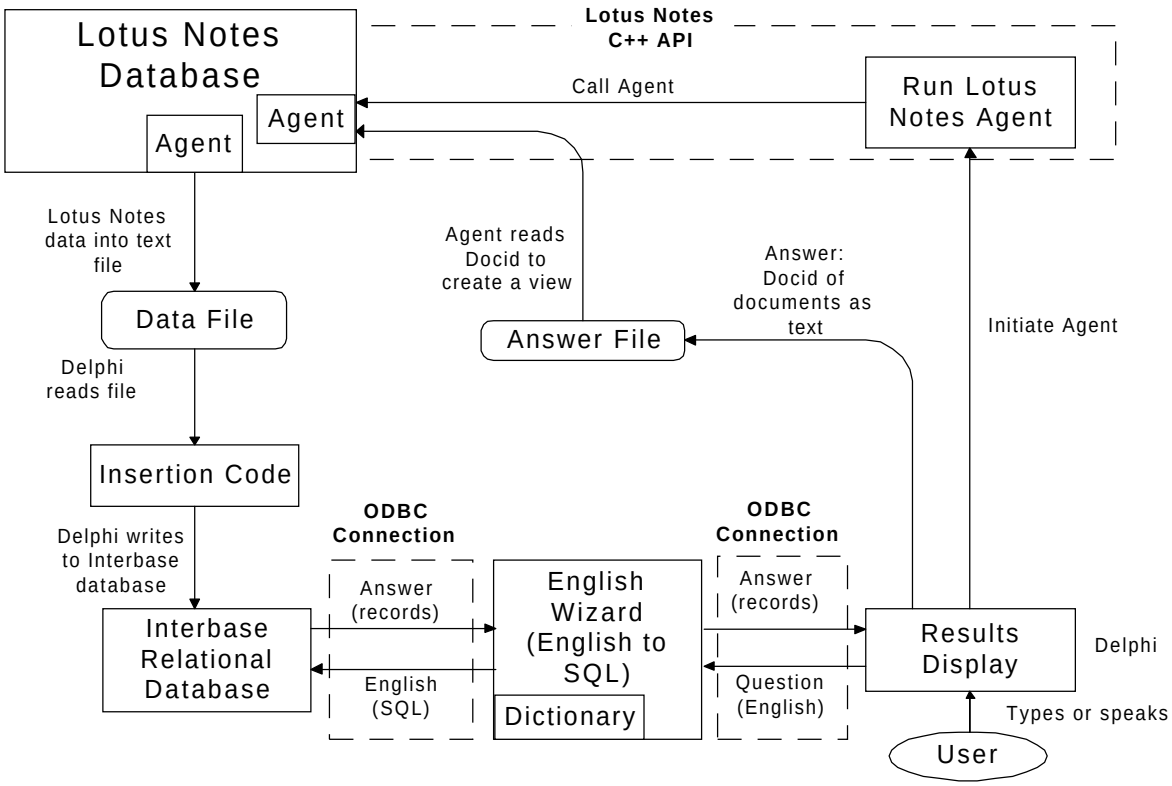

Figure 5. A block diagram of Lotus Notes Query Program.

The user interface of the Lotus Notes Query Program is shown in Figures 6a and 6b. There are basically two types of questions that the user can ask to interrogate the relational form of the Lotus Notes database. Queries like ‘What documents were written about "RED CROSS"' return records. Aggregate questions similar to ‘what is the average response time’ or ‘how many documents did author "CN=J401" write’ return a value.

The user can open, read and modify the data in any document within a Lotus Notes view, if the answer is represented in the form of a view. On closing a modified document, the Lotus Notes Query Program will update the document data in the Lotus Notes Query Program InterBase database. The document will also be updated in the Lotus Notes database.

