Designing an Intelligent User Interface for Contact Centres

Akash Singh, Janet Wesson
Department of Computer Science and Information Systems
Nelson Mandela Metropolitan University, PO Box 77000, Port Elizabeth, 6031
Tel: 041-504-2323 Fax: 041-504-2831
Email: {Akash.Singh, Janet.Wesson}@nmmu.ac.za

Abstract—Contact centres (CCs) are organisational units that aim to achieve an elevated sense of loyalty and satisfaction from customers and employees whilst keeping the operational cost at a minimum. Currently seventeen percent of all calls received by CCs are not resolved on the first call. This is due to the inability of the contact centre agents (CCAs) to diagnose the problem and find an adequate solution. The aim of this paper is to discuss the development of a model and prototype that employs an intelligent user interface (IUI) in order to improve CC operations.

Index Terms — Contact Centres, Intelligent User Interfaces, Task Modelling, User Modelling.

I. INTRODUCTION

CONTACT centres (CCs) are often the first form of interaction that customers experience with a company [1]. The role of a CC can vary from logging and solving customer queries and transactions to broadcasting marketing propaganda [2].

The Merchant Global Benchmarking Report in 2005 [3] revealed that only 71% of all customer queries were resolved on the first call by the agent who answered the call. Ten percent of all queries from calls answered were escalated to second and third tier agents and 17% of all customer queries were not resolved on the first call. The inability to solve these queries reside in the ineffectiveness of the contact centre agents (CCAs) to diagnose the problem and find an adequate solution. Thus, there is a need for an efficient and effective solution that will enhance the problem solving process.

Intelligent user interfaces (IUIs) aim to amplify the rate of information flow between humans and computers [4]. This is acomplished by delivering interaction between the user and the computer that is efficient, effective and natural [5]. The application of IUIs to the domain of CCs may therefore improve the efficiency and effectiveness of call resolution.

Section II of this paper discusses the problem domain in more detail. Section III describes the typical tasks and processes that need to be performed in the process of logging and resolving a customer's query. The design of an IUI model for CCs is discussed in Section IV. Section V describes how the components of the proposed model were implemented in order to validate the proposed model. Section VI focuses on future work.

II. RELATED WORK

A. Contact Centres

A contact centre (CC) refers to a call centre that provides a value added service using more than one electronic medium [6]. CCs are beneficial as they provide increased accessibility for customers and allow for transactions and queries to be handled in an efficient manner [2].

CCs can be classified based on their area of operations as either being a help desk or a service desk. Help desks can be defined as a single point of contact which deals with the resolution of customer problems and requests [7], [8]. Service desks are a central hub of contact between the customer and all IT-related areas, providing communication, information and resources to customers with IT-related problems [1], [9]. The distinguishing factor between a help desk and service desk is that a service desk utilises multiple channels in order for customers to report problems and receive assistance [7].

B. Field Study

A field study of three South African CCs was conducted in May 2006 in order to determine the cause of the problems currently experienced [10]. The study consisted of observing and interviewing CCAs in their natural working environment.

A questionnaire was issued to the CCAs in order to obtain information relating to the use of systems in their everyday activities. One of the factors identified as influencing call resolution rates was the number of systems with which the CCAs need to interact in order to complete a task. The interviews revealed the use of up to 13 different systems to log and verify customer details as well as to find relevant information.

The next subsection discusses IUIs and how these could be used as a possible solution to the problems currently experienced by CCAs.

C. Intelligent User Interfaces

IUIs differ from traditional interfaces due to the inclusion of artificial intelligence within the interface [5]. IUIs make use of user, discourse, task and context models to assist in supporting interaction between the user and the interface. Intelligent or software agents (IAs) are also used in IUIs to deliver autonomous actions based on the user's habits, preferences and interests [11].

Intelligent interfaces need to employ one or more intelligent techniques in order to make them intelligent [12]. User modelling, natural language technology, dialogue modelling and explanation generation are some of the techniques that can be employed.

An IUI could assist in refining and enhancing the process of diagnosing queries and providing support at the first line of contact. Although the interaction mode between the CCA and the customer would remains the same, the manner in which information is queried, validated and retrieved can be done through an integrated IUI as opposed to several systems (Figure 1).

