Nelson Mandela Metropolitan University

for tomorrow

Department of Computer Science and Information Systems

An Adaptive User Interface Model for Contact Centres

Bronwin Anastasia Jason

Supervisor: Prof André Calitz

December 2008

The Department of Computer Science & Information Systems

Submitted in partial fulfilment of the requirements for the degree of Magister Scientiae in Computer Science and Information Systems at the Nelson Mandela Metropolitan University

Acknowledgements

I would like to first and foremost thank God the Almighty, for granting me the opportunity to do this research.

I would like to express my greatest gratitude to my supervisor, Prof. André Calitz, for his valuable support, advice, enthusiasm and encouragement over the duration of this research. I would especially like to thank him for the many hours spent reading through research papers and this document and for the useful suggestions provided to improve its content and structure.

A special thanks is extended to Mr Danie Venter, Senior Lecturer and Statistical Consultant at the Nelson Mandela Metropolitan University, for providing valuable information and suggestions with the statistical analyses of this research. His valuable assistance was instrumental in the completion of this dissertation.

I would like to express my gratitude to Mrs Bronwen Kaplan, language editor at the Nelson Mandela Metropolitan University. I would like to thank her for proof-reading this dissertation and providing me with valuable feedback which enabled me to improve its content.

A special thanks is extended to the Nelson Mandela Metropolitan University ICT Service Desk for their valued support and assistance.

Staff and students at the Department of Computer Science and Information Systems at the Nelson Mandela Metropolitan University provided me with valued feedback and suggestions.

My family and friends have provided me with continued support, patience and confidence and I extend my gratitude to them.

This research would not be possible without the necessary funding and technology from the Telkom Centre of Excellence programme.

Summary

Contact centres (CC), are the primary interaction point between a company and its customers and these are rapidly expanding in terms of both workforce and economic scope. An important challenge for today's CC solutions is to increase the speed at which CCAs retrieve information to answer customer queries. CCAs, however, differ in their ability to respond to these queries and do not interact with the computer user interface (UI) in the same way as they each have different capabilities, experience and expertise. Studies have provided empirical support that user performance can be increased when the computer UI characteristics match the user skill level. Adaptive user interfaces (AUIs) are the key to creating personalised systems. Their sole task is to provide an interface most suitable to users' needs whilst facilitating the users' varying skill levels.

The aim of this research was to develop an AUI model for CCs to support and improve the expertise level of CCAs. A literature review of CCs, user expertise, AUIs and existing AUI models resulted in the proposal of an AUI model for CCs. The proposed AUI model was described in terms of its architecture, component-level and interface design.

An AUI prototype was developed as a proof-of-concept of the proposed AUI model. A literature review on existing AUI evaluation approaches resulted in an evaluation strategy for the proposed AUI model. The AUI prototype was evaluated according to the evaluation strategy that was identified. User testing incorporating eye-tracking and a post-test questionnaire was used to determine the usefulness and usability of the AUI prototype. Significant results were found with regards to user satisfaction ratings, the learnability of the AUI prototype and its effectiveness.

This dissertation makes an important contribution in the design of an AUI model that supports and improves the expertise level of CCAs. The model could be used to assist the development of CC applications incorporating AUIs. Future research is however needed to evaluate the effect of the proposed AUI model in a larger CC environment.

Keywords: Contact centres, Adaptive user interfaces, model-based design, novice users, expert users, user interfaces

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACD	Automatic Call Distributor
AI	Artificial Intelligence
ASD	Adaptive Service Desk
AUI(s)	Adaptive User Interface(s)
AOI(s)	Area(s) of Interest
CC(s)	Contact Centre(s)
CCA(s)	Contact Centre Agent(s)
CS & IS	Computer Science and Information Systems
CSR(s)	Customer Service Representative(s)
CTI	Computer-Telephony Integration
HCI	Human-Computer Interaction
ICT	Information and Communications Technology
IDE	Integrated Development Environment
IT	Information Technology
IUI(s)	Intelligent User Interface(s)
IVR	Interactive Voice Response
NMMU	Nelson Mandela Metropolitan University
QUIS	Questionnaire for User Interaction Satisfaction
SQL	Structured Query Language
UI(s)	User Interface(s)
XML	Extensible Markup Language

Chapter 1: Introduction

1.1 Background

"If the customer experience is at the centre of how companies compete, then customer contact is at the centre of that experience" (Avaya, 2005).

Every moment of customer contact is an important moment for companies because the company's customer interaction is a valuable asset for building lasting and profitable customer relationships. Call centres or their successors, contact centres (CCs), are the primary interaction point between a company and its customers and these are rapidly expanding in terms of both workforce and economic scope. The trend towards CCs has been inspired by the internet, by customer demand for channel variety and by acknowledged potential for efficiency gains (Koole and Mandelbaum, 2001). Companies are expecting more from their CCs in order to remain competitive and cut costs while providing premium customer service.

The contact centre personnel handling the calls logged by the customers are referred to as contact centre agents (CCAs). The interactions that take place in a CC are between a customer and a CCA, between a CCA and a computer and, indirectly, between a customer and a computer (Steel, 2003). One of the important challenges for today's CC solutions is to increase the speed at which CCAs retrieve information to answer customer queries. CCAs, however, differ in their ability to respond to these queries and do not interact with the computer user interface (UI) in the same way as they each have different capabilities, experience and expertise.

In the past, due to the limitations of technological capabilities, software systems used to be relatively simple. They were only appropriate for specific tasks and this meant that a limited amount of people used them. Today, systems have become comparatively complex and the systems' users have become a more diverse user group. These users have different characteristics, needs, abilities, preferences and interests. As a result of this change, software systems have to become more individualised in order to cater for the differences found in users (Baldwin *et al*., 2000).

Adaptive user interfaces (AUIs), a sub-field of intelligent user interfaces (IUIs), have been proposed as a promising attempt to overcome the above-mentioned problems of human-computer interaction (HCI) complexity. The basic concept of an AUI involves changing the UI based on some user characteristics; therefore, AUIs are a promising approach to facilitate the varying skill levels of computer users.

1.2 Relevance of Research

Studies have provided empirical support that user performance can be increased when the computer UI characteristics match the user skill level (Jettmar, 2000). As a result, software systems have to become more individualised and cater for these differences. An investigation into CCs revealed their UIs do not currently cater for CCAs' varying skill levels.

AUIs are the key to creating personalised systems. AUIs' sole task is to provide an interface most suitable to users' needs whilst facilitating the users' varying skill levels. It is envisaged that by utilising AUIs to provide a more personalised UI according to the CCA's expertise level, an improved customer experience could be achieved. From a theoretical standpoint, it is evident that AUIs have significant benefits to offer in the domain of CC operations. From a practical standpoint, an AUI that adapts the interface according to the CCA's expertise level assists CCs with agent training which could possibly reduce training costs and time.

1.3 Research Outline

This section will discuss the research outline in terms of the problem statement (Section 1.3.1), thesis statement (Section 1.3.2), scope and constraints (Section 1.3.3), research objectives (Section 1.3.4), questions (Section 1.3.5) and design (Section 1.3.6).

1.3.1. Problem Statement

The main objective of this research is to develop an AUI model and an AUI prototype for CCs. The AUI model will be used to accommodate the varying skill levels of CCAs in order to improve CC operations. The model needs to show how the call-resolution process at the first-level of support can be improved when the UI matches the CCA's skill level.

1.3.2. Thesis Statement

Hofstee (2006) suggests that a dissertation should be formulated in terms of a thesis statement as it guarantees a structure and provides clarity on the research which will be undertaken. The thesis statement of this research is:

> *An AUI model that can be adapted according to a CCA's UI skill level can improve that CCA's productivity and enhances the CCA's interaction with the UI.*

1.3.3. Scope and Constraints

The problem domain of this research project will be limited to the logging of customer queries for a specific CC, namely, the Nelson Mandela Metropolitan University (NMMU) Information Communications Technology (ICT) Service Desk. Telkom SA Ltd. and Dimension Data CCs were considered, however, due to logistical reasons and confidentiality of data, they were not available. Customers interact directly with the CCAs logging their queries; therefore, limiting the scope to this task was deemed the most appropriate.

The UI skill levels of users can range from novice to intermediate to expert. This research will, however, be limited to the two most commonly used expertise levels, namely, novice and expert. The UI skill level of CCAs escalates from novice to expert very quickly and therefore limiting this research to these two skill levels is most appropriate.

1.3.4. Research Objectives

In order to achieve the main objective of developing an AUI model for CCs, the following research objectives need to be realised:

- An investigation into CCs;
- An investigation into the different user expertise levels and the UIs which facilitate the different levels;
- An investigation into AUIs;
- An investigation into existing AUI models;
- Designing an AUI model for CC operations;
- The implementation of a prototype to evaluate the proposed model; and
- An evaluation of the proposed AUI model.

1.3.5. Research Questions

The primary research question to be answered in this study is, therefore:

How can an AUI model for CCs improve CCAs' productivity and usability?

Several research questions were then identified from the primary research question. These questions are listed in Table 1.1, together with the research methods used to answer the different questions. The chapter in which each research question is addressed is also included in this table.

Research Question	Research Methods	Chapter
1. *What are the factors involved with facilitating the skill level of CCAs?*	Literature study	2
2. *What considerations should be made when designing UIs that support the varying skill levels of users?*	Literature study	3
3. *How can AUIs be designed for CCs to facilitate CCAs' skill levels?*	Literature study	4
4. *How can an AUI model be designed to support CC operations?*	Iterative Design	5
5. *How should the proposed AUI model be implemented?*	Implementation, Evaluation	6
6. *How useful is the proposed AUI model?*	Model Evaluation, User Testing	7
7. *What recommendations and conclusions can be made?*	Analysis	8

Table 1.1: Research Questions and Methodology

1.3.6. Research Design

The primary objective of this research study is to propose an AUI model for CCs. A model-based approach is a useful and powerful tool to develop UIs (Lopez-Jaquero *et al*., 2003). Model-based research provides clarity on the identified research problem and captures the essential aspects of a system (Olivier, 2004). The primary method which will, therefore, be used for this research study is the design of a model and will be supported by secondary methods. The secondary methods include a literature survey, a prototype construction and a prototype evaluation (Olivier, 2004).

The literature survey will be conducted to gain a better understanding of CCs, the varying skill levels of users and possible interfaces which can support them, AUIs and existing AUI models. A prototype will be constructed as a proof-of-concept for the proposed AUI model and evaluation thereof will be conducted to evaluate whether the proposed AUI model improves the performance of CCAs and if it enhances a CCA's user experience.

1.4 Dissertation Structure

This dissertation consists of eight chapters. Each chapter attempts to meet a research objective by answering the questions posed in Table 1.1. The structure of the dissertation is visually depicted in Figure 1.1.

Chapter 1 (Introduction) provides background information about CCA behaviour as well as the motivation for this research and its direction. The research outline is described in terms of the research objectives and the research questions posed to meet these objectives. The scope and design of the research is also explained.

Chapter 2 (Contact Centres) comprises the first literature study and focuses on an investigation into CCs. CCs are discussed in terms of the various types available as well as the way they are structured. A discussion on CC personnel is provided, detailing the induction training process and providing an overview of the most commonly used training methods. A discussion is provided on how UIs can aid training, and existing CC software is analysed in order to identify the extent to which adaptivity is supported. The process of logging a customer's query is also discussed.

Chapter 3 (User Expertise and User Interfaces) comprises the second literature study which focuses on an investigation into novice and expert users and how UIs could support users' expertise levels. Design considerations for designing interfaces for both the novice and expert user are discussed. Multi-layer UIs are discussed as a possible approach to support users' expertise levels. AUIs are, however, considered to be a more ideal approach; therefore, a discussion on IUIs (AUIs' broader research area) is provided.

Chapter 4 (Adaptive User Interfaces) comprise a detailed literature study of the field of AUIs, a discussion on what AUIs are and their components. These components are discussed to determine how AUIs could possibly benefit CC operations. The stages of the adaptation process are also discussed as well as the various challenges which AUIs encounter.

Chapter 5 (Adaptive User Interface Models) commences with a literature study of an existing IUI model for CCs and proceeds to examine existing AUI models. These models are analysed in order to identify an AUI model which could be specialised for the CC domain. The literature study results in a proposed AUI model, the design of which is discussed further.

Chapter 6 (ASD Prototype Implementation and Pilot Study) details the implementation of the AdaptiveServiceDesk (ASD) prototype based on the proposed AUI model discussed in Chapter 5. The implementation of ASD will serve as a proof of concept that the model could be used to develop AUIs for CC applications. A pilot study will be discussed and results will be used to conduct the main evaluation in Chapter 7.

Chapter 7 (Main Evaluation and Results) discusses the evaluation of the ASD prototype and the proposed AUI model implemented in Chapter 6. The evaluation will use results obtained in the pilot study discussed in Chapter 6. The results of the evaluation are presented.

Chapter 8 (Recommendations and Conclusions) concludes the dissertation by evaluating the achievements and results of the research. This is followed by ideas for future research.

Appendices are provided which include detailed statistical results of the pilot study discussed in Chapter 6, and statistical results of the main study which is based on the model evaluation discussed in Chapter 7.

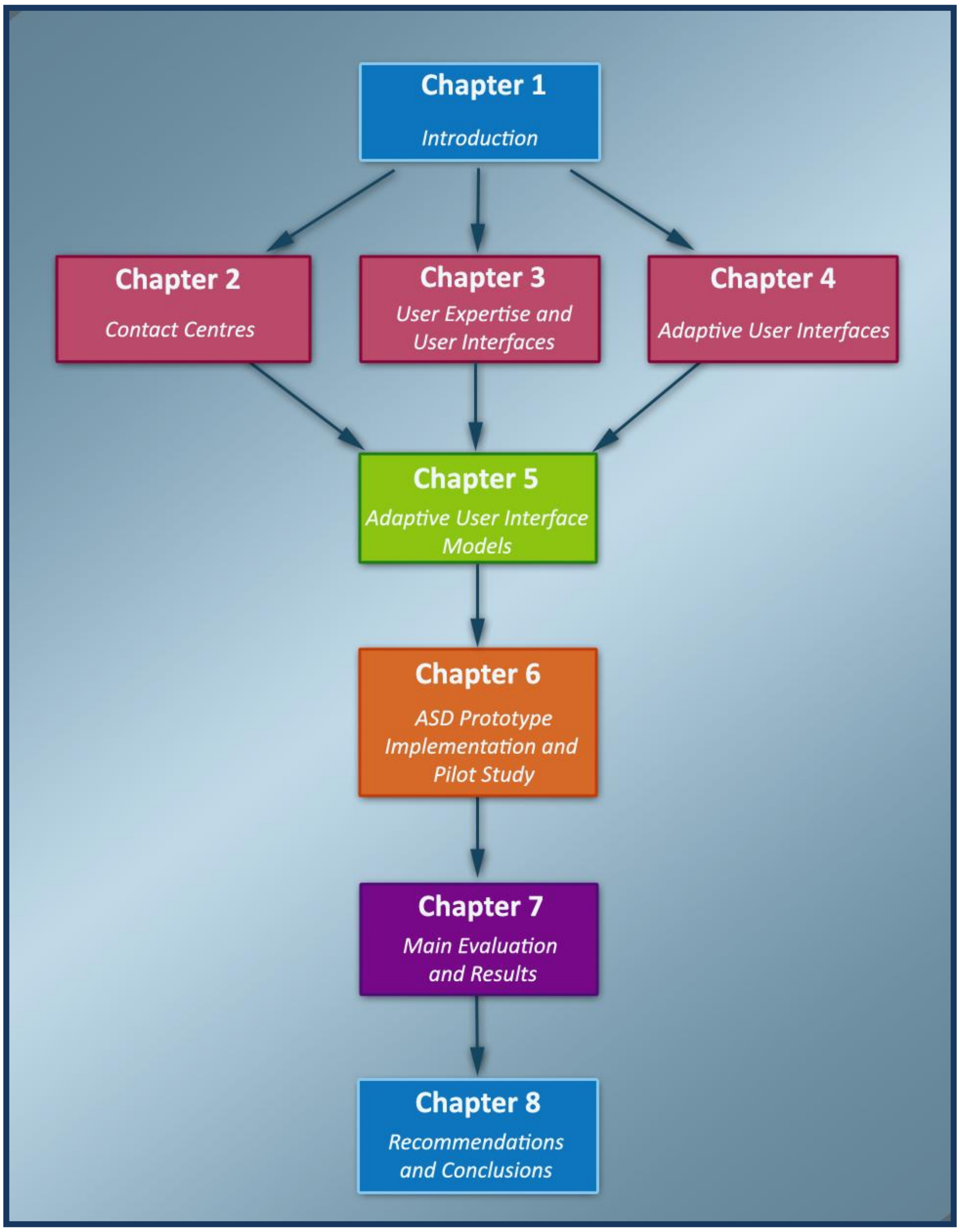


Figure 1.1: Structure of Dissertation

Chapter 2: Contact Centres

2.1 Introduction

Contact centres (CCs) are the primary interaction point of a company with its customers and are rapidly expanding in terms of both workforce and economic infrastructure (Koole and Mandelbaum, 2001, Mandelbaum, 2004). CCs need to reduce costs by taking advantage of the latest technological advancements. CC personnel managing calls logged by customers, act as intermediaries between a customer and the CC. There are training costs associated with CC personnel. Designing good user interfaces (UIs) is one possible approach to reduce CC personnel training costs (Schumacher, 2004).

The purpose of this chapter is to provide an overview of the domain of CCs. This is done to establish what CCs are and how they are structured, the importance of the CC personnel managing customer queries and how to reduce training and operational costs. A discussion on designing UIs for CCs is provided as they are a possible solution to mitigate training costs. The discussion on designing UIs results in an investigation into existing CC software and provides a basis for a comparison of existing CC software. The call-resolution process is further explained to identify how an adaptive user interface (AUI) could be used to personalise and customise the contact centre agent's (CCA) UI.

2.2 Overview

A call centre can be defined as an operation that uses telephone, personnel and computer technology in sophisticated ways to deliver a variety of services to customers (Australian National Audit Office, 1996, Parbhoo, 2002). CCs are the contemporary successors of call centres. CCs are defined as operations (Koole and Mandelbaum, 2001, Mandelbaum, 2004):

> *...whereby in addition to the traditional telephone services, they interface with customers via some additional multi-media customer-contact channels such as Interactive Voice Response (IVR) units, email, fax, internet and chat.*

CCs provide many business advantages, including: improved efficiency, increased hours of operation, reduced costs and greater flexibility (Robertson, 2002). CCs also deliver improved customer service through increased access for customers (Australian National Audit Office, 1996, Robertson, 2002). Customers can communicate with CCAs, also known as customer service representative (CSR) agents, via a variety of channels such as telephones, e-mails, faxes, person to person, internet, etc. These CCAs are specially trained in how to respond to customer queries in a quick and efficient manner (Rathod, 2001). Most major companies communicate with customers via one or more CCs, which are either internally-managed or outsourced (Koole and Mandelbaum, 2001).

Communication with CCAs can commence once the customer is either greeted with a recorded message followed by the placement in a telephone queue managed by an Automated Call Distribution System (ACD), or after being offered various caller options by an Integrated Voice Response (IVR) unit, or immediately after the call is directed by an ACD. Whatever process is executed mainly depends on the size of the CC (Timbrell *et al*., 2005).

CCs vary greatly in size and geographic dispersion, ranging from small sites (up to 12 CCAs) with only a few personnel assisting customers, to much larger CCs utilising hundreds or thousands of CC personnel. The working environment of a larger CC can be described as an endless room, with numerous open-space cubicles, in which CC personnel sit in front of computer terminals, assisting customers with their specific queries (Timbrell *et al*., 2005, Koole and Mandelbaum, 2001).

Irrespective of their geographic dispersion, CCs are most commonly classified by functionality and the various areas of operations they support. The next sections discuss the various types of CCs.

2.3 Types of Contact Centres

The previous section defined CCs as call centres that additionally enable customer support via a variety of channels. There are various CC types and structures (Singh, 2007) and these are depicted in Figure 2.1. The areas of operations are the help desk and service desk CCs, where either incoming (inbound) or outgoing (outbound) calls are handled. This sub-section discusses these various types.

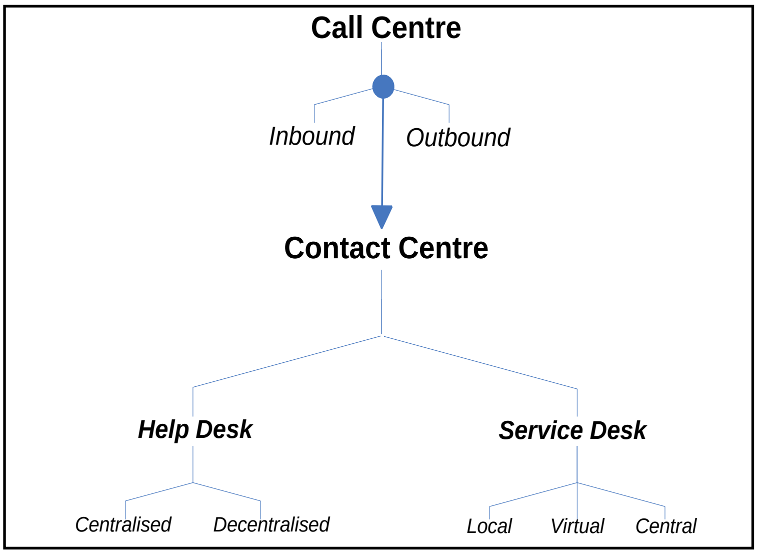


Figure 2.1: Contact Centre Types and Structures (Singh, 2007)

2.3.1. Inbound Contact Centres

A central characteristic of CCs is whether they handle inbound or outbound traffic (Koole and Mandelbaum, 2001). An inbound CC is primarily responsible for receiving incoming calls from customers and handles either queries or transactions.

Query-based, inbound CCs are dependent on some sort of knowledge base that contains the necessary information which can provide possible solutions to customer queries. These can range from very simple to very complex queries (Robertson, 2005).

Transactional, inbound CCs are dependent on a set of procedures that could be used to assist the customer with a transactional process. Thus, CCAs focus on procedural knowledge and utilise complex front-end systems to record transactions. Similarly to a query-based, inbound CC, these recorded transactions can also range from simple to more complex, involving more detailed and careful use of front-end applications (Robertson, 2005).

2.3.2. Outbound Contact Centres

Outbound CCs handle outgoing calls that are initiated by the CC personnel themselves. These types of CCs have traditionally been associated with sales, support, advice and other business transactions. There has been some development in inbound CCs whereby outbound calls are initiated to high-value customers who have abandoned their calls before being served (Gans *et al*., 2003). *This research will focus on inbound CCs*.

2.3.3. Help Desks

A help desk can be defined as a single point of contact for customer problem resolution (Frantz, 2006). Help desks can be either centralised or decentralised; each approach having its advantages and disadvantages (Table 2.1). A particular advantage that decentralised help desks have over centralised help desks is their ability to provide on-site services (Table 2.1).

A centralised help desk acts as a single location within an organisation to provide support to all its users. Due to logical reasons, most centralised help desks are located within information technology (IT) departments. Locating IT specialists in close proximity maximises the efficiency of the help desk. CC personnel quickly become specialists as they form the only technical support within an organisation. CC personnel share knowledge of problems and solutions with each other, contributing to a more knowledgeable staff and thus, faster problem resolution (Sanderson, 2003).

Decentralised help desks are generally distributed throughout multiple sites within a single organisation. Multiple sites are necessary for companies with offices in different geographic locations which want to provide on-site support to their field offices. There might also be a need to offer support during hours when a centralised help desk might be closed. Multiple sites also provide an opportunity for specialisation: each division can have its own help desk staffed with specialists (Sanderson, 2003).

	Advantages	Disadvantages
Centralised	• Easy for users to locate. • Better communication among specialists. • Easier to enforce standards. • Better use of resources. • Exposure of specialists to a broad range of issues.	• Inability to provide on-site support to remote locations. • Difficulty of understanding business needs. • Challenge of providing support in multiple time zones.
Decentralised	• Ability to address local, site needs (time zone, language, products). • Availability of on-site services.	• Difficulty of providing standardised information. • Need for common mission and goals. • Challenge of measuring performance.

Table 2.1: Advantages and Disadvantages of Centralised and Decentralised Help Desks (Sanderson, 2003)

2.3.4. Service Desks

Service desks are defined as a central point of contact between customers and all IT-related areas, where customers can use multiple channels for requesting services (Frantz, 2006; Microsoft, 2002). Service desks can either have local, central or virtual structures.

Local service desks are situated at each location or department within an organisation (Microsoft, 2002) and were initially created to meet local business needs. This approach is, however, unpractical in situations where multiple locations requiring support services are needed (Great Britain Treasury *et al*., 2000).

Central service desks are responsible for support to an entire organisation unit, regardless of geographic location (Microsoft, 2002). All service requests are logged to a central physical location providing the most benefits to organisations with multiple locations (Great Britain Treasury *et al*., 2000).

A virtual service desk is a combination of both the local and central service desk allowing for information to be centrally stored and accessed globally (Microsoft, 2002). This structure poses a prime operational restriction in the need for a physical presence by a specialist or replacement engineer (Great Britain Treasury *et al*., 2000).

The main difference between a help desk and service desk is that a service desk enables the usage of multiple channels to service customers (Frantz, 2006). There are both advantages and disadvantages associated with the various service desk structures. These are illustrated in Table 2.2. Central and virtual service desks are more advantageous with regard to cost reduction, whereas local service desks might be inefficient and expensive to resource (Table 2.2). *The CC used as a case study in this research operates as a central service desk.*

	Advantages	Disadvantages
Local	• Provide customised support for specific location-based groups or staff. • Each service desk provides backup to other service desks in the event that one should become unavailable (disaster, and so on).	• Might be inefficient and expensive to resource.
Central	• Users know where to call for support. • Fewer staff may be required, which reduces training, equipment, and facility costs.	• Knowledge obtained by staff not as detailed as knowledge of local service desk staff.
Virtual	• 24-hour support is provided. • Combines advantages of both local and central service desks.	• More difficult to provide on-site support.

Table 2.2: Advantages and Disadvantages of Local, Central and Virtual Service Desks (Microsoft, 2002, Great Britain Treasury *et al*., 2000, Bon *et al*., 2007)

2.4 Contact Centre Structures

All inbound or outbound CCs must have an underlying organisational structure. This structure may vary from a flat (pool) to a multi-layered one. The structure implemented depends on the needs of the organisation. The five primary structures are pool, dispatch, tiered, specialised and method. Most CCs utilise only one structure although a combination could be used. The five primary structures will now be discussed briefly.

2.4.1. Pool

Pool structures are the simplest and involve all CC personnel supporting the same technology, serving the same customers and performing the same job duties. Thus, CCAs handling calls tend to have a more general idea of products and services offered but not an in-depth knowledge to solve complex problems (Sanderson, 2003).

2.4.2. Dispatch

In a dispatch structure, the first-line personnel act as dispatchers. Their sole purpose is to receive customer requests and then dispatch/refer them to an appropriate group without attempting to solve the problem themselves. The advantages of a dispatch structure are that there is a minimal waiting period before calls are answered at the first-level and a high level of problem resolution. Dispatchers require little training; however, there can be a loss of productivity if callers have to describe their problems more than once. First-level CCAs also have limited career advancement (Sanderson, 2003).

2.4.3. Tiered

Tiered structures divide the CC into several groups/levels where each level provides a different degree of support (Figure 2.2). The amount of levels needed depends on the organisation. The first- level CCAs receive customer queries in an attempt to resolve them; thus, they need to familiarise themselves with a variety of products and services. A predetermined time is usually allocated to the first-level for resolving the call. It is not vital, however, that they have the in-depth knowledge that a higher-level will possess (Sanderson, 2003).

The ideal situation would be for the query to be resolved at the first-level without the need for query escalation, but this depends greatly on the expertise of the CCA handling the call. If the CCA is equipped to resolve the call, then ownership of the query is taken by the CCA. Ownership refers to taking responsibility for a problem and monitoring it until it is resolved (Sanderson, 2003). This results in high customer satisfaction.

Second- level and upward CCAs are specialists and do not generally take incoming calls directly from customers. These specialists are required to have an in-depth knowledge of a limited area and are expected to solve complex problems in their area of expertise.

The advantages of a tiered structure include a high call-resolution rate at the first-level, as well as a high level of customer and employee satisfaction. There could, however, be a lack of problem solving at the first level before referring calls. Personnel at higher levels are specialists; thus, there could be higher costs for these skilled personnel and this could result in possible internal friction between the higher and lower levels (Sanderson, 2003).

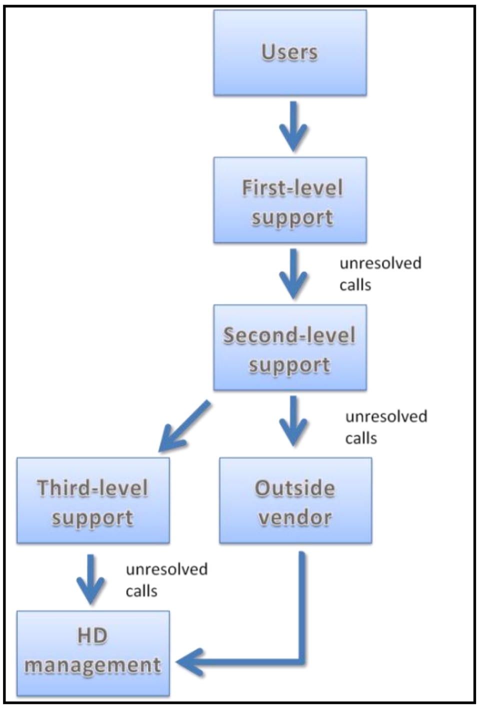


Figure 2.2: Contact Centre Tiered Structure (Sanderson, 2003)

2.4.4. Specialised

A specialised structure routes an incoming call to a support group. There are two types of support models: one is based on the product the problem pertains to (the product model), and the other on the business unit that is making the request (the business model).

The advantage of a product model is that it enables CCAs to become experts in a particular area. There can be a difficulty, however, in knowing where to assign calls if there are too many specialty groups. A shortage of specialty groups may also lead to insufficient knowledge to solve queries. The advantage of a business model is that it enriches CCAs' understanding of queries from a business standpoint; however, this means that support staff are exposed to only a limited range of technologies (Sanderson, 2003).

2.4.5. Method

In a method structure, CCs are organised by the manner in which support is provided. One group handles all incoming telephone queries while another responds to e-mail. This structure is most effective when requests are evenly distributed across several inquiry methods but are ineffective when CCs receive one form of inquiry more than another (Sanderson, 2003).

Regardless of their organisational structure, all CCs rely on people to act as the intermediaries between the CC and their customers. Studies show that customers prefer personal support rather than impersonal support from other methods such as mailing systems, automatic answerers, SMS and WAP (Torre, 2002). The next section discusses the people employed by CCs and the various training methods employed to equip them to service customers.

2.5 Contact Centre Personnel

A variety of personnel are employed at CCs: some operate in non-customer facing positions (e.g., managers) while others interact with customers on inbound and outbound calls. The main focus is on those personnel in customer facing roles, namely CCAs, as these individuals have a greater chance of impacting on the operational performance of the CC, and therefore, directly affect customers' satisfaction levels.

CCAs are the most expensive resources within a CC; thus, the measurement of their effectiveness and productivity is always a concern of management (Rathod, 2001). Some of the costs involved with CCAs are costs of work, training and frequent turnover (Torre, 2002). It is imperative that the costs based on an agent's productivity and associated network expense be balanced with the quality of service offered by him or her when handling a customer.

CCAs need to be trained, motivated and equipped to deal with customers. Employee satisfaction, increased by training and empowerment tools, is the key to providing customer and, ultimately, shareholder satisfaction as well as playing a crucial role in job satisfaction and performance for CCAs (Rathod, 2001).

Some of the reasons why training remains a valuable investment for organisations are (Heathcote, 2003):

- Employees of all skill levels need training to obtain the skills, attitude and knowledge necessary to competently do their jobs.
- New employees need to be trained before they can be reasonably effective within an organisation.
- Due to the constant change of organisations brought about by new technologies and procedures, existing employees still require training.
- Well-trained employees are likely to be better motivated and have a more promising career path.
- Training can result in increased sales, better service to customers leading to improved customer satisfaction and low staff turnover.

2.5.1. Agent Induction Training

CCAs transition through a series of stages during their careers. There are two primary stages which they transition through: accelerated learning and competence. The first stage of accelerated learning is undertaken by newly employed CCAs. After the initial learning stage, CCAs are more competent to solve day-to-day customer queries. There will be situations where the CCA lacks the expertise to resolve the customer’s request and, depending on the organisational structure implemented, the request will be referred to a CCA on a higher level (Section 2.4). At this stage, there is still consistent learning of CC technology and other areas of the business (Sanderson, 2003).

Focus will be placed on the initial learning stage of induction. Induction training is the opportunity for newly appointed CCAs to become able and confident in their roles. A good induction follows a logical structure built around a funnelling technique (Calvert, 2004).

The funnelling technique, as depicted in Figure 2.3, ensures that newly appointed CCAs first learn about the organisation through corporate induction, followed by an overview of the CC. Learning about the necessary products, processes and systems ensues, after which the CCA learns how to practically apply the acquired skills (Calvert, 2004).

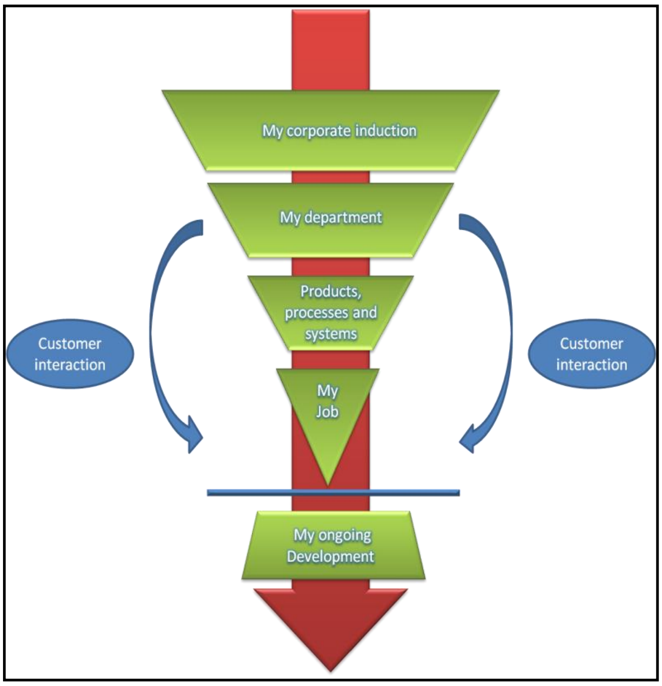


Figure 2.3: The Funnelling Technique (Calvert, 2004)

The most included content areas of agent induction training are (Dimension Data, 2006):

- The corporate induction;
- Systems training;
- Customer product knowledge; and
- Communication skills training.

The Merchants Global Contact Centre Benchmarking Report (2006) indicated an increase from the previous (2005) report in the time taken for novice CCAs to become competent (i.e., an average of 41 to 47 days). One of the key drivers for this increase is reported as being increased systems and technology knowledge requirements; thus, systems' training is one of the core areas to ensure early effectiveness on the job, performance in and engagement with the organisation.

It can be deduced that newly appointed CCAs have a lot to learn in a very short time. Improving the CCA induction process provides some real benefits (Robertson, 2003); thus, CCs need to tackle these challenges by using a range of effective training methods.

2.5.2. Training Methods

The majority of training methods used to train newly appointed CCAs have been focused on ensuring that agents learn about the CCs' most basic issues. Certain organisations utilise self-service components (e.g., IVR and web self-service) that have targeted these basic issues, taking away the easy end of service issues. Incoming calls received by CCAs are more complicated and, therefore, not the kind of calls that can be learnt in a few weeks of novice agent training (eGain, 2005).

Training constitutes one of the major operating costs of a CC (Australian National Audit Office, 1996). Instead of forcing CCAs to engage in intensive learning, new interactive process software can help capture and codify service best practices that can be interactively used by novices (eGain, 2005). This section discusses the most popular training methods used in CCs.

2.5.2.1. Classroom

The Merchants Report (2006) indicated the most popular training method used by CCs has been the classroom style, fully supported by trainers. This method is most appropriate when there are cost constraints, a shortage of trainers or a high number of delegates. This method is found to be more appropriate for induction training with large groups of new CCAs. The disadvantage of the classroom training style is that unless CCAs are assessed at intervals throughout the training or formally at the end of the session, there is no evaluation evidence that the agent is fully competent to begin handling calls (Dimension Data, 2006).

2.5.2.2. Web-based / E-Learning

The idea of utilising software such as web-based or e-Learning, to assist with the induction training of novice CCAs seems like a promising approach; however, in reality, not many CCs employ these training methods. The Merchants Report (2006) indicates that less than half (45 percent) of the CCs investigated utilise web-based and eLearning training methods. A possible reason for this involves the challenges in the technology investment required, as many products have either limited flexibility or require a high degree of maintenance to keep the information up-to-date, utilising valuable training resource time (Calvert, 2004).

2.5.2.3. Buddying

Another popular training method used is the concept of 'Buddying'. Buddying involves new CCAs monitoring experienced CCAs and listening in on calls (Dimension Data, 2006). Learning on the job results in increased skill levels (Gans and Zhou, 2002). The survey conducted by Dimension Data found that using "live" calls was by far the most popular training method. Buddying positively uses focused interaction and provides the opportunity to immediately influence performance improvement in a relevant rather than generic and delayed manner (Dimension Data, 2006).