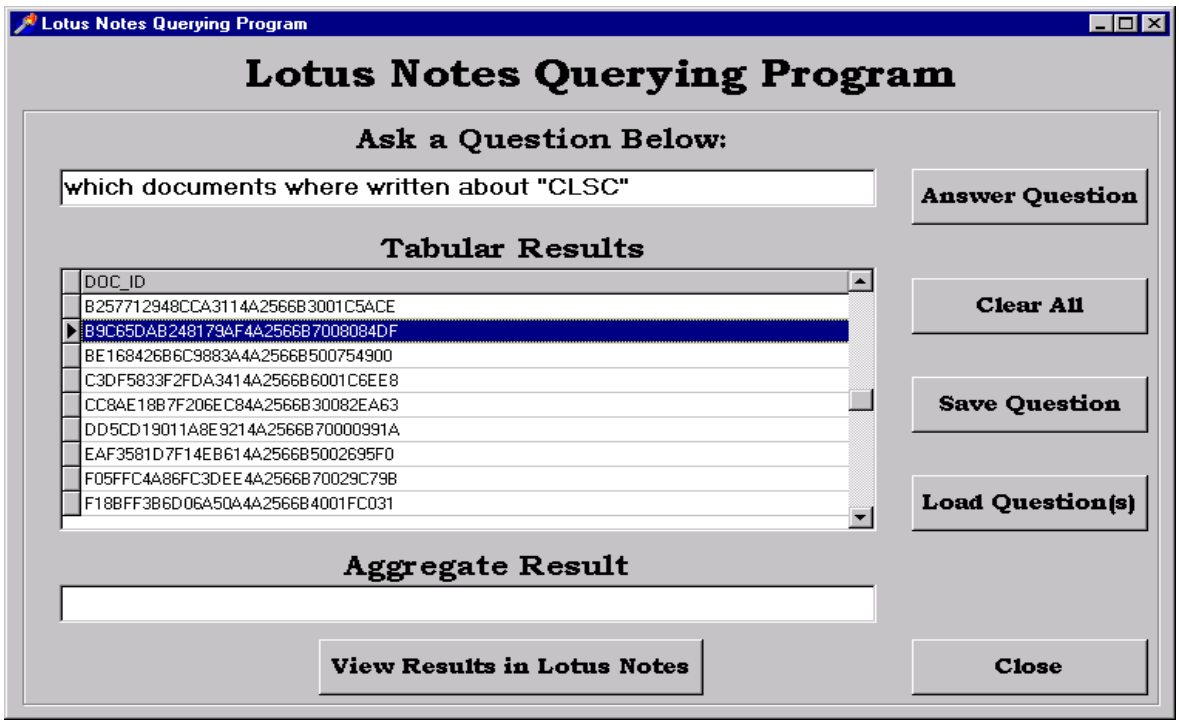

Figure 6a. Lotus Notes Query Program showing tabular results.

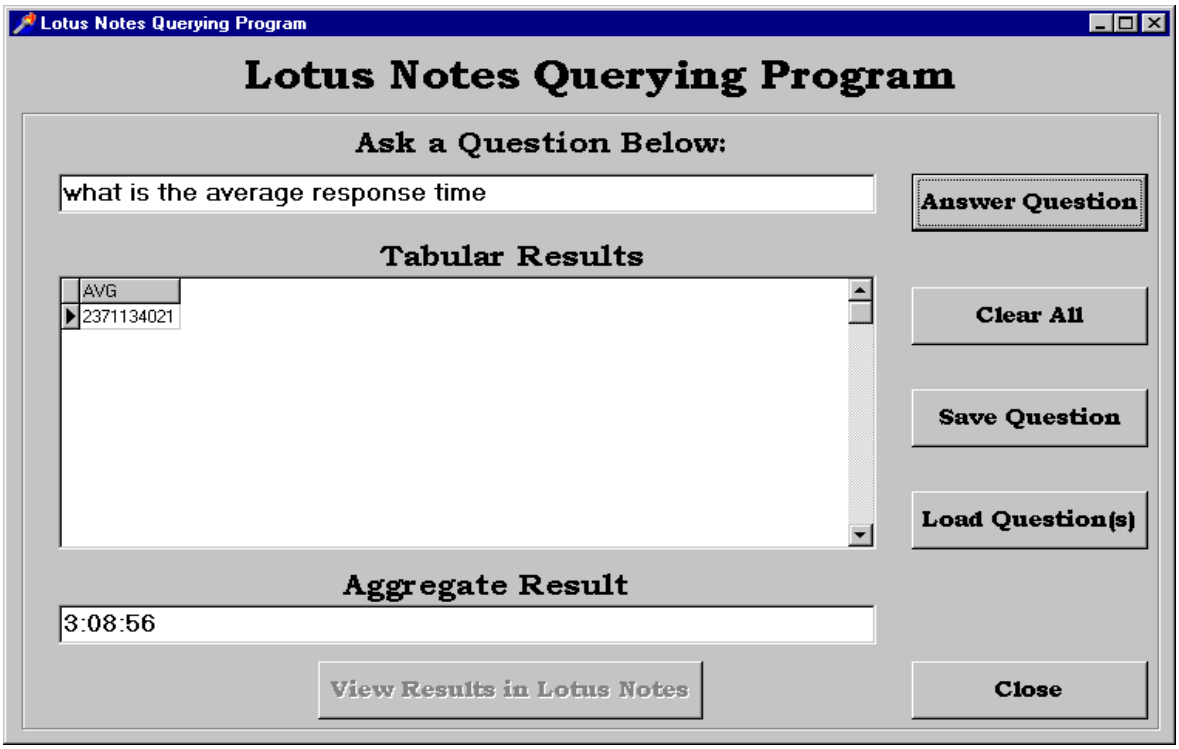

Figure 6b. Lotus Notes Query Program showing an aggregate result.

