THE AUTOMATION OF HELPDESKS

L.M. Coventry and T.B. Kane

Department of Computing Science,
University of Stirling, Stirling,
FK9 4LA.
telephone: (0786) 67432
fax: 0786 64551 International +44786 64551
email: (JANET) lmc@uk.ac.stir.cs tbk@uk.ac.stir.cs

ABSTRACT
Helpdesks are becoming increasingly more important in large organisations. These organisations may be commercial or academic. However, management of such helpdesks is becoming increasingly more difficult as the amount of knowledge that must be acquired and maintained by the advisor becomes rapidly unmanageable. The sources of information are highly distributed and the domains are becoming larger. No one person can be an expert in all necessary domains. Many attempts are being made to increase the effectiveness of these helpdesks. The solution may lie in an intelligent assistant for the advisor. This paper describes such a system. The advisors remain central to the process. They may train the system and their ability to communicate effectively with a variety of users, each with different experience and knowledge, modes of learning and information requirements cannot be replicated by a software system. However, an intelligent system can help the advisor to maintain their knowledge, acquiring knowledge from a variety of sources and assisting in the diagnosis of problems. The advisor formulates queries from users for the system, receives the information from the system and communicates this in a manner appropriate to the user who had the initial problem.

KEYWORDS: Agents, blackboard architecture, advisory system, interface, answer filters.

INTRODUCTION
Helpdesks have traditionally been employed by large organisations to provide assistance or information in person or via the telephone to customers, employees and other interested parties.

Managing helpdesks is becoming increasingly difficult because of the following trends:

- the volume of calls to helpdesks is increasing,
- the complexity of knowledge required is increasing,
- the diversity of knowledge required is increasing,
- the cost of operating the helpdesks is increasing,
- the variability of the user community is increasing.

Helpdesks are especially important in providing technical information about procedures and regulations (eg answering questions regarding...) The advisor must build an awareness of each client's capabilities and experience, and must have the ability to absorb information about new systems and updates of old systems.

Many attempts are being made to increase the efficiency and effectiveness of helpdesks; automation is the most serious one. Traditional computerised solutions include searching databases. However, because of the increased complexity and quantity of the stored information, this approach is becoming less effective.

This paper describes a semi-automatic helpdesk system. The functionality of the system has been developed and the interface to the system is currently being implemented.

A SOLUTION
The solution may lie in the creation of an expert system that will "assist the assistors". The developing areas of Distributed Artificial Intelligence and Knowledge Based Systems offer radical new ways to approach the problem of alleviating the workload on the advisor. Barr (1990), a consultant specialising in the automation of helpdesks, believes that intelligent helpdesk systems provide one of the most cost-effective means of knowledge distribution.

An example is a helpdesk that provides a hotline for employees to troubleshoot problems in computer hardware and networks. The requests are made by phone. Helpdesk operators call an Expert System for help. The information provided by the employee is keyed in using a simple menu system. The Expert System then asks a few questions and provides brief procedures to help diagnose the problem. The Expert System provides diagnosis and instructions on what to do next. The consultation takes about 10 minutes (Karon 1988). The system expediates the work of the helpdesk operators, and also allows the operators to diagnose problems that would previously be directed to a higher level of expert. The system can also be used to train the helpdesk workers.

THE FUTURE

The advisor will remain central to the process, to which the system will be an intelligent assistant. The advisor will interact with the system, formulating user requests into meaningful queries. The advisor can also train the system to improve its responses and interpret the responses for the inquirer. Abraham *et al.* (1991) provide a complete discussion and suggest the following basic features for any such system:

- provide assistance to a advisor,
- collect information from the user network,
- store the user information in a database,
- use stored data from past consultations to personalise the current session,
- use data from past consultations to induce new rules and thus facilitate learning over time,
- Use historical data to produce management reports (highlight specific problems and trends).