The disadvantage of using the buddying approach is that there may be ineffective up-skilling of new agents, if the CCA being asked to buddy a new agent has had no formal experience of training others. Another disadvantage is the transference of poor work habits due to the naturalness of this training method (Dimension Data, 2006).

User-centred design principles dictate that good designs will reduce the need for training. This principle can be used as the basis for investing in the user experience within a CC domain. Training costs can be one of the most expensive elements of a CC and although training remains vital to producing knowledgeable agents, usability specialists can help eliminate redundant training (Schumacher, 2004).

2.6 Designing User Interfaces for Contact Centres

A CC consists of several interactions. The customer does not directly interact with the computer system, whereas the CCA interacts with both customer and computer system (Steel *et al*., 2002). Studies have also shown a very high correlation between agent satisfaction and customer satisfaction (Instranet, 2003). So, in order to retain customers, investment needs to be placed in employees and the technology they utilise. Improving employee (CCA) satisfaction is key to providing the customer with a good customer experience.

Modern CCs are complex socio-technical systems and have widely been recognised as requiring a fine balance between people, processes and technology (Mandelbaum, 2004, Dimension Data, 2006). Both the people within a CC and the technology need to work together interdependently; therefore, the essence of CCs does not entirely lie in the technology but rather in the way technology is used to support the CCAs in their daily activities. The technology cannot replace the impact of the well-trained CCAs but it can improve their productivity, quality and performance. The Merchants Report (2006) report believes that organisations have not fully leveraged CC applications to the extent that they could, and it should be carefully considered how technologies could enhance improved customer-service levels.

The integration of CC applications has been minimal, requiring agents to tab between screens, increasing call-handling time and frustrating customers. The Merchants Report (2006) suggests making front-end tools process-driven, intuitive and agile in terms of customer services, in order to derive real value. Improvements such as increased first-call-resolution, reduction in call length and less need for system training can only be met once these requirements have been addressed (Dimension Data, 2006).

Section 2.5.2.3 stated that good UI designs will reduce the need for training. Computer UIs can be quite difficult to use and require a learning curve to become used to the product. Many conventional human computer interaction (HCI) issues such as interface learnability and system feedback apply to CC systems as well. Designing UIs for CCs is, however, a balancing act that involves the ability to weigh multiple considerations, issues and pressures. Designing CC UIs must be approached somewhat differently than the way one would approach the design of more typical UIs such as web sites and applications. The core design principles remain the same, but specific matters such as business and user issues require a more specialised and detailed approach (Schumacher, 2004).

The following issues needs to be taken into consideration when designing UIs for CCs (Schumacher, 2004):

- *High-volume transaction environment*
  The CC domain differs from other domains due to its transactional nature, resulting in thousands (depending on the CC size) of customer interactions per day.
- *Potential for impact*
  Small design changes (e.g., reducing type fonts) have the potential to have

profound user and business impacts; thus design changes need to adhere to the "Do no harm" principle.

- *Interdependence*

  The notion of the CCA (the user) controlling, but the customer directing the interaction is unique to CC design. This implies that the design needs to satisfy the needs of both the CCA and the customer; thus, they need to anticipate what callers (customers) will say and design for the next step.

- *Demands of multiple systems*

  CCAs' desktop are often more "cluttered" than typical users. Due to the inconsistencies of multiple applications, CCAs often have to remember different methods for finding and entering information. Designs need, therefore, to map out plans for cross-application consistency.

- *Business issues*

  Organisational pressures often drive the usage of applications by CCAs; thus, designs need to account for the organizational realities and shifting business priorities.

- *Performance pressures*

  Management often places tremendous pressure on CCAs such as decreasing call handling time. Applications can either facilitate or impede a CCA's ability to get commissions; thus, designs need to maximize both organizational goals as well as personal performance.

- *Expert behaviour*

  In CCs, novice users become experts very quickly. It is more beneficial to expert users to have more information at once rather than less. A common design flaw is disregarding the expert and designing solely for the novice user. A good design principle would be to design for the expert but still support the novice user.

- *Design control*

  Studies show that experts are not UI design experts; thus, CC UI design should be according to solid task analysis, user interviews, and recognised design practices.

- *High costs of training*

  Training costs are high due to CCs demanding expertise by CCAs and as previously mentioned, good UI designs will reduce the amount of training required by CCAs.

2.6.1. Existing Software

One of the main driving forces behind CC development is the software it utilises. Traditionally, CCs have been telecom entities, but the growth and maturation of computer telephony integration (CTI) have led to computing-based centres and applications. Software has become one of the best and most widely used tools for translating business parameters into technological terms.

Software is used by CCs as a means to (Singh, 2007, Sharp, 2003):

- Retrieve customer information through the use of an IVR;
- Provide customer profiles through customer relationship management (CRM);
- Manage queues through the use of an ACD;
- Integrate operations performed on the computer and calls received on the telephone through CTI;
- Act as an interconnection to back-office applications; and
- Support the retrieval and storage of information in a knowledge base.

2.6.1.1. SiteHelpDesk-IT

SiteHelpDesk-IT is a web-based help desk solution which was specifically designed for internal IT service management (SiteHelpDesk.com, 2008). SiteHelpDesk-IT provides the following features:

- User Call Logging and Self Help;
- Helpdesk Administration;
- Operator Call Management; and
- Reports, Graph and 3D Charting.

Figure 2.4 illustrates the call management’s screen which enables the logging of customer queries. As depicted in Figure 2.4, there are various list selections that need to be made within this interface, such as call type and sub call type, and text areas are provided to the user in which the call’s problem and resolution should be entered. The interface does not support the CCA’s understanding of the CC process of logging a customer’s query and appears too complex for novice CCAs.

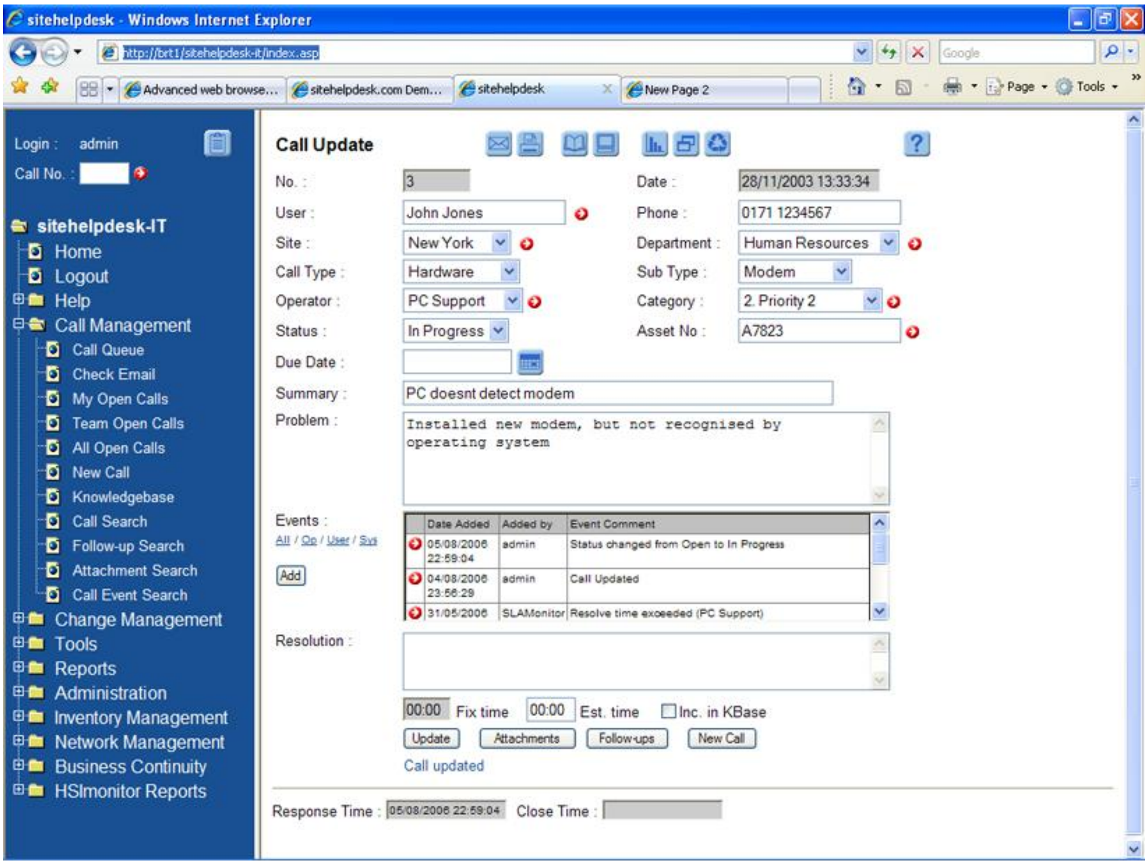


Figure 2.4: Call Management Screen from SiteHelpDesk-IT (SiteHelpDesk.com, 2008)

2.6.1.2. SysAid Web-Based Help Desk

SysAid is an easy to use web-based help desk or service desk software solution which combines help desk, remote control, asset management and monitoring and IT activity analysis tools into one simple, accessible solution (Sysaid.com, 2008).

The SysAid system consists of eight logical modules:

- Help Desk Module;
- Asset Management Module;
- Monitoring;
- Report and Analysis Modules;
- Manager Dashboard;
- Tasks and Projects;
- Remote Control Module; and
- Communication Module.

The Help Desk Module, visually represented by Figure 2.5, enables the logging of calls. Different views can be added to reflect customised outlooks on the help desk requests. This interface does not, however, support the different skill levels of CCAs.

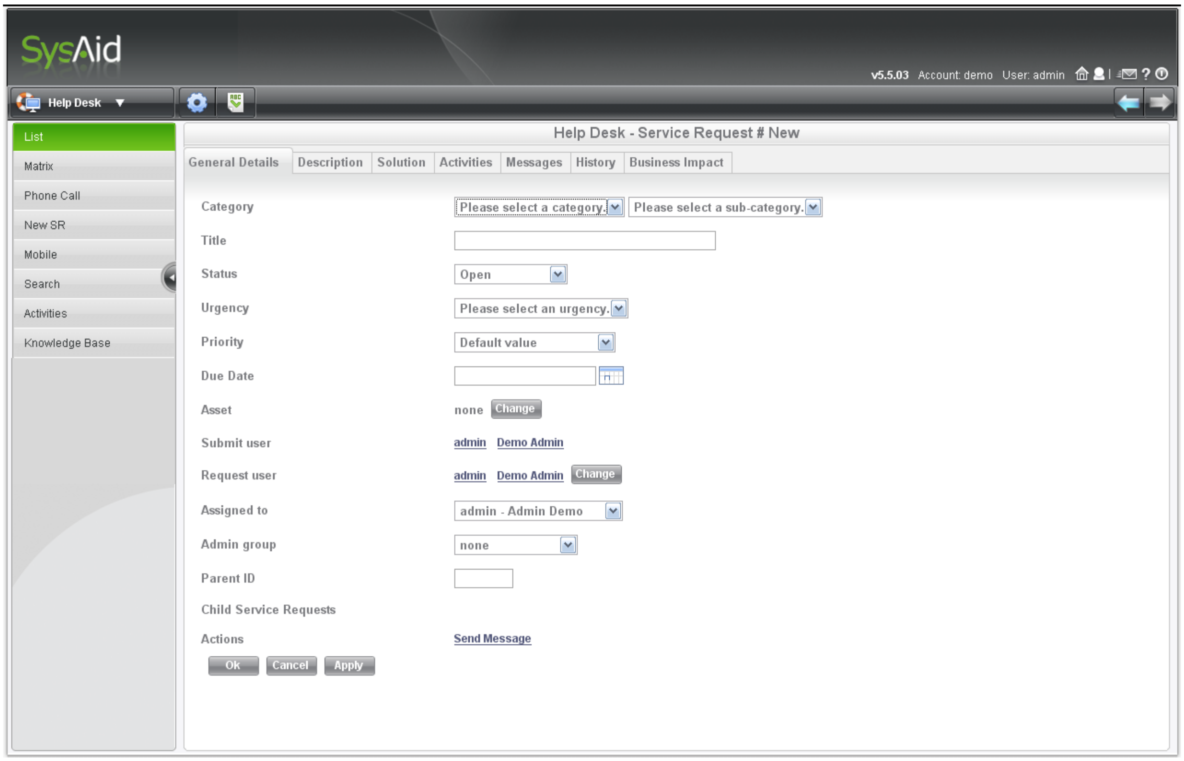


Figure 2.5: SysAid Help Desk Module (Sysaid.com, 2008)

2.6.1.3. LiveTime

LiveTime is a Java-based help desk or service desk that supports all major operating systems, databases and internet browsers. LiveTime uses sophisticated Artificial Intelligence (AI) techniques to promote first-time call-resolution. It boasts an intuitive UI ensuring that all relevant information is accessible when it is needed. LiveTime can thus be considered as having an intelligent user interface (IUI). The LiveTime help desk has recently announced its availability for the iPhone (Figure 2.6), which is ideal for technicians, enabling them easy access to LiveTime's configuration management and providing them with true mobility (LiveTime.com, 2008).

LiveTime provides the following main functionality:

- Incident Management;
- Configuration and Asset Management;

- Service Level Management;
- Management Reporting;
- Knowledge Management; and
- Self-service.

Figure 2.6: LiveTime Support for Mobility (LiveTime.com, 2008)

2.6.1.4. FrontRange Solutions-Heat

The Nelson Mandela Metropolitan University (NMMU) Information Communications Technology (ICT) Service Desk uses the HEAT Product Suite as its service desk application. HEAT provides tools to log and resolve calls, track call activity, manage progress and process of calls, store information about customers and their problem history, track information on the helpdesk's performance, spot trends and generate detailed reports. It consists of different modules for assisting both managers and HEAT users to work with calls (FrontRange.com, 2008).

These modules are (FrontRange.com, 2008):

- Call Logging;
- Auto Ticket Generator;
- Business Process Automation Module (BPAM);
- HEAT Manager's Console;
- HEAT Quick Start Wizard; and
- HEAT Answer Wizard.

HEAT enables the customisation of its UIs. The customised Call Log UIs of NMMU's ICT Service Desk can be seen in Figure 2.7. The HEAT call log screen (Figure 2.7) appears cluttered and the process of logging a customer's query remains unclear. An interview conducted with the CCAs at the NMMU ICT Service Desk revealed that the interface is overwhelming for novice CCAs (Vermaak, 2008).

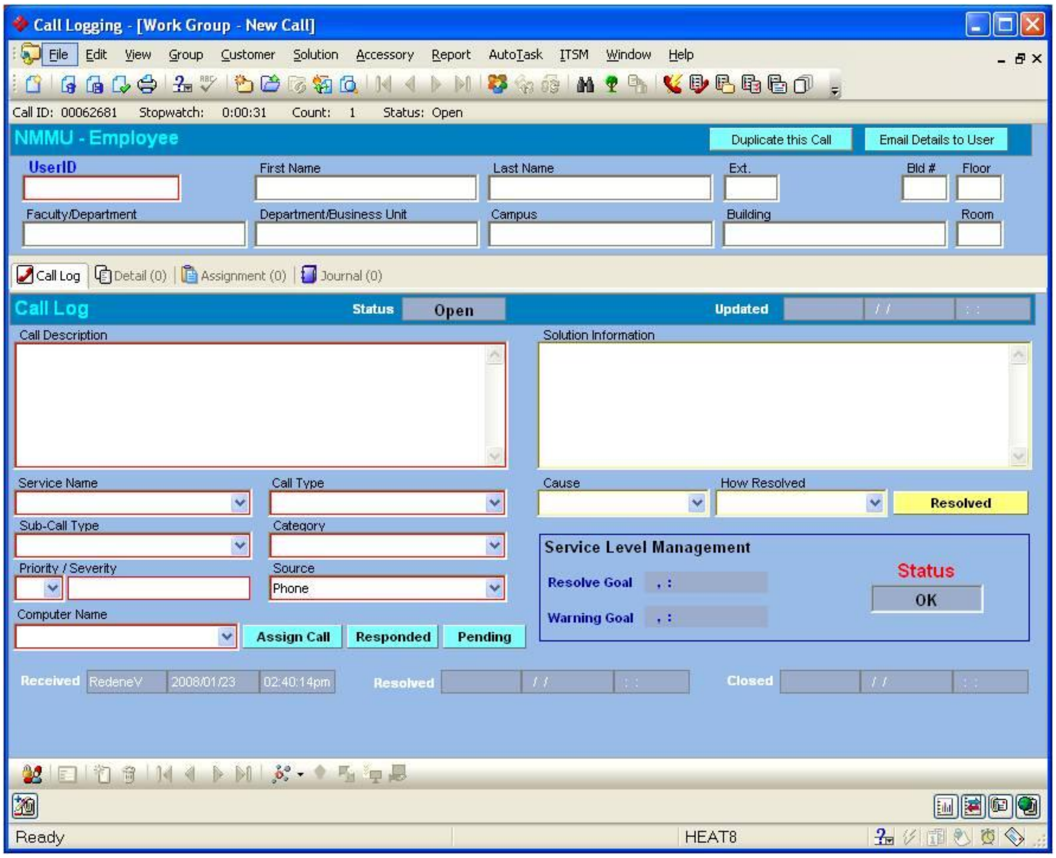


Figure 2.7: HEAT Call Log Screen

2.6.1.5. Intelligent Service Desk

Singh (2007) implemented an IUI prototype, Intelligent Service Desk (ISD), which provided the capabilities required to log, diagnose and resolve a customer's query, retrieve task-based information and deliver dynamic feedback. The NMMU ICT Service Desk was used as a case study. Singh (2007) implemented two windows. The first window (Figure 2.8) supports the logging and diagnosing of customer queries and the second window supports the resolution of those queries (Singh, 2007). As seen in Figure 2.8, the logging of customer queries' UI is divided into three sections, namely A, B and C.

Section A contains task status information. The name of the current task, a description of the current step and the total number of steps for the current task are displayed in this section. This section contains a visual step indicator indicating the completion status of the current task through the use of colour. Section B represents the direct manipulation section. This allows for user input in the form of customer, incident and resolution details.

Section C displays system feedback in the form of information on demand to help reduce the time in which information is found. System feedback displayed here can either take the form of customer details, incident categorisation keywords or possible solutions. If the user has entered an incorrect data type, an error will be displayed in the system feedback section.

Thus, intelligence was added to the windows through the delivery of task-based support in the task status section (Figure 2.8 (A)) and intelligent feedback in the system feedback section (Figure 2.8 (C)).

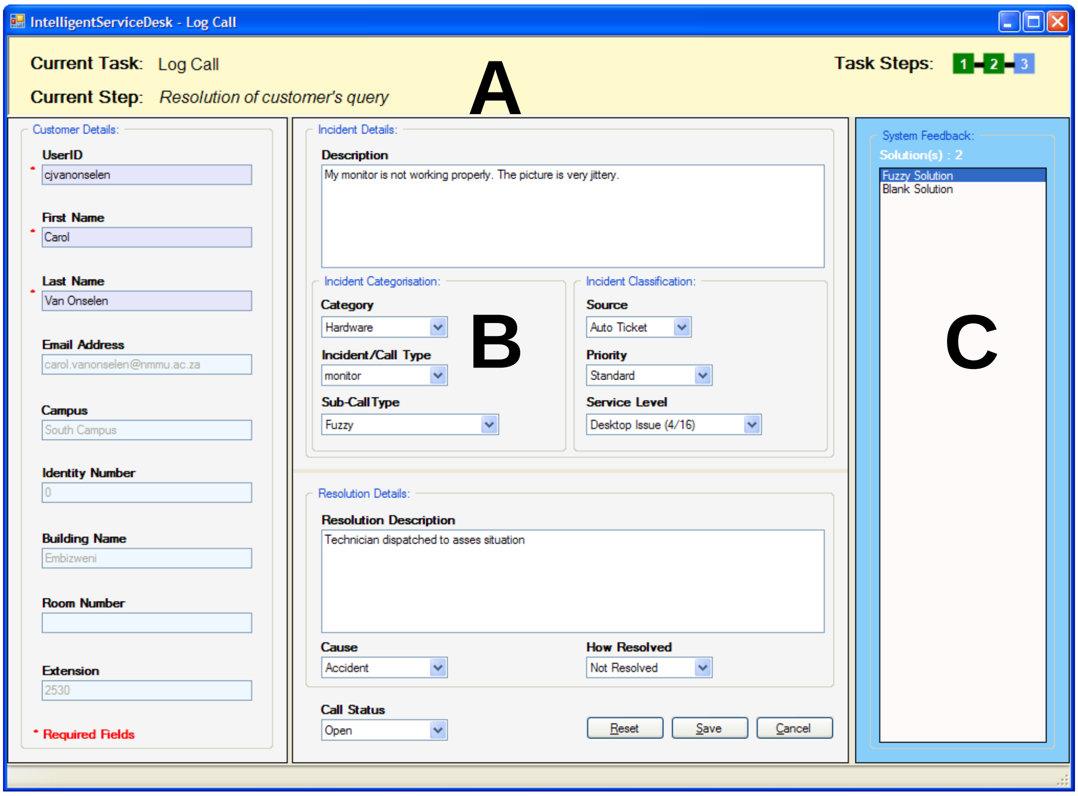


Figure 2.8: ISD Query Diagnosis Window

2.6.1.6. Software Comparison

Section 2.6.1.1 to 2.6.1.5 presented examples of some existing software illustrating the functionalities and benefits provided by help desk and service desk applications. This section provides a tabulated (Table 2.3) comparison of the above mentioned CC software with regards to the following criteria:

- Clear indication of call logging process/steps;
- Customisability of the UI;
- Support for both novice and expert CCAs with regards to using the UI; and
- Inclusion of an IUI.

The first three selected criteria were established by reviewing the issues in designing UIs for CCs (Section 2.6). The last criterion, namely the inclusion of an IUI, was selected due to studies performed by Singh (2007). Singh (2007) found that an efficient and effective interface for CCs could be achieved through the application of IUIs, offering significant benefits to the domain of CCs (Singh, 2007).

	CONTACT CENTRE SOFTWARE				
	SiteHelpDesk-IT	**SysAid**	**LiveTime**	**HEAT**	**ISD**
Clear Call Logging Process	✗	✗	✗	✗	✗
Customisable	✓	✓	✓	✓	✓
Novice and Expert Support	✗	✗	✗	✗	✗
Intelligent Interface	✗	✗	✓	✗	✓

Table 2.3: Contact Centre Software Comparison

None of the CC software mentioned considers the varying skill levels of CCAs with respect to using UIs. Newly appointed CCAs do not remain at the same skill level; thus it is important to build the change dynamic into the system at the beginning. Adapting the interface (Chapter 4) according to the user’s need facilitates the varying skill levels of the

CCAs. None of the CC UIs previously mentioned exploits the advantages which an AUI could offer and this provides motivation to conduct this research study.

2.7 Call-Resolution Process

The main goal of any CC is to resolve their customer's queries in order to improve customer satisfaction. The Merchants Report (2006) defines a resolved call as:

> *"A customer enquiry or transaction that is resolved or completed to the customer's satisfaction by the initial agent, or another resource that the call has been escalated to, and on which no further manual action needs to be taken by the initial agent or any other resource within the organisation after the call has been completed, other than initiating an automated process or standard post-call administration."*

The call-resolution process is initiated by the logging of a customer's query. There are many paths the call can follow once placed by a customer. The simplest case involves the CCA resolving the customer's query. In reality, however, there are situations in which the CCA is not able to assist the customer and, depending on the organisational structure (Section 2.4) implemented, the call must be transferred to another CCA (Gans *et al*., 2003).

The primary steps involved when logging a customer's query are (Vermaak, 2008):

1. Provide customer details;
2. Provide the call's details;
3. Assign the call; and
4. Provide the call's solution (resolution) details.

It is critical to find a balance between productivity, performance, quality and customer satisfaction in order to optimise the performance of CCAs (Fluss, 2005). In order to improve customer satisfaction, the speed at which CCAs assist customers is vital. Speedy call resolution by the initial CCA handling the call is an ideal approach to ensure customer satisfaction. (Gans *et al*., 2003). Possible factors which affect the call-resolution process are (Dimension Data, 2006):

- *Complexity of calls*

One can expect calls which are long in duration to be more complex.

- *System integration*

  A CC with little integration will require the CCA to toggle between systems to find appropriate information to help customers.

There is a possibility that there will be an increase in call duration as the complexity of calls is increased. A focus on system integration may mitigate this (Dimension Data, 2006). An efficient and effective CC system that facilitates faster call logging is a viable approach to improve customer satisfaction.

Singh (2007) illustrated how the inclusion of an IUI could be used to improve the call-resolution process within a CC (Section 2.6.1.6). The use of an IUI within a CC allows a CCA, at the first- level of support, to be both a call taker and a problem solver (Figure 2.9 (a)). Escalation of a query is only done if the query cannot be resolved here. Inclusion of the IUI could reduce the time and costs for query resolution. Time could be reduced by potentially having resolved the customer's query on the first call. Costs would be reduced by eliminating the need to outsource the problem. This process reserves second- and third-level support for critical issues (Singh, 2007).

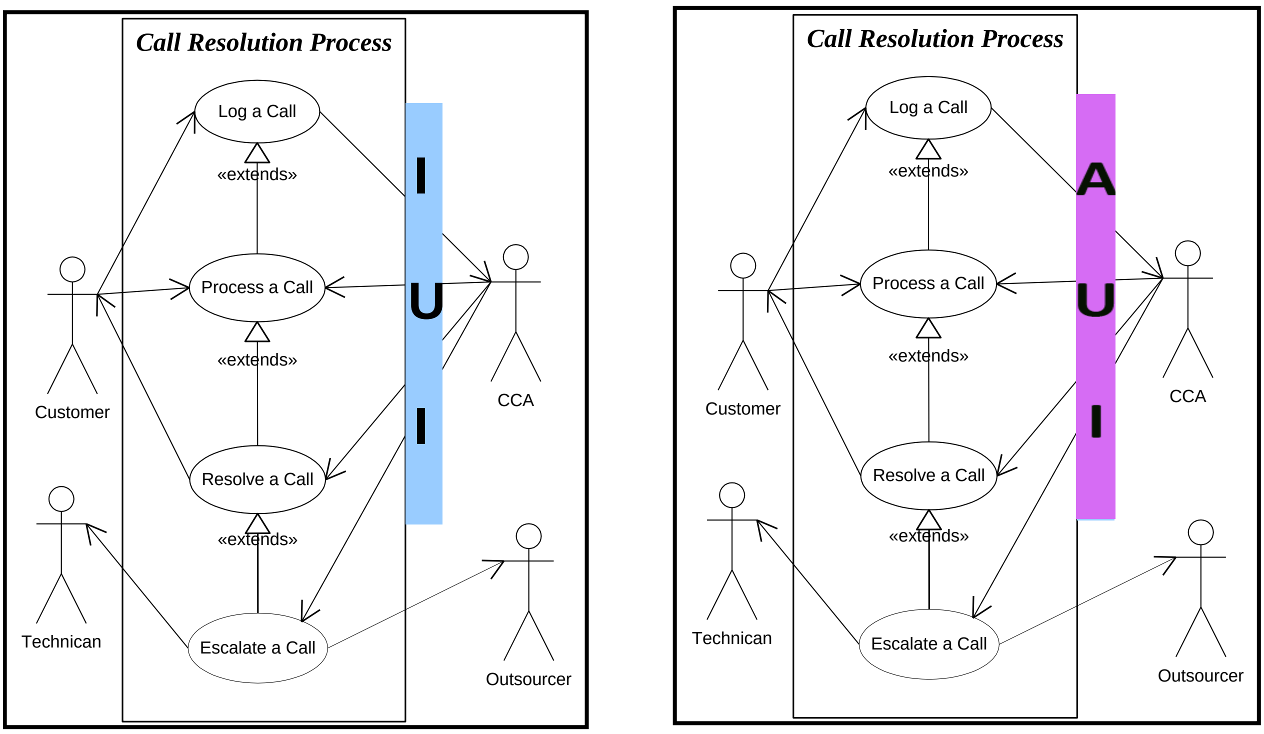


Figure 2.9: Call-Resolution Process Using an (a) IUI (Singh, 2007) and (b) AUI

CCAs can, however, have different expertise levels. Newly appointed CCAs are still novices until they become more competent in handling calls (Section 2.5.1). The importance of training CCAs has been reiterated as well-trained CCAs can resolve queries in a timelier manner. Well-designed CC UIs have been linked directly with reducing training time (Section 2.6). Figure 2.9 (b) shows how an AUI could be used to personalise and customise the CCA's experience.

2.8 Summary

CCs provide many business advantages by delivering an improved customer service (Section 2.2). The areas of operation provided by CCs are either a help desk or a service desk whereby either incoming or outgoing calls are handled (Section 2.3). Customers interact with CCAs and the way the customer's call is handled depends largely on the CC's underlying organisational structure (Section 2.4).

Even though contact centres utilise state-of-the-art technology, all CCs rely on CCAs to act as the intermediaries between the information in the database and the servicing of the customers. These CCAs need to be trained and skilled in customer service (Section 2.5.1). There are various training methods used; however, good UI designs can help mitigate costs involved in training (Section 2.5.2 and 2.6).

Existing CC software does not support CCA's different expertise levels and none have a UI to assist novice CCAs (Section 2.6.1). AUIs are a possible way of alleviating this problem. An IUI (the broader research area of AUIs) has been shown to be a promising approach when included in the call-resolution process of a CC (Section 2.7). The next chapter further investigates UIs, users' differences in expertise and IUIs in order to understand how UIs can be designed for CCAs' differing skill levels and acts as a background to the research area of AUIs.

Chapter 3: User Expertise and User Interfaces

3.1 Introduction

The previous chapter, Chapter 2 on Contact Centres, deduced that good computer user interfaces (UIs) can reduce contact centre agents' (CCAs) response time and training costs. An approach to improve the CCA's productivity is to provide them with a UI that will increase the speed at which they assist customers, by supporting their different capabilities and skills. Designing for both novice and expert users is not however a simple task. The difference in behaviour between novice and expert users needs to be investigated before the design of interfaces can commence.

The purpose of this chapter is to provide a literature review of user expertise levels and interfaces which cater to users' varying skill levels. This literature review is necessary to provide background information on how adaptive user interfaces (AUIs) could be used within the domain of CCs and how these interfaces can cater for the differing skill levels of CCAs. The evolution of UIs from multi-Layer UIs to intelligent user interfaces (IUIs) will be discussed.

3.2 User Interfaces

The following definition of an interface is given by Koelle (2004):

> *An interface provides a means of communication between two or more entities.*

Specifically related to computer systems, a human-computer interface is that part of the computer system with which a person interacts to achieve some task (Koelle, 2004). It is regarded as the most important part of any system as it represents its capabilities to most users. UI design is seen as a subset of the field of study called, human computer interaction (HCI). HCI is regarded as the study, planning and design of how people and computers work together so that a person's needs are satisfied in the most effective way (Galitz, 2007).

In the early days of personal computers, UIs consisted of text-based interfaces referred to as command-line interfaces (Figure 3.1). An inexperienced user, however, found command driven programs difficult to use because of the number of commands that had to be learnt. UIs have come a long way since the simple command-line interface, as seen in Figure 3.1.

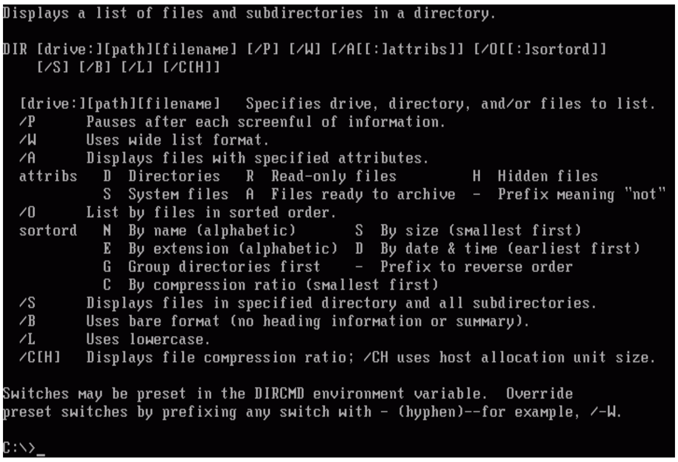


Figure 3.1: Command-line Interface

Graphical User Interfaces (GUIs) have since replaced command-line interfaces. GUIs use graphic icons and controls in addition to text (Figure 3.2). In addition to their visual components, GUIs also make it easier to move data from one application to another. Despite the advancement from command-line interfaces, some GUIs are still hard to learn for inexperienced users.

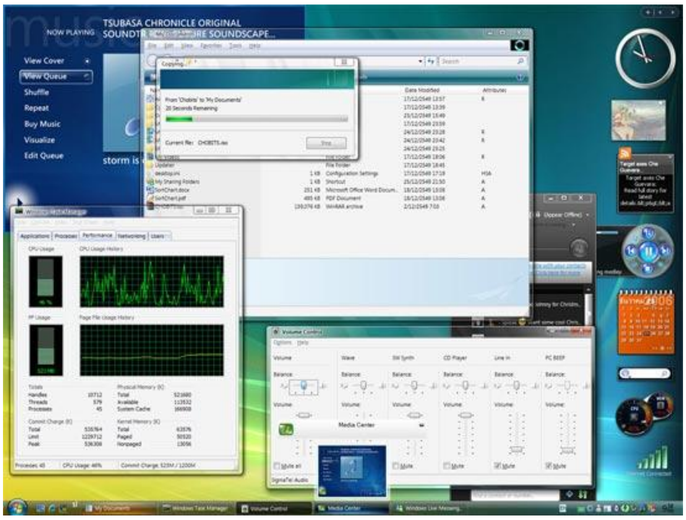


Figure 3.2: Graphical User Interface

Researchers working on future UIs often put the same or even more emphasis on tactile control and feedback or sonic control and feedback rather than on visual feedback obtained from GUIs. The term, GUIs, has become inadequate in these situations and as a result, direct manipulation is the term that was more widely used. Direct manipulation is a HCI style that was coined by Ben Schneiderman and allows users to directly manipulate objects presented to them. This interaction style offers the user control and predictability in their interfaces and is closely related to windows, icons, menus, and a pointing device (WIMP GUI) (Kumar, 2005).

People interact with computers quite differently than they did 20 years ago when one typically had one user to each computer and everything was in a particular place. Computer users were mostly professionals, and there was a limit to the amount of information on a computer. It was completely structured and well organised and was completely static as nothing changed unless the user directly manipulated the UI. Inflexible UIs did not reflect the diversity among users as users differ in their preferences, working methods and levels of expertise.

3.3 Differences in Expertise

There have been various methods used to classify the different, and at times, changing characteristics of users as they gain experience with computer systems. A new, relatively new or infrequent user has been given various names such as naïve, casual, inexperienced or novice; whereas, more skilled users have been given names such as experienced, full-time, frequent, power or expert (Galitz, 2007). This research uses the terms novice and expert to differentiate between inexperienced and experienced users.

Prumper (1991) states that there are usually two overlapping criteria that are used to differentiate novice and expert users: knowledge and the time spent working with a particular system (Prumper *et al.*, 1991). Nielsen (1993) supports Prumper (1991) and declares three main dimensions along which users' experiences differ (Figure 3.3). These three dimensions are (Wu, 2000, Nielsen, 1993):

1. *Experience with the system (application software).*

   This refers to the degree to which users have worked with similar software applications.

2. *Experience with computers in general.*

   This refers to the amount of experience users have working with computers and their computer literacy.

3. *Experience with the task domain (task experience).*

   This refers to the amount of experience the user has of the tasks the system will be performing.

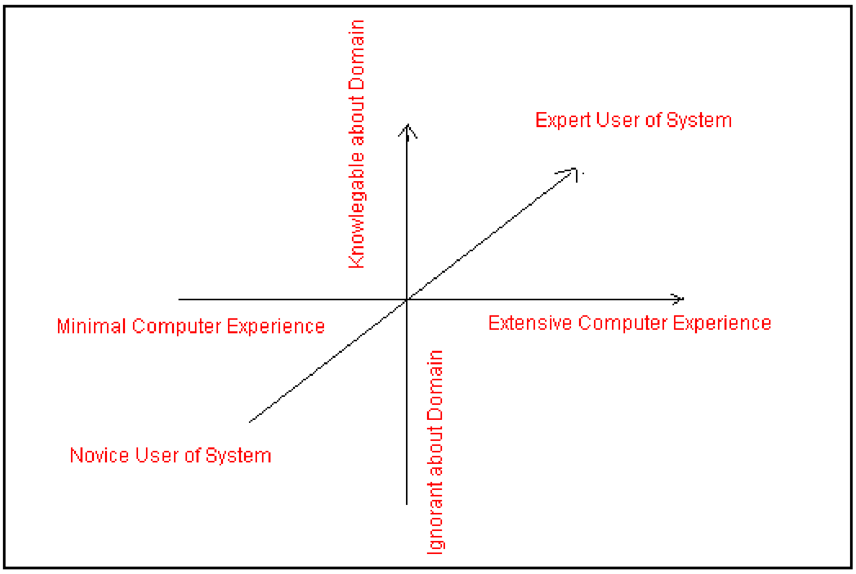


Figure 3.3: User Cube of Expertise (Nielsen, 1993)

The most common dimension used when discussing user expertise is the user's experience with a specific UI. Early work focused on a one-dimensional range of user expertise, namely novice and expert users. Focusing on only novice and expert users is a simplistic approach as users tend to be expert in some features and novice in others (Saxena, 1993). Evidence is available to support the fact that novice and expert users behave differently (Hurst *et al.*, 2007).