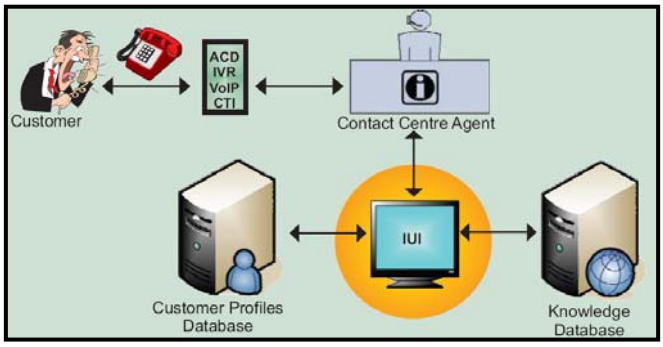

Figure 1: Extended CC Model Using an IUI [10]

This section has discussed how IUIs could be applied to the domain of CCs and what benefits they could present. The next section presents an investigation into existing IUI models.

D. Existing Models

An investigation into existing IUI models was undertaken in order to determine the most suitable model that could be specialised and adapted to the domain of CCs. Determining the effectiveness and extensibility of the various models was based on the set of evaluation criteria specified in Table 1 [13].

Four IUI architectures were evaluated, namely The Intelligent Interface Architecture [14], The Focus Architecture [15], an Adaptive Intelligent Interface Architecture [16] and the L-CID Model of a Self Adapting Intelligent Interface [17].

The Intelligent Interface Architecture [14], illustrated in Figure 2, meets all of the requirements of an IUI model, according to Table 1. This makes it the most suitable architecture to be used within the domain of CCs.

TABLE I
INTELLIGENT INTERFACE COMPONENTS AND PURPOSE

Interface Modules	Purpose
Input Analysis	Facilitates multimodal means of input.
Output Generation	Delivers multimodal output tailored to the user's specifications.
Dialog Control	Responsible for tailoring the flow and controlling the interactions of the user and facilitates for error detection and correction.
Agent / User Modelling	Utilising interface agents to build and maintain user models to deliver a personalised and natural means of interaction.
Knowledge Base	Repository(ies) for storing the User Model, Task Model, Domain Model and Vocabulary.
API	Methods called to execute the underlying functions of the system.

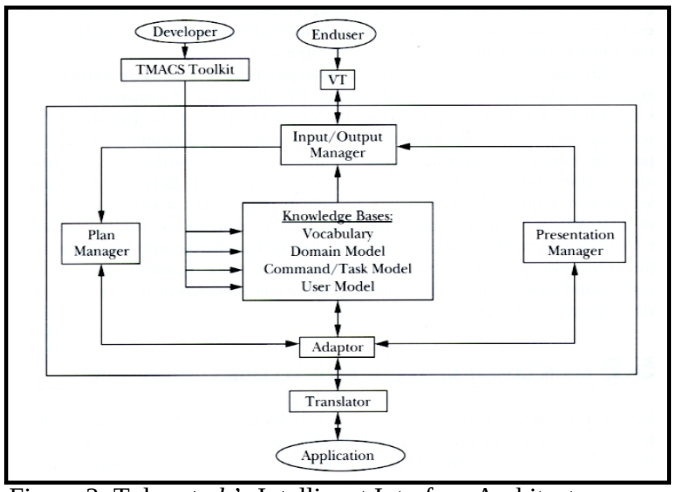

Figure 2: Tyler *et al.'s* Intelligent Interface Architecture [14]

The next section, Section III, describes the typical tasks and processes required in order to log and resolve a customer's query.

III. REQUIREMENTS ANALYSIS

The field study in Section II identified that a service desk would benefit the most through the implementation and use of an IUI. This is due the variety of mediums and modalities in which service desks receive and handle calls. Thus, the scope of this research is limited to that of a service desk and the focus to diagnosing customer queries and generating information in order to solve these queries. Figure 3 illustrates the process of current service desk operations, which was identified through the field study discussed in Section II.