3. **Discussion**

The integration of both speech recognition and database query technologies as described has raised many issues. Some of these issues are technical and will be addressed with continued development of the technological concepts. However, other issues are of a socio-technological nature. These have arisen from the military domain for which these concepts have been prototyped and developed. In order to produce an effective user interface with a speech recognition capability, studying human to human conversations in the application domain is required [Yankelovich *et al.*, 1995]. The issues of concern include developing a vocabulary of military acronyms and jargons, user interface design for communication by speech, and system scalability. Addressing these issues will be paramount in the development of speech recognition and database querying technologies for future military messaging environments.

The Australian Defence Force (ADF) produces and maintains a document, called ADFP 123 - Acronyms and Abbreviations, which contains the definitions of all the acronyms and abbreviations used within the ADF. An important issue in this research has been to gain knowledge on how the users pronounce specialised words or utter acronyms and jargon to eachother and to the speech user interface. It is noted that in agreement with [Ellis, 1999] the unpredictability of the technology continues to result in user frustration. However accurate domain specific vocabularies will result in less user frustration by limiting the user vocabulary development.

The success of a speech recognition capability within a military messaging environment will depend on the success of research into training and enrolment, error detection and correction techniques [Noyes and Frankish, 1989]. The enrolment text used for the Dragon NaturallySpeaking software was a selection that Dragon Systems provides from Arthur C. Clarke "3001 – The Final Odyssey". It is possible to change this training text to be more relevant to the application domain, thus increasing the correct recognition rate for the users. Thus, knowledge of the application domain will in many ways influence the success of a speech recognition capability in military messaging systems.

The ease of error correction and recovery determines the user acceptance of a speech recogniser. In a recent study [Karat *et al.*, 1999] the most common command used by users of continuous speech recognisers was the command that reversed the immediately preceding action. In this study the average correction time for words and phrases was three times as long as dictating those words or phrases. Therefore, although it would often be most efficient to enter data by a combination and speech and manual input modalities, nevertheless error correction should be done through either manual input alone or through an interactive correction mechanism by switching between modalities [Suhm *et al.*, 1999].

In order to create new entries in a Lotus Notes database, the Lotus Notes database Client is used When the Lotus Notes Query Program creates a view in Lotus Notes as a result of user querying the system, the view is not updated in the Lotus Notes Client. The user is forced to close and reopen the Lotus Notes Client in order to re-display the view created as a result of a query.

While the Lotus Notes Query Program was built as a concept demonstrator, the scalability of the technology used imposes another limitation. The system as described exports the Lotus Notes

data to a text file that is imported into an InterBase relational database. This approach would not be feasible for a large scale distributed system due to the slow speed of a Lotus Notes agent writing the text file, and the considerable processing time required to import data from the text file to the relational database. Security is another consideration resulting from the use of text file. Both Lotus Notes and (most) relational databases have control mechanisms to prevent unauthorised access to the data they contain. A text file has no control mechanisms and is easily read.

Another scalability issue relates to the relational database itself. Since the data in the relational database mirrors the data in the Lotus Notes, the size and complexity issues in the Lotus Notes data may be passed on to the relational database. The relational database is designed for the types of queries the users require. Thus, the more complex the Lotus Notes database is the more difficult designing the relational database would become.

The use of Java, namely its Java DataBase Connection (JDBC) to communicate directly with a Lotus Notes database via the Java classes will solve the problems outlined above.

4. **Future Directions**

Future research and development of the concept demonstrator described in this report will be done using the Java programming language to improve the integration of speech technology with Lotus Notes databases so that the integration will be robust and context sensitive. Context awareness should result in a higher recognition accuracy and better error recovery.

Improving the user interface and the mechanisms for querying Lotus Notes databases will also be attempted. It is envisaged that a user interface for a Lotus Notes database that allows the user to create, edit and view entries in the database will be produced. The user will be able to enter information by speech as well as keyboard and mouse. An improved searching and querying capability will also be provided to the user through the speech enabled user interface. Internet browsers (e.g. Internet Explorer 4.0) are being used at the DJFHQ to read Lotus Notes databases. Since this GUI will be written in Java it can be used within an Internet browser such as the Internet Explorer 4.0.

Naturally, a user-orientated speech recognition interface improves the useability of that interface [Kloosterman, 1994]. For natural and efficient speech user interfaces careful attention must be applied to the transaction and dialogue design. A speech interface can be improved by using directive prompts and confirmation protocols. These prompts could be visual or auditory by incorporating text to speech synthesis [Peckham, 1984]. Furthermore, providing user control to access instructions and correct errors would provide further enhancement [Kamm, 1994; Yankelovich *et al.*, 1995]. The use of Java will allow the speech recognition software to provide visual and auditory feedback mechanisms to aid error recovery and correction.