3.3.1. *Qualitative Differences*

Novice and expert users exhibit noticeable differences in their respective behaviours, i.e., they have different ways of thinking. This section explores some of their key characteristics.

3.3.1.1. Novice Behaviour

Novice users are generally concerned with how to do things instead of how fast they can do them (Buxton *et al.*, 1993). They require the interface to be easy to learn so that they could become expert users quickly (Wu, 2000). Novice users have been found to possess the following characteristics (Galitz, 2007):

- They have a fragmented conceptual model of the system;
- Their knowledge is ordered less meaningfully, orienting it towards surface features of the system;
- They structure their information into fewer categories;
- They have difficulty in making inferences and relating new knowledge to their objectives and goals; and
- More attention is paid to low-level details and to surface features of a system.

3.3.1.2. Expert Behaviour

Expert users do not only know more than novice users, they know “differently”. They have the ability to take large amounts of information and see it as connected units. Expert users are goal orientated. When using an interface, they quickly deduce goals and actions to achieve those goals. They want a highly efficient interface and would, therefore, like the number of interactions to be reduced (Wu, 2000). Expert users have mental models that are closer to the system’s model (Kellogg and Breen, 1987). In summary, expert users have been found to possess the following characteristics (Galitz, 2007):

- They have an integrated, conceptual model of the system;
- Their knowledge is ordered more abstractly and more procedurally;
- Information is organised more meaningfully, orienting it towards their task; and they structure their information into more categories;
- They have a better ability to make inferences and can relate new knowledge to their objectives and goals; and
- Less attention is paid to low-level details and surface features of a system.

3.3.2. *Quantitative Differences*

Sections 3.3.1.1 and 3.3.1.2 convey the varying characteristics of both novice and expert users. These differences in behaviour should manifest themselves in measurable differences in user actions (Hurst *et al*., 2007). This section discusses these differences, particularly with regards to performance and searching mechanisms. Finally, these differences are considered in order to design UIs for novice and expert users (Section 3.3.3).

3.3.2.1. Performance

Novice and expert users' behaviour usually differs dramatically at the physical level of interaction. Novice performance builds the skills that transition to expert performance when the basic actions of the novice and expert are the same (Buxton *et al*., 1993). The performances of novice and expert users were compared in a study of information retrieval through searching a database. It was found that expert users performed significantly faster than novice users (Dillon and Song, 1997).

Oka and Nagata (1999) recorded the history of the keyboard and mouse operations of both novice and expert users. They found that novice users perform many mouse operations where the mileage of each operation is long. Contrasting to the novice users, expert users perform fewer mouse operations where the mileage of each operation is shorter than the novice's one.

3.3.2.2. Searching Mechanisms

Novice and expert users have different methods of selecting the correct menu item. An expert user typically knows the menu item he or she wants to select and is able to memorise (i.e., recall) the location of menu items. Contrasting with expert users, novice users do not tend to know what menu item they want to select or its location and have to search for it. This means that expert users generally tend to make faster menu selections. This also suggests that features that differentiate searching behaviour from other types of motion may help to differentiate novice and expert use. When searching menus, features that approximate searching include, but are not limited to (Hurst *et al*., 2007):

- The number of submenus open; and
- How often the cursor "dwells" over a menu item.

Supporting the work done by Hurst *et al*. (2007), a usability evaluation conducted on a nursing-assessment system revealed that novice users had difficulty locating where information should be entered into the system, whereas the experts could complete the tasks and had learned to use the system as a checklist for collecting the necessary information (Bourie *et al*., 1997).

A well-designed system must support both novice and expert users, taking into consideration their differences (Galitz, 2007). This is a challenging aspiration: the UI design needs to accommodate experts while, at the same time, reduce its complexity for novice users.

3.3.3. *Designing User Interfaces for Novice and Expert Users*

Padilla (2003) states the efficiency of using an application is limited by the motor load for expert users and by the cognitive load for novice users. Novice users can be categorised by the fact that they are unfamiliar with a UI. The UI can incorporate rich explanatory elements to assist with novice user's cognitive understanding of the computer screen. This explanatory value does, however, come at an expense for expert users (Figure 3.4) (Padilla, 2003).

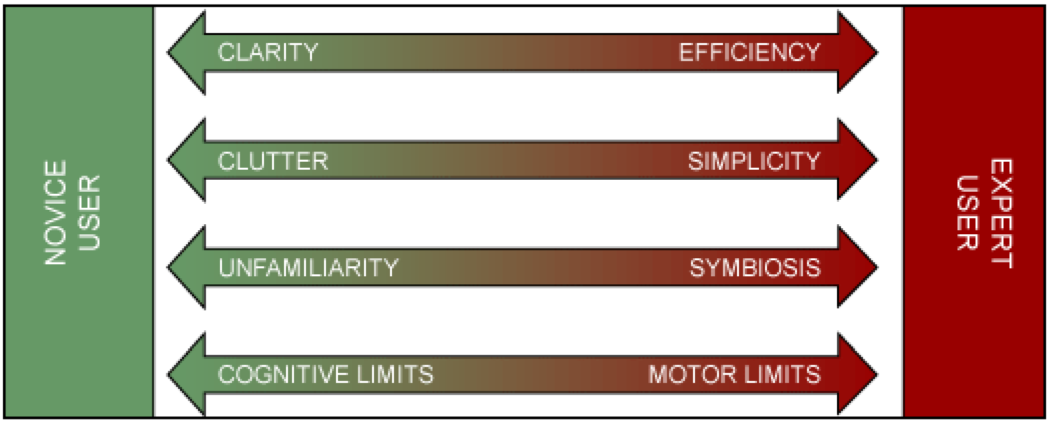


Figure 3.4: The Spectrum of Users' Needs (Padilla, 2003)

3.3.3.1. Designing for the Novice User

Interfaces that are intentionally designed for novice users may not need special help systems, as they should include all the necessary user assistance in the primary interface itself. The system will have to explain to users, with little domain knowledge, what it is doing and what different options mean (Nielsen, 1993).

Novice users have been found to need overviews, buttons for selections, and guided tours within web-page design. In business systems in particular, novice users have been found to (Galitz, 2007):

- Depend on system features such as menus, prompting information, and help screens that assist recognition memory;
- Need restricted vocabularies, simple tasks, small number of possibilities, and very informative feedback; and
- View practice as a means to reach expert status.

Some common ways of assisting novice users are simplifying options that limit the number of features, while making those features more comprehensible. Design strategies to reduce complexity for first-time and novice users have been proposed and demonstrated to be advantageous even in early systems such as the training wheels' interface (Shneiderman, 2003).

The training wheels' concept, a term initially coined by Carroll and Carrithers in 1984, is an example of a multi-layered interface (Section 3.4). It involves displaying the basic functions to the user while blocking the typical new-user errors, making them unreachable (Carroll and Carrithers, 1984).

3.3.3.2. Designing for the Expert User

As previously mentioned, it is possible that an expert user may be a novice in certain parts of a system, especially if it is a complex one with many features. Expert users still require help with those parts of the interface they are not accustomed to and they will benefit from increased learning of those features (Nielsen, 1993). In particular to using systems, experts have been found to (Galitz, 2007):

- Rely upon recall rather than recognition;
- Expect rapid performance;
- Need less informative feedback; and
- Avoid novice memory aids, reduce the number of keystrokes, perform chunking and summarisation of information and introduce new vocabularies such that they could become more efficient.

Expert users can be catered for by including accelerators in the interface to allow them to use faster, but less obvious, interaction techniques (Nielsen, 1993). Accelerators satisfy the expert user's need for rapid performance and reduce the number of keystrokes used.

There are some graphical element aspects that are found to be desirable expert shortcuts/accelerators. These are (Galitz, 2007):

- Mouse-double clicks;
- Pop-up menus;
- Tear-off or detachable menus; and
- Command lines.

Expert users prefer smooth navigation paths, compact but in-depth information, fast page downloads and extensive services to satisfy their varying needs, as well as the ability to rearrange the interface within web-page design (Galitz, 2007). Computer environments are completely different than they were 20 years ago, and more flexible UIs are the state of the art. (Shneiderman and Maes, 1997, Ross, 2000). Flexible systems provide users with a greater freedom and improve users' efficiency as well as the correspondence between user, task and system characteristics. The next section discusses a particular type of UI, namely multi-layer UIs, which provide users with some kind of flexibility and cater to both novice and expert users.

3.4 Multi-Layer User Interfaces

Multi-layer UIs initially equip novice users with the basic set of functions (layer 1) enabling them to move to higher levels as they become more accustomed to the UI. These higher levels contain more functionality. The users are given control over which layer they feel most comfortable. The concept of the separation of layers, brought about by multi-layer UIs, empowers users to learn features in a meaningful sequence and also limits complexities brought about by menus and help screens (Shneiderman, 2003).

Figure 3.5 depicts the visual representation of a multi-layer design and illustrates its various possibilities. Numerous current designs (Figure 3.5 (A)) make all features available to the user without offering guidance about where to begin. Modularity is somewhat provided by the pull-down menu groupings but there remains little guidance about how to sequence learning. In contrast, multi-layer design (Figure 3.5 (B)) provides a clear sequence for learning and approximately has the same amount of features at each level, rendering this design too restrictive. An extension to this multi-layered design is to either first have a small layer and then multiple layers, with additional functionality

(Figure 3.5 (C)), or to have two to three thin layers followed by a modular design enabling users to choose relevant features, much like a mushroom (Figure 3.5 (D)) (Shneiderman, 2003).

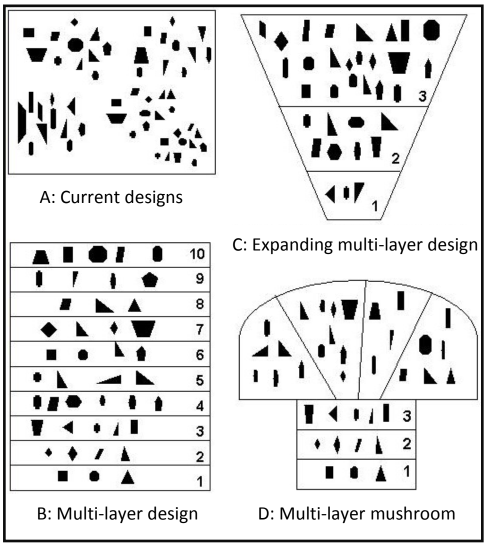


Figure 3.5: Multi-layer Designs (Shneiderman, 2003)

Commercial software often implements multi-layer UIs as professional or lite versions. The following figures (Figures 3.6 and 3.7) illustrate the use of the multi-layer design. Layer 1 (getting started) only provides typing and a small set of buttons without any pull-down menus (Figure 3.6 (A)). There are no error messages as this layer promotes safe exploration. Layer 1 does, however, offer simple help with animated instructions (Figure 3.6 (B)).

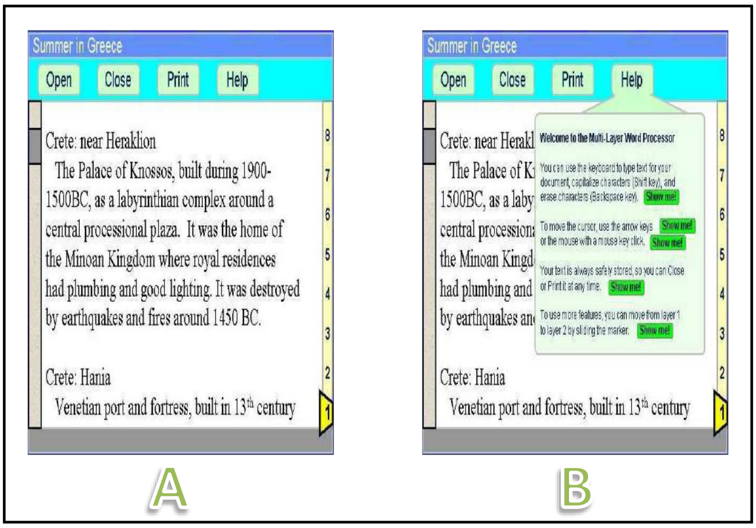


Figure 3.6: Multi-layer User Interface (Layer 1) (Shneiderman, 2003)

Layer 2 (basic editing) offers users additional features such as additional buttons, fonts, a ruler and a status bar (Figure 3.7 (A)). Layer 3 (formatting) provides users with even more additional features boasting pull-down menus with columns, paragraph controls, headers and footers, comments, and find and replace (Figure 3.7 (B)) (Shneiderman, 2003).

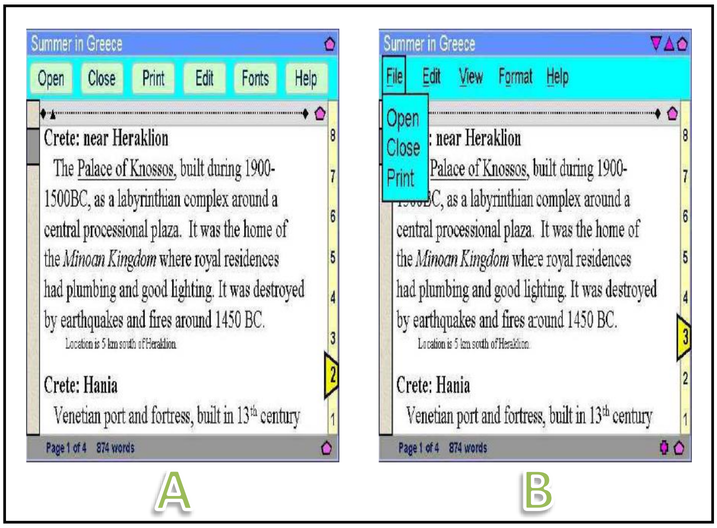


Figure 3.7: Multi-layer User Interface (Layer 2 and 3) (Shneiderman, 2003)

Multi-layer interfaces facilitate the learning of UIs and are a viable approach to get users started with an interface. Multi-layer interfaces also accommodate both novice and expert users. An improved approach would be to make these multi-layer interfaces adaptive: the user automatically proceeds to the next level based on their performance in the current level. Limited research has been done on adapting multi-layer UIs (Clark and Matthews, 2005). Clark and Matthews (2005) provided two techniques for adapting multi-layer UI: the first technique provided adaptation based on the user's use of a similar application; and the second provided adaptation based on the user's application start-up style (Clark and Matthews, 2005). This research is, however, limited as layers are added only with first use of the application and not dynamically at runtime. Before introducing the concept of adaptivity, there is a brief review of its broader research area, namely IUIs.

3.5 Intelligent User Interfaces

IUI can be considered the next wave of interfaces and constitute a major direction of current HCI research towards the provision of high-quality user-computer interaction. They are considered especially important if the main requirement of the system is to support heterogeneous user groups with variable and diverse needs, abilities and preferences since they facilitate a more 'natural' interaction (Maybury and Wahlster, 1998, Karagiannidis *et al*., 1995, Horgen, 2001).

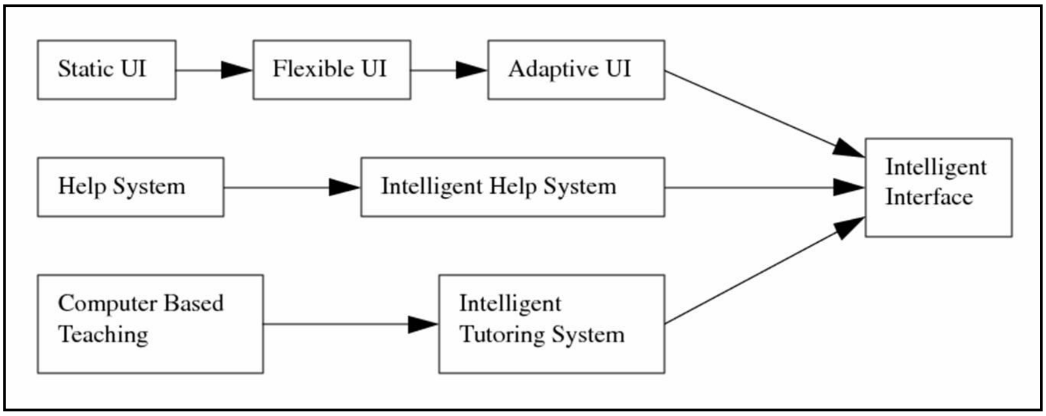


Figure 3.8: Intelligent User Interface (Dieterich *et al*., 1993)

Dieterich *et al.* (1993) define an IUI (Figure 3.8) as the integration of an AUI (Chapter 4) with an Intelligent Help System and an Intelligent Tutoring System (Dieterich *et al*., 1993). For the purpose of this research, the following formal definition of an IUI as proposed by Maybury and Wahlster (1998) is used:

> *IUIs are human-computer interfaces that aim to improve the efficiency, effectiveness, and naturalness of human-machine interaction by representing, reasoning, and acting on models of the user, domain, task, discourse, and media (e.g. graphics, natural language, gesture).*

Efficient interaction means the ability to complete tasks within a shorter amount of time. Effective interaction means doing the right thing at the right time such as tailoring content according to context. A more natural interaction means support for natural language (Horgen, 2001).

IUIs are a multi-disciplinary research area, which has been influenced by other research fields. Some of these research fields (e.g., psychology, cognitive sciences and artificial intelligence (AI)) as well as their relationships to other IUI research topics are visually depicted in Figure 3.9. The largest research field which has been related to IUIs is HCI (Figure 3.9) where IUIs are considered to be a subfield of HCI (Alvarez-Cortes *et al*., 2007).

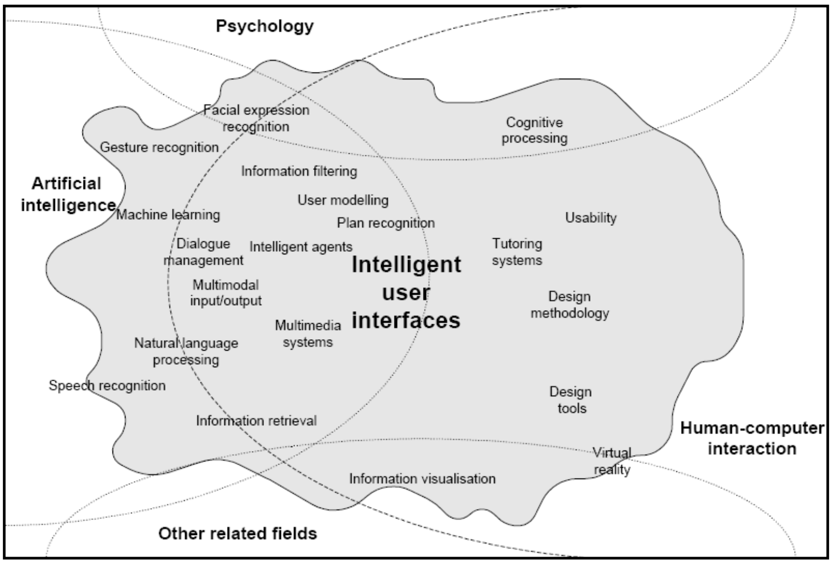


Figure 3.9: The Multi-disciplinary Research Area of Intelligent User Interfaces (Ehlert, 2003)

3.5.1. Intelligent User interfaces and Direct Manipulation

IUIs have been proposed as a means to overcome some of the problems that direct manipulation interfaces cannot handle, for example (Hook, 2000, Alvarez-Cortes *et al.*, 2007):

- *They reduce information overflow problems.*
  Irrelevant information can be filtered out by IUIs whilst the user searches for information in complex systems or large databases. This reduces the user's cognitive workload.
- *They provide help on how to use complex systems or real-time cognitive overload problems.*
  Some applications are complicated to use at first without obtaining help. An intelligent help system could explain certain functions and provide information to simplify tasks.
- *They create personalised systems.*
  As mentioned in Section 3.3, users are diverse and have different needs. IUIs could utilise these differences to create a more personalised user experience.
- *They take over tasks from the user.*
  An IUI could detect which task the user is currently working on, recognise the user's intent and take over the user's task, enabling the user to focus on other activities.

3.5.2. Intelligent Techniques

The most important properties of IUIs are their ability to improve user and machine communication, and provide personalisation and flexibility of use (Ehlert, 2003, McTear, 2000). These improvements could be achieved by utilising a variety of intelligent techniques (Waern, 1997). Some of these are (Ehlert, 2003):

- *Intelligent input technology*

  Innovative techniques such as natural language and gaze and eye tracking are used by IUIs to obtain user input.
- *User modelling*

  User modelling techniques allow the system to obtain information about users and maintain or infer this obtained knowledge. User modelling is covered in more detail in Section 4.5.1.
- *User adaptivity*

  User adaptivity includes all techniques which enable the system to adapt to different users and different usage situations; hence, creating more personalised systems (Section 3.5.1).
- *Explanation generation*

  Explanation generation techniques, such as information visualisation enable systems to explain their outcomes to users.

3.6 Adaptive User Interfaces

AUIs and IUIs are not independent from each other but rather share a mutual relationship. A UI has to make several communication decisions while interacting with a user. These decisions may concern several aspects of the interaction, such as *what*, *when* and *how* to communicate. In this context, a UI can be called intelligent to the measure that it adapts itself, and makes these communication decisions dynamically at run-time, based on the requirements of the interaction. Thus, adaptivity is recognised as a central component of an IUI (Horgen, 2001). The definition of an IUI, given by Dieterich (1993), further supports the concept of AUIs being a part of the IUI research field (Figure 3.8).

AUIs are the key to creating personalised systems. Their sole task is to provide an interface most suitable to users' needs whilst facilitating the users' varying skill levels. Multi-layer UIs (Section 3.4) were discussed in this chapter as a possible approach to facilitate users' varying skill levels, however, these types of interfaces are not as flexible as AUIs. Within the context of CCs, AUIs are ideal for both novice and expert CCAs; therefore they will be discussed thoroughly in the next chapter.

3.7 Summary

UIs and interactions have changed considerably since their first inception (Section 3.2). Interfaces in the past did not reflect the diversity among users. Users have differences in expertise when using interfaces and these differences are both qualitative and quantitative in nature. Careful considerations need to be in place when designing UIs for novice and expert user (Section 3.3).

Multi-layer UIs (Section 3.4) enable users to switch between layers as they become more accustomed to the UI. These interfaces cater for both novice and expert users and facilitate learning of UIs; thus, they could be a viable approach within the CC domain. Multi-layer UIs are, however, not dynamic and they place more workload on the user.

IUIs have already proven to be a successful approach for CCs. Sections 3.5.1 and 3.5.2 clearly show adaptivity as being a central characteristic of IUIs. AUIs provide the personalisation needed for UIs; thus, it is proposed that an AUI incorporating the benefits of multi-layer interfaces could be used to facilitate learning done by CCAs, improving their performance. The next chapter further investigates the research area of AUIs.

Chapter 4: Adaptive User Interfaces

4.1. Introduction

The previous chapter, namely User Expertise and User Interfaces, discussed the various design considerations of novice and expert user interfaces (UIs). This was done to investigate how the varying skill levels of contact centre agents (CCAs) could be accounted for. Multi-layer UIs were discussed as an appropriate technique for incorporating novice and expert CCAs; however, these interfaces are not dynamic. Chapter 2 showed that an intelligent user interface (IUI) has been successful in assisting CCAs with the call-resolution process.

Adaptive user interfaces (AUIs), which are considered a sub-field of IUIs, are ideal in providing personalised systems. AUIs can cater to the various skill levels of users. The purpose of this chapter is to provide a thorough literature review of previous work done with AUIs in order to determine how such interfaces can be applied to the contact centre (CC) domain, in a dynamic way, assisting novice CCAs.

The chapter commences with a discussion on the definition of AUIs (Section 4.2). A discussion on the components of adaptivity will follow (Section 4.3). The components of adaptivity are discussed in terms of the afferential (Section 4.4), inferential (Section 4.5) and the efferential component (Section 4.6). Stages of the adaptive process (Section 4.7) are discussed before a discussion on the challenges facing AUIs (Section 4.8) can resume.

4.2. Definition of an Adaptive User Interface

Adaptive and adaptable UIs have been a theme of HCI for a long time (Mitchell and Shneiderman, 1989, Dieterich *et al*., 1993). However, before embarking on the extensive research field of AUIs, a clear distinction needs to be made between an adaptable and an adaptive UI.

A system is called adaptable if the user is provided with tools to customise the user interface. This is an attractive objective: to provide the user with facilities for tailoring the system according to his/her personal tasks and needs. This kind of individualisation provides control over the adaptation to the user. Even though the user is given full control, adaptations handled by the user are often restricted to a low level (Jameson, 2002).

While AUIs could possibly overcome limitations brought about by adaptable UIs, they introduce new problems of their own (Section 4.8). Benyon and Murray (1993) formally define AUIs as:

> *Systems which can automatically alter aspects of their functionality and/or interface in order to accommodate the differing needs of individuals or groups of users and the changing needs of users over time.*

Similar to the definition given above, Langley (1999) has defined an AUI as being:

> *A software artefact that improves its ability to interact with a user by constructing a user model based on partial experience with that user.*

The common theme presented by the above definitions is that adaptivity occurs *automatically* according to the *user's needs*; where the user's needs are formally specified in the second definition as the user model. For the purpose of this research, a combination of the above two definitions will be used, defining an AUI as:

> *A software artefact which can automatically alter aspects of its functionality and/or interface and improves its ability to interact with a user by constructing a user model based on partial experience with that user.*

The above definition distinguishes the difference between an adaptable and an adaptive interface. The core difference is that adaptable interfaces enable the user to provide the personalisation; whereas, with adaptive interfaces, the system provides the personalisation. Table 4.1 displays a more detailed comparison between adaptive and adaptable interfaces. The main advantage of adaptable interfaces is the control given to users. Users of adaptive interfaces have no control; however, little effort is required by them (Table 4.1).

	ADAPTIVE	ADAPTABLE
Definition	System dynamically adapts to the current task and current user.	User changes the functionality of the system.
Knowledge	Contained in system and projected in different ways.	Extended knowledge.
Strengths	• Little (if any) effort by the user. • No special user knowledge required.	• User is in control. • User knows her/his task best. • System knowledge will fit better. • Success model exists.
Weaknesses	• User has difficulty developing a coherent model of the system. • Loss of control. • Few (if any) success models exist.	• Systems become incompatible. • Substantial work required by users. • Complexity is increased due to user's learning of adaptation component.
Mechanisms Required	• User, task and dialog models. • Incremental update of models. • Knowledge base of goals and plans. • Powerful matching capabilities.	• Layered architecture. • Domain models and domain-orientation. • "Back-talk" from the system. • Design rationale.
Application Domains	• Active help systems. • Critiquing systems. • Differential descriptions. • User interface customization. • Information retrieval.	• Information retrieval. • End-user modifiability. • Tailorability. • Filtering. • Design in use.

Table 4. 1: A Comparison between Adaptive and Adaptable Systems (Fischer, 2001)

An AUI does not exist in isolation and is designed to interact with human users (Figure 4.1). Figure 4.1 shows the human user's involvement in the process of adapting the UI. The user model is built using past user interaction data and the user model is then used for the UI adaptation effect which is displayed to the human user (Figure 4.1). Furthermore, as stated in the definition, the interface is only adaptive if it improves its interaction with the specific user. Simple memorisation of such interactions does not suffice. Improvements should also rather result from generalisation over past experiences and carry over to new interactions (Langley, 1999).

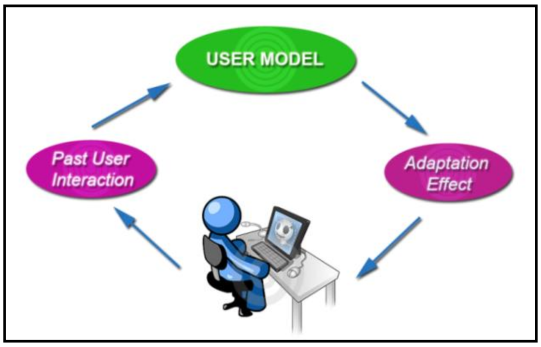


Figure 4.1: The Interaction of an AUI with a Human User

The definition of an AUI will seem familiar to some researchers, as it takes the same form as common definitions for machine learning. The main differences are that the user plays the role of the environment in which learning occurs, the user model takes the place of the learned knowledge base, and interaction with the user serves as the performance task on which learning should lead to improvement. In this view, AUIs constitute a special class of learning systems that are designed to aid humans, in contrast to early work on machine learning, which aimed to develop knowledge-based systems that would replace domain experts (Langley, 1999).

4.3. Components of Adaptivity

The architecture, behaviour, application and limitations of AUIs can generally be described as consisting of three components, namely, (Oppermann, 1994b):

1. Afferential (Section 4.4);
2. Inferential (Section 4.5); and
3. Efferential components (Section 4.6).

Figure 4.2 depicts an overview of these components by introducing some concepts that can be applied to any AUI and illustrating the general schema for the processing that occurs in AUIs. As seen in Figure 4.2, ovals represent input or output; rectangles represent processing methods; cylinders represent stored information; dotted arrows represent use of information and solid arrows represent production of results.

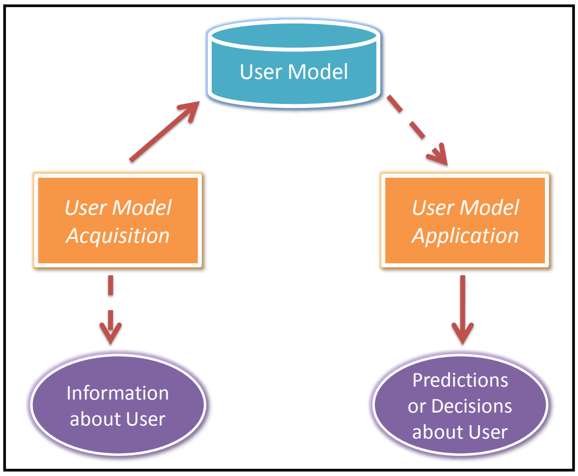


Figure 4.2: General Schema for Processing in an AUI (Jameson, 2002)

It was previously mentioned that an AUI makes use of some type of *information about the current individual user* (Figure 4.2). The information obtained about the user corresponds to the afferential component of adaptivity, discussed further in Section 4.4.

The process of *user model acquisition* (Figure 4.2), involves the system performing some type of learning and/or inference using the information about the user in order to arrive at some sort of user model. The process of *user model application* (Figure 4.2) involves the system applying the user model to the relevant features of the current situation in order to determine how to adapt its behaviour to the user (Jameson, 2002). The processes of *user model acquisition* and *user model application* correspond to the inferential component of adaptivity, discussed in Section 4.5.

The conclusion of the *user model application* process, results in *predictions or decisions about the user* (Figure 4.2) being displayed (Jameson, 2002). The display of the predictions and decisions made correspond to the efferential component of adaptivity, discussed further in Section 4.6.

4.4. Afferential Component of Adaptivity

An AUI can only personalise an interface if it contains some user-related data. The afferential component of adaptivity is responsible for acquiring information about individual users (Oppermann, 1994b). This information can either be implicitly and/ or explicitly obtained (Alvarez-Cortes *et al.*, 2007).

Explicitly acquired information involves information that the user (explicitly) supplies to the system. This information can be obtained from user self-reports on personal characteristics, proficiencies and interests; and on user evaluations of specific objectives. Self-reports on objective personal characteristics involve information about the user, such as their age and gender, facts that have the advantage of changing relatively infrequently. User evaluations of specific objectives involve the user providing system responses to certain items and the system evaluating them (Jameson, 2002). Explicitly acquired information does, however, place a heavy load on users.

Implicitly acquired information places a lesser burden on users and involves information that the system (implicitly) obtains from the user during the interaction sequence. The broadest category of implicit information is naturally occurring actions. These actions may range from mouse movements to more task-specific information. Naturally occurring actions may not require additional investment by the user. A limitation of acquiring information by using these naturally occurring actions is that they may be difficult to interpret; however, the UI can be designed in such a way that these actions are easy to interpret. Implicit information can also be obtained from previously stored information. Previously stored information involves all information stored for reasons unrelated to any adaptation, e.g., previously used web home pages. A limitation of using this previously stored information is that the usefulness of the information in context to the current application might be limited; however, the advantage is that the information can be applied right at the start of the interaction. Other sources of implicit information that should be briefly mentioned are sensing devices, such as microphones; and devices that receive explicit signals about the user's surroundings, such as GPS (Jameson, 2002).

The afferential component of adaptivity needs to store the acquired information. AUIs utilise models to store necessary information. The most important models appropriate for AUIs are the user, task and domain and the system models (Krogsaeter and Thomas, 1994). The following four sections will explore these models, in particularly the user model, to gain a better understanding.

4.4.1. The User Model

The importance of the user model has been mentioned and it has especially been noted in the definition of an AUI (Section 4.2). The user model lies at the heart of AUIs. The reason for this is that adaptivity requires the system to have a certain amount of knowledge about the user and the user model fills this void (Krogsaeter and Thomas, 1994). The terms "user profile" and "user model" have, however, often been used interchangeably.

Ghorbani and Zhang (2007) define a user profile as a collection of demographic (user name, sex, age, etc.) and usage information as well as user interests or goals achieved either explicitly or implicitly. User-profile data can describe individual users as well as groups of users (Ghorbani and Zhang, 2007).

Certain authors (Dieterich *et al.*, 1993, Domik and Gutkauf, 1994, Oppermann, 1994a, Krogsaeter and Thomas, 1994) define a user model as the collective information that a system has of the properties of an individual user (as well as a user group) in order to tailor the interaction or adapt the system and user's dialog. This definition of a user model appears to be similar to the definition of a user profile given by Ghorbani and Zhang (2007).

Ghorbani and Zhang (2007), however, define the user model as an abstract representation, which contains a collection of information and explicit assumptions about an individual user (as well as a user group) on relevant aspects of the user, which is needed in the adaptation processes. They distinguish the user model from a user profile by stating that the user model is a more abstract representation and that it contains explicit user assumptions. The user model in this context uses the various user information it contains to deduce the current goals and interests of the user. The user profile, therefore, is an instance of a user model for a particular user. The definitions of the user profile and user model, as defined by Ghorbani and Zhang (2007), will be used for the purpose of this research.

4.4.1.1. Information Needed

It was previously mentioned, that the user model represents the characteristics of the users with which the system interacts. There are a variety of attributes that can be maintained in the user model (Figure 4.3) and these attributes are dependent on the concerned application, their purpose and how they are used. The typical attributes are (Kules, 2000):

- User preferences, interests, attitudes and goals;
- Proficiencies (e.g., task-domain knowledge, proficiency with the system);
- Interaction history (e.g., interface features used, tasks performed/in progress, goals attempted/achieved, number of requests for help); and
- User classification (stereotype).

Detailed knowledge of various aspects of the editor
1.1 believes that killing the window is a good way to quit
1.2 knows how to quit safely and sometimes does
1.3 has been told twice about the benefits of the safe quit method
1.4 probably does not know how to make multiple windows on a file
1.5 has been told once how to make multiple windows on a file
Some general attributes of the user
2.1 writes C programs
2.2 dislikes using a mouse
2.3 seems to only want to know the minimum about **sam**
2.4 claims to be a sophisticated **user** of the text editor **vi**
2.5 fast typist
2.6 prefers terse explanations and descriptions
User's current goal
3.1 currently typing a set of additions to a large program
3.2 currently adding code to a function (exprn) in the file parse.c

Figure 4.3: Example of Parts of a User Model (Kay, 1993)

4.4.1.2. User Model Classification

User models are classified along the following four dimensions (Ross, 2000, Rich, 1998):

- *What is modelled?*

  This dimension specifies whether the user model is for a canonical or an individual user. Canonical-user models are built for one typical user, whereas an individual model, as its name revels, is built for an individual user and should be able to tailor its behaviour to a heterogeneous variety of users.
- *The source of modelling information*

  This dimension specifies whether the user model is constructed either explicitly or implicitly. Explicit user models are built by the user; whereas, implicit user models are built by the system. Acquiring information implicitly and explicitly has been mentioned in Section 4.3.1.

- *The time sensitivity of the model*

  This dimension specifies whether the user model is short or long term. Short-term user models contain highly specific information and focus on building the user model during a single session. Long-term user models contain more general information that changes more slowly over time, or rather over a whole series of sessions.

- *Update methods*

  Update methods are either static or dynamic and are usually concluded based on the first three dimensions. The most basic type of user model is one that remains the same, i.e., static, and is modelled for a canonical user. User models constructed by user behaviour for individual users over short periods of time are usually dynamically updated. Models that contain very short-term information can be task models as they are dependent on the task and not the individual user. Task models are further investigated in the next section.

4.4.2. The Task Model

A task model can be static or dynamic. A static task model represents tasks that the user can perform with the system. A dynamic task model represents tasks that the user performs at a particular time. Dynamic task models can be viewed as an aspect of user modelling as it is dependent on the user (Krogsaeter and Thomas, 1994). When a user's goal is recognised, the system can adapt accordingly, making it simpler for the user to accomplish his / her goals. The system can also offer to complete a user's task automatically, providing the user with task completion (Krogsaeter and Thomas, 1994).

4.4.3. The Domain Model

The domain model represents the domain outside the adaptive system. The domain and task models are not always clearly distinct such as when a user's tasks involve manipulating the modelled domain. The difference between the two is that the domain model describes the domain; whereas, the task model describes how to operate on the domain. Domain models are generally static in nature compared to user and task models (Krogsaeter and Thomas, 1994).

4.4.4. The System Model

The system model represents a computer system such as its architecture and UI. The system model used in AUIs will specialise into a dialogue or interaction model. This dialogue model describes how and when modification of the UI should occur. A dialogue model should be able to display system information to the user in a readable and understandable manner (Krogsaeter and Thomas, 1994).