The role played by the CCA in Figure 3 is that of a call taker. The tasks that are performed by CCAs typically comprise logging a call placed by a customer and processing the details of the call. In the task of processing the call, the CCA has to identify the customer and capture the problem as described by the customer. Currently the task of resolving a query is handled either by a technician or an outsourcer. This process is currently time consuming and costly. An IUI for CCs would typically sit between the CCA and the system (Figure 1).

This would allow the CCA not only to process and log a call but to potentially solve the customer's query on the first call. Providing these capabilities at the first level of support would not only reduce call resolution time but also reduce costs in having second level support, technicians or outsourcers attempting to resolve the problem.

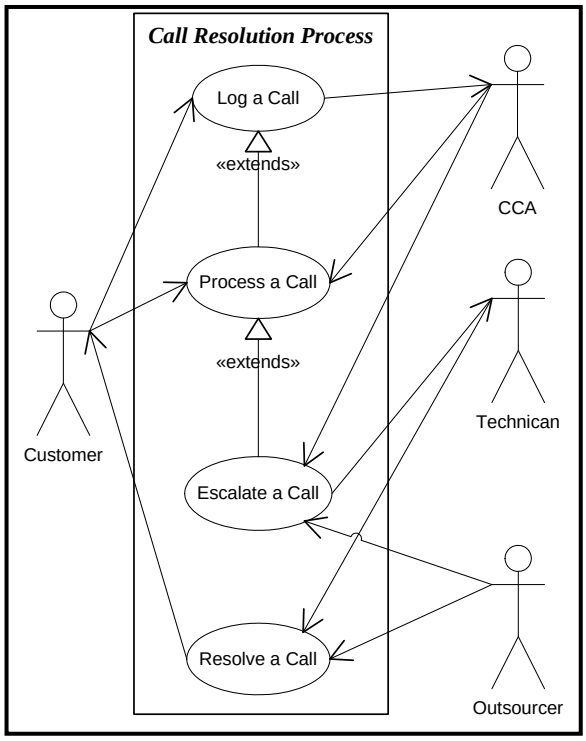

Figure 3: Use Case Analysis of Service Desk Operations

The next section, Section IV, will discuss the design and adaptation of a model for CC operations.

IV. Model Design

This section aims to describe the design of an IUI model for CCs in terms of its architecture, component–level design of the *Knowledge Base* and interaction design.

A. Architecture

This section describes the various components of the IUI Architecture and the interaction between these components. Figure 4 represents the key components of the architecture illustrated in Figure 2.

1) Input/Output Manager: The primary role of the *Input/Output Manager* is to provide multimodal capabilities to the user (CCA) via the interface. High-level goals are received from the *Presentation Manager* and are translated in lower-level operations before being displayed to the user [14].

2) Plan Manager: The *Plan Manager* contributes to the architecture by assisting the users to achieve their high-level goals. It operates based on knowledge of the user's current goals and plans. Through the use of the *Plan Manager*, the interface will attempt to detect and try to correct any errors in the user's plans.

The *Plan Manager* retrieves low-level commands from the *Input/Output Manager* and compares them to the task's parameter values, thereby detecting any errors. A limitation with Tyler *et al.'s* IUI architecture [14] (Figure 2), is that the *Plan Manager* is unable to infer user's goals based on the low-level commands entered. This has been eliminated (as illustrated in Figure 4) by allowing the *Plan Manager* to access the Task Model directly from the *Knowledge Base*.

3) Knowledge Base: A knowledge base is a key component of any IUI as it serves as a repository for application knowledge, domain based knowledge, knowledge about the current user, instructional knowledge and communicational knowledge. This is necessary so that the interface can make decisions using intelligent techniques and knowledge in order to determine how to adapt the interface to the current needs of the user [14].

4) Agent Manager: The original architecture in Figure 2 showed the *Agent Manager* as the *Adaptor*. This component has been renamed due its autonomous nature in delivering an environment that is truly intelligent and adaptive to the user. The *Agent Manager* receives interface events from the *Plan Manager;* these events are then validated against various models stored in the *Knowledge Base*. The *Agent Manager* then either updates an existing model residing within the *Knowledge Base* or retrieves the required results.