THE KCM PROJECT

An semi-automated intelligent helpdesk is currently under development at Stirling University as part of the Knowledge and Constraint Management (KCM) project. It is an interdisciplinary undertaking, with team members coming from the fields of Artificial Intelligence and Human-Computer Interaction. The system's task is to assist advisors in a centralised Computer Support Service. Advisors are highly skilled individuals. An advisor is required to have knowledge of a number of different systems and applications available on those systems. The advisors must continually update their knowledge to keep up with developments and to be helpful. To do this, the advisors must have access to system information held in diverse forms at different locations. Therefore it must be capable of operating over a computer network. These forms include databases, CD-ROMS and online manuals, computing officers and other human experts. They also have access to a history of queries if the enquirer has asked for information on a previous occasion. An intelligent helpdesk, to assist in the advisor's task, must also be able to acquire information from such sources. It must be able to assist in the diagnosis of problems, in the maintenance of client profiles and to do all this in a manner which results in an intelligent explanatory report for the advisor.

The System

The helpdesk is based on a network of communicating asynchronous blackboard systems (Englemore and Morgan, 1988) written in SEPIA Prolog. A number of system agents are used to interface application programs on remote machines (eg ORACLE used to log client–advisor question–answer information, BRS used to store advisory newsletters in an easily accessible free text retrieval format). This was deemed the most promising structure to implement the system as it must be capable of problem solving, of integrating information from a variety of knowledge sources held in diverse forms and of permitting incremental development and modification. A blackboard, partitioned and layered as required, holds the current state of the system. A scheduler calls upon the knowledge sources to write on the blackboard using their own expertise. That expertise may be of any kind, possibly to generate a hypothesis concerning the solution to part of the problem, to access a remote database for information relevant to the current problem or to construct an explanation of the solution to a problem. The tasks of interfacing with the user and carrying out the required computations have been separated. This allows for independent development of both aspects of the system.

At present, the helpdesk blackboards are functional and the interface blackboard (XBS) is being written.

The Blackboards

The elements of the helpdesk system architecture are

BEBS: Back End Blackboard System. This coordinates the Client–Advisor dialogue and issues commands and messages for the other blackboards. It also waits for responses from the other blackboards and processes any immediately.

FEBS: Front End Blackboard System. Each conceptual task that the helpdesk can perform is assigned to a specially constructed system agent. FEBS deals with all the calls into the distributed agent system which must be made as a result of the Client–Advisor dialogue. FEBS can also examine the progress of these agents and interfere with their execution.

XBS: X-interface Blackboard System. This is the advisor's interface. It mediates between the advisor and the helpdesk system by providing an X window editor for input of queries by the advisor.

DBS: Diary Blackboard System. This is responsible for checking previous Client–Advisor dialogues and making profiles, judging system performance, developing explanations and learning.

For further system details see Bland *et al* (1992).

This architecture allows for high level separation of concerns and orthogonality of concepts. The most important elements are BEBS and FEBS. The system related functions are handled by FEBS, which acts as an agent activation controller, and coordinator of requests from other blackboards. This leaves BEBS free to formulate questions, construct messages, deal with conflicts and make new proposals. The system functions are separated from the high-level problem solving functions.

The Advisor's Interface

The advisor uses an X windows interface to the system. The interface allows a combination of graphical interactions and restricted natural language (Guindon *et. al.* 1987; Guindon 1988). The interface design was derived from an analysis of two years of advisory logs. Queries were split into types and a shorthand notation for the advisor and functional notation for the helpdesk devised.

We are currently working on a scheme for graphical query input which presents the advisor with an X window editor for constructing standard structured queries. These structures reflect the underlying nature of the client's query and so provide the helpdesk with a clue as to how to answer it.

Standard Query Types

From an analysis of session logs developed at the Computer Support Service over the first year of the helpdesk's function, the basic query types to emerge are:

"hows"
- how to display file on screen
- how to print on psychology's printer
- how to get greek letters
- how to read mail

"cans"
- can i have unix on a PC
- can cricket graph plot missing values

"wants"
- want advice on which mouse to buy
- want copy of kermit
- want to transfer files to supervisor

"unable"
- unable to send mail to BERKELEY.QAL.EDU
- ps7ee1 unable to log in

"problems"
- problem formatting 2Mb disk
- problem with 31T6 setup
- problem joining files with cat

"whats"
- what spreadsheet is available on the mainframe
- what plotter to use on VAXB
- what equipment to buy

These types are listed in descending order of frequency of occurrence in the session log. The examples given along side each type of request are examples from the log. The examples used illustrate the vagueness of requests, diversity of knowledge required by an advisor and forms of language used when advisors log the problem.