Not all of the above-mentioned models are necessary for adapting the UI; however, they are the most common used. The most important model contained in the knowledge base is the user model and the next section further discusses the creation and maintenance of user models.

4.5. Inferential Component of Adaptivity

The inferential component of adaptivity uses user- data acquired to identify possible indicators for adaptation. It is the most important component of adaptivity, implying that a basis must be specified (theory, set of rules) for drawing inferences. Another implication of the inferential component is that the kind of data to be recorded (afferential component) and how the system should be adapted (efferential component) must be defined (Figure 4.4). The inferential component acts as the switch box of an AUI (Oppermann, 1994b). User modelling provides the role of the inferential component of adaptivity: a concept which will be further discussed in the next section.

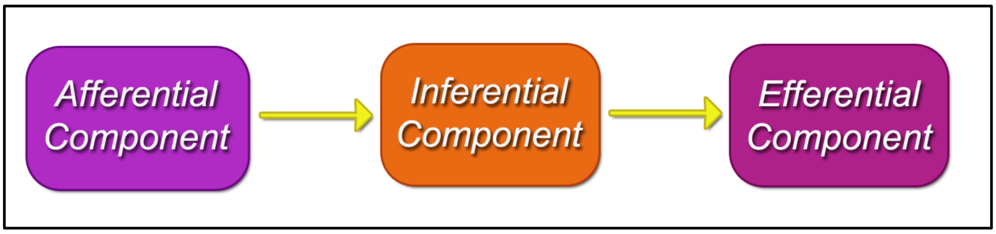


Figure 4.4: The Three Components of Adaptivity

4.5.1. User Modelling

The user's perspective of personalisation involves the presentation of an interface based on some data (Figure 4.5). The generation of this data, typically stored in a user model, is known as user modelling (Alvarez-Cortes *et al.*, 2007). User modelling is the whole process of constructing and maintaining user models, including creating, updating and deleting user profiles. It contains the functions which incrementally build up the user model, store, update and delete entries in instantiated user profiles and maintain the consistency of the model (Ghorbani and Zhang, 2007).

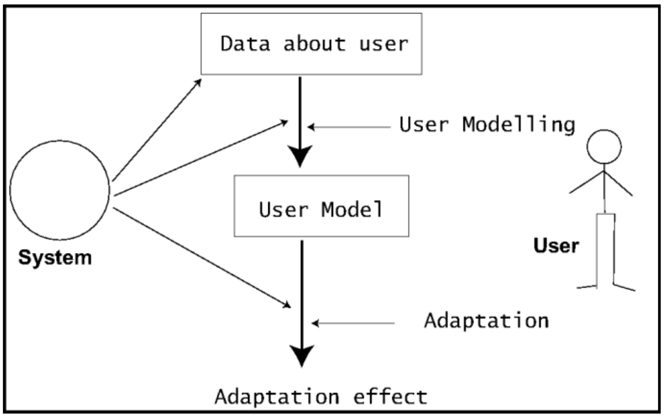


Figure 4.5: System and User Perspectives of a User Model (Alvarez-Cortes *et al.*, 2007)

Figure 4.5 indicates that user modelling uses data about the user in order for an adaptation to take effect; therefore, the inferential component of adaptivity is achievable. It is, however, difficult to map user characteristics represented in a user model to appropriate system responses and without this connection, the user's characteristic should not be modelled (Oppermann, 1994a).

User modelling corresponds to well-accepted UI design principles as it focuses on the user's needs and involves a detailed analysis of the task domain (Kules, 2000). According to literature, most user modelling approaches begin with either a default user model or obtained information from the user to generate a user model. After the initial user model has been constructed, explicit or implicit changes can occur to update it (Oppermann, 1994a, Ghorbani and Zhang, 2007).

4.5.2. User-Modelling Techniques

User-modelling techniques have been developed to represent various kinds of information and assumptions about the user in appropriate, formal representation schemes; for inferring additional assumptions about the user, based on initial hypothesis; for maintaining consistency in the user model, and for inferring conclusions about the user, based on the user's interaction history with the system (Kobsa, 1995).

This sub-section discusses one of the most commonly used user-modelling techniques, namely, stereotypes. The other commonly used user-modelling techniques follow this discussion.

4.5.2.1. Stereotypes

A popular approach to user modelling is stereotypes. Rich (1979) first proposed the use of stereotypes for a recommender system she built, called Grundy (Figure 4.6). Grundy acts as a librarian, recommending books to its users by asking questions about the user's likes and dislikes (Rich, 1998).

The stereotype approach has proven to be successful in situations where a quick assessment, which is not necessarily completely accurate, of the user's background knowledge is required (Kobsa, 1993). The designer categorises users into a single unit (user group), simplifying the design and processing load at run time. Systems that utilize stereotypes require the collection of characteristics, or facets, as well as a set of triggers. The characteristics/facets refer to any information concerning the user. The set of triggers are occurrences which signal whether a suitable prototype could be used or not (Kules, 2000).

FACET		*VALUE*	*RATING*
Activated-by		*Athletic-w-trig*	
Genl		*ANY-PERSON*	
Motivations			
	Excite	3	600
Interests			
	Sports	4	800
Thrill		5	700
Tolerate-violence		4	600
Romance		-5	500
Education		-2	500
Tolerate-suffering		4	600
Strengths			
	Physical-strength	4	900
	Perseverance	3	600
SPORTS-PERSON			

Figure 4.6: Sample Stereotype Used by Grundy (Rich, 1998)

Kobsa (1993) identifies three tasks that need to be fulfilled by the user-model developer:

1. *User subgroup identification*

   The user-model developer needs to identify various user subgroups wherein each subgroup contains users with the same set of characteristics.

2. *Identification of key characteristics*

   The user-model developer should identify the key characteristics which will classify the users into their respective subgroups. These key characteristics are referred to by Jameson (1999) as the body and by Rich (1979) as the traits of the stereotype. The computer system should be able to recognise whether these key characteristics are present or absent. This recognition done by the computer system is referred to as the trigger.

3. *Representation in (hierarchically ordered) stereotypes*

   The characteristics of the various user groups must be formalised in an appropriate representation system. The collection of all the key characteristics of a particular subgroup is called the stereotype of this subgroup. There could be situations in which the contents of one stereotype form a subset for the contents of another stereotype. When it comes to these situations, it is useful to construct stereotype hierarchies. Stereotype hierarchies allow the contents of the super-ordinate stereotype to be inherited from the subordinate stereotypes. An example of a stereotype hierarchy in the medical domain is displayed in Figure 4.7. The contents of the topmost stereotype (i.e., system users) are inherited from its subordinate stereotypes (e.g., hospital managers, scientists, medical professionals, etc.).

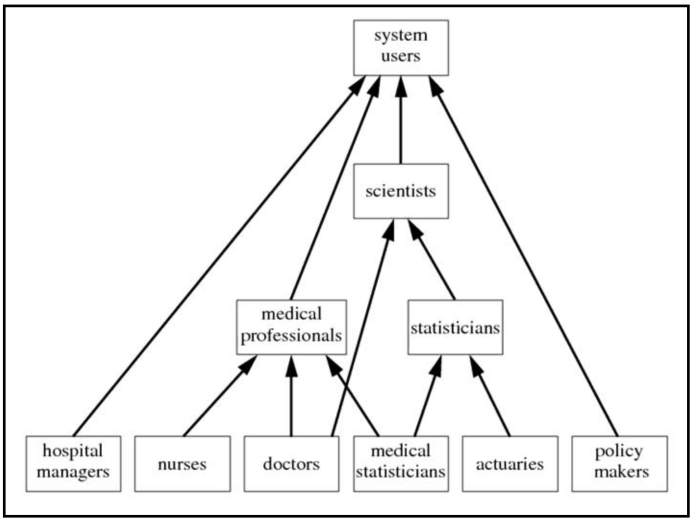


Figure 4.7: Example of Stereotype Hierarchy in a Medical Domain (Kobsa, 1993)

In her PHD thesis, Rich (1979) noted that stereotypes have the advantage of space-efficiency. This means that qualities that apply to multiple users need only to be stored once but can be used by all stereotype members when required. Researchers question the stereotype approach, based on the argument that there is no such thing as a stereotypical user, and users may find themselves being modelled on an inappropriate stereotype (Crow and Smith, 1993, Kobsa, 1993). A possible solution for this problem could be to allow the modification of assumptions within stereotypes for individual users (Oppermann, 1994a).

A variety of computational techniques can be used for the inference processes in stereotype-based systems. The emphasis, however, is less on the sophisticated computational approaches employed, and more on realistic specification of the content of the stereotypes and the rules activating them (Jameson, 2002).

4.5.2.2. Other User Modelling Techniques

The other commonly used user-modelling techniques, classified according to category, are:

- *Classification Learning*

  Classification learning involves using a set of training examples characterised in terms of its features, where each training example has been classified and, based on the examples, a classifier is built. Some learning methods which have been developed within this paradigm include decision trees, probabilistic classifiers, neural networks, case-based reasoning and specialised text-classification. The disadvantages of this approach are that it may not be that easy to categorise items in terms of features and there is possibility that the system might not be able to process an adequate number of training examples before it begins classifying items (Jameson, 2002).
- *Collaborative Filtering*

  Collaborative filtering involves predicting the user's future behaviour by using data from a like-minded group of users (Albrecht and Zukerman, 2001). Collaborative filtering differs from classification learning due to the fact that each item is characterised in terms of other user's responses and not in terms of its features (Jameson, 2002). This approach is used when it can be assumed that the user behaves similarly to other users (Albrecht and Zukerman, 2001). The disadvantage of this approach is that there might not be enough responses available by similar users and users also might not be willing to provide responses (Jameson, 2002).

- *Decision-Theoretic Methods*

  The above two user-modelling techniques are almost entirely data-based: they do not utilise general knowledge about users, their goals or the items they are dealing with. Decision-theoretic methods are more theory- based as the system designers incorporate knowledge about the relevant variables into their models. The advantage this approach has is that the system can make useful inferences about a user without first acquiring a long-term user model. The major challenge with this approach involves the construction of suitable general models, as they are almost always complex (Jameson, 2002).

- *Techniques for Plan Recognition*

  Plan recognition involves interpreting a user's actions as steps in the execution of a plan that is intended to achieve some goal. This is done so that assistance can be provided to the user. Plan recognition may assist the user by taking over routine actions. Help and tutoring systems could use plan recognition to identify problems with the user's plan or to remind the user of steps that need to be taken (Jameson, 2002).

All the approaches mentioned (Sections 4.5.2.1 and 4.5.2.2) have advantages and disadvantages but whatever approach is chosen, users need to be satisfied with the conclusions they deduct and the inferences they make. Regardless of how sophisticated the user-modelling techniques are, what ultimately matters is how well the larger "system" that includes the user works (Jameson, 2002). The advantages brought about by adaptation are minimal if expensive mechanisms only achieve slight improvements in usability and usefulness (Fischer, 2001).

4.6. Efferential Component of Adaptivity

AUIs require automatic modifications of a system's behaviour. The efferential component is responsible for handling the various types of adaptations, or rather specifying how the system should be adapted (Oppermann, 1994b). The adaptation can take place on four different levels (Reichenbacher, 2003, Jameson, 2002):

1. *Information Level*

The information level adapts the content of the information.

2. *Presentation Level*

   The presentation level adapts the visualisation of the information.

3. *User Interface Level*

   The UI level adapts the user interface to the user.

4. *Functionality Level*

   The functionality level adapts the system's functionality to the user.

These levels provide many benefits and support many forms of adaptation. The next sections discuss how they could be beneficial.

4.6.1. Benefits and Functions

The primary goal of an AUI is to provide the user with an easy-to-use interface whilst improving the efficiency and effectiveness of the user's interaction. AUIs aim to make complex systems more usable as well as providing users with an interface that they want to see (Dieterich *et al*., 1993). Other goals of AUIs aim to speed up and simplify interactions as well as improving the user's satisfaction (Dieterich *et al*., 1993, Karagiannidis *et al*., 1995). These goals can only be realised if the UI accommodates heterogeneous user groups and considers the increasing experience of the user (Dieterich *et al*., 1993). An important goal of interface adaptation, which should also be mentioned, is its ability to take into account special perceptual or physical impairments of individual users so as to allow using a system more efficiently, with minimal errors and frustration (Jameson, 2002).

AUIs have been used in a variety of applications and domains to support a range of activities. Jameson (2002) defined a list of forms of adaptations, outlining the benefits of AUIs in different domains supporting different tasks. These forms of adaptations, presented by Jameson (2002), satisfy four main functions, namely (Jameson, 2002):

1. Supporting system usage;
2. Supporting information acquisition or decision making;
3. Supporting learning; and
4. Supporting collaboration.

The forms of adaptations supporting the above mentioned functions of adaptation will now be further discussed.

4.6.1.1. Supporting System Usage

AUIs could be helpful in supporting a user's efforts to operate a system successfully and efficiently. The forms of adaptation, as seen in Figure 4.8, which support this function are (Jameson, 2002):

- Helping with routine tasks;
- Adapting the interface; and
- Giving advice about system use.

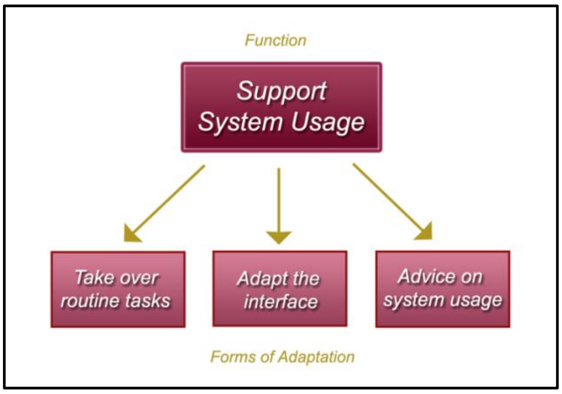


Figure 4.8: Support System Usage Function and Forms of Adaptation

Help with routine tasks

AUIs could support frequently occurring (routine) interactive tasks that place heavy demands on the user's time, though, typically, not on their intelligence or knowledge. This could save the user time and effort. Examples of routine tasks include email management and appointment scheduling (Jameson, 2002).

An AUI which helps with routine tasks is Lookout. Lookout was initially a prototype which was later employed by Microsoft. It monitors a user's interactions with the Microsoft Outlook messaging and calendar systems and decides whether, when, and how to best assist users with the tasks involving their calendar and scheduling appointments. An example of Lookout in action can be seen in Figure 4.9, where Lookout could automate the task of scheduling based on a background analysis of the e-mail being reviewed (Horvitz, 1999).

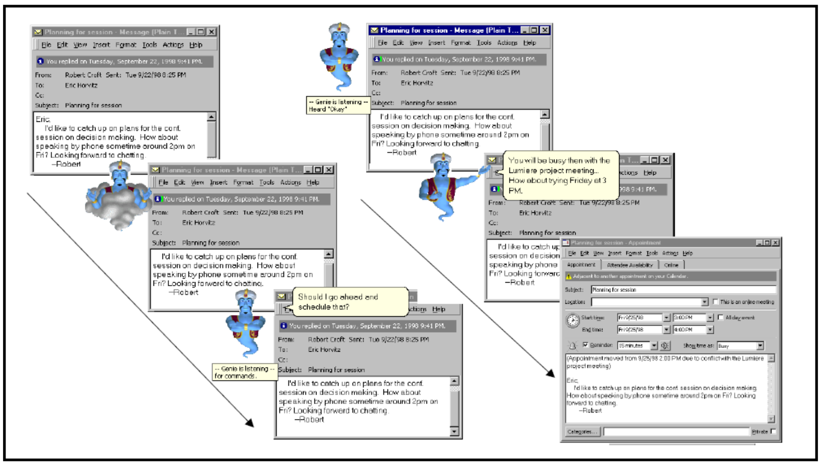


Figure 4.9: Lookout in Action (Horvitz, 1999)

Adapt the interface

AUIs can adapt the visual interface such as its menus, icons, and the system's processing of signals from input devices such as keyboards (Jameson, 2002). Previous AUI research has mostly focused on adapting menus (Findlater and McGrenere, 2004, Gajos *et al.*, 2008, Findlater and McGrenere, 2008, Gajos *et al.*, 2006, Tsandilas and schraefel, 2005). Early research done by Greenberg and Witten (1985) showed that an adaptive menu structure, which displayed the most frequently used items first, was faster than a static structure. Contrastingly, Mitchell and Schneiderman (1989) compared static to adaptive menus that reordered during usage based on frequency, and found that users performed faster using static menus and also preferred the static to the adaptive menus. Menu-based adaptive interfaces have, however, been the subject of much research since and many improvements have been made (Findlater and McGrenere, 2004, Gajos *et al.*, 2008, Findlater and McGrenere, 2008, Gajos *et al.*, 2006, Tsandilas and schraefel, 2005).

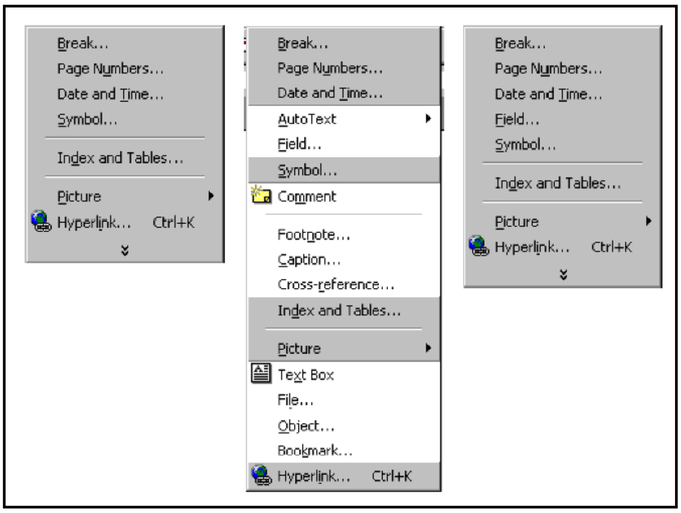


Figure 4.10: Screenshot of a Smart Menu (Jameson, 2002)

The Smart Menu feature was introduced by Microsoft in 2000. An infrequently used menu option is initially hidden from view and it only appears in the main part of a menu after the user has selected it for the first time. It will be removed later if the user does not select it often enough. The main idea is that eventually, the menus should only contain the items that the user accesses regularly so that less time is spent searching within menus (Jameson, 2002). As seen in Figure 4.10, the user accesses the “Insert” menu within Microsoft Word and not finding the desired option, clicks on the extension arrows and selects the “Field” option. When the user later accesses the same menu, “Field” now appears in the main section.

A popular algorithm employed in adaptive menus is the Base adaptive algorithm (Figure 4.11). The Base adaptive algorithm, as used by Findlater and McGrenere (2008), dynamically changes the selection list based on the user’s previous list selections made. This algorithm incorporates both recently and frequently used items and is commonly used in Microsoft Office 2003’s adaptive menus.

1. set top section to the *most recently* selected item and the *two most frequently* selected items (as pre-calculated from the selection stream)
2. if there is overlap among these three slots or if this is the first selection in the stream (i.e., no recently selected item exists)

   then the third most frequently selected item is included so that 3 unique items appear in the top
3. order top items in the same relative order as they appear in the bottom section of the menu

Figure 4.11: Base Adaptive Algorithm (Findlater and McGrenere, 2008)

Provide advice about system use

An AUI could provide assistance by adaptively offering the user information and advice about how to use the application (Jameson, 2002). A well-known example of this is the Lumiere project (Ehlert, 2003), which uses information about the user’s goals and needs, observations about the current program state, a representation of sequences of actions over time, and the words in a user’s query and then generates a probability distribution over areas that the user might need assistance with, as well as the likelihood of the user requiring assistance (Ehlert, 2003). This project has provided the basis for the office assistant in MS Office 97 (Figure 4.12).

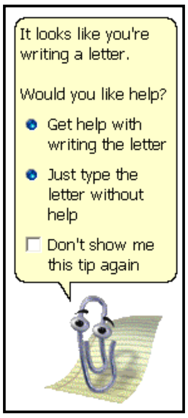


Figure 4.12: Ms Office Assistant (Nicknamed Clippy)

4.6.1.2. Supporting Information Acquisition

Users are often overwhelmed by the wealth of information obtained from the vast number of electronic documents. AUIs could be helpful in assisting users with finding information they need, and presenting this information to them in a suitable format. The forms of adaptation, as seen in Figure 4.13, which support this function are (Jameson, 2002):

- Helping users find information;
- Tailoring information presentation; and
- Recommending products.

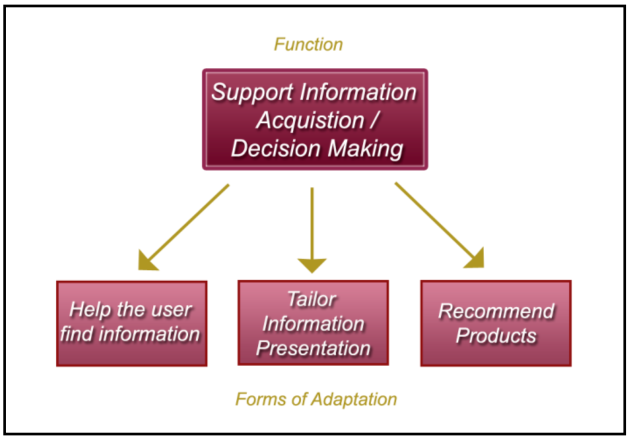


Figure 4.13: Support Information Acquisition Function and Forms of Adaptation

Assist the user to find information

AUIs could assist the user with finding relevant information. AUIs within this category typically utilise existing techniques for analysing information (usually textual) in the field of information filtering; however, they employ user models to provide the personalisation when filtering information. There are forms of adaptive support for helping users find information, namely (Jameson, 2002):

- Support for browsing;
- Support for query-based search or filtering; and
- Spontaneous provision of information.

An example of such an AUI is the Adaptive News Server (Figure 4.14), which delivers news stories to small, portable computing devices such as mobile phones and personal digital assistants. In Figure 4.14, A represents an overview screen consisting of three stories, where the first two involve American football and the last involves horse racing. Once the user has selected the third story (horse racing), the screen depicted in B appears. The overview screen, C, now consists of mostly horse racing stories (first two stories), as the system inferred that the user has an interest in horse racing (Jameson, 2002).

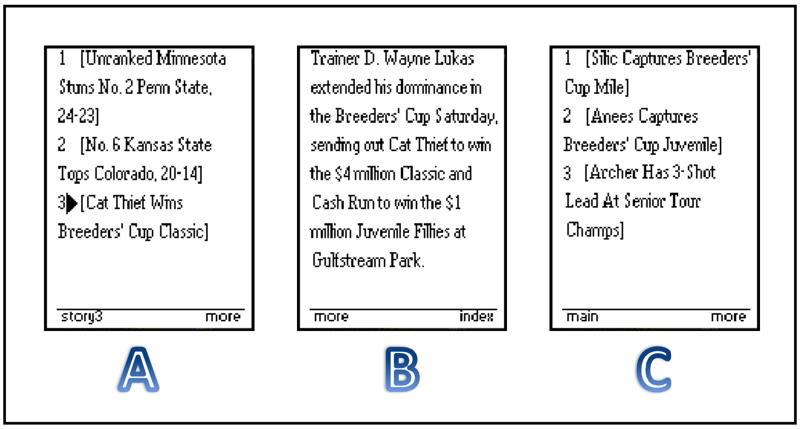


Figure 4.14: Sequence of Three Screens Presented by Adaptive News Server (Jameson, 2002)

Tailor information presentation

Even when relevant information can be presented to the user, AUIs could present this information in a form most suitable to the specific user and could consider a variety of user properties when tailoring documents, namely (Jameson, 2002):

- The user's degree of interest in particular topics;
- The user's knowledge about particular concepts or topics;
- The user's preference or need for particular forms of information presentation; and
- The display capabilities of the user's computing device.

An example of an AUI which tailors information presentation is the Adaptive Diagnostics and Personalised Technical Support (ADAPTS) system (Brusilovsky and Cooper, 2002). ADAPTS provides an intelligent, adaptive Electronic Performance Support System (EPSS) that integrates an adaptive diagnostics engine with adaptive access to technical information. ADAPTS adjusts the diagnostic strategy to each individual technician, thus dynamically adapting all sequences of setups, tests, and repair or replace procedures on the basis of the technician's response. Depending on the technician's responses, new activities might be planned. Information is assembled on the fly and dynamically selected. Technical support information is displayed to the technician. The ADAPTS system consists of two main processes: adaptive diagnostics and adaptive interaction. C in Figure 4.15 shows how the adaptive diagnostics' process uses a diagnostic engine, the main function of which is to select the most relevant task for the user to perform. The adaptive interaction process is maintained by the adaptive hypermedia interface shown in A and B. The adaptive hypermedia interface consists of two windows: the outline frame, A, and the content presentation frame, B.

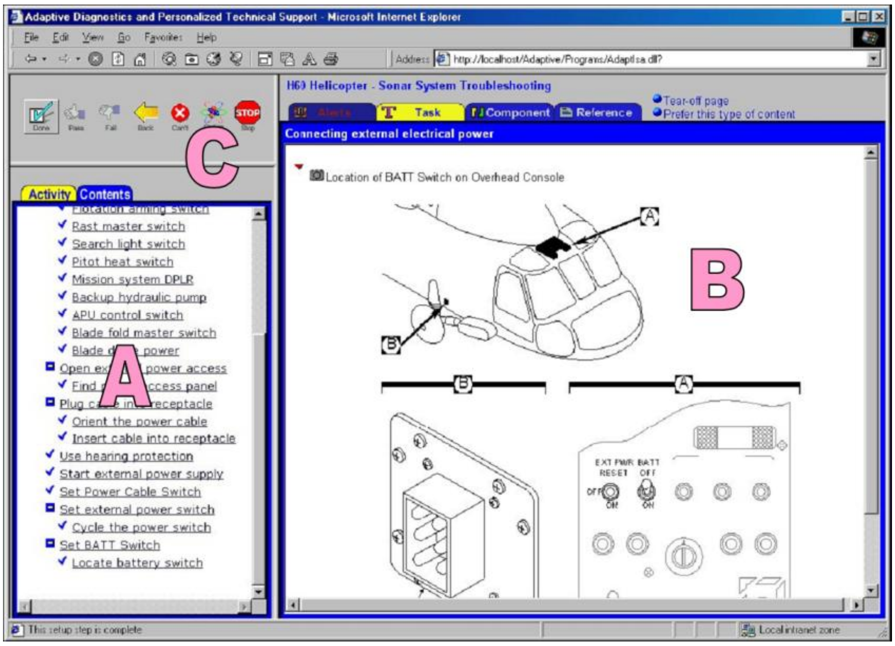


Figure 4.15: The ADAPTS Interface: (a) Adaptive Outline Frame; (b) Adaptive Content Presentation Frame; (c) An Applet for Communication with Adaptive Diagnostic Engine (Brusilovsky and Cooper, 2002)

Recommend products

Recommending products is the most practically important category of AUIs, employed by many commercial web sites (Langley, 1999). AUIs can be beneficial to e-commerce sites enabling the customisation of the sales interaction and suggesting suitable products. Amazon.com is an example of a commercial web site that recommends products frequently purchased by customers who purchased a selected product (Figure 4.16).

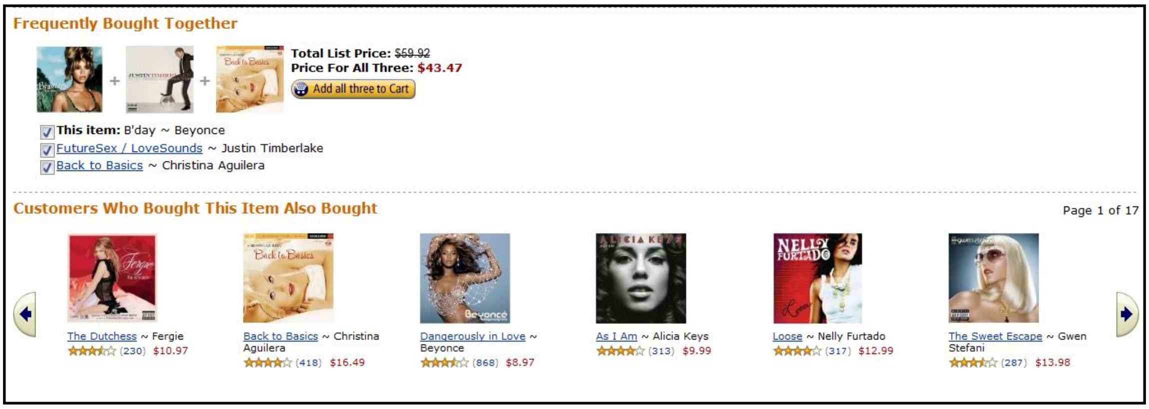


Figure 4.16: Amazon.com *Customers Who Bought* Section

Pazzani *et al*. (1996) proposed an AUI that acts as a recommender system, recommending web pages that interest the user. This system is called Syskill & Webbert and acts like a search engine in locating web pages according to user input. The difference with this system is that it also highlights what pages the user will or will not be interested in. This system also enables the user to indicate his/her desired web pages (Pazzani *et al*., 1996).

4.6.1.3. Supporting Learning

AUIs could support learning by incorporating personalisation in computer-based tutoring systems and learning environments. The forms of adaptations previously mentioned (Sections 4.6.1.1 and 4.6.1.2) are relevant to this function as well (Figure 4.17). The aspects of the AUI which could be adapted include (Jameson, 2002):

- Taking over routine tasks which are not crucial for learning;
- Adapting the interface to facilitate learning;
- Providing adaptive help on both the interface usage as well as the learning tasks;
- Helping the user find information;
- Recommending learning material (e.g., lessons and exercises);
- Tailoring content and/or the presentation of learning material; and
- The selection and the form of the instructional information presented.

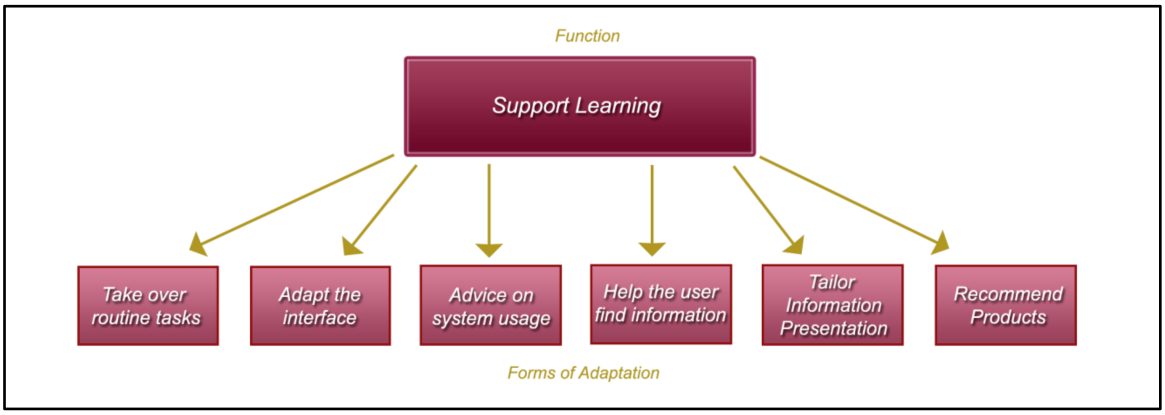


Figure 4.17: Support Learning Function and its Forms of Adaptation

Oka and Nagata (1999) developed menus that support learning by shifting a novice user to an expert user. They proposed redundant menus which are easy to use by novice users. As the novice uses the redundant menus, the redundancy gradually get reduced until eventually the redundant menus become the current menus of Microsoft Windows (Figure 4.18). The shortcut operations are presented to the novice users in an attempt to aid them to become expert users. After observing the operations of a particular user, the system also presents better operations such as (Oka and Nagata, 1999):

- If the user inputs many spaces in the head of a line, the command of the indentation will be suggested; and
- If the user uses menus and icons for decorating characters, a dialog for characters will be suggested.

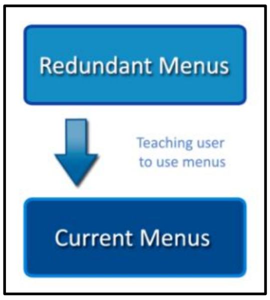


Figure 4.18: From Redundant Menus to Current Menus (Oka and Nagata, 1999)

4.6.1.4. Supporting Collaboration

AUIs could use a number of user's user models to facilitate collaboration by incorporating the ways in which users match or complement each other. Similarly to the AUI function, *Support Learning* (Section 4.6.1.3), the forms of adaptations previously mentioned (Sections 4.6.1.1 and 4.6.1.2) are relevant to this AUI function as well (Figure

4.19). The aspects of the AUI which could be adapted to support collaboration include (Jameson, 2002):

- Adapting the interface to facilitate collaboration;
- Recommending suitable collaborators (Figure 4.16); and
- Tailoring content and/or the presentation to facilitate collaboration.

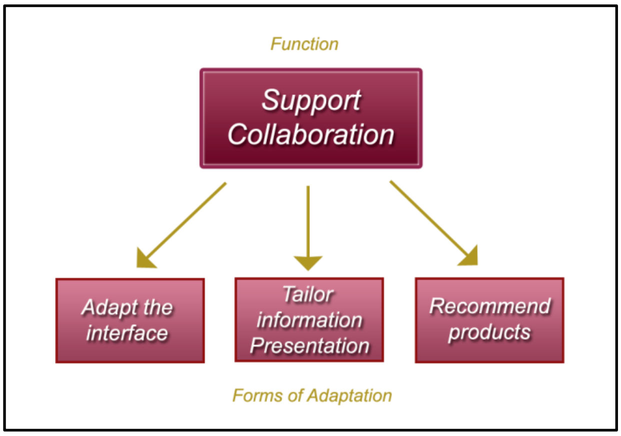


Figure 4.19: Support Learning Function and its Forms of Adaptation

4.7. Stages of the Adaptive Process

Dieterich (1993) proposes four stages of the adaptive process from the user's perspective. These stages are controlled by the system designer, administrator, a local expert, the user or the system itself. The system designer, administrator and local expert will only cater for a group and individual adaptation can only be achieved if the stage of the adaptive process is controlled by either the user or the system. The four stages are:

- *Initiative*

  At this stage, either the system or the user suggests an adaptation. This stage is the first stage in the adaptive process.
- *Proposal*

  At this stage, possible changes or alternatives are recommended by either the system or the user.
- *Decision*

  This stage involves choosing the most suitable proposals put forth in the previous stage.
- *Execution*

  This stage involves executing the chosen adaptation.

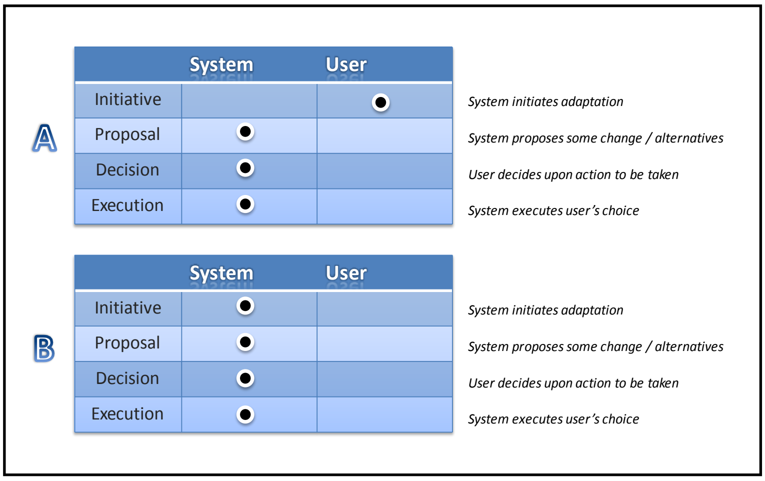


Figure 4.20: AUI Configurations (Dieterich *et al*., 1993)

In Figure 4.20, u*ser-initiated self-adaptation*, (A), and *Self-adaptation*, (B), are considered AUIs, as other configurations are related to adaptable UIs and not AUIs. Researchers suggest that a system should not, however, be limited to one type of adaptation but would be more beneficial if it employed various types of adaptations (Dieterich *et al*., 1993). A combination of adaptations for both adaptive and adaptable systems would, therefore, be more beneficial to AUIs.

4.7.1. Timing Strategy of Adaptation

In order to execute the chosen adaptation, the timing strategy of adaptation needs to be decided. The timing strategy refers to when the adaptation should take place. There are three timing strategies, namely (Dieterich *et al*., 1993):

1. Adaptation before the first session;
2. Adaptation during use; and
3. Adaptation between two sessions.

Adaptation *before the first session* involves adapting the UI before the user first uses the application. These timing strategies usually involve classification of the user according to a pre-test. Adapting only before the user initially uses the application appears too simplistic and is not sufficient as the user's needs might change during interaction. Adaptation *during use* involves adaptation that takes place continuously as the user interacts with the application. This timing strategy is best to accommodate the user's changing needs but can, however, bring about confusion. Adaptation *between sessions* facilitates complex adaptation strategies, due to the adaptation only taking place at the end of a session. If the user has not used the application for a while, this timing strategy might become inconsequential (Dieterich *et al*., 1993).

4.8. Challenges

AUI have received criticism as adaptation and automatic assistance contradict the principles of direct-manipulation interfaces. Some of the typical properties of adaptive interfaces can lead to usability problems that may or may not outweigh the benefits of adaptation to the individual user. These usability problems have been mainly noted by researchers in the community of HCI. A more detailed overview of usability challenges can be found in Jameson (2002). The usability goals affected by AUIs are (Jameson, 2002, Hook, 2000):

- Predictability and transparency;
- Controllability;
- Unobtrusiveness;
- Privacy; and
- Breadth of experience.