A key aspect of the *Agent Manager* is its ability to interact with the User Model and to either consult or update this model based on the current step and task of the user. The *Agent Manager* also needs to keep track of various performance data such as, how long it takes the user to complete a task, how many errors have been committed and how many times the user consulted the online help.

5) Presentation Manager: The role of the *Presentation Manager* is to retrieve the results from the *Agent Manager* and, based on the results, to determine the best possible modality in which these results could be presented.

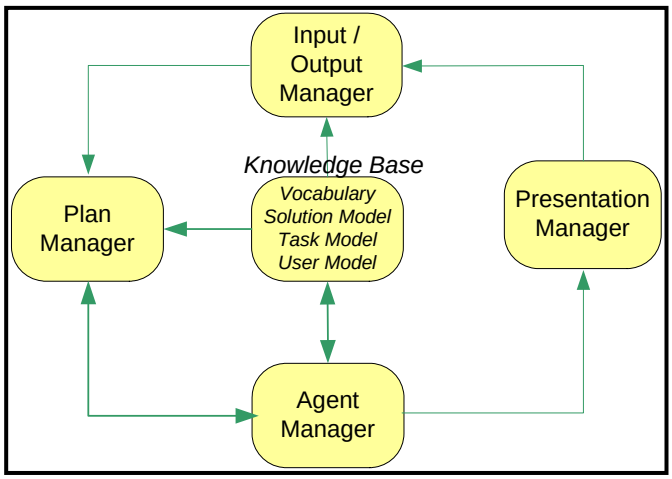

Figure 4: Components of the IUI Architecture

The next subsection further investigates the design of these components for the domain of CCs.

B. Knowledge Base Component Design

Specialising the IUI architecture in Figure 2 for the domain of CCs requires that the *Knowledge Base* contains information relating to the domain of CCs. The other components of the architecture can then be configured to operate within the new domain.

1) User Model: A user model is pivotal in contributing towards the intelligence of an IUI as it allows for the system to adapt to the needs of the user. A User Model will be present for every user of the application. It will contain the profile of the user and various task related data. Including task related data within the User Model will provide an indication of whether using the IUI actually reduces the time taken to diagnose and solve customer queries. Storing a list of categories of the problems solved by a user within the User Model, will provide an indication of the expertise that a particular user has in solving specific types of queries.

2) Task Model: Using task models to explicitly store the high and low–level goals will allow the interface to deliver task-based descriptions of the various steps and sub-steps required by the user to achieve their high–level goals. The Task Model will be stored in the *Knowledge Base* and will be a representation of the tasks typically performed by CCAs in the diagnosing and resolving customer queries. The Task Model will need to store the name of the task, a task description reflecting the high–level goals, the fields that need to be completed, the field types, description of the fields and the sub-steps that need to be completed in order to complete the task.

3) Solution Model and *Vocabulary:* A Solution Model was not present in the original IUI architecture (Figure 2) and was added to improve call resolution at the first point of contact for the customer. Based on keywords from the description of the customer's query, the user will be presented with a list of possible solutions. These keywords will be matched against keywords stored in the Vocabulary part of the *Knowledge Base.* The comparison of the keywords and identification of the possible solutions will be done by the *Agent Manager*. A Solution Model would need to contain the following details: the name of the solution, a brief description of the solution, the problem category associated with the solution and the steps required to execute the solution.

The next sub-section aims to describe and illustrate how all of these components work together to provide an IUI that will support CC operations.

C. Interaction Design

This section aims to describe how the various components of the architecture could be presented to the user in the form of an IUI.

Figure 5 contains a low-fidelity template illustrating the interface design of an IUI. Section A in Figure 5 will display the high-level goals of the current task and the steps required to be completed by the user. Section B will be used to provide user input to the application and Section C is the section where the feedback generated by the *Agent Manager* will be displayed. This section needs to be separated from the normal direct manipulation (DM) section of the user interface [18]. The *Agent Manager* needs to observe the actions performed by the user in the DM section and provide the necessary feedback in its own section.