We envisage a simple input routine based on these main query types, inputting in this way means that the query is already broken down into semantic parts. Given that we are able to isolate the important parts of the query and use a vector information retrieval method (Dunlop, 1991), we will have successfully taken a dialogue between the client and advisor and created from it a query which can be passed to the system. At this point, the links between the query formulation and FEBS are made. Agents in FEBS deal with each question type. Agents will know where to look for the answers and how to deal with the possibly large quantity of information returned. Here the advisor has a key role, deciding on what level of explanation will be most suitable for the particular client. This role brings in the notion of *Answer Filters*. These act on the same information to extract different aspects pertinent to individual clients.

The Advisors' Connection with the Helpdesk

The advisor begins a session by logging on to an X terminal which is attached to the system via XBS. The advisor now has access to a UNIX operating system which can be used to pursue his own enquiries and also access to a set of icons which can awaken the Helpdesk and ask it to perform some function.

The advisor communicates directly with BEBS by means of a menu and typing information at the terminal. This blackboard is the first level of abstraction offered to the advisor and it deals directly with the advisor while instructing FEBS to carry out system specific operations. While these operations are carried out, the advisor is free to do other tasks. An icon is sent to the screen when the reply is ready to be displayed. XBS is also responsible for packaging the results for presentation on the advisor's screen. Error states, such as loss of functionality in the distributed network can also be reported in a similar manner, keeping the advisor informed about the current state of investigation.

Long Term Diary

An agent will maintain a history of each query–answer success or failure in a long term diary file. Such files would then provide the raw data for DBS.

CONCLUSIONS

This paper summarizes ongoing work on the automation of a helpdesk. The design and implementation put forward is for a semi-automatic system with the advisor maintaining a key role in the process. We are providing a practical way to deal with the query input stage and also working with Answer Filters to apply to the total information returned by the system in order to extract a view of that information pertinent to each individual user.

This system has been developed with the full cooperation of the advisors in the Computer Support Service and has been designed with them and the tasks they carry out in mind. However, there is substantial research to be undertaken to evaluate the effectiveness of such systems. The views of the advisor should be paramount, this will take place when the system is fully operational. Then as a result of these usability studies, the system can be further refined.

At present, the functionality of the helpdesk has been implemented and the prospects for an effective helpdesk look promising. However, the interface to the system is crucial and once this is fully implemented the consequences for the future of the system will be fully realised.

REFERENCES

[1] A. Barr. *The Evolution of Expert Systems.* Aldo Ventures, Palo Alto, 1990.

[2] R. Bland, J.R. Cowie, T.B. Kane, C. Rattray, and I.R. Wilson. The architecture of an intelligent helpdesk system. In *Proceedings of the 5th Florida Research Symposium*, pages 181–185, 1992.

[3] M. Dunlop. Multimedia information retrieval. Technical report, University of Glasgow, 1991.

[4] R. Englemore and T. Morgan. *Blackboard Systems.* Addison-Wesley, Reading, Massachusetts, 1988.

[5] Abraham D.M. et al. Expertech: Issues and development of an intelligent help desk system. *Expert Systems with Applications*, 2(4), 1991.

[6] R. Guindon. How to interface to advisory systems? Users request help with a very simple language. In *Proceedings of the ACM SIGCHI Conference: Human Factors in Computer Systems*, 1988.

[7] R. Guindon, K. Shuldberg, and J. Conner. Grammatical and ungrammatical structures in user-advisor dialogues: Evidence for sufficiency of restricted languages in natural language interfaces to advisory systems. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 41–44, 1987.

[8] P. Karon. Help desk gets help from trouble-shooting system. *PC Week*, January 26 1988.