Predictability refers to the degree to which the user can predict the consequences of their actions. Transparency refers to the degree to which the user understands system events or the operations of the system. There are several different reasons why AUIs could make systems unpredictable and non transparent: AUIs have a complexity of inference and decision processes; some AUIs have an anthropomorphic appearance; and there is an incompleteness of relevant information for adaptation. Some ways in which this usability challenge can be overcome is to allow the user to inspect the user model and to explain system actions to the user (Jameson, 2002, Hook, 2000).

Controllability refers to the extent to which the user has control of any system events. There are several different reasons why AUIs provide a lack of control to the user: AUIs require an implicit acquisition of information about the user by the system, and control is also taken away from the user because AUIs take over the work from the user. Some strategies to ensure the usability goal of controllability is met involve submitting actions to the user for approval and allowing the user to set parameters that control the system's behaviour (Hook, 2000).

Obtrusiveness refers to the extent to which the user is disturbed by the system while they are performing a task. There are several different reasons why AUIs could make systems obtrusive: some AUIs have an anthropomorphic appearance and take over the work from the user. The usability goal of unobtrusiveness can be met by adapting the timing of messages to the user's activities and context (Jameson, 2002).

The concept of AUIs involves personalising the UI based on some user information stored in a user model. The user might, therefore, have privacy concerns that their data might be inappropriately used. There are several different reasons why AUIs could cause privacy concerns due to the visibility of adaptation in AUIs and because AUIs require an implicit acquisition of information about the user by the system. Some strategies to ensure the usability goal of privacy is met involve allowing users to control the visibility of adaptations, using data acquisition methods that support privacy and applying general privacy-protection measures (Jameson, 2002, Hook, 2000).

AUIs that help users with information acquisition involve the system doing most of the work of examining various information sources. This results in the user learning much less about the domain, than he/she would in an environment with no adaptivity. AUIs narrow the user's experience. There are several different reasons why AUIs could affect the user's breadth of experience. These are due to the system taking over the work from the user and an incompleteness of relevant information for adaptation. Some strategies to ensure the usability goal of breadth of experience is met involve explaining the system's actions and intentionally introducing diversity (Jameson, 2002).

Implementation of AUIs comes at a cost, both at the early design stages as well as in coding. The cost has, however, to be justified for the adaptation to be worthwhile, and if adaptation significantly improves usability and the quality of interaction, it can then be deduced that the adaptation cost has, in fact, been justified. The cost of adaptation can be minimalised if the adaptation arises as a natural consequence of better attention and metrics being applied to an interactive system design (Benyon and Murray, 1993).

In contrast to the challenges brought about by adaptation, it must be noted that AUIs serve many usability guidelines better than static or rather non-adaptive interfaces. These benefits can be seen in Section 4.6.

4.9. Conclusion

The purpose of this chapter was to decide how AUIs can be applied to the domain of CCs in order to assist novice CCAs becoming expert CCAs with regards to using the CC UI. AUIs provides personalisation by incorporating a user model which uses a user's past experiences to decide how to adapt the UI (Section 4.2). This chapter has highlighted three components of adaptivity, namely: afferential, efferential and inferential components of adaptivity (Section 4.3), and these should be included when designing an AUI for CCs.

The afferential component of adaptivity (Section 4.4) collects information from the user, either explicitly or implicitly. It stores this user information in a user model (Sections 4.4.1 and 4.4.1.1). The user model is classified according to four main categories (Section (4.4.1.2). Other models which this component utilises are task, system and domain models. CCAs perform repetitive tasks to achieve their goals. Incorporating an AUI into a CC domain therefore requires both a user and task model to model CCAs' behaviour. Implicit information places a lesser burden on users and therefore the information needed to build a user model for a CCA should be implicitly obtained, such as information about the CCA's interaction history.

The inferential component of adaptivity (Section 4.5) uses information stored in the user model to make possible inferences. User modelling (Section 4.5.1) involves the construction and maintenance of the user model and consists of many techniques (Section 4.5.2), the most popular being the stereotype approach (Section 4.5.2.1). Whatever user-modelling technique is employed, the main goal is that the user has to be satisfied with the inference decision made. Assisting novice and expert CCAs requires the system to have a collection of characteristics of each user group and therefore a stereotype-user modelling technique would be ideal.

The efferential component of adaptivity (Section 4.6) decides how to display information to the user. Besides user satisfaction, the main goal of AUIs is that they should provide the user with an easy-to-use interface which improves their efficiency and effectiveness (Section 4.6.1). The main functions supported by AUIs are as supports for system usage, information acquisition or decision making, learning, and collaboration. These functions can be realised through a variety of forms of adaptations (Sections 4.6.1.1 to 4.6.1.4). The functions supported by assisting novice CCAs are the support for system usage and the support for learning. The support for system usage will assist CCAs with operating a complex CC system and the support for learning will assist novice CCAs becoming expert CCAs by incorporating personalisation.

The various stages involved with AUIs are initiative, proposal, decision and execution, and these could be performed by either the system or the user (Section 4.7). AUIs consist of the configuration whereby the system either performs all stages or the system performs all but the first stage (initiative). Once the configuration is known, the timing strategy of adaptations must be decided, i.e., when should adaptation take place (Section 4.7.1). Adaptation during use and between sessions are most appropriate to support novice and expert CCAs because CCAs are escalated from novice to experts within a matter of weeks and therefore their skill level can change dramatically from when they first use the CC application.

AUIs pose many challenges (Section 4.8) that can be overcome. The usability problems affected by AUIs can, however, be justified if the benefits of adapting the interface significantly improves usability and the quality of the user interaction. This research proposes that the significant benefits posed by AUIs within the CC domain will outweigh any challenges faced.

The next chapter will compare several AUI models to determine whether any of these models could be specialised to the domain of CCs.

Chapter 5: Adaptive User Interface Models

5.1. Introduction

Contact centres (CCs) and adaptive user interfaces (AUIs) have been discussed so far. CCs face many challenges and it was deduced that a viable approach to assist with these would be the design of good computer user interfaces (UIs). These UIs would, however, - have to cater to the diversity of contact centre agents' (CCA)' skill levels. AUIs offer various levels of support and, in particular, have the ability to cater for these variations.

Model-based approaches are useful and powerful tools to develop UIs. AUI models, on the other hand, are incomplete and currently there is no AUI model specifically designed for CCs. The purpose of this chapter is to discuss a proposed AUI model for CCs. The discussion of the proposed AUI model commences after reviewing existing ones (Section 5.2). An AUI model for CCs is proposed to facilitate the design and implementation of an AUI for CCs (Section 5.3).

5.2. Existing Models

A model can be defined as the representation of a system which includes details of its structure and processes. A model hides unnecessary detail so as to put more emphasis on the essential components (Mohagheghi and Aagedal, 2007). There are certain elements which need to be present when designing a model for the purposes of system design and implementation. These are (Pressman, 2005):

- Data design;
- Architectural design;
- Component-level design; and
- Interface design.

Data design transforms the data model into the data structures that will be required to implement the system. Architectural design identifies the key components within the systems as well as the relationships between these modules. Component-level design specifies the architecture's components into more detail, resulting in a procedural specification. Interface design is concerned with the design of interfaces in-between modules within a system (Pressman, 2005).

Model-based approaches are useful and powerful tools to develop UIs (Lopez-Jaquero *et al*., 2003). The purpose of the section is to review existing AUI models to determine which model is most appropriate for the domain of CCs. Before an investigation of existing AUI models resumes, a discussion of the model proposed by Singh (2007) needs to be undertaken as it was found to be suitable for CCs.

5.2.1. An IUI Model for Contact Centre Operations

Singh (2007) conducted a thorough investigation into existing IUI models in order to select the most suitable one (Singh, 2007). These investigations lead to the proposal of an IUI model which consists of three major elements (Figure 5.1):

1. Architectural;
2. Component-level; and
3. Interface element.

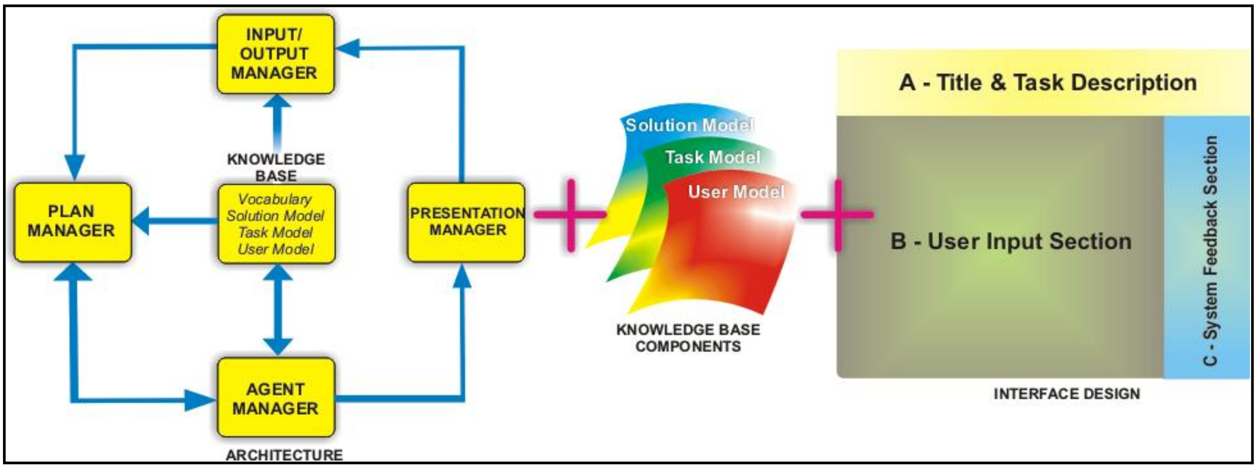


Figure 5.1: Proposed IUI Model for CCs (Singh, 2007)

The architecture of the Singh's (2007) model (*Architecture* in Figure 5.1) was adapted from Tyler *et al* (1991). The main components of the architecture are:

1. *Input / Output Manager*

   The main purpose of this component is to provide the user with a multimodal means of input and output. Translated input is sent to the Plan Manager. The output component receives high level commands from the Presentation Manager which is then translated into low level commands before being presented to the user (Tyler *et al*., 1991).

2. *Knowledge Base*

   An IUI acquires its intelligence by employing various intelligent techniques and knowledge. The Knowledge Base is a repository that contains application knowledge, domain-based knowledge, communication knowledge and instructional knowledge as well as knowledge about the user. The Knowledge Base, therefore, is a key component of IUIs as it allows the interface to make decisions, using intelligent techniques, in order to determine how to adapt the UI and deliver information to meet the user needs (Tyler *et al*., 1991). Singh (2007) specialised the model for CCs by incorporating CC knowledge within the Knowledge Base (Singh, 2007).

3. *Plan Manager*

   The main purpose of this component is to assist the user in achieving high-level goals by using knowledge of the users' current goals and plans. The Plan Manager can provide error detection and correction and interpret ambiguous requests as well as having the ability to help users map high-level goals into low-level application commands. The Plan Manager receives low-level commands from the Input/Output Manager and compares these with constraints on the task parameter values in order to detect global errors (Tyler *et al*., 1991). Singh (2007) improved Tyler *et al*.'s Plan Manager by using a Task Model to infer user goals from the user's low-level commands (Singh, 2007).

4. *Agent Manager*

   The Agent Manager, known as the Adaptor in Tyler *et al's* architecture, receives interface events from the Plan Manager and then consults the various models within the Knowledge Base. The Agent Manager then either updates the Knowledge Base or obtains the required information from it. The Agent Manager can keep track of various user performance data due to its ability to interact with the user model. Other tasks which the Agent Manager is responsible for include error checking and the delivery of required information for the automation of steps (Singh, 2007).

5. *Presentation Manager*

   The main purpose of this component is to determine the most suitable modality and modality techniques to display to the user based on information obtained from the Agent Manager (Tyler *et al*., 1991). This model plays an important role in how the UI could be adapted to a user. Besides deciding how a set or results or an interface could be displayed to a user, it caters for the customization of output by users based on their specifications (Singh, 2007).

The Component-level design of Singh's (2007) model described the structures of the models residing within the Knowledge Base. The Task, User and Solution models reside within the Knowledge Base (*Knowledge Base Components* in Figure 5.1).

The Interface element consisted of a design of a low-fidelity IUI template (*Interface Design* in Figure 5.1), which included sections for providing task-based information, user input and intelligent feedback in the form of system feedback.

Singh (2007) constructed a prototype as proof-of-concept to demonstrate its effectiveness and, through the use of a model evaluation, successfully showed that the proposed model could be used to develop IUIs for CCs. Only three of the five components of the model's architecture were implemented, namely the Plan Manager, Agent Manager and Knowledge Base.

5.2.2. Existing AUI Models

Certain core components should be included when designing an AUI for CCs. Chapter 4 revealed these core AUI components as being:

- Afferential (Section 4.4);
- Inferential (Section 4.5); and
- Efferential (Section 4.6) components of adaptivity.

In addition to these core components, a fundamental component of an AUI is the repository which holds the User Model and any other models necessary for adaptation. This repository is known as the Knowledge Base. The three components of adaptivity mentioned above, as well as the Knowledge Base, were used as a basis to investigate various AUI models. There are currently various other AUI architectures/models proposed by researchers but the following four models were selected as the most appropriate to satisfy the criteria of AUIs.

5.2.2.1. High-level Model of Adaptation

Paramythis *et al.* (2001) proposed the High-level Model of Adaptation which consists of various stages and components of an AUI (Figure 5.2). The components of this model and how they relate to the criteria will now be discussed.

The Interaction Monitoring module captures various interaction data when the user interacts with the UI. The Interpretation/Inferences module refers to parts of the AUI that interpret information and make inferences in order to update the system (e.g. User Model). Explicitly Provided Knowledge refers to information obtained explicitly. The Interpretation/Inferences and Explicitly Provided Knowledge components relate directly to Opperman's (1994a) afferential component of adaptivity.

Modelling refers to explicit or implicit representations of the users, their plans, tasks that could be performed, etc. Adaptation Decision Making refers to making decisions of what can be adapted, how it can be adapted and why it is adapted. This component uses information obtained to make the adaptation decisions. Modelling and Adaptation Decision Making components relate to Opperman's (1994a) inferential component of adaptivity.

The Applying Adaptations module refers to actually applying the adaptations. This component, in theory, serves as a subset of the Adaptation Decision Making component but it may be varied independently of the decision making process, e.g., taking into account the various adaptation strategies. This component relates to Opperman's (1994a) efferential component of adaptivity.

The Transparent Models & Adaptation "Rationale" component enables users to review models or the "rationale" that underlies the adaptation decisions. The Automatic Adaptation Assessment component refers to the run-time assessment of the adaptation effects by evaluating their success. The proposed High-level Model of Adaptation, however, does not explicitly define the Knowledge Base (Paramythis *et al*., 2001).

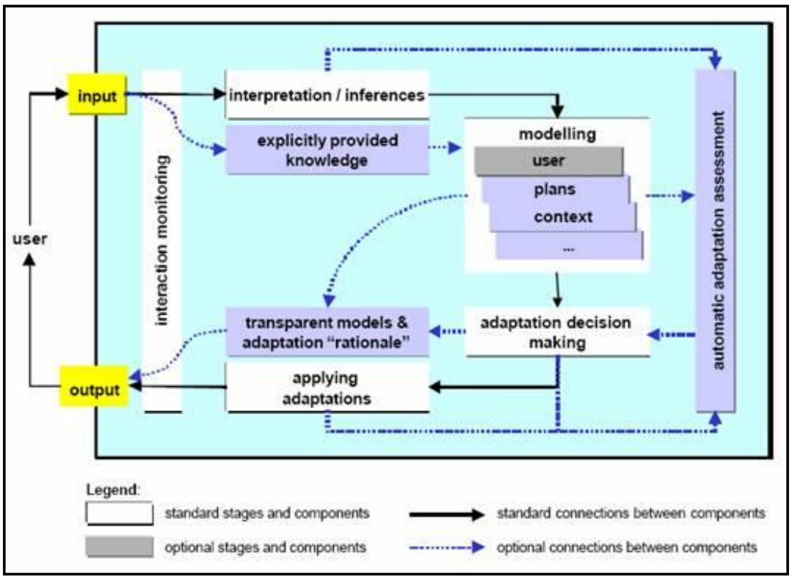


Figure 5.2: High-level Model of Adaptation in AUIs (Paramythis *et al.*, 2001)

5.2.2.2. UbiquiTO

Cena *et al*. (2006) proposed the architecture of UbiquiTO, an agent-based system that acts as an expert tourist guide for mobile users (Figure 5.3). The components of this architecture and how they relate to the criteria will now be discussed.

An explicit Knowledge Base is defined and is represented by the User Model Module in order to adapt the UI. The Recommender and Presentation Adapter components are responsible for personalisation. The Recommender makes use of personalisation to adapt the content whereas the Presentation Adaptor makes use of adaptation rules to adapt the presentation to the user preferences, the device characteristics and to the context. The Recommender and Presentation Adapter components correspond to the efferential component of adaptivity. The Watcher collects information implicitly or explicitly and thus corresponds to the afferential component of adaptivity. The inferential component of adaptivity can be made up from rules found in the user model, but the inferential component is not explicitly defined (Cena *et al*., 2006).

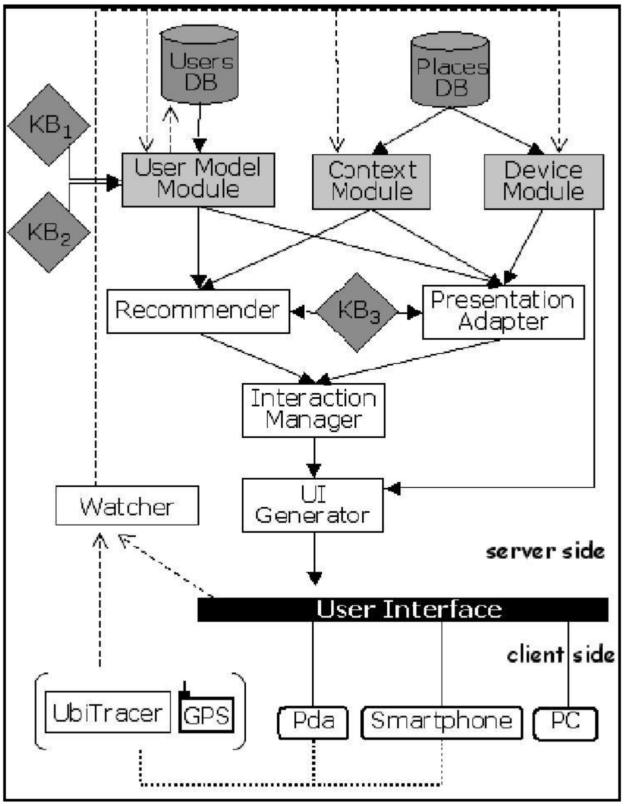


Figure 5.3: The Architecture of UbiquiTO (Cena *et al*., 2006)

5.2.2.3. eHealth Systems

Pechenizkiy *et al*. (2005) proposed a framework for an AUI for eHealth systems (Figure 5.4). The components of this model and how they relate to the criteria will now be discussed. As seen in Figure 5.4 the arrows emphasise information flows crucial to the AUI process. There are three major groups of framework components: Participants, Data Repositories and Different Engines.

The Participants are the various users of the system. The Data Repository is a repository of all information needed. The Adaptation Engine serves as the core component for adaptation and consists of the Knowledge Base, a Model (user, task, and environment) Generator and an Adaptation Effect provider. The inferential component of adaptivity can be made up from rules found in the User Model but it is not explicitly defined. The Adaptation Effect provides various kinds of adaptations such as adaptation to content, to presentation and to navigation. This component corresponds to the efferential component of adaptivity. The Model Generator generates the user /task / environment model by either implicitly or explicitly acquiring user information. The acquiring of information evident in this component, therefore, corresponds to the afferential component of adaptivity but the afferential component of adaptivity is also not explicitly defined (Pechenizkiy *et al*., 2005).

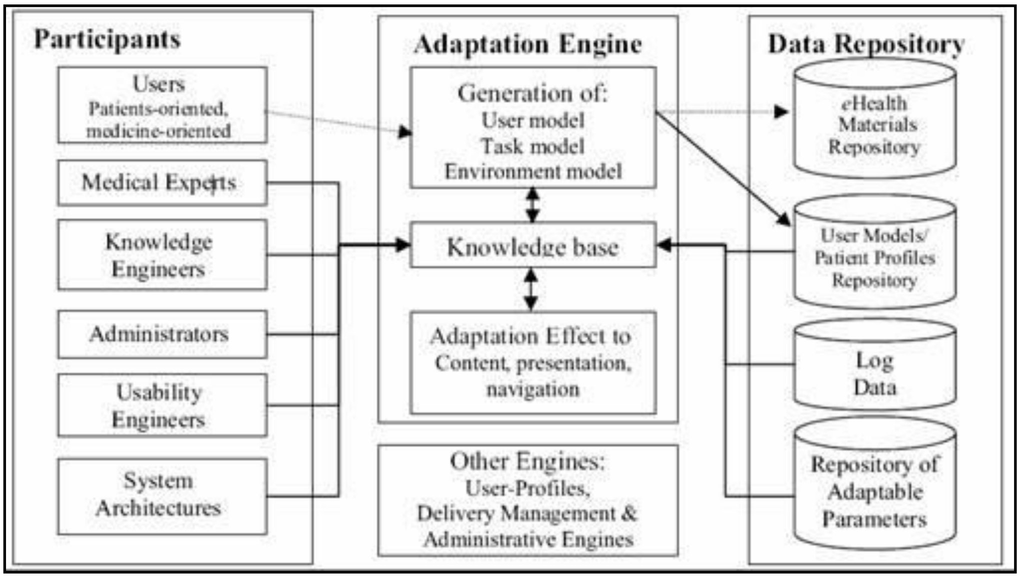


Figure 5.4: General Framework of Adaptive eHealth System (Pechenizkiy *et al*., 2005)

5.2.2.4. Adaptive SPSE

Zudilova-Seinstra (2007) proposed a possible architecture of an adaptive, scientific, problem-solving environment (SPSE) (Figure 5.5). The components of this model and how they relate to the criteria will be further discussed.

The adaptive SPSE architecture is formed by three major components: a Problem-Solving Framework, an Adaptation Engine and a Data Repository. The Problem-Solving Framework consists of Simulation, Visualisation and User Interaction tools. The Adaptation Engine is similar to the Adaptation Engine found in the eHealth system proposed by Pechenizkiy *et al.* (2005) serving as the core component for adaptation. It consists of three major compounds namely a User Model Generator, a provider of the Adaptation Effect and a Knowledge Base. The User Model guides and controls the adaptation process. The User Model Generator initialises and updates the User Model and the inferential component of adaptivity is thus satisfied with this component. The acquiring of information evident in this component also corresponds to the afferential component of adaptivity but the afferential component of adaptivity is not explicitly defined. A provider of the Adaptation Effect is responsible for configuring the UI based on information obtained from the User Model. This component corresponds to the efferential component of adaptivity (Zudilova-Seinstra, 2007).

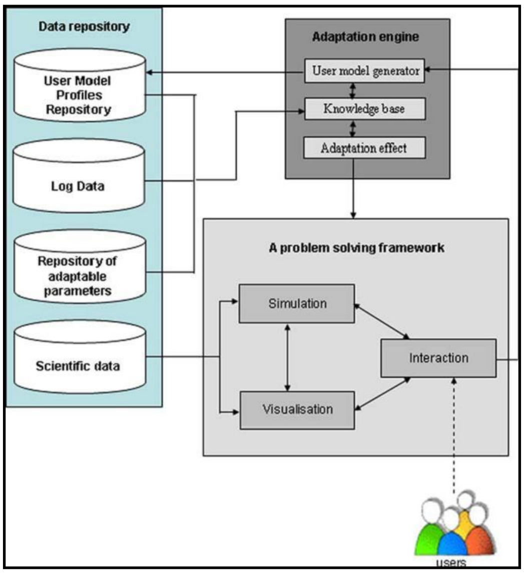


Figure 5.5: Possible Architecture of an Adaptive SPSE (Zudilova-Seinstra, 2007)

5.2.2.5. AUI Model Comparison

Sections 5.2.2.1 to 5.2.2.4 assessed AUI models based on whether or not they contained the three components of adaptivity. A comparison of the investigated AUI models (Table 5.1) reveals that the High-level Model of Adaptation is the only AUI model which satisfies all three components of adaptivity (Section 5.2.2.1). The only component which the High-level Model of Adaptation does not formally specify is the Knowledge Base.

The other models (Section 5.2.2.2 to 5.2.2.4) investigated all consist of the efferential component of adaptivity and formally specify a Knowledge Base. The afferential and inferential components of adaptivity appear, however, to be implied but not formally specified.

Existing AUI models, with the exception of the High-level Model of Adaptation (Section 5.2.2.1), only specify an architectural design. The High-level Model of Adaptation consists of both an architectural design and a component-level design as a more procedural design of the components is specified (Section 5.2). The High-level Model of Adaptation does not, however, specify an interface design element. The interface element of a model is important to provide a technical specification of the interface design

(Pressman, 2005). Existing AUI models cannot be viewed as complete models due to them not consisting of the key elements of a model, and none has been specifically designed for the CC domain. However, these models do specify the basic architecture of an AUI model and so could be used to produce an AUI model which supports CCAs.

	ADAPTIVE USER INTERFACE MODELS			
	High-level model of adaptation	UbiquiTO	AUI for eHealth systems	Adaptive SPSE
Afferential	✓	✓	✗	✗
Inferential	✓	✗	✗	✓
Efferential	✓	✓	✓	✓
Knowledge Base	✗	✓	✓	✓

Table 5.1: A Comparison between Adaptive User Interface Models

5.3. Proposed Model

Singh (2007) proposed an IUI model which was found to be suitable for CCs (Section 5.2.1). The purpose of this research is to propose an AUI model for CCs which caters to the skill level of CCAs. Similar to the IUI model proposed by Singh (2007), the proposed AUI model comprises three of the four key elements of a model, namely:

1. An architectural element;
2. A component-level element; and
3. An interface element.

Singh (2007) modified Tyler *et al.*'s (1991) intelligent interface architecture by addressing its limitation. This limitation was its inability to infer user's goals. The architectural element of the proposed AUI model, as depicted in Figure 5.6, thus comprises of Tyler *et al.*'s modified intelligent interface architecture. This architecture, hereafter known as the IUI architecture, was discussed in Section 5.2.1. The component-level element and the interface elements of the proposed model will be discussed in the subsequent sections.

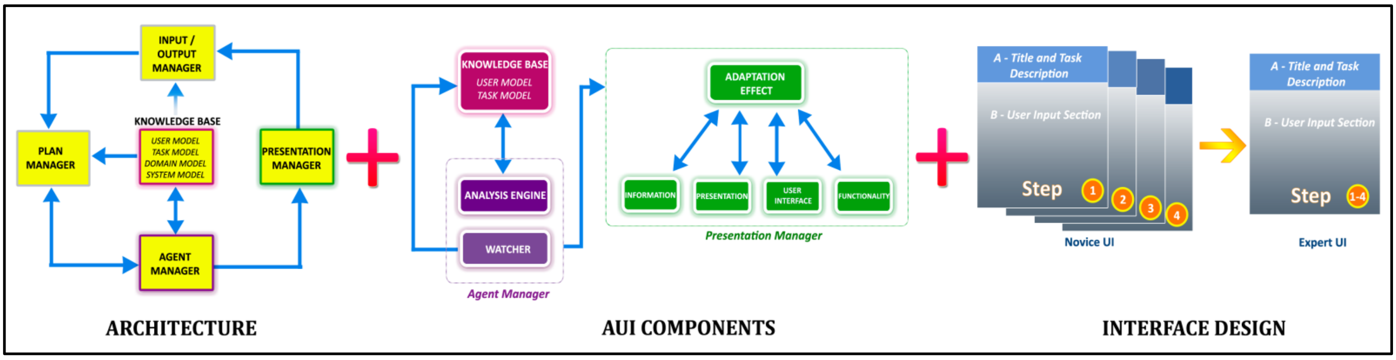


Figure 5.6: Proposed AUI Model

5.3.1. *Component-Level Design*

As previously mentioned, the component-level element of models specifies the model's architecture components in more detail, resulting in a procedural specification (Section 5.2). The component-level design element of the proposed model consists of the following AUI components:

- Knowledge Base;
- Analysis Engine;
- Watcher; and
- Adaptation Effect.

It was also previously mentioned that AUIs are recognised as being a central component of IUIs (Section 3.6). The components of the IUI architecture (Figure 5.6) that support AUIs are the Agent Manager, the Knowledge Base and the Presentation Manager (Section 5.2.1) and the component-level design element will detail these components. The manner, in which the component-level design element (the above-mentioned AUI components) conforms to the architectural element (IUI architecture) of the proposed model, will be discussed.

The Agent Manager component of the IUI Architecture, as discussed in Section 5.2.1, receives input from the Plan Manager, updates the User Model, and ensures that the UI is modified according to the user's needs. As seen in Figure 5.7, the Analysis Engine and the Watcher AUI components serve as the Agent Manager component of the IUI Architecture. These AUI components collectively provide the same function if not more, as that provided by the Agent Manager.

The Presentation Manager component of the IUI Architecture is discussed in Section 5.2.1. This component receives input from the Agent Manager component and uses knowledge obtained to decide how the UI can be adapted to the user's needs. As depicted in Figure 5.7, the Adaptation Effect AUI component serves as the Presentation Manager component of the IUI Architecture. The Information, Presentation, User Interface and Functionality components are contained within the Adaptation Effect.

Therefore, the AUI components, mentioned in this section, exhibit the relationship between IUIs and AUIs. The most essential components of AUIs, namely the afferential, inferential and efferential components of adaptivity need to be present in any AUI model. The sections that follow further discuss the AUI components and how they satisfy the essential components of adaptivity.

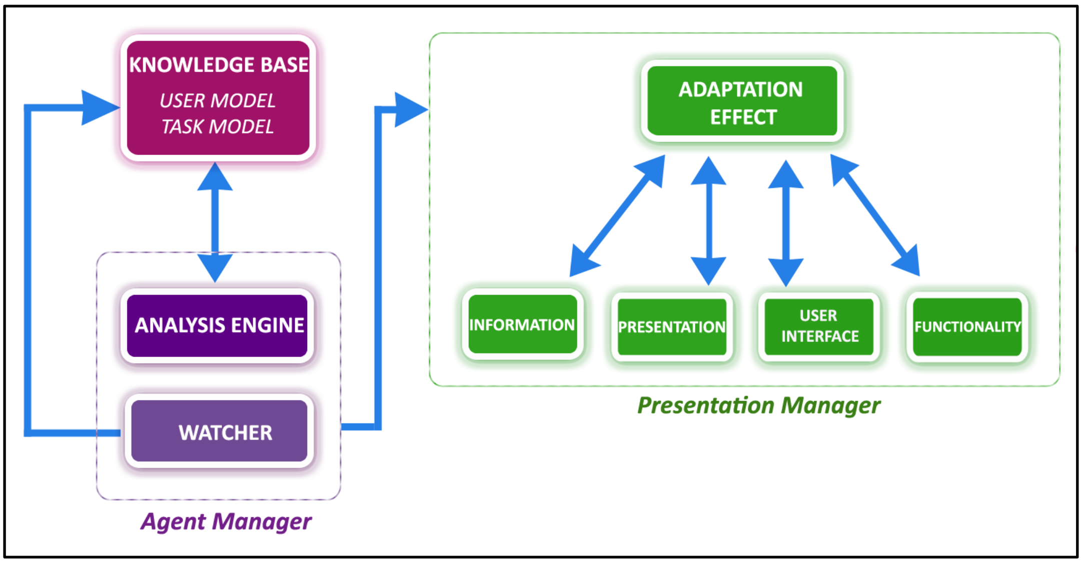


Figure 5.7: AUI Model Components

5.3.1.1. Knowledge Base

The Knowledge Base was previously discussed in Section 5.2.1. It serves as the core component for AUIs and acts as a repository by making use of various models (user, task model, etc.). The Knowledge Base, however, is not limited to these models. Based on the definition of AUIs, the Knowledge Base should at least contain a user model as it provides the personalisation needed for AUIs.

The proposed model can be specialised for the domain of CCs by incorporating CC information in the Knowledge Base (Singh 2007). The other components can then be configured to operate within the new domain. The CC information needed to be stored in the Knowledge Base, necessary for an AUI, is:

- Customer information;
- Customer query information;
- User models of the CCAs using the application; and
- Task models relating to the tasks of logging customer calls.

A discussion of User and Task models will follow.

User Model

User models are the core components for AUIs (Section 4.4.1) as they provide the necessary user information needed for adaptation. The switch from guiding a novice CCA to facilitating an expert CCA requires the system to keep track of the CCAs' expertise levels. It was previously mentioned that novice and expert users' behaviour differs dramatically at the physical level of interaction (Section 3.3.2.1). Intuitively, the difference between novice and expert CCAs would be how fast they log calls, i.e., the speed at which they make list selections; thus, the User Model will contain performance-related information, which will consist of data for each Informative Moment (IM) and the potentially Predictive Features (PF) associated with that IM. Figure 5.8 depicts an extract of the user model schema. A discussion of IMs and PFs will follow.

```
<UserModelComboBox>
    <InteractionFeatures>
        <Low-levelMotionCharacteristics>
            <TotalTime></TotalTime>
            <YMouseVelocity></YMouseVelocity>
            <YMouseAcceleration></YMouseAcceleration>
            <DwellTime></DwellTime>
        </Low-levelMotionCharacteristics>
        <InteractionTechnique>
            <AverageDwellTime></AverageDwellTime>
            <NrItemsVisited></NrItemsVisited>
            <UniqueItemsVisited></UniqueItemsVisited>
            <SelectionTime></SelectionTime>
        </InteractionTechnique>
        <PerformanceModels>
            <KLMDifference></KLMDifference>
            <KLMRatio></KLMRatio>
        </PerformanceModels>
    </InteractionFeatures>
</UserModelComboBox>
```

Figure 5.8: User Model Schema

Hurst *et al*. (2007) define IMs as user actions which can be readily isolated, are indicative of the phenomena they wish to study, model or predict, and can be easily and accurately labelled. They gathered data obtained from menu operations and used it as IMs. For each IM, data for a number of possibly PFs is captured (Figure 5.9). Hurst *et al*. (2007) developed potential features that could be predictive of a user's skill level. These features are not based on a task model but rather on low-level mouse and menu data which could be used in any application as they are not application specific (Hurst *et al*., 2007).

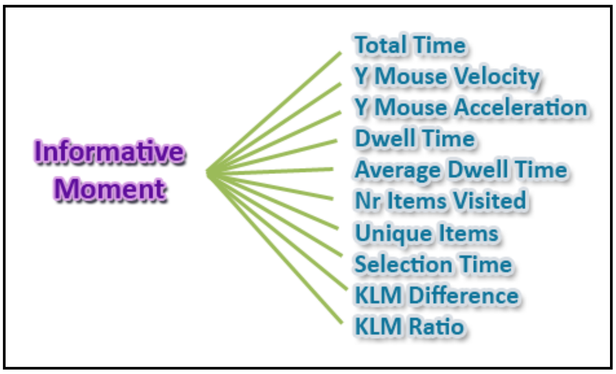


Figure 5.9: Potentially Predictive Features

The tasks undertaken by CCAs when logging customer queries constitute as mostly list selections, which are similar to performing menu selections, and the work done by Hurst *et al*. (2007) could prove to be useful here. For the purpose of this research, IMs are defined as list selections made by CCAs. List selections classified as IMs are when the CCA selects a customer, service name, call type, priority, source, campus, contact, cause and resolved option. The following potentially PFs, organised by category, were selected from Hurst *et al*. (2007) as most appropriate for CCA's list selections (Figure 5.8 and 5.9):

Features derived from low-level motion characteristics

- *Total Time (seconds)*: Total elapsed time within the action (starting when the list opened and ending when it closed). This is a summative value of all the selection times for the list selection.
- *Y Mouse Velocity* (pixels/second): Average velocity of the mouse during a list operation in the Y direction.
- *Y Mouse Acceleration* (change in velocity/second): Average unsigned acceleration of the mouse during a list operation in the Y direction.

- *Dwell Time* (seconds): Time spent dwelling (not moving) during the interaction sequence. Dwell time is only accumulated after the time spent dwelling lasts longer than one second.

Features related to the interaction technique

- *Average Dwell Time* (seconds/count): Time spent dwelling divided by the number of menu items visited.
- *Number of Items Visited* (count): Total number of list items that were visited during the list selection.
- *Unique Items* (count): Total number of unique list items visited during the list selection.
- *Selection Time* (seconds): Elapsed time (starting when the list opened and ending when it closed) within the item that was ultimately selected.

Features related to performance models

- *KLM Difference* (seconds): Difference between Keystroke Level Model (KLM) predicted time and selection time for the action. The KLM involves constructing a detailed, task-specific model of expert behaviour (Hurst *et al.*, 2007). Obtaining the KLM predicted time involved using a modified version of the KLM model (Figure 5.10) created by Hurst *et al*. (2007). This model is useful as it is presumed that expert users would perform at speeds closer to the predicted time than novices would.
- *KLM Ratio* (dimensionless): KLM-predicted time divided by the actual time for the action.

(LM + T) + C

= (1.35 + 1.1) + 0.2

= 2.65

where:

LM is the *Look and Mental Operator*,

T is the *Travel* to move the mouse to the target, and

C is the mouse *Click* when selecting a item.