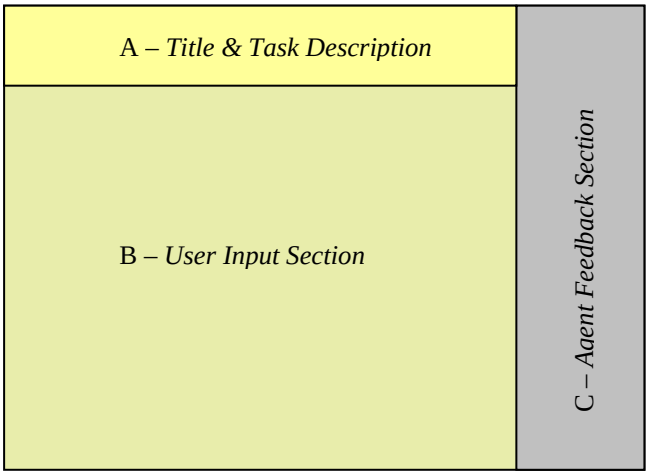

Figure 5: Components of the IUI Template

The next section will discuss the implementation of the proposed model as a proof of concept.

V. Implementation

For the purpose of this research the *Input/Output Manager* and the *Presentation Manager* components were not implemented (Figure 4). The reason for this is that these components provide multimodal and adaptive capabilities and do not directly support the query diagnosis and resolution process of CCAs (Figure 3).

The implemented prototype will allow the user to successfully log a call placed by the customer, deliver task related information from a Task Model and deliver dynamic feedback through the use of an intelligent agent (Figure 6).

The Nelson Mandela Metropolitan University (NMMU) ICT Service Desk was used as a case study for this research. At present the service desk uses the FrontRange Solutions HEAT Product Suite.

Implementation of the *Knowledge Base* was done using SQL Server 2005 combined with an extract of the existing HEAT database. The User, Task and Solution Models were implemented using XML and currently reside within the *Knowledge Base*. The data in the User Model was encrypted using a Triple Data Encryption Standard (DES) in order to ensure the integrity and reliability of the data.

The *Plan Manager* and *Agent Manager* components were implemented in C#.

Upon selecting a task, the *Plan Manager* accesses the *Knowledge Base* to retrieve the relevant Task Model. This Task Model is then decrypted and instantiated. The current task and step information is then displayed to the user in a section allocated to the delivery of task-based information (Section A in Figure 6).

Depending on the section of the screen the user focuses on, the current step will change. Input entered by the user (Section B in Figure 6) is sent to the *Agent Manager* and validated. The validation is based on the type of data required by the particular field which is stored in the Task Model. Validation is done in order to support plan recognition and to identify whether the user needs assistance in the form of intelligent help.

Feedback based on the validation from the *Agent Manager* is delivered in the Agent Feedback section (Section C in Figure 6). If the user has entered an invalid data type the error will be displayed along with a possible solution. This form of validation can be classified as intelligent help. Provided that the validation is successful, the *Agent Manager* will search the *Knowledge Base* for all the possible candidates that could be entered in that particular field.

On hovering over the list of possible candidate results, the user will see more details; which will prevent the user from going outside of the current interface to perform a search.

The User Model is instantiated and stored in memory where it can be referenced and updated as needed with task related information.

In terms of diagnosing the customers query, an efficient parser still needs to be implemented that matches the description of the problem with various keywords stored in the *Knowledge Base*. This would allow for efficient diagnosis in terms of categorising the problem and finding a possible solution. Once a list of possible solutions is identified, the user can then select a possible solution which then loads that particular Solution Model. This would guide the user through a sequence of steps and would assist in determining the best possible solution for the customer's query.

VI. Conclusion and Future Work

This paper has highlighted the potential of applying IUIs to the domain of CCs in order to provide an environment in which CCAs can diagnose and resolve customer queries efficiently and effectively.