5. **Conclusions**

The language and database technologies described in this report combine to form a concept demonstrator of how various filed in forms from a Lotus Notes database can be filled through direct speech input and how natural language queries can be made from a Lotus Notes database.

Dragon NaturallySpeaking version 3.52 has been integrated with a Lotus Notes database known as Command Support System database. Log operators from the Logistics and Administration Cell (J14) within the Deployable Joint Force Headquarters have been trained to use Dragon NaturallySpeaking speech recogniser for data entry. In doing so information about how acronyms are pronounced has been elicited and clues have been gathered as to when, why and how the operators use the Lotus Notes databases. Speech integration has provided flexibility and naturalness in user interaction with the forms within the CSS database and has indicated that direct speech input can provide a potentially faster and more accurate means of user interaction with messaging and communication systems.

The query interface to the Lotus Notes database has enabled a user to ask questions in plain English and obtain documents containing information or numerical values related to a certain action and response. Although the possible range of queries is limited, the prototype model has thus far enabled the operators at the DJFHQ to monitor their internal workflow in an efficient manner. It has improved resolution of the data that can be searched, has reduced the system response time, and has allowed operators to ask a broader range of questions than they had been able to do from the Lotus Notes databases.

Further development of the language technology concept demonstrator will attempt to improve the querying and data extraction from a Lotus Notes database to a relational database in order to meet the client technology and security requirements. It is intended that a natural language query capability with a Lotus Notes database via a Java based user interface will be integrated in the concept demonstrator so as to improve user searching and querying the databases. The integration of database and language technologies will enhance the efficiency of messaging and communication in future military command and control environments.

6. **References**

[Bates, 1994] M. Bates. *Models of Natural Language Understanding*. In: D.B. Roe, J.G. Wilpon (eds.): *Voice Communication between Humans and Machines*. National Academy of Sciences, 238-253, 1994.

[Dragon Systems, 1997] Dragon Systems. *DragonDictate Macro Language - Guide and Reference*. Dragon Systems Inc., Second Edition, 1997.

[Dragon Systems, 1998] Dragon Systems. *Dragon NaturallySpeaking ActiveX Components – Guide and Reference*. Dragon Systems Inc., Version 1.0, 1998.

[Ellis, 1999] D. Ellis. *Dragon NaturallySpeaking Preferred*, Australian Personal Computers, March 1999, p. 42.

[Linguistic Technology, 1997a] Linguistic Technology. *English Wizard - Dictionary Administrator's Guide*. Linguistic Technology Corp., Littleton, MA, USA, 1997.

[Linguistic Technology, 1997b] Linguistic Technology. *English Wizard - Programmer's Reference*. Linguistic Technology Corp., Littleton, MA, USA, 1997.

[Kamm, 1994] C. Kamm. *User Interfaces for Voice Applications*. In: Roe, D.B., Wilpon, J.G. (eds): Voice Communication between Humans and Machines. National Academy of Sciences, 422-442, 1994.

[Karat *et al*, 1999] C-M. Karat, C. Halverson, D. Horn, J. Karat: *Patterns of Entry and Correction in Large Vocabulary Continuous Speech Recognition Systems*. Proceedings of Computer Human Interaction ’99 Conference. May 568-575, 1999.

[Kloosterman, 1994] S.H. Kloosterman. *Design and Implementation of a User-Orientated Speech Recognition Interface: the Synergy of Technology and Human Factors*. Interacting with Computers **6**(1) 41-60, 1994.

[Noyes and Frankish, 1989] J.M. Noyes, C.R. Frankish. *A Review of Speech Recognition Applications in the Office*. Behavior and Information Technology **8**(6) 475-486, 1989.

[Peckham, 1984] J.B. Peckham. *Automatic Speech Recognition - a Solution in Search of a Problem?*. Behavior and Information Technology **3**(2) 145-152, 1984.

[Suhm *et al*, 1999] B. Suhm, B. Myers, A. Waibel. *Model-Based and Empirical Evaluation of Multimodal Interactive Error Correction*. Proceedings of Computer Human Interaction ’99 Conference. May 584-591, 1999.

[Weinstein, 1994] C.J. Weinstein. *Military and Government Applications of Human-Machine Communication by Voice*. In: D.B. Roe, J.G. Wilpon (eds.): *Voice Communication between Humans and Machines*. National Academy of Sciences, 357-370, 1994.

[Yankelovich *et al*, 1995] N. Yankelovich, L. Gina-Anna, M. Marx. *Designing SpeechActs: Issues in Speech User Interfaces*. Proceedings of Computer Human Interaction '95 Conference. ACM, 1995.

[Zue, 1994] V. Zue: *Human Computer Interaction using Language based Technology*. Proceedings of IEEE International Symposium on Speech, Image Processing and Neural Networks. April I-VII, 1994.