Figure 5.10: KLM Design for a List Selection

Task Model

Task models represent tasks that the user can perform with the system at a particular time (Section 4.4.2). A user's goal can only be recognised by the utilisation of a task model. An AUI utilises a task model to offer the user task support. There are various areas of task support that the proposed AUI model provides.

Task support can be provided by the display of task status information. Current task and step information is displayed to the user in a section allocated to the delivery of task-based information, similar to that implemented by Singh (2007).

Task support can also be provided in the form of error checking. Error messages can be displayed to the CCA based on incomplete tasks. A consolidation with the user model enables error messages to be displayed in accordance with the CCA's skill level.

Providing the task support mentioned above requires the task model to contain certain information. The information contained within the task model is the name of the task well and its status, i.e., whether the task is complete or not. Figure 5.11 depicts an extract of the task-model schema.

```
<TaskModel>
  <Steps>
    <SearchCustomer>
      <Search_By>Incomplete</Search_By>
      <Search_Value>Incomplete</Search_Value>
      <Select_Customer>Incomplete</Select_Customer>
    </SearchCustomer>
    <LogCall>
      <Call_Description>Incomplete</Call_Description>
      <Call_Categorisation>
        <ServiceName>Incomplete</ServiceName>
        <CallType>Incomplete</CallType>
        <SubCallType>Incomplete</SubCallType>
      </Call_Categorisation>
      <Call_Classification>
        <Priority>Incomplete</Priority>
        <Source>Incomplete</Source>
      </Call_Classification>
    </LogCall>
    <AssignCall>
      <Select_Campus>Incomplete</Select_Campus>
      <Select_Contact>Incomplete</Select_Contact>
    </AssignCall>
    <ProvideSolution>
      <Solution_Description>Incomplete</Solution_Description>
      <Solution_Classification>
        <Cause>Incomplete</Cause>
        <HowResolved>Incomplete</HowResolved>
      </Solution_Classification>
    </ProvideSolution>
  </Steps>
  <CurrentStep>
  </CurrentStep>
</TaskModel>
```

Figure 5.11: Task Model Schema

5.3.1.2. Analysis Engine

The Analysis Engine uses the user model and other models in the Knowledge Base to derive new user information (Figure 5.12). The Analysis Engine can update the user model based on new information learned about the user, or it can initiate an event such as suggesting something to the user. It also responds to queries from the application. Thus, the inferential component of adaptivity is satisfied by the Analysis Engine. As previously stated, the inferential component of adaptivity requires a basis, such as a set of rules, to be specified for drawing inferences made (Section 4.5).

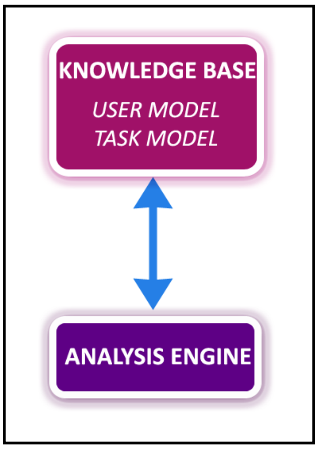


Figure 5.12: Analysis Engine AUI Component

The Analysis Engine fulfils the essential role of user modelling (Section 4.5.1). Various user-modelling techniques such as stereotypes (Section 4.5.2.1) and classification learning (Section 4.5.2.2), can be employed to construct and maintain the user models needed for adaptation. A stereotype user-modelling technique facilitates the differing skill levels of CCAs. Section 2.6 mentioned that novice CCAs become skilled quickly; therefore, a stereotype user modelling technique employing only novice and expert user groups is appropriate for CCs. The Analysis Engine can, however, only accomplish its crucial role once it has obtained data needed for the adaptation.

5.3.1.3. Watcher

The Watcher takes the same role of the watcher component found in the architecture of UbiquiTO (Section 5.2.2.2) proposed by Cena *et al.* (2006). The Watcher's role is to acquire user information implicitly (by observing the user's behaviour), and/or information could be provided by the user explicitly. The afferential component of adaptivity is thus satisfied by the Watcher (Section 4.4).

Once the Watcher has collected the necessary information from the user, it can update the user and task models stored within the Knowledge Base (Figure 5.13). Information obtained is only useful if the Analysis Engine can map it on to a particular adaptation effect.

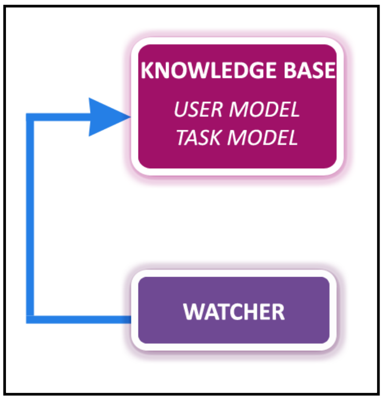


Figure 5.13: Watcher AUI Component

5.3.1.4. Adaptation Effect

The Adaptation Effect is similar to the Adaptation Effect component found in the architectures proposed by Pechenizkiy *et al.* (2005) (Section 5.2.2.3) and Zudilova-Seinstra (2007) (Section 5.2.2.4). The Adaptation Effect decides how to adapt the UI to the user's behaviour based on data obtained from the Knowledge Base. The efferential component of adaptivity is satisfied by the Adaptation Effect component (Section 4.9). The adaptation can take place on different levels, i.e., various kinds of adaptations could be provided (Figure 5.14). As previously mentioned (Section 4.9), the various levels of adaptation are:

1. *Information Level;*
2. *Presentation Level;*
3. *User Interface Level; and*
4. *Functionality Level.*

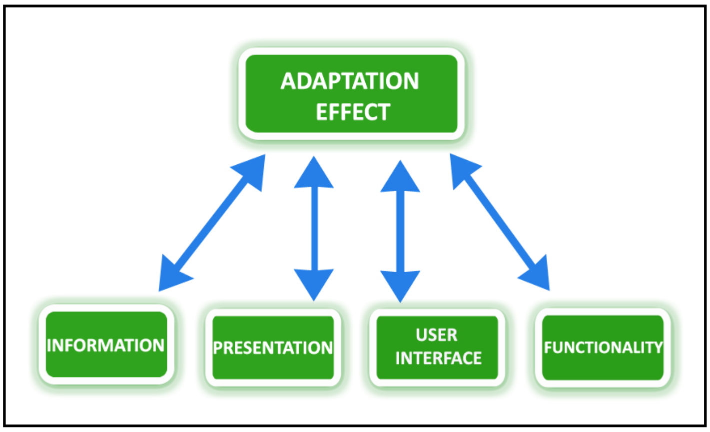


Figure 5.14: Adaptation Effect AUI Components

The Information, Presentation, User Interface and Functionality components within the Adaptation Effect component thus map directly to above-mentioned levels of adaptation. Specialising the proposed model to CC domain involves an Adaptation Effect which employs the User Interface component. The UI design will be further discussed in the next section.

5.3.2. Interface Design

As previously mentioned, the interface design element specifies the GUI design of the interfaces (Section 5.2). Interface design can be divided into three different types of design: internal, external and human-computer interface design (Pressman, 2005). The proposed AUI model focuses on human-computer interface design.

The human-computer interface design needs to support the CC steps involved when logging a customer's queries. The primary steps involved when logging a customer's query, as previously mentioned in Section 2.7, are capturing the customer's details, capturing the call's details, assigning the call and/ or capturing the call's resolution details.

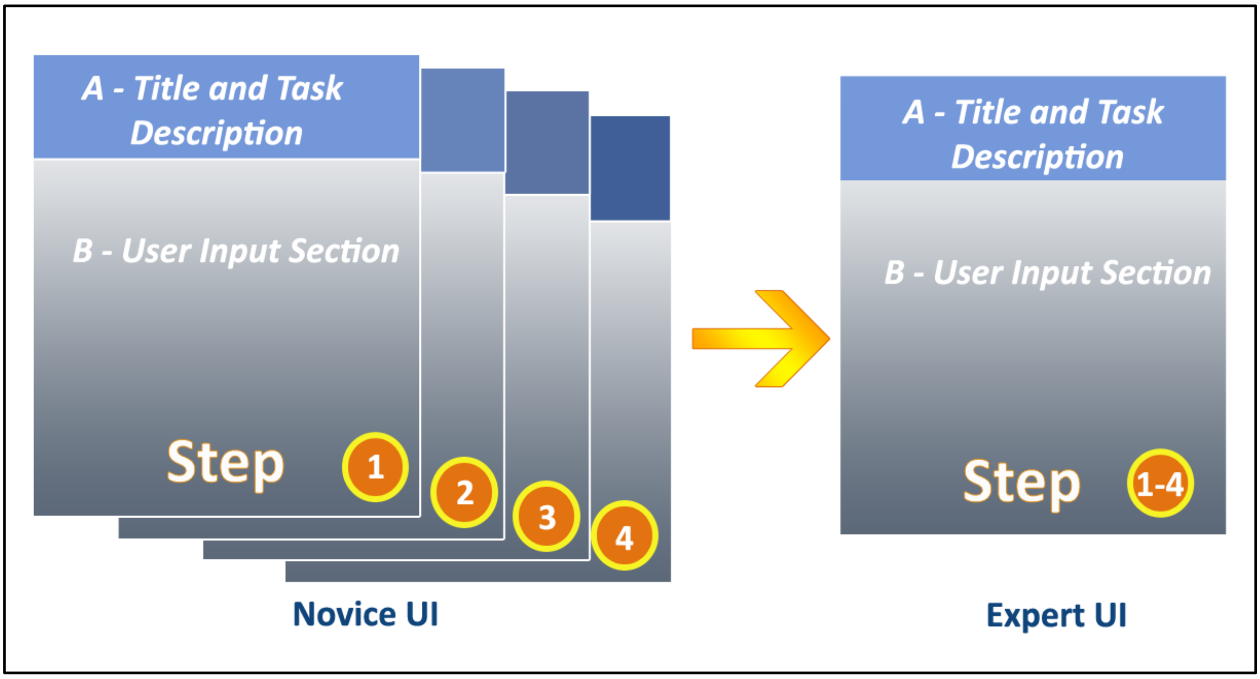


Figure 5.15: Low Fidelity Prototype of the Novice and Expert UIs

Section 2.6 discussed a common design flaw when designing UIs for CCs. The design flaw is that the expert user is disregarded and design is only catered to the novice user. CCAs have different skill levels and these differences need to be accounted for when designing UIs. Multi-layer UIs cater to the differing skill levels of users and facilitate the learning of UIs by empowering users to learn the UI in a meaningful sequence (Section 3.4). *The interface-design element of the proposed AUI model consists of a two-level, multi-layer design* (Figure 5.15). The first level consists of a sequence of screens which is designed to cater to novice CCAs. This first-level UI will hereafter be known as the novice UI. The second level consists of a single screen which is designed to cater to expert CCAs. This second-level design will hereafter be known as the expert UI. The CCA will be transitioned from using the novice UI to using the expert UI based on rules contained within the Analysis Engine.

The novice UI and the expert UI contain the same UI design. The UI design contains two sections, namely section A and section B (Figure 5.15). Section A, similarly to Singh (2007), is a section dedicated to the delivery of task-based information. This section utilises the task model contained in the Knowledge Base (Section 5.3.1.1.b). Section B serves as the direct manipulation section requiring user input. The Watcher AUI component monitors the user's input within Section B of the novice UI.

Novice users are more concerned with how to do things instead of how fast they can do them (Section 3.3.1.1). This suggests that novice users need to be guided through their tasks. The novice UI (Figure 5.15) displays a separate screen for each of the call-logging steps, guiding the CCA step-by-step through the call-logging process.

Expert users require an efficient UI whereby the number of interactions is reduced (Section 3.3.1.2). The expert UI (Figure 5.15) displays only one screen for logging a call and users are not constricted to do tasks in a step-by-step manner as the novice users are. This means they are given more freedom when performing tasks.

5.4. Summary

Singh (2007) successfully implemented an IUI model which supports CC operations (Section 5.2.1). Existing AUI models (Section 5.2.2) are incomplete and do not cater for the relationship between AUIs and IUIs. Therefore, an AUI model was proposed which incorporates the relationship between AUI and IUIs and is specifically designed for CC operations.

The proposed AUI model consists of an architectural element, a component-level design element and an interface element (Section 5.3). The components of the architectural component which the component-level design element expands upon are the Knowledge Base, Agent Manager and the Presentation Manager. The component-level design elements (Section 5.3.1), namely the Analysis Engine and the Watcher, comprise the Agent Manager, and the Adaptation Effect component-level design element comprises the Presentation Manager.

The Knowledge Base component (Section 5.3.1.1) contains a user model and a task model. The user model contains performance-related data in the form of IMs, where each IM is associated with PF data. The user model thus contains the data which will be used by the Analysis Engine to determine the user’s skill level. The task model contains task status information needed for the UI design section that is dedicated to the delivery of task based information and also needed for error checking.

The Analysis Engine (Section 5.3.1.2) component employs a stereotype user-modelling technique which uses the information stored in the knowledge base to infer the user’s skill level. The Watcher (Section 5.3.1.3) collects the information both implicitly and explicitly and stores it in the Knowledge Base. The Adaptation Effect (Section 5.3.1.4) component of the proposed model provides an adaptation to the UI.

The interface element (Section 5.3.2) of the proposed model consists of a multi-level UI comprising two layers, where the first layer consists of a sequence of screens designed for the novice CCA and the second layer consist of a single screen design for the expert CCA. This chapter has only discussed the design of the proposed model. The success of the proposed AUI model can, however, only be evaluated once an implementation of the model has been undertaken. The next section, therefore, discusses the implementation of the proposed model.

Chapter 6: ASD Prototype Implementation and Pilot Study

6.1. Introduction

Implementation of a model can be demonstrated by means of a prototype. A prototype, as defined within information technology (IT), refers to a simplified system that serves as a guide or example for a complete system. The prototype is implemented to prove that the concept presented by the model has merit (Olivier, 2004). Chapter 5 concluded with the design of a proposed adaptive user interface (AUI) model for contact centres (CC). The AUI model proposed in the preceding chapter has yet to be implemented in practice.

The purpose of this chapter is to discuss implementation of the proposed AUI model. A prototype of the proposed model was implemented as a proof-of-concept. The prototype was named the AdaptiveServiceDesk (ASD). Implementation of the ASD is discussed in terms of the AUI components of the proposed model. The AUI components, as mentioned in the previous chapter, consist of a Knowledge Base, Watcher, Adaptation Effect and Analysis Engine. ASD supplies each contact centre agent (CCA) with the capability to log a customer's query via the novice UI until he or she has become skilled. When a CCA has reached a level which is considered to be that of an expert user, he or she transitions from the novice UI to the more appropriate expert UI. Implementation of the AUI components enables this transitioning and a discussion of the implementation of the AUI components is provided in this chapter.

The chapter commences with a discussion on the implementation of the Knowledge Base AUI component (Section 6.2). A discussion on the implementation of the Watcher AUI component will follow (Section 6.3). Implementation of the Watcher AUI component will entail a discussion on how both the User Model (Section 6.3.1) and the Task Model (Section 6.3.2) are updated. Implementation of the Adaptation Effect AUI component (Section 6.4) will be discussed in terms of the novice UI (Section 6.4.1) and the expert UI (Section 6.4.2). The pilot study (Section 6.5) will be discussed before a discussion on the Analysis Engine (Section 6.6) can resume. The discussion of the pilot study precedes that on the implementation of the Analysis Engine, because results obtained from the pilot study directly affect implementation of the Analysis Engine.

6.2. Knowledge Base

The Knowledge Base acts as a repository that stores CC information such as customer and query information as well as the User and Task models required for adaptation (Section 5.2.1). The CC information stored within the Knowledge Base supports the primary steps involved with logging a customer's query, which, as specified in Section 2.7, are:

1. Providing customer details;
2. Providing the call's details;
3. Assigning the call; and
4. Providing the solution (resolution) details.

The CCA handling the customer query always performs the first three steps mentioned above and only performs the last step of *Providing the solution details* if he or she assigns the query to themselves. The Knowledge Base was implemented using SQL Server Management Studio and XML. CC information stored within the SQL Server database consisted of an extract of the NMMU ICT Service Desk's HEAT database. Stored Procedures were used to access data from the SQL Server database as they can offer performance gains, when used, instead of regular queries.

The User Model was stored in the SQL Server database and contained the informative moments (IM) which were embedded as XML files. IMs, discussed in Section 5.3.1.1, are list selections made when interacting with the UI. Figure 6.1 depicts the list of IMs (nine), where each IM occurs at a particular step of logging a customer's query. The *Provide customer details* query logging step only contains the *search customer* IM, the *Provide the call's details* query logging step contains the *service name*, *call type*, *priority* and *source* IMs, the *Assign the call* contains the *campus* and *contact* IMs and the *Provide solution details* query logging step contains the *cause* and *resolved* IMs.

The Task Model was not stored in the database but rather separately as a XML file. This is due to its nature of storing short-term task information. The Watcher AUI component obtains and updates the relevant information in the Task and User Models. This will be discussed further in the next section.

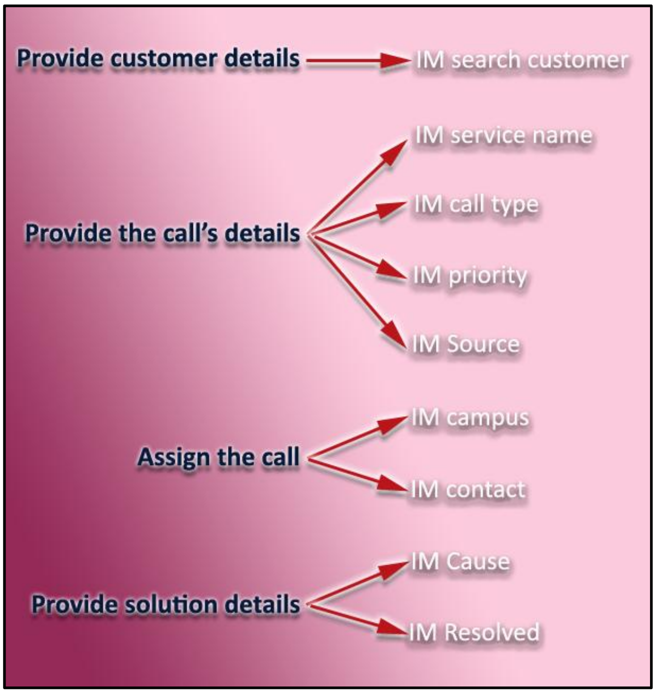


Figure 6.1: Informative Moments and their Corresponding Call Logging Steps

6.3. Watcher

Implementation of the Watcher, Adaptation Effect (Section 6.4) and Analysis Engine (Section 6.6) AUI components was achieved through the Microsoft Visual Studio 2008 integrated development environment (IDE) in Microsoft C#. The Watcher AUI component's core responsibility lies in data acquisition and then updating the Knowledge Base with the acquired data (Section 5.3.1.3). The Watcher needs to collect data for the User and Task model components of the Knowledge Base. Implementation of the Watcher component concerning the User and Task model components will be discussed in more detail.

6.3.1. Updating the User Model

The Watcher can only update the User model once it has obtained the necessary data. Design of the User Model (Section 5.3.1.1) indicated that it consists of data for each IM (list selection) and the predictive features (PF) associated with that particular IM. All nine IMs contain the same ten PFs and, as previously mentioned (Section 5.3.1.1), the categories to which the PFs belong are:

- Low-level motion characteristics;
- Interaction technique; and
- Performance models.

Implementation of the Watcher thus decomposes the PF categories into three classes: each class of category is responsible for acquiring the PFs associated with that category. These three classes will now be discussed further.

The *LowLevelMotionCharacteristics* class (Figure 6.2) is responsible for obtaining PFs which belong to the low-level motion characteristics' category. The PFs belonging to this category include the *Total Time*, *Y Mouse Velocity*, *Y Mouse Acceleration* and *Dwell Time*. As depicted in Figure 6.2, the *getTotalTime()*, *getMouseVelocity()*, *getYMouseAcceleration()* and *getDwellTime()* methods of the *LowLevelMotionCharacteristics* class obtain the PFs of the low-level motion characteristics' category.

The *Total Time* and *Dwell Time* PFs within this category utilised the *Windows.Forms.Timer* class. The *Total Time* PF required implementation of a *Windows.Forms.Timer* in order to calculate the (total) time, in seconds, that the user opened the specific list selection (IM). The *Dwell Time* PF required implementation of a *Windows.Forms.Timer* in order to calculate the time spent not moving for a period greater than one second. Calculations performed for the *Dwell Time* and *Total Time* included all actions taken with the list from when it was opened and closed and not merely the actions which involved the item that was ultimately selected. The *Y Mouse Velocity* was calculated by dividing the *Selection Time* by the Y distance moved within the list selection. This distance was calculated by counting the number of items visited during the list selection and multiplying this count with the item's pixel height. The *Y Mouse Acceleration* was calculated by dividing the *Y Mouse Velocity* by the *Selection Time*.

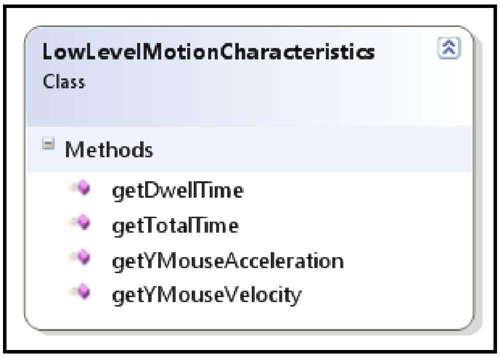


Figure 6.2: Low-level Motion Characteristics Class

The *InteractionTechnique* class (Figure 6.3) is responsible for obtaining PFs which belong to the interaction technique category. The PFs belonging to this category include the *Average Dwell Time*, *Nr of Items visited*, *Unique Items* and *Selection Time*. As depicted in Figure 6.3, the *getAvgDwellTime()*, *getNrItems()*, *getSelectionTime()* and *getUniqueItemsVisited()* methods of the *InteractionTechnique* class obtain the PFs of the interaction technique category. The *Selection Time* PF within this category utilised the *Windows.Forms.Timer* class in order to calculate the time in seconds that the user spent within the list selection to decide on the item that was ultimately selected. The *Nr of Items visited* PF was calculated by counting the number of items selected. The *Average Dwell Time* PF was calculated by dividing the *Dwell Time* by the *Nr of Items Visited*. The *Unique Items* PF utilised a hash table to store the number of items that were selected only once for the specific list selection (IM).

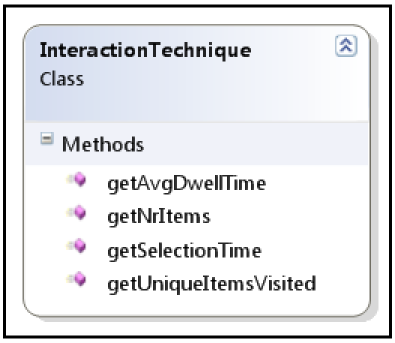


Figure 6.3: Interaction Technique Class

The *PerformanceModels* class (Figure 6.4) is responsible for obtaining PFs which belong to the performance models category. The PFs belonging to this category include the *KLM Difference* and *KLM Ratio*. As depicted in Figure 6.4, the *getKLMDifference()* and *getKLMRatio()* methods of the *PerformanceModels* class obtains the PFs of the performance models category. The *KLMPredictedTime* field (Figure 6.4) pertained within the *PerformanceModels* class is a constant which equates to 2,65 seconds. This constant was calculated by implementing the design of the KLM for a list selection (Figure 5.5). The *getKLMDifference()* method thus calculates the absolute difference between the user's *Selection Time* and the *KLMPredictedTime*. It is predicted that expert users will have a smaller *KLM Difference* than novice users. The *KLM Ratio* PF was calculated by dividing the *KLMPredictedTime* by the *SelectionTime* for the specific list selection (IM).

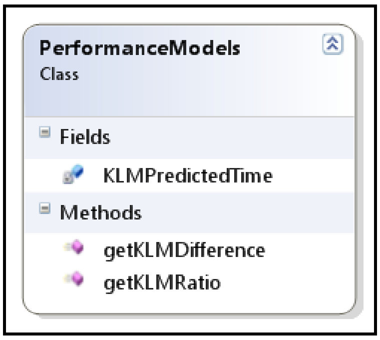


Figure 6.4: Performance Models Class

Once the Watcher obtains values for all the PFs necessary, the User Model needs to be updated with this information. The *UpdateUserModelComboBox* class is responsible for retrieving each IM's PF values and then storing the IM into XML format in the Knowledge Base. The *getUserModelComboBoxXML()* method is responsible for calling the appropriate stored procedure which retrieves the IM in XML format and loads it into memory.

The various update methods (Figure 6.5) within the *UpdateUserModelComboBox* class are responsible for updating the IM with the PF values which are obtained through instantiations of the *LowLevelMotionCharacteristics*, *InteractionTechnique* and *PerformanceModels* classes (classLowLevelMotionCharacteristics, classInteractionTechnique and classPerfomanceModel fields in Figure 6.5). The *updateUserModelComboBoxXML()* (Figure 6.5) finally saves the IM to the Knowledge Base.

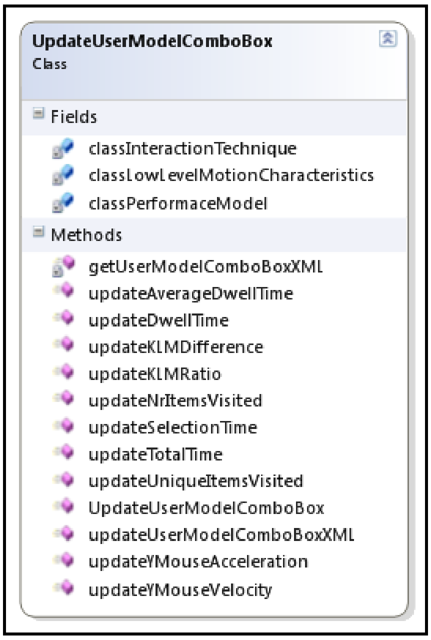


Figure 6.5: Update User Model Class

6.3.2. Updating the Task Model

The Task Model, stored within the Knowledge Base, facilitates the delivery of the task status information in the form of complete and incomplete steps. The Task Model, together with the CCA's experience level stored in the CCA's User Model, is used to deliver appropriate error messages. The Watcher thus updates the Task Model by obtaining the status (incomplete or complete) of the current task and then modifying it accordingly. In contrast to data stored within the User Model, the Task Model contains session specific data and is stored as a XML file. The *GetTaskModel* class is responsible for obtaining a task's status by traversing the XML file and it is also responsible for obtaining the steps involved in a particular task. All task steps are initially classified as incomplete. As the CCA concludes a task's step, the *UpdateTaskModel* class (Figure 6.6) overwrites the step's status as complete. The various update methods, seen in Figure 6.6, perform this function of updating the step's status.

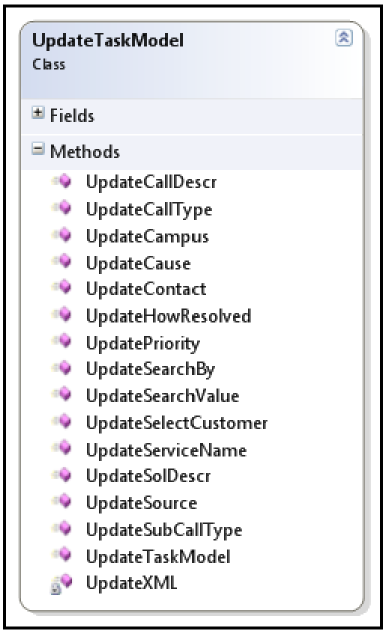


Figure 6.6: Update Task Model Class

Completion by a CCA logging a customer's query, results in the *RestoreTaskModel* class being called. The *RestoreTaskModel* class's reponsibility resets all task steps' statuses as incomplete. This is due to the Task Model being session specific.

6.4. Adaptation Effect

The Adaptation Effect AUI component utilises the User Interface component, thus providing adaptation to the UI. A multi-layer UI design (Section 3.4) is employed, consisting of two levels of interfaces: namely, the novice and the expert UI. Adaptation to the UI is provided by initially providing the CCA with the novice UI and then, when the Analysis Engine detects that the CCA is behaving like an expert, the CCA is provided with an expert UI. This sub-section will discuss the design of the novice UI (Section 6.4.1) and provides a scenario detailing how the novice UI functions when logging a customer's query (Section 6.4.1.1). This sub-section also includes a discussion on the design of the expert UI (Section 6.4.2).

Both the novice and expert UIs need to support the steps involved in logging a customer's query and consist of two sections: namely, the Title and Task description and the User Input section (Section 5.3.2). Previously, Figure 5.10 depicted a low-fidelity prototype of the interface design of these two sections and implementation thereof can now be seen in Figure 6.7.

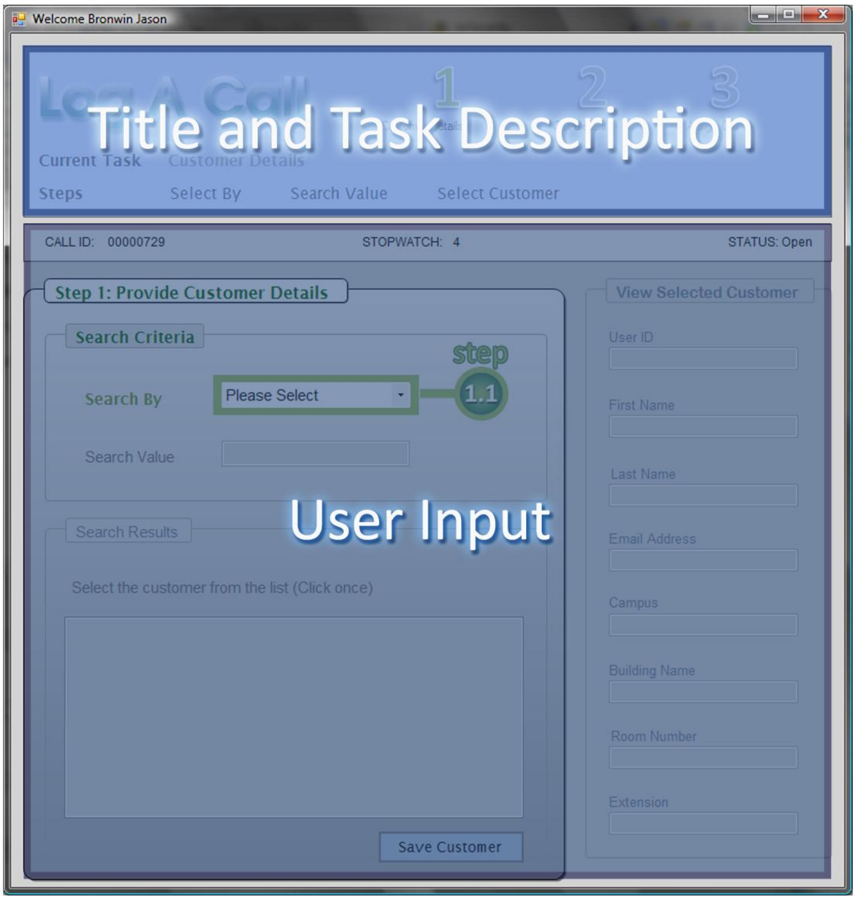


Figure 6.7: Title and Task Description and User Input Sections

The novice UI and expert UI both consist of the same Title and Task Description section, depicted in Figure 6.8 which displays the name of the task which the user is currently undertaking as well as the necessary steps involved. The Title and Task Description section also contains a Visual Task Indicator (Figure 6.8) which displays the total amount of tasks involved in logging a customer's query. The Visual Task Indicator colour codes the call logging tasks as an indication of the user's progress. The status of a task is represented by three distinct colours, namely, **grey**, **green** and **blue**, where **grey** represents a task which has not yet been attempted by the CCA, **green** represents the task which the CCA is currently busy with and **blue** represents a task which the CCA has completed (Figure 6.9). These colours are also utilised when displaying the *Current Task Name* and the *Steps of Current Task* (Figure 6.8).

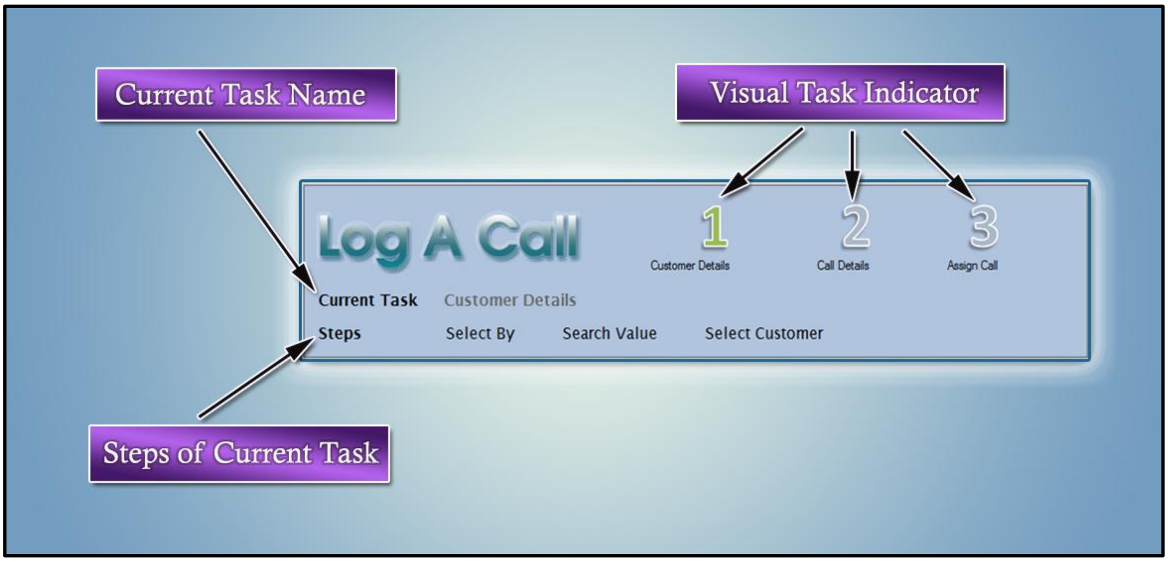


Figure 6.8: Title and Task Description

Colour, when properly utilised, has the ability to communicate facts and ideas more quickly and aesthetically to the user (Wright *et al.*, 1997). **Green** and **blue** were the colours chosen as step indicators because they are easy to learn and remember. **Green**, in particular was chosen to represent the current task being performed, as it is used to show positive progress (Wright *et al.*, 1997). The meanings of the colours chosen should be commonly held in the population-at-large; therefore, another reason **green** was chosen to represent the current task, is because it is commonly associated with "go" (Brown, 1998). **Blue** is perceived as a colour in the background which does not compete for attention; therefore, it was chosen to represent completed tasks (Wright *et al.*, 1997). **Grey** was chosen to represent incomplete tasks, as it was successfully utilised to represent incomplete tasks in the prototype developed by Singh (2007).

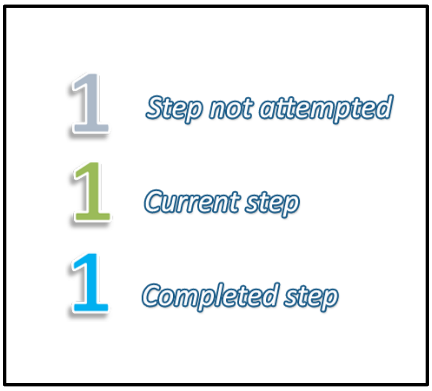


Figure 6.9: Colour Chart

The Title and Task Description section of the novice and expert UIs are similar: the only difference is that the novice UI further provides tooltips that appear as the user's mouse hovers over this section. If the user's mouse cursor hovers over the Title and Task Description section, a description of the section is presented to the user (Figure 6.10 (A)). Hovering over the task number (eg., 2 or 3 in Figure 6.10 (B)), in particular, provides the user with a detailed description of the status of the call logging task. This is done to aid the novice user's unfamiliarity with the UI (Section 3.3.3).

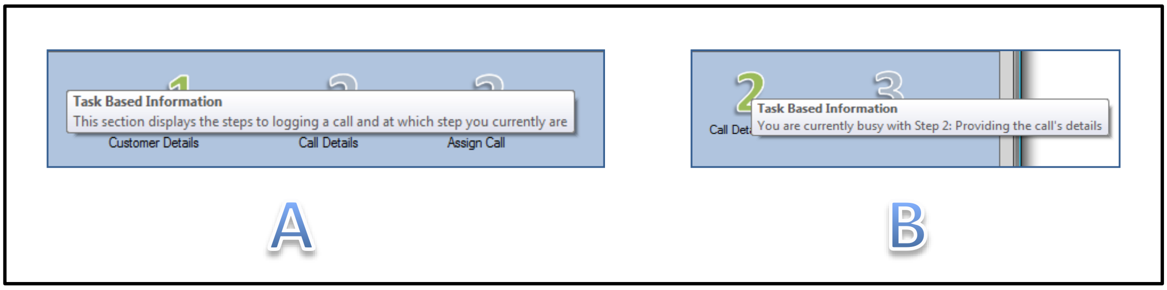


Figure 6.10: Visual Task Indicator Tooltips

The task-based information displayed in the Title and Task Description section represents the information residing within the Task Model, and the methods used when accessing the Task Model are responsible for maintaining this UI section. The user's interaction with the User Input section of the UI directly affects the Title and Task Description section.

The User Input section, however, unlike the Title and Task Description section of the UI, is different for the novice UI and expert UI. The novice UI includes a separate screen; thus, it contains a separate User Input section for each of the call logging steps. The expert UI contains a single screen; thus, it contains a single User Input section, where all query logging steps are tabbed into one interface. Further details of implementation of both the novice and expert UIs follow.