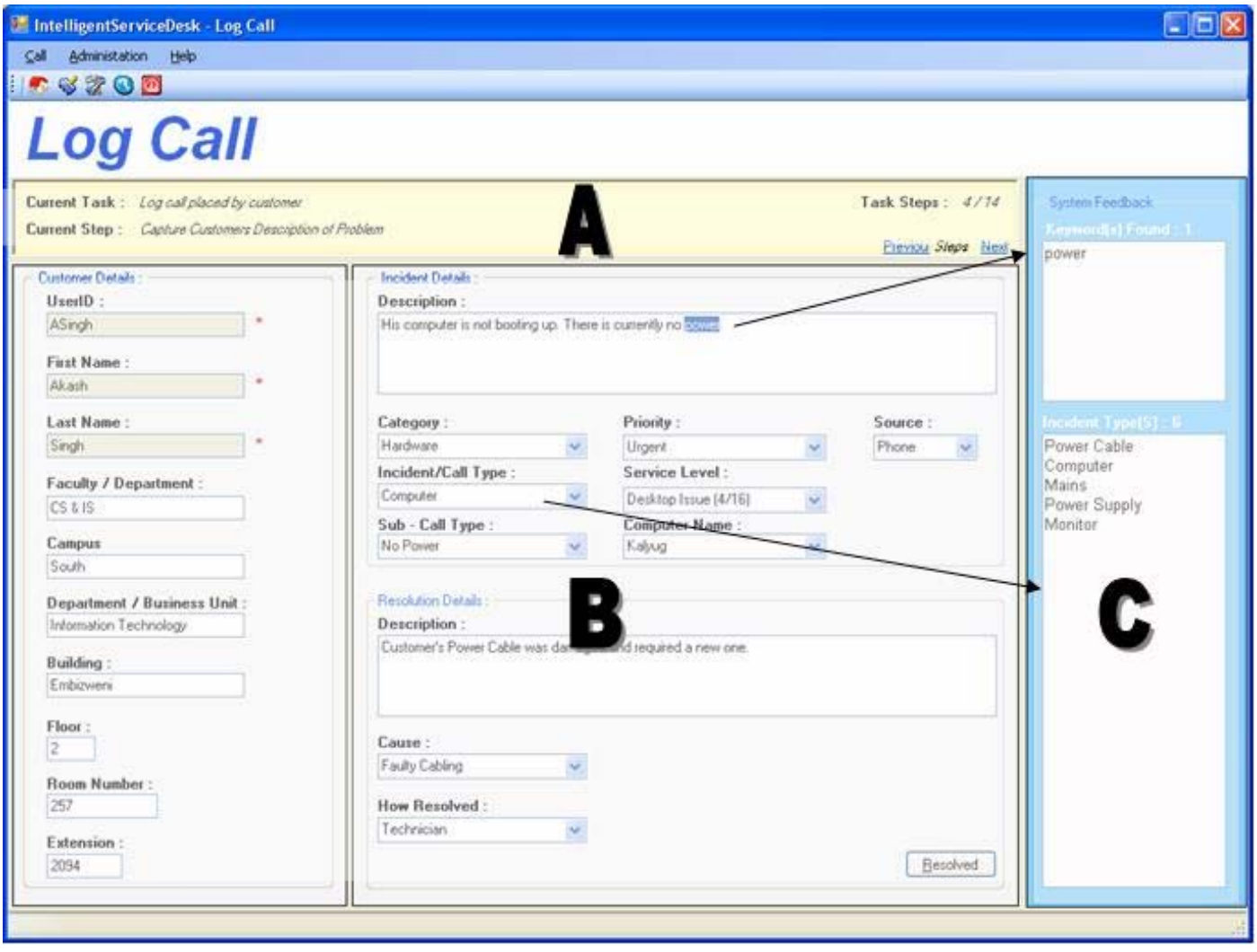

Figure 6: IUI Prototype for CC Operations

An IUI model was proposed for CCs consisting of an IUI architecture, component-level design and interaction design (Section IV). The proposed model was validated by implementing an IUI prototype to support the key processes of logging and resolving customers' queries for the NMMU ICT Help Desk (Section V).

The next stage will be to evaluate how the IUI prototype can assist in logging and processing customer calls at the NMMU ICT Service Desk. This will involve usability testing with actual users to evaluate the usability and usefulness of this prototype.

The envisaged benefits of this research would be a reduction in costs and improvements in efficiency for CC operations as the time spent diagnosing and resolving customer problems could be significantly reduced.

ACKNOWLEDGMENT

Acknowledgements are due to the Telkom Centre of Excellence Programme and the Department of Computer Science and Information Systems at the Nelson Mandela Metropolitan University for making this research possible.