6.4.1. Novice User Interface

Novice users are more concerned with how to do things instead of how fast they can do them; therefore, they require the UI to be easy to learn so that they can quickly become experts (Section 3.3.1.1). CC UI designs, in particular, need to anticipate what the customer will say and design for that next step (Section 2.6); thus, the design of the novice UI entails a separate screen for each of the call logging steps, familiarising the CCA with the process of logging a customer's query.

It was previously noted that the novice user's unfamiliarity with the UI could be aided by incorporating explanatory elements into it (Section 3.3.3). UIs designed for novice users should include all assistance in the primary interface since novice users depend on system features to assist their recognition memory (Section 3.3.3.1); thus the User Input section of the novice UI contains a Visual Step Indicator (Figure 6.11), which guides the user, step-by-step, through the process of logging a customer's query. This Visual Step Indicator gradually highlights the area of the screen requiring user input. The highlighting is represented by a green border surrounding the appropriate interaction area. The Visual Step Indicator also visually indicates the step with which the highlighted area is associated, by means of a green circle (Figure 6.11). The naming convention which the Visual Step Indicator uses is *[call logging step number (1 to 4)].[sub-step number]*. The Visual Step Indicator, represented in Figure 6.11, corresponds to step 1.1 which indicates that it is the first sub-step of the first call logging step. The Visual Step Indicator, in summary, emphasises the step to be completed next by moving to the next step as the user handles the current step.

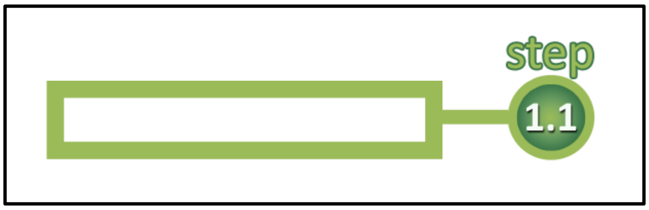


Figure 6.11: Visual Step Indicator

Novice users need informative feedback (Section 3.3.3.1) to assist them with their unfamiliarity with the CC UI. Informative feedback is provided with regards to error messages (Figure 6.12 (A)) and confirmation dialogues (Figure 6.12 (B)). Error messages provide an indication of the incompleted step and how to rectify the error made. Consolidation with the Task Model with regards to completed and incompleted steps enables the provision of error messages. Confirmation dialogues provide novice users with more detailed confirmations.

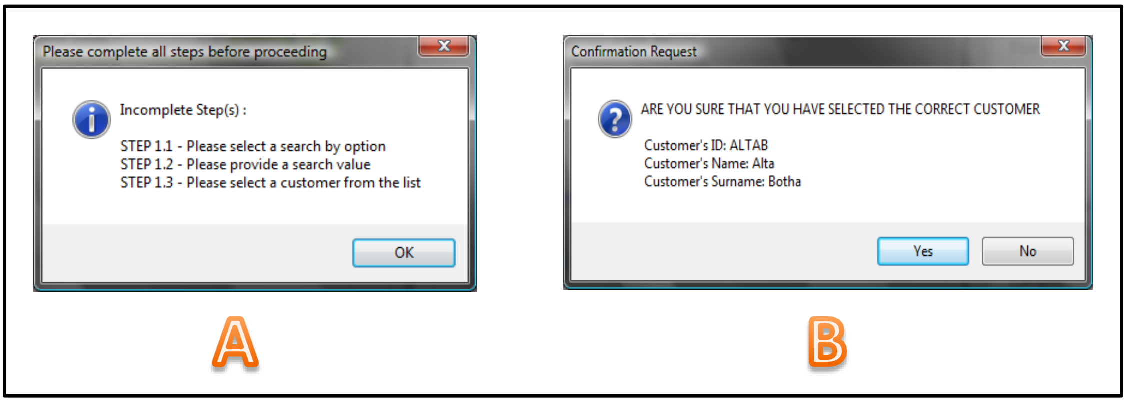


Figure 6.12: Error Message (A) and Confirmation Dialogue (B)

This sub-section has provide a discussion on the design elements supported by the novice UI. The Visual Step Indicator, which was discussed in this section, is the key design element of the novice UI. The functionality of the novice UI can be more effectively illustrated by demonstrating its ability to support the task of logging a customer’s call. The next section will provide a discussion on the design of the novice UI screens.

6.4.1.1. Task-based User Interface Design

The purpose of this sub-section is to illustrate the functionality of the screens which the novice UI comprises. The scope of this research is limited to the task of logging a customer’s call (Section 1.3.3). The novice UI, as previously mentioned, supports the task of logging a customer’s call by comprising a separate screen for each of the call logging steps. There are four call logging steps (Section 6.2), and the fourth call logging step (*Providing solution’s details*) is only performed if the call is assigned to the CCA receiving the call. Thus, the novice UI consists of four screens.

The order in which a CCA performs the call logging steps is sequential: the second step (*Providing the call’s details*) cannot be executed until the first step (*Identifying the customer*) has been completed and so forth. CC UIs needs to be designed according to solid task analysis (Section 2.6); therefore, to accustom the novice user to the CC process, the novice UI screens are presented in a sequential manner.

Figure 6.13 shows a hierarchical task analysis for logging a customer's call. The call logging sub-steps, which are highlighted as white boxes in Figure 6.13, will be demonstrated in this sub-section. Screenshots of call logging steps 1.2 (*Search value*) and 1.3 (*Select customer*) effectively illustrate the functionality provided by the novice UI screens, particularly with regards to the support provided by the Visual Step Indicator (Figure 6.11). Screenshots of call logging sub-steps 2.1 (*Call description*), 3.1 (*Selecting assignee's campus*) and 4.1 (*Solution description*) are provided to demonstrate the design of the remaining novice UI screens.

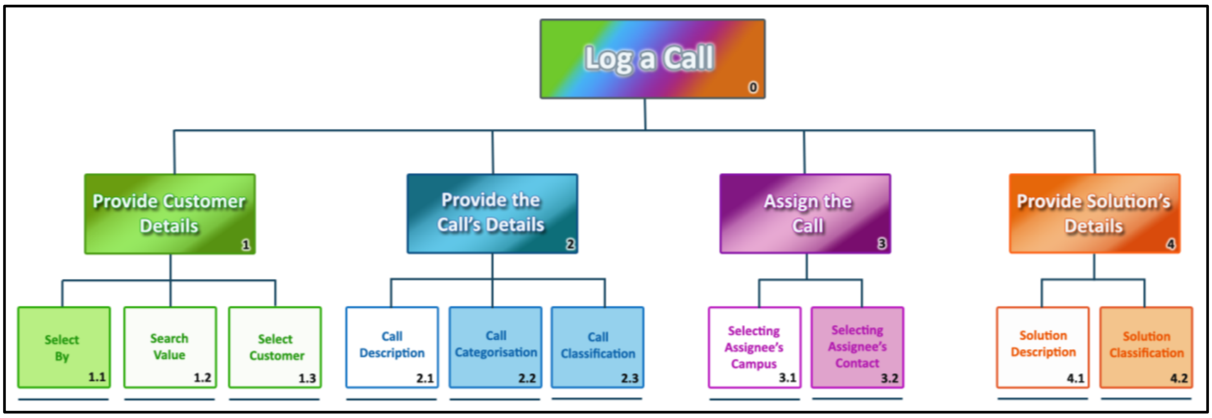


Figure 6.13: A Graphical Representation of the Task Analysis for Logging a Call

Step 1: Provide Customer Details

Figures 6.14 and 6.15 represent screenshots of the first screen of the novice UI. This supports the first call logging step: namely, *Provide Customer Details*, which entails three sub-steps:

1. Select by;
2. Search value; and
3. Select customer.

The above-mentioned sub-steps are obtained due to methods residing in the Task Model. A design of the Task Model schema containing all the call logging steps and their corresponding sub-steps can be seen in Figure 5.11. Users are restricted in the order in which they perform the sub-steps.The User Input section of the novice UI supports the Title and Task Description section by utilising the same colours. The screenshot of the first novice UI is as a result of the user selecting to *Search by* the customer's first name (Figure 6.14). As depicted in the User Input section of Figure 6.14, the *Search by* sub-step has been completed due to its forecolour text being **blue**. The *Search by* sub-step in the Task and Description section of Figure 6.14 is also depicted as being complete, illustrating the collaborative nature of the Task and Description and the User Input UI sections. The Visual Step Indicator (Figure 6.14) indicates the next step to be completed and, as seen in the area highlighted by the Visual Step Indicator, the novice UI trains the user by providing instructions to the interface ('*Enter User ID here*' in Figure 6.14). The area of the User Input section in which the user needs to select the customer remains **grey** as this step has not yet been attempted (Figure 6.14).

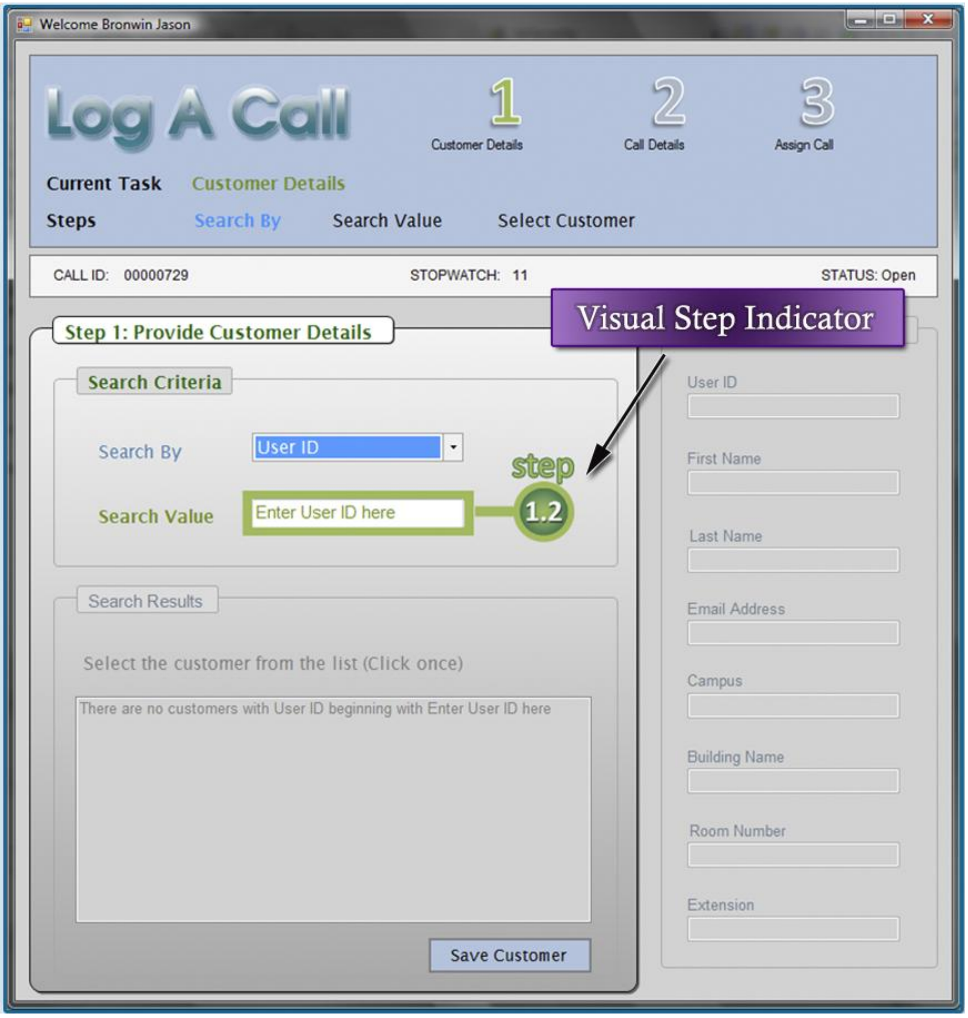


Figure 6.14: Novice UI (Step 1.2)

Once the user has entered text in the *Search value* section ('*ann*' in Figure 6.15), The Visual Step Indicator relocates its position to the *Select customer* section (*search results* area in Figure 6.15) emphasising the next sub-step which needs to be completed for the current call logging step. The *search value* sub-step is depicted as being completed due to its **blue** forecolour in both the User Input and the Title and Task Description sections (Figure 6.15).

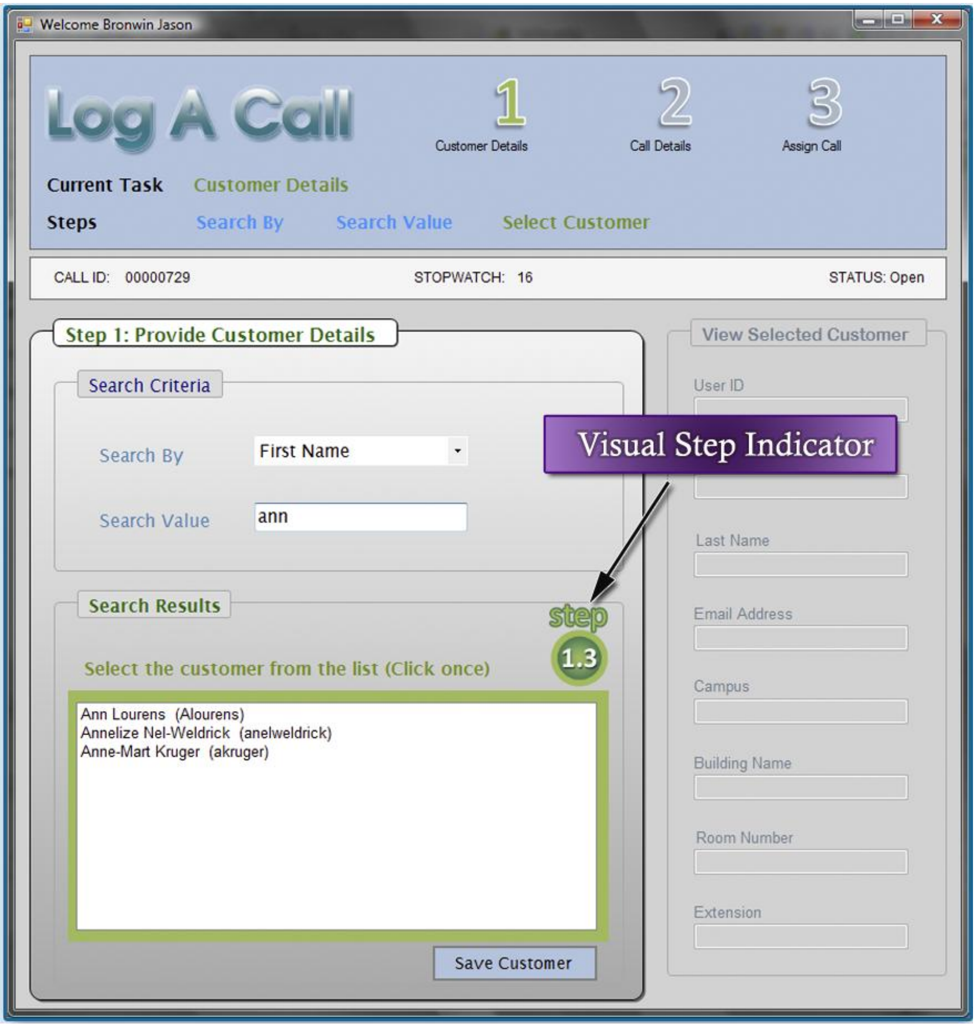


Figure 6.15: Novice UI (Step 1.3)

Selecting the customer from the list, as instructed by the novice UI ('*Select the Customer from the list (Click Once)*' in Figure 6.15), results in all three sub-steps of *Providing the Customer's Details* to be complete. This status of completion is represented in the User Input section as well as the Title and Description section and, in particular, the Visual Task Indicator which is now visually displayed as being complete (Figure 6.16). The Visual Step Indicator is absent due to the completion of all sub-steps and the user is instructed to proceed to the next call logging step by means of a **green** arrow (Figure 6.16). Hovering the mouse cursor over the button (which enables the user to proceed to the next call logging step), provides a tool tip conveying information about its use. If the user has not performed all of the sub-steps needed, an error message appears which is similar to the one depicted in Figure 6.12 (A). If the user has, however, successfully completed all necessary sub-steps, a confirmation dialogue appears which is similar to the one depicted in Figure 6.12 (B).

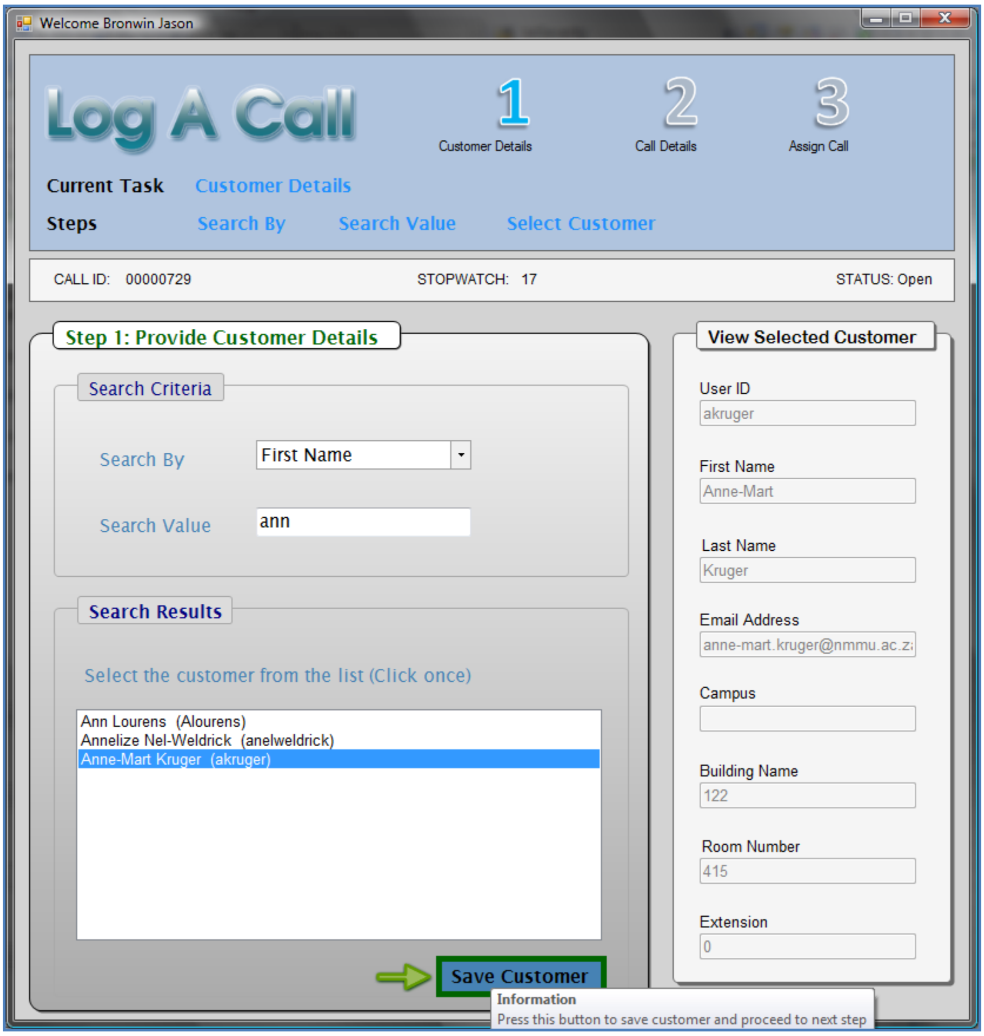


Figure 6.16: Novice UI (Step 1)

Figures 6.14 to 6.16 illustrated the Visual Step Indicator's role on the first screen of the novice UI and visualised the *Provide customer details* call logging step. The following three figures (Figures 6.17 to 6.19) are screenshots of the novice UI's remaining screens which support the remaining three call logging steps: *Providing the call's details*, *Assign the call*, and *Provide solution details*. These screens are presented to the user in a sequential manner.

Step 2: Provide the Call's Details

Figure 6.17 supports the *Providing the call's details* screen which is comprised of three sub-steps:

1. Call description;
2. Call categorisation; and
3. Call classification.

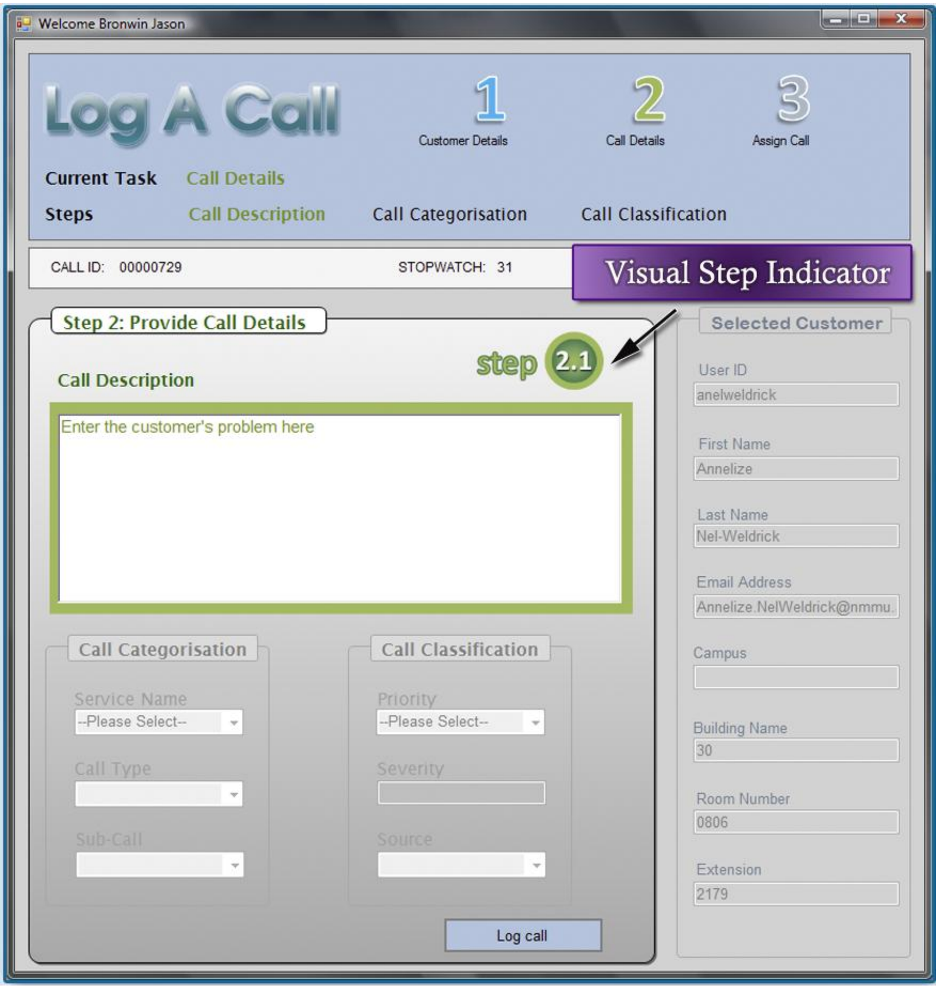


Figure 6.17: Novice UI (Step 2)

Step 3: Assign the Call

Figure 6.18 supports the *Assign the call* screen which is comprised of two sub-steps:

1. Selecting the assignee's campus location; and
2. Selecting the assignee's (contact).

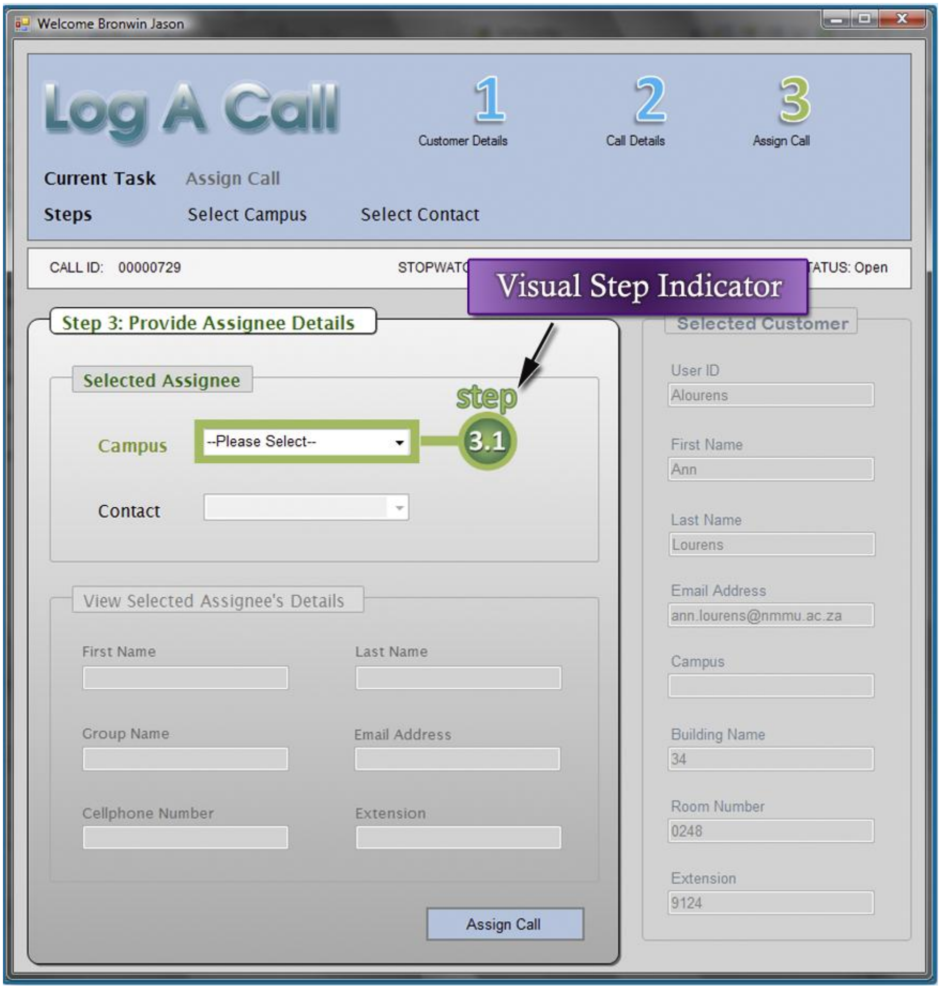


Figure 6.18: Novice UI (Step 3)

Step 4: Provide Solution Details

Figure 6.19 supports the *Providing solution details* screen which is comprised of two sub-steps:

1. Solution description; and
2. Solution classification.

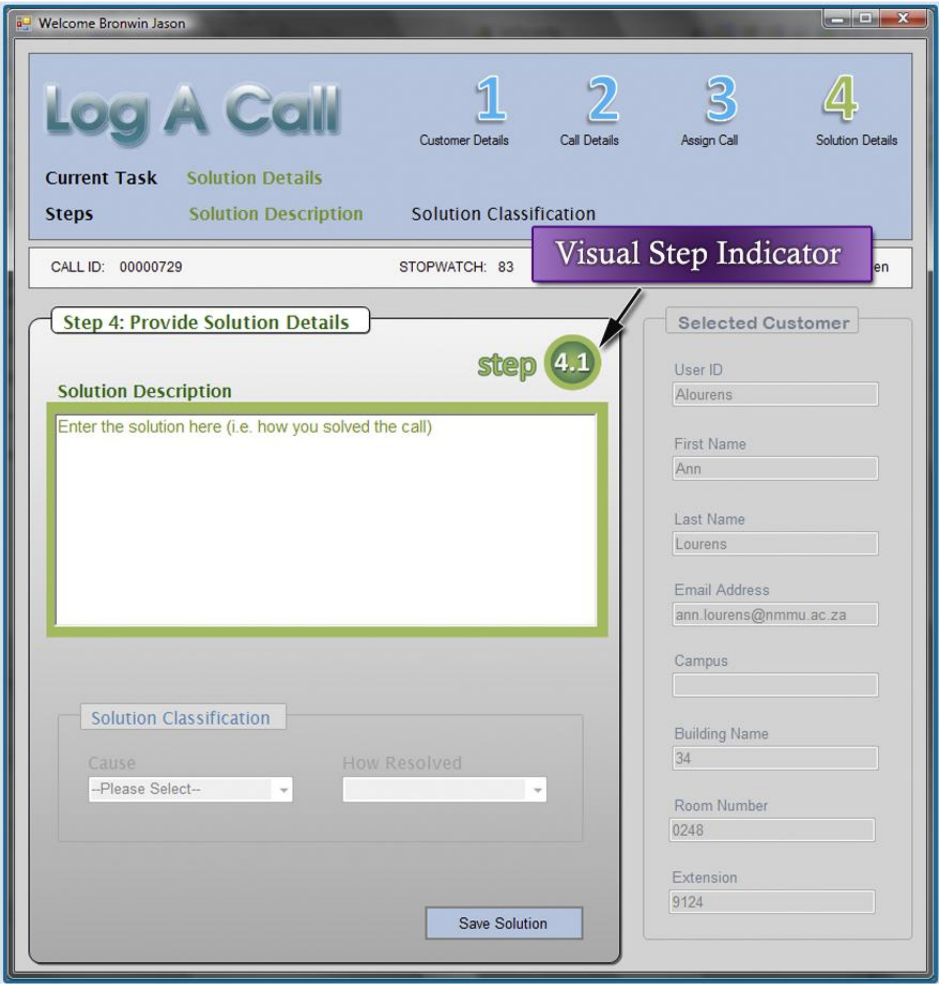


Figure 6.19: Novice UI (Step 4)

6.4.2. Expert User Interface

Expert users are more goal oriented; therefore, they require a highly efficient interface which limits the number of interactions needed (Section 3.3.1.2). Thus, the design of the expert UI entails a single screen which accomodates all the call logging steps. Figure 6.20 represents a screenshot of the expert UI where the first call logging step is located on the far left (*Step 1)* and the remaining call logging steps are tabbed within the interface (*Step 2*, *3* and *4)*. The design of the tab pages follow the same design as the individual novice screens, so that consistency is maintained.

The expert UI does not provide the user with a Visual Step Indicator, because the expert users' knowledge is more abstractly and procedurally ordered (Section 3.3.1.2). Designing UIs for expert users further supports the absence of a Visual Step Indicator by recommending avoidance of novice memory aids (Section 3.3.3.2).

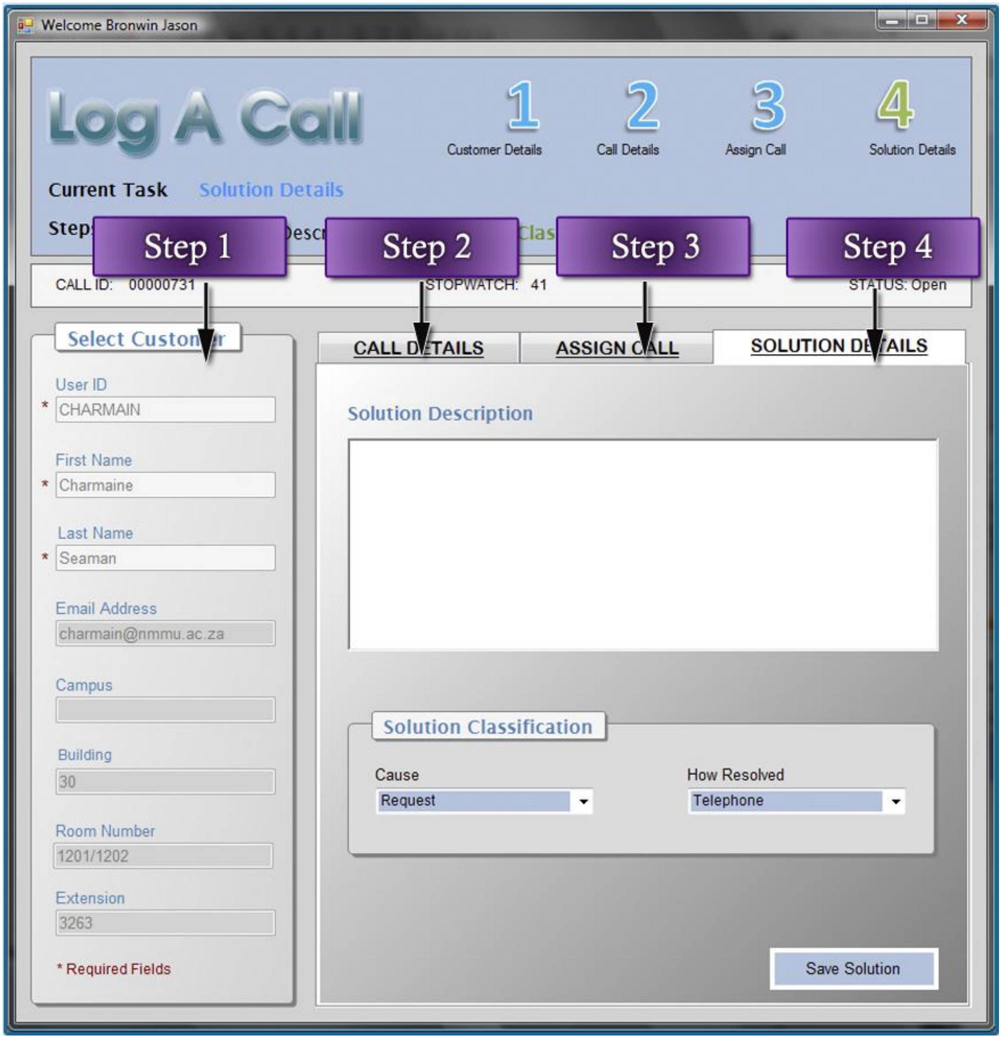


Figure 6.20: Expert UI (Step 1 – 4)

Expert users require less informative feedback (Section 3.3.3.2): therefore, unlike the novice UI, no instructions are provided on the expert UI as to what tasks need to be performed (‘*Enter User ID here*’ in Figure 6.14). The expert UI also contains less informative error messages and confirmation dialogues (Figures 6.21 (A) and 6.21 (B)).

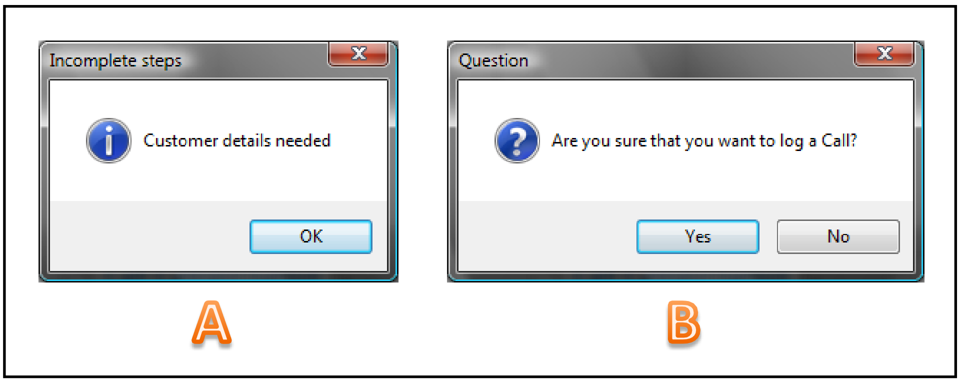


Figure 6.21: Error Message (A) and Confirmation Dialogue (B)

Experts users can be catered for by including accelerators in the interface which reduce the number of keystokes needed to enable rapid performance (Section 3.3.3.2). Accelerators are provided in the form of a pop-up list utilised when performing the *Provide customer details* call logging step and by incorporating adaptive lists.

The *Provide customer details* call logging step is simplified and provides the expert user with a faster method to search for the required customer. As the user types in either the *User ID*, *First Name* or *Last Name* text areas (Figure 6.22 (A)), a pop-up list appears which contains potential candidates (Figure 6.22 (B)) from which the user selects the required candidate. This way, selecting the required customer follows a more efficient method than the step-by-step one provided by the novice UI (Figures 6.14 to 6.16)

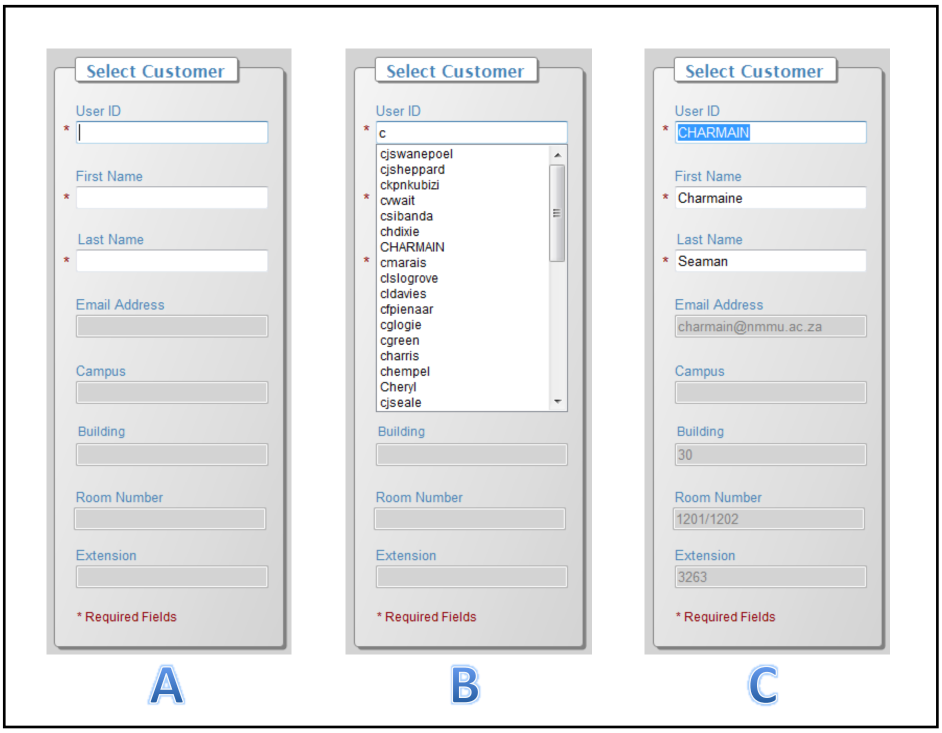


Figure 6.22: Identifying the Customer Call Logging Step

Adaptive lists are provided on the expert UI, whereby the user's three most frequently and most recently used items are selected to appear at the top of the list, termed the Adaptive Section. This is differentiated from the rest of the list by a blue background colour (*Adaptive Section* in Figure 6.23). The algorithm used for adapting the list is the Base Adaptive Algorithm (Figure 4.11). The data stored within the Knowlede Base needed to calculate the frequency, include the *List Name* (Figure 6.23), the *Item Name* (Figure 6.23) belonging to the specified list and a count of the number of times the list's item was selected. The data stored within the Knowlede Base needed to calculate the recency include the *List Name* and the name of the most recently accessed item. Adaptive lists are not encountered on the novice UI to avoid possible confusion novice users might have with the list constantly changing.