REFERENCES

[1] ITIL. (2006). *Service Desk* [Online]. Available: http://www.itil.org/itil_e/index_e.html

[2] Australian National Audit Office (1996). *Telephone Call Centres* [Online]. Available: http://www.anao.gov.au/WebSite.nsf/Publications/4A256AE90015F69B4A2569120006FCD1

[3] Dimension Data. Merchants Global Contact Centre Benchmarking Report 2006. Dimension Data, 2006, pp. 83-120.

[4] M.T Maybury and W. Wahlster, "Intelligent User Interfaces" i*n Readings in Intelligent User Interfaces*, M.T Maybury and W. Wahlster, Eds. San Francisco*:* Morgan Kaufmann Publishers, Inc, 1998, pp. 1-13.

[5] M.T. Maybury, "Intelligent user interfaces : an introduction," in *Proc. 4th Annual International Conference on Intelligent User Interfaces*, Los Angeles, California, United States, 1998, pp. 3-4.

[6] Avaya, Inc. (2006). *Call Center vs Contact Centre* [Online]. Available: http://www.avaya.com/gcm/master-usa/en-us/wowtopics/callctrvscontactctr/callctrvscontactctr.htm

[7] M.G. Frants. (2006). *The Evolution of the IT Help Desk to the Service Desk. CompuCom* [Online]. Available: http://www.compucom.com/wwwimages/Corp/Images/CompuComServiceDeskWhitePaper0106.pdf

[8] R. Davis and S. Maxwell., "The comprehensive IT help desk: a single point-of-contact for the university community". in *Proc. of the 32nd Annual ACM SIGUCCS Conference on User Service*, Baltimore, MD, USA, October 10-13, 2004. SIGUCCS '04, ACM Press, New York, pp. 37-39.

[9] Microsoft. (2002). *Service Management Functions: Service Desk* [Online]. Available: http://www.microsoft.com/technet/itsolutions/cits/mo/smf/smfsvcdk/mspx

[10] A. Singh and J.L. Wesson, "Using Intelligent User Interfaces to Support Contact Centre Operations", in *Proc. 9th Annual SATNAC Conference*, Spier, September 2006.

[11] N.R. Jennings and M.J. Wooldridge, "Applications of Intelligent Agents", in *Agent Technology: Foundations, Applications and Markets*, N.R. Jennings and M.J. Wooldridge, Eds.1998. pp. 3-28.

[12] A. Waern. (1997). *What is an Intelligent Interface?* [Online]. Available: http://www.sics.se/~annika/papers/intint.html

[13] M.T. Maybury. (2001). *Intelligent Interfaces for Universal Access: Challenges and Promise* [Online]. Available: http://www.mitire.org/work/tech_papers/tech_papers_01/maubury_intelligent/maubury_intelligent.pdf

[14] S.W. Tyler , J.L. Schlossberg, A.R., Gargan, K.L. Cook, and J.W. Sullivan, (1991). An Intelligent Interface Architecture for Adaptive Interaction. in *Intelligent User Interfaces*, J.W. Sullivan and S.W. Tyler, Eds. ACM Press, New York, NY, pp. 85-109.

[15] E. Edmonds, "The Future of Intelligent Interfaces: not just "how?", but "what?" and "why?"". in *Proc. 1st International Conference on Intelligent User Interfaces*. Orlando, Florida, United States, 1993, pp. 13-17.

[16] W.E. Hefley and D. Murray, "Intelligent User Interfaces," in *Proc. 1st International Conference on Intelligent User Interfaces*. Orlando, Florida, United States, 1993, pp. 3-10.

[17] A.R. Puerta, "The Study of Models in Intelligent Interfaces," in *Proc. 1st International Conference on Intelligent User Interfaces*. Orlando, Florida, United States, 1993, pp. 71-78.

[18] H. Lieberman, "Autonomous Interface Agents," in *Proc. SIGCHI Conference on Human Factors in Computing Systems*, Atlanta, Georgia, 1997, pp. 67-84.

Akash Singh is currently pursuing a Masters degree in the field of Computer Science and Information Systems at the Nelson Mandela Metropolitan University.