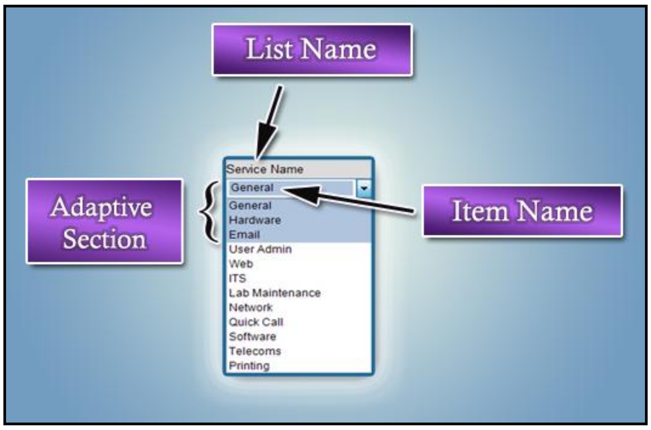


Figure 6.23: Adaptive List

Methods pertained in the *GlobalMethods* class (Figure 6.24) are responsible for implementation of the adaptive lists. The *BaseAdaptiveAlgorithm()* method of the *GlobalMethods* class is responsible for implementation of the Base Adaptive Algorithm used. After the user has made a list selection, the *UpdateFrequentlyUsedItem()* and the *UpdateOrInsertRecentlyUsedItem()* methods are responsible for updating the Knowledge Base.

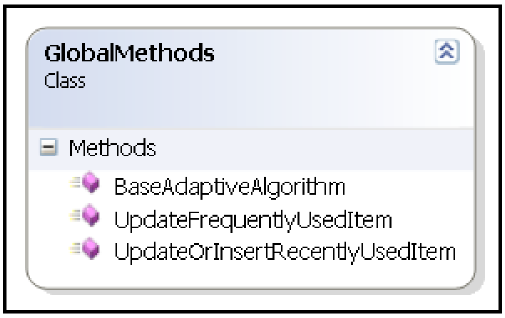


Figure 6.24: Adaptive List Methods

Figure 6.25 (A), (B) and (C) represents screenshots of the user performing a list selection. The '*Service Name*' list depicted in Figure 6.25 (A) has an Adaptive Section which initially conists of the '*General*', '*Hardware*' and '*Email*' list items. The *BaseAdaptiveAlgorithm()* is responsible for displaying these three items in the Adaptive Section of the '*Service Name*' list. After the user has slected the '*Software*' list item

(Figure 6.25 (B)), the *UpdateOrInsertRecentlyUsedItem()* and *UpdateFrequentlyUsedItem()* methods are called and the Adaptive Section is modified accordingly (Figure 6.25 (C)).

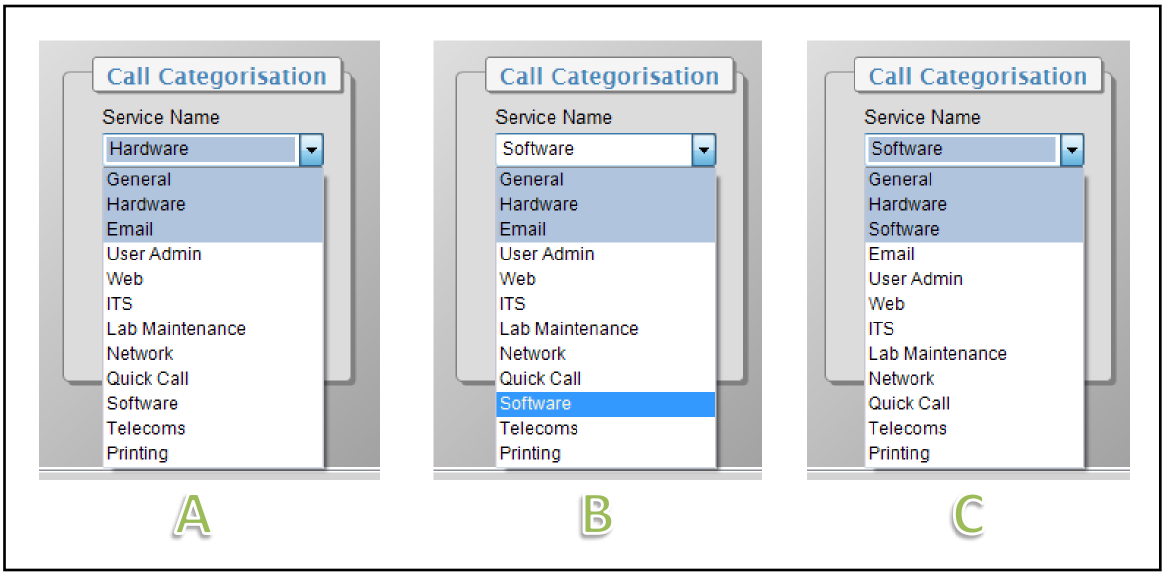


Figure 6.25: Sequence of a List Selection Made

6.5. Pilot Study

A pilot study was conducted on the ASD prototype to gather interaction data from users classified as novice and expert CCAs in order to determine the moment when a CCA using the novice UI will receive the expert UI. This moment, hereafter known as the Adaptation Moment, could only be determined by analysing if a difference in interaction data between novice amd expert users existed, and if this difference was significant enough. A secondary goal of the pilot study was to identify any potential problems before conducting the main study (Chapter 7).

The following sub-sections will investigate whether there is a significant difference between novice and expert users and how this difference could be used during implementation of the Analysis Engine. Commencement of the pilot study resumed only once a group of suitable participants was obtained. These participants needed to consist of both expert and novice users (Section 6.5.1). The procedure used to conduct the pilot study will be discussed in Section 6.5.2. Section 6.5.3 will discuss the data collection and thereafter, a discussion on the pilot study's results (Section 6.5.4) will resume. The next section will discuss further the selection of the participants used in the pilot study.

6.5.1. Participant Selection

The participants selected should be representative of the actual users (Tullis and Albert, 2008), and for this reason, all participants had a sound knowledge of the domain of information technology (IT) such that they could assist customers with IT-related queries. Figure 6.26 illustrates that 35 percent of the participants (n=8) were NMMU ICT Service Desk personnel (first-, and second-line CCAs, and a technician) and 65 percent of the participants (n=15) were students from the Department of Computer Science and Information Systems at NMMU.

The scope of this research study is limited to the NMMU ICT Service Desk (Section 1.3.3); therefore, the number of CCAs who were available for testing was limited. The NMMU ICT Service desk currently employs six CCAs (three first-line and three second-line support), however due to restructuring, seven CCAs (four first-line support and three second-line support) and a technician were available for testing. A population of only eight CC personnel was thus available for testing. All CC personnel were compensated for their participation as the usability test took them away from their working environment. A convenience sample of 15 postgraduate participants was selected from the Computer Science & Information Systems Department at NMMU. These students volunteered to be part of the usability test and were not compensated for their time.

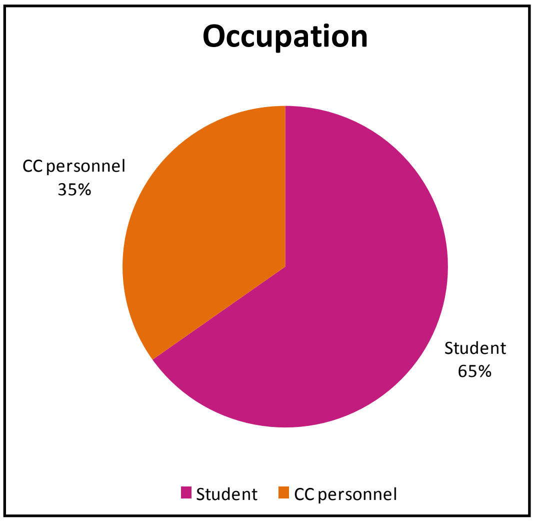


Figure 6.26: Occupational Profile of Test Participants (n=23)

Figure 6.27 illustrates the gender and age profile of the test participants. As depicted in Figure 6.27 (A), Forty-three percent (n=10) of the participants were female and the remaining 57 percent (n =13) of the sample were males. As depicted in Figure 6.27 (B), the majority (82 percent) of the participants were between 20 – 29 years old. The remaining 18 percent were between 30 – 50 years old: nine percent were between 30 – 39 years old and nine percent between 40 – 50 years.

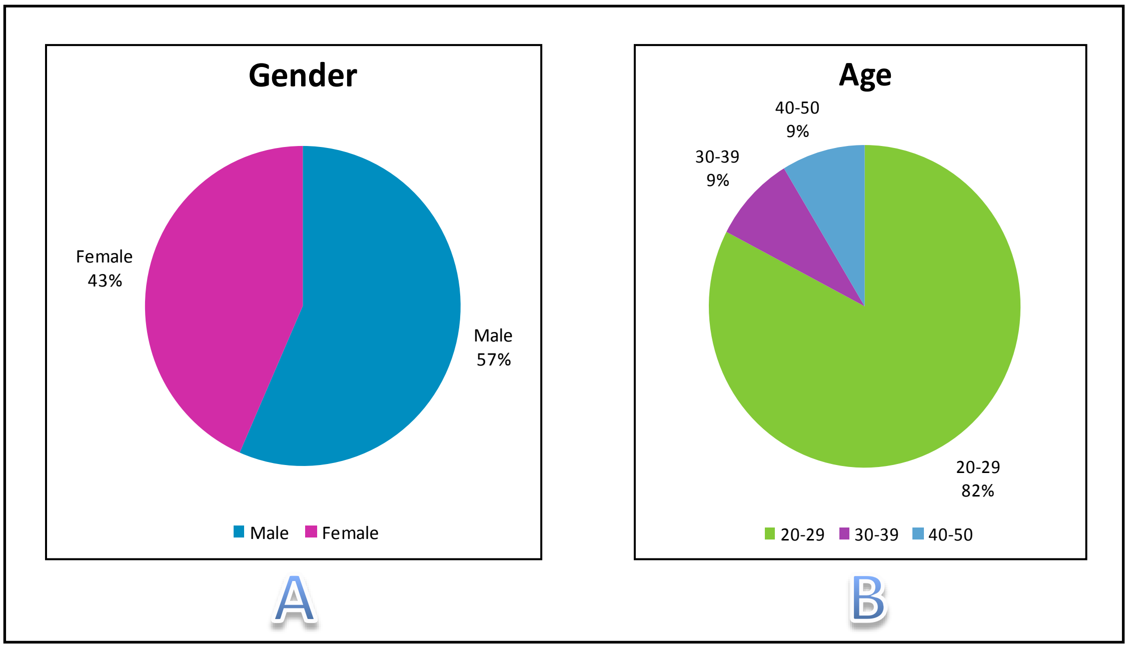


Figure 6.27: Gender (A) and Age (B) Profile of Test Participants (n=23)

A background questionnaire (Appendix A) was used to classify the participants for this evaluation as either expert or novice users. The questionnaire was divided into four parts. Section A of the questionnaire reflected the prospective participant's biographical details. Section B, C and D comprised two questions each.

Classification of user expertise was along three dimensions (Section3.3):

1. Experience with the system;
2. Experience with computers in general; and
3. Experience with the task domain (task experience).

Section B, C and D of the background questionnaire mapped to these three dimensions by reflecting the prospective participant's contact with the HEAT product suite (experience with the system), previous experience with computers (experience with computers in general), and previous experience with any other Service Desk software (experience with the task domain). Questions asked in the background questionnaire are depicted in Table 6.1.

Questions	Median
Section B: HEAT product suite	1.67
1.1. How many years have you worked with the application?	
1.2. Approximately how many days of the week do you use the application?	
Section C: Computer Experience	
2.1. How often do you work on a computer?	
2.2. How many years of computer experience do you have?	
Section D: Service Desk Experience	
3.1. How many Service Desk / Help Desk Software have you used?	
3.2. How many years experience do you have working with Helpdesk Software	

Table 6.1: Background Questionnaire Results (n=23)

The background questionnaire used a three-point Likert scale to obtain the participants' expertise results, where a 1 = "Novice" and 3 = "Expert". The median of all 23 particpants for the three dimensions of expertise was 1.67 (Table 6.1); therefore, particpants were classified as novice users if they received a score of between 1 and 1.67, and as expert users if they received a score of between 1.68 and 3. The detailed results of the background questionnaire can be viewed in the Appendix D.

Figure 6.28 illustrates the results of this classification, and, as expected, the 35 percent of the particpants classified as expert users represent the CC personnel and the 65 percent of particpants classified as novice users represent the student particpants.

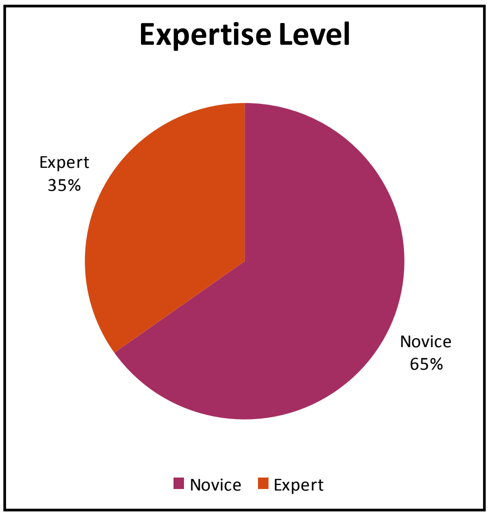


Figure 6.28: Experience Level Profile of Test Participants (n=23)

6.5.2. Procedure

User testing was conducted in the usability laboratory at the Computer Science and Information Systems (CS & IS) Department at the NMMU. The usability laboratory is a controlled environment and consists of two rooms: an observer and a participant room, separated by a one-way mirror (Figure 6.29). The laboratory was designed using the classic testing laboratory setup (Van Greunen, 2002). The two separate rooms allow for communication between the observer and testing participant without disturbing the participant.

Figure 6.29: The Participant Room (A) and the Observer Room (B) of the NMMU Usability Laboratory

One participant was tested at a time. The participant was welcomed and briefed, on arrival, about the experiment. The participant was required to complete a consent form (Appendix H). An overview (Appendix B) was given to each participant briefing them about the ASD prototype and providing them with enough CC information necessary to log a call. Participants were asked to work through a test plan (Appendix C) for each UI evaluated. This was done in the participant room.

All participants (novice and expert) were asked to evaluate both the novice and expert UIs and were to log two calls of medium complexity, chosen to mimic real-world activities. There are either three or four steps involved with logging a customer's query (Section 6.2). There are four call logging steps if the call is assigned to the CCA logging the call, and three if the call is assigned to someone else. The first task required by the particpants to log, hereafter referred to as Task 1, comprised three of the four call logging steps, and entailed a call which was assigned to someone other than the CCA logging the call. The second task required by the particpants to log, hereafter referred to as Task 2, comprised all four call logging steps, and entailed the CCA solving the call and providing its solution details. An interview conducted on 23 January 2008 with the manager of NMMU ICT Service Desk, Mrs R. Vermaak (Vermaak, 2008), revealed that newly appointed CCAs (novice CCAs) are only required to solve simple type customer queries, such as assistance with software. Task 2 was a Microsoft Word customer query and the participants were shown how to solve it. Similar to the evaluations conducted for the training wheels concept (Chapter 2), and due to the assistance provided by the novice UI, the participants were not given instructions on how to log these calls.

Novice users were initially given the novice UI and logged two calls. At a later stage, they were given the expert UI where they were asked to log two similar calls. Expert users were first given the novice UIs on which they performed the same two tasks the novice users were given. Immediately after completing the post-test questionnaire for the novice UI, they were given the expert UI and were required to log the same two calls logged by the novice users when evaluating the expert UI.

Data was only captured while both novice and expert users logged the two calls using the novice UI. This was because the adaptation transitions a user from using the novice UI to using the expert UI; thus, only data captured while the user logs calls via the novice UI would be useful. The Watcher AUI component monitored each user logging calls on the novice UI and, for each IM encountered, captured data for the PFs.

6.5.3. Data Collection

Nine IMs were used (Figure 6.1), each consisting of ten PFs which needed to be compressed into a single IM value and were expressed in different units (e.g., seconds, pixels/second, count). In order for the PFs to represent a single IM value, it was necessary to standardize the PF values. *Z*-scores are linear transformations of raw values. *T*-scores, which are linear transformation of *Z*-scores, were used for the standardization of the PF values.

T-scores, which have no units, express how far a score is from the mean in standard deviations. A *T*-score (mean = 50 and SD = 10) is calculated as:

$$T\text{-score} = 50 + 10 * (\text{z-score}).$$

The *Z*-score is calculated as:

$$Z\text{-score} = (x - \mu) / \sigma$$

Where

x = the score to be transformed (raw value)

μ = the mean of the distribution of those scores

σ = the standard deviation of the distribution of those scores

Averaging the *T*-scores together requires each of the scales to be going in the same direction. In this case, if lower values meant better, the *T*-score was multiplied by (-1) to reverse its scale. Table 6.2 depicts the various PFs and their associated direction.

	Predictive Features	Direction
1	Total Time	(-1)
2	Y Mouse Velocity	(+1)
3	Y Mouse Acceleration	(+1)
4	Dwell Time	(-1)
5	Average Dwell Time	(-1)
6	Nr Items Visited	(-1)
7	Unique Items Visited	(-1)
8	Selection Time	(-1)
9	KLM Difference	(-1)
10	KLM Ratio	(+1)

Table 6.2: PFs and Associated Direction

An IM *T*-score comprising the ten PFs had to be computed once the PF *T*-scores were computed from the raw data values and appropriate directions given. The IM *T*-score for each participant was calculated as follows:

IM *T*-score = weighted mean of PFs = $\sum w_i T_i / (\sum w_i)$

Where

w_i = weight for particular PF

T_i = *T*-score for particular PF

i = 1 to 10 (there are 10 PFs)

As depicted above, the IM *T*-score was calculated by applying weights to the PF *T*-score values. The PFs were first ranked according to importance. The rankings were determined in an interview conducted on 21 August 2008 with a statistician, Mr D. Venter (Venter, 2008). In this interview, he suggested that the weights of the PFs be computed by inversing the PF value's rank. The rankings and weights of the PF values can be seen in Table 6.3 below. The IM *T*-Scores for all participants for Task 1 and Task 2 can be seen in Appendix D.

	Predictive Features	Rank	Weight
1	Total Time	1	1
2	Y Mouse Velocity	9	0.11
3	Y Mouse Acceleration	9	0.11
4	Dwell Time	2	0.5
5	Average Dwell Time	3	0.33
6	Nr Items Visited	6	0.167
7	Unique Items Visited	7	0.143
8	Selection Time	3	0.33
9	KLM Difference	5	0.2
10	KLM Ratio	8	0.125

Table 6.3: PFs and Associated Weights

Correlations are used as a measure of the strength of a relationship between two variables and have a range from -1 to +1. The stronger the relationship, the closer the value is to -1 or +1, and the weaker the relationship, the closer the correlation coefficient is to 0. Correlations were used to measure the strength of the relationship between the PF *T*-scores and their associated IM's *T*-scores. The relationship between the PF *T*-scores and their associated IM *T*-score was reasonably strong due to most correlation coefficients being closer to +1 and it was a positive relationship (as the PF value increases, so does the IM value). There were a few weak relationships.

The *Search Customer* IM within the first call logged had a weak relationship between the *Unique Items Visited* PF and also between the *KLM Difference* PF, as the correlation coefficients were -0.195 and -0.254 respectively. The negative correlation coefficient also indicated a negative relationship. The *Search Customer* IM within the second call logged also had a weak relationship between the *Nr Items Visited* PF, as the correlation coefficient was -0.007. The *Search Customer* IM was thus given a low ranking when computing the total *T*-score for each participant. The *Source* IM within the second call logged had a weak relationship between the *Nr Items Visited* PF, as the correlation coefficient was -0.145. The *Source* IM was thus also given a low ranking when computing the total *T*-score for each participant. The correlations between the IMs and their respective PFs generally, however, indicated strong relationships.

Similiar to the manner in which the ten PFs were consolidated into a single IM value, a final score was computed for each particpant comprising a weighted average of the IM *T*-scores and the total task time *T*-score. The total task time represents the time from when the call logging form is first opened to the time the user completes the process of logging a call. This is not the same as the PF, total time. A *T*-score was computed for the total task time and this was computed in the (-1) direction, meaning a lower value meant better.

Weights were thus applied to the nine IMs and the total task time by inversing their respective rankings. The rankings were determined in interviews conducted on 25 August 2008 with a statistician, Mr D. Venter (Venter, 2008) and on 23 January 2008 with the manager of NMMU ICT Service Desk, Mrs R. Vermaak (Vermaak, 2008). The rankings and weights of the IM values and the total task time can be seen in Table 6.4 below.

	Informative Moments and Total Task Time	Rank	Weight
1	Total Task Time	1	1
2	Search Customer	8	0.125
3	Service Name	2	0.5
4	Call Type	2	0.5
5	Priority	9	0.111
6	Source	10	0.1
7	Campus	5	0.2
8	Contact	5	0.2
9	Cause	4	0.25
10	Resolved	5	0.2

Table 6.4: IMs and Associated Weights

The final *T*-scores for all participants for Task 1 and Task 2 can be seen in Appendix D. The final *T*-scores, hereafter known as the overall performance data, of all 23 participants were used to compare the differences between the novice and expert users. The descriptive statistics results for these two groups will now be discussed.

6.5.4. Results

Literature suggests that novice and expert users' behaviour differs dramatically at the physical level of interaction (Section 3.3.2.2). Consultation with Venter (Venter, 2008) on 25 August 2008, suggested the use of a Mann-Whitney non-parametric test, to determine whether there was a significant difference in performance data between the novice and expert CCAs.

A Mann-Whitney test is utilised when the design is a between-groups' design with one independent variable and two levels (Gliner and Morgan, 2000). A between-groups' design is used to compare results for different participants (Tullis and Albert, 2008), which is ideal for this research study, since it compares differences in performance data between novice and expert users (two levels). An independent variable of a study is an aspect which could be manipulated (Tullis and Albert, 2008), and in this research study, only one independent variable is identified, namely the user's expertise level.

A Mann-Whitney test analyses ordinal data from two independent samples that do not necessarily have a normal distribution (Gliner and Morgan, 2000). The performance data used in this research is classified as ordinal data. A Mann-Whitney test is particularly useful when the sample sizes are small and the group sizes are uneven (Pett, 1997). This research study consists of two small uneven sample sizes (eight expert users and 15 novice users). A Mann-Whitney test, as suggested by Venter, was thus utilised.

In order to determine whether there was a significant difference in performance data between the novice and expert CCAs, the following null hypothesis was tested using a Mann-Whitney test:

H_o: There is no difference in interaction data between novice and expert CCAs.

A Mann-Whitney test was conducted in the data analysis software package, STATISTICA V8.0. A level of significance of 0.05 was used to avoid a type I error, which occurs if a hypothesis is wrongly rejected. A significance level of 0.05 (5%) indicates that with a 95 percent confidence level, a hypothesis that is actually correct, will not be rejected. The null hypothesis will be rejected if the p-value falls within the lower five percent of possible values. The p-value (number between 0 and 1) reflects the strength of the data used to evaluate the null hypothesis. A small p-value indicates strong evidence against the null hypothesis (Rumsey, 2003).

The significant results of the Mann-Whitney test for the overall performance data of the novice and expert users are depicted in Table 6.5. Detailed results of the Mann-Whitney test can be viewed in Appendix D. Based on the results of the Mann-Whitney test, there are significant differences between novice and expert users for both Task 1 and Task 2, due to the p-value being 0.00 for both tasks, which is less than the significant level of

0.05. The null hypothesis can thus be rejected with a 95 percent confidence level; therefore, it can be concluded that:

There is a difference in interaction data between novice and expert CCAs.

	Overall Performance Data			
	Novice Users (N=15)	Expert Users (N=8)		
	Rank Sum	Rank Sum	U	p-value
Task 1	128	148	8	.000
Task 2	126	150	6	.000

Table 6.5: Mann-Whitney Test Results

ASD starts CCAs with the capability to log a customer's query via the novice UI until they have become skilled. When a CCA has reached a level which is considered to be that of an expert user, he or she is transitioned from the novice UI to the more appropriate expert UI. It was shown that a significant difference exists between novice and expert users. This encourages a shift from the novice to the expert UI. The shift can, however, only be determined by analysing the overall performance data of novice and expert users in order to identify a possible transitioning period.

Figure 6.30 illustrates the percentage of users (novice and expert) for Task 1, whose overall performance data fall within a given category. Figure 6.30 shows that 60 percent of novice users obtained an overall performance *T*-score of between 0 and 49.99. Among novice users, 33.33 percent obtained an overall performance *T*-score of between 50 and 54.99 and only 6.7 percent performed with a *T*-score larger than 55. Contrasting to the novice users' performance, none of the expert users obtained an overall performance *T*-score of between 0 and 49.99. Expert users obtained *T*-scores larger than 50, where 25 percent obtained an overall performance *T*-score of between 50 and 54.99 and the majority (75%) expert users obtained *T*-scores larger than 55.

Figure 6.31 illustrates the percentage of users (novice and expert) for Task 2, whose overall performance data fall within a given category. Figure 6.31 shows that 73 percent of novice users obtained an overall performance *T*-score of between 0 and 49.99. 26.7% Novice users obtained an overall performance *T*-score of between 50 and 54.99 and no novice users performed with a *T*-score larger than 55. Similar to the expert users' performance in Task 1, none of the expert users for Task 2 obtained an overall performance *T*-score of between 0 and 49.99. The expert users obtained *T*-scores larger than 50, where half (50%) obtained an overall performance *T*-score of between 50 and 54.99 and the other half (50%) obtained *T*-scores larger than 55. The overall performance data for expert users for both Tasks 1 and 2 always exceeded a *T*-score of 50.

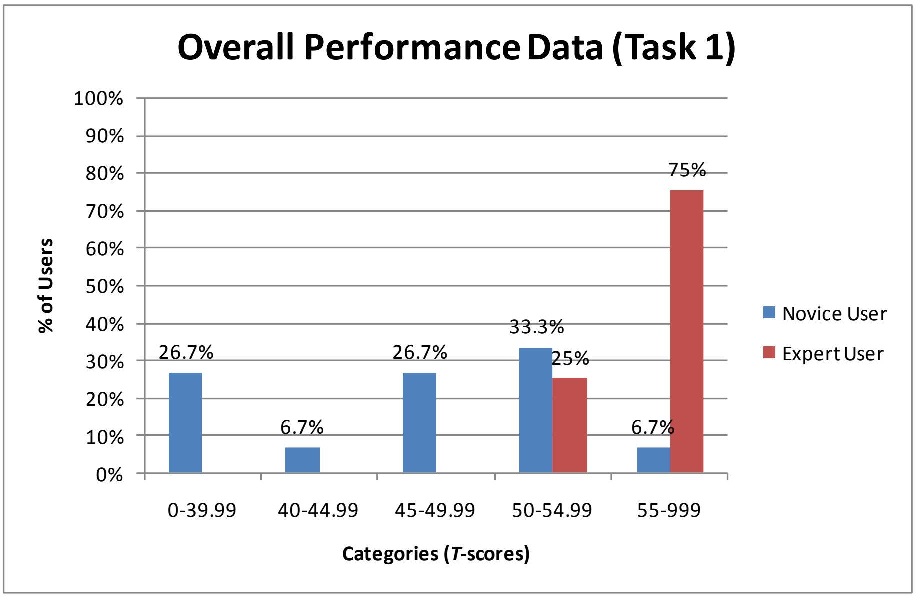


Figure 6.30: Overall Performance Data for Task 1 (n=23)

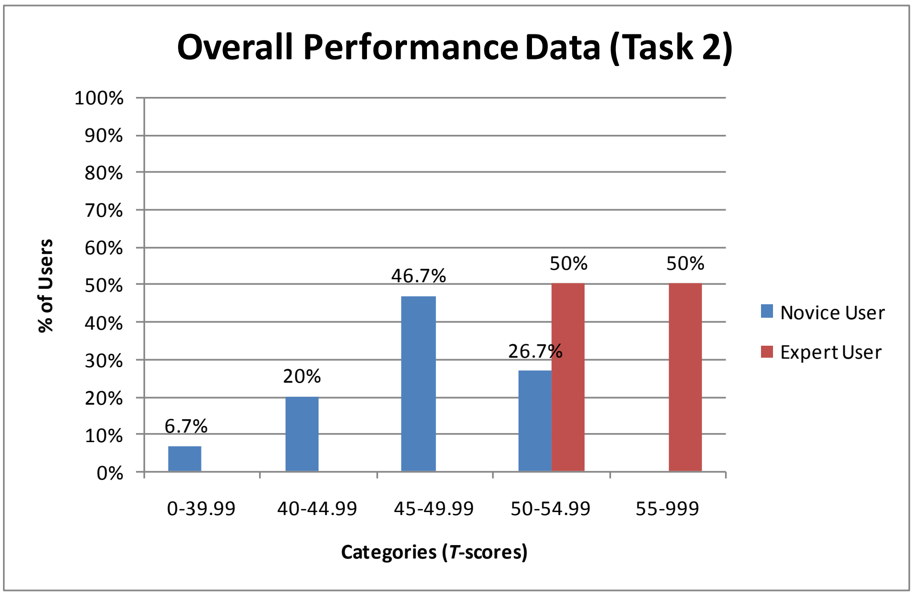


Figure 6.31: Overall Performance Data for Task 2 (n=23)

Statistics such as means, medians, standard deviations and quartile values of the overall performance data were calculated for both the novice and the expert user group. These statistics can be viewed in Table 6.6 and 6.7. There is quite a difference in overall performance means between the novice and expert users for both Task 1 and Task 2. Expert users had means of 54.88 and 55.09 for Task 1 and Task 2 respectively with small standard deviations (1.04 and 2.51). These deviations indicate a small amount of variability in the expert user group. Novice users had means of 47.4 and 47.29 for Task 1 and Task 2 respectively with relatively larger standard deviations (6.14 and 3.98). These deviations reflected a larger amount of variability in the novice user group.

Table 6.6 shows that expert users' minimum *T*-score for the overall performance data for both Task 1 and 2 were 52.46 and 52.17 respectively. Novice users' upper quartile (Quartile 3 in Table 6.7), which lies three quarters of the way through the values, for both Task 1 and Task 2, were 51.99 and 50.18, which does not even equate to expert user's minimum values. Figure 6.30 and 6.31 previously showed that expert users' overally performance for both Tasks 1 and 2 always had a score greater than 50. A suggestion provided by Venter (Venter, 2008), on 27 August 2008, was that the expert users' lower quartile (*Quartile 1* in Table 6.6) be used as the Adaptation Moment (i.e. if a user performs at scores greater than the expert user's quartile 1 scores, the user transitions to the expert UI). However, Venter did advise against using the expert users' mean scores for the Adaptation Moment, due to the possibility that novice users might take a long time to perform, if ever, at those scores. The mean value for expert users for Task 1 was, however, 54.88, which is close to the expert users' quartile 1 value (54.95).

The research study decided to use an integer value of 52 (expert users' minimum value for Tasks 1 and 2) as the Adaptation Moment. This value was decided upon using advice given by Venter (Venter, 2008) and by analysing the statistics of the overall performance data of both the novice and expert user groups. The chosen Adaptation Moment provides novice users with an opportuinty to reach a higher skill level and obtain the expert UI; however, the disdvantage is that the switch to the expert UI might be too quick. The evaluation of the switch and the users' satisfaction results in the main study will be investigated in Chapter 7. The next sub-section will discuss the implementation of the Analysis Engine, which provides the switch from the novice UI to the expert UI.

	Expert Users (n=8) Overall Performance Data	
	Task 1	Task 2
Mean	54.88	55.09
StdDev	1.04	2.51
Min	52.46	52.17
Quartile 1	54.95	52.48
Median	55.20	55.27
Quartile 3	55.42	57.53
Maximum	55.67	57.88

Table 6.6: Descriptive Statistics for Expert Users (n=8)

	Novice Users (n=15) Overall Performance Data	
	Task 1	Task 2
Mean	47.40	47.29
StdDev	6.14	3.98
Min	37.13	39.49
Quartile 1	41.18	44.81
Median	48.71	47.81
Quartile 3	51.99	50.18
Maximum	55.37	53.04

Table 6.7: Descriptive Statistics for Novice Users (n=15)

6.6. Analysis Engine

The Analysis Engine uses the User Model which is stored in the *Knowledge Base* to derive new user information and then updates the User Model with this new information (Section 5.3.1.2). The Analysis Engine is responsible for determining when the CCA will receive the expert UI based on the CCA's performance whilst using the novice UI. The CCA's performance data which the Analysis Engine analyses is obtained through the Watcher component.

The *AnalyzeSkillLevel* class (Figure 6.32) is responsible for determining whether a novice user has gained enough skill using the novice UI to begin logging calls via the expert UI. This Adaptation Moment was determined based on results obtained in the pilot study (Section 6.5).

Consultation with Venter (Venter, 2008), on 27 August 2008, suggested that the user's overall performance score for a task be calculated in the same manner as the final *T*-score was calculated in Section 6.5.3. *T*-scores, however, require a mean and standard deviation of scores and an interview with another statistician, Mr. J. Pietersen (Pietersen 2008), on 15 September 2008, advised incorporating the expert users' mean and standard deviation from the pilot study with the user's mean and standard deviation so that the user's overall performance score could be compared to that of expert users. The previous sub-section mentioned that this research study decided to use an integer value of 52 (the minimum value) as the Adaptation Moment. The rule employed to transition a user from a novice UI to a expert UI is:

If (*T*-score of all the user's tasks with a solution) > 52 and
If (*T*-score of all the user's tasks without solution) >52
Then User expertise Level = Expert

The *getArrayUserModelComboBoxXML()*method (Figure 6.32) is responsible for obtaining the user's User Model from the Knowledge Base and the *getArrayUserModelComboBoxXMLExperts()* (Figure 6.32) method is responsible for obtaining the expert users' User Models from the Knowledge Base. The *CalcFinalT()* method (Figure 6.32) calculates the user's overall performance scores for all tasks completed by the user and the *isExpert()* method (Figure 6.32) uses the rule above to determine whether the user's expertise level is that of an expert. Section 4.7.1 discussed three timing strategies for when the adaptation could take place. This research utilises the *between sessions* timing strategy as it facilitates complex adaptation strategies. After a user has logged a call, the *AnalyzeSkillLevel* class is called. If the user is identified as behaving like an expert, the next call logged by the user transpires on the expert UI.

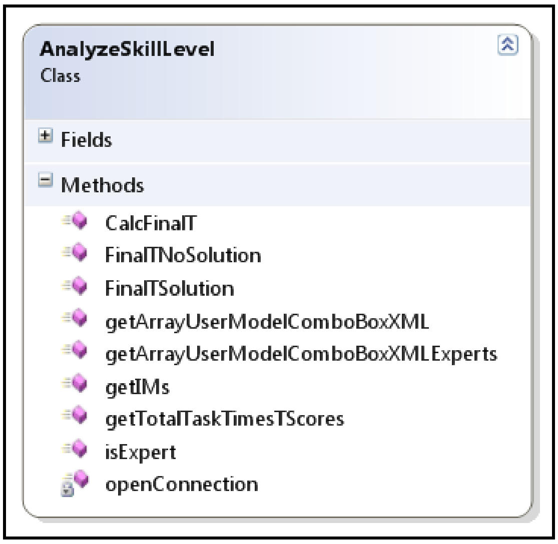


Figure 6.32: AnalyzeSkillLevel Class

6.7. Summary

This chapter illustrated the successful implementation of the proposed AUI model defined in Chapter 5. Implementation of the proposed model required implementation of the AUI components: namely, the Knowledge Base, the Watcher, the Adaptation Effect, and the Analysis Engine.

The User Model and Task Models, stored in the Knowledge Base, were updated by implementing of the Watcher component. The Knowledge Base was implemented using SQL Server Management Studio and XML. The Adaptation Effect AUI component provided implementation of the two types of UIs; namely, the novice UI and the expert UI. These two types of UIs were implemented to support the call logging process.

The novice UI entailed a separate screen for each of the call logging steps, guiding the user step-by-step through the process of logging a call. A Visual Step Indicator provided the user with even more guidance on the novice UI. The expert UI entailed a single screen accommodating all call logging steps tabbed into the single screen and additionally facilitated the user with a more efficient interaction by incorporating adapted lists.

A pilot study was conducted on the ASD prototype to gather interaction data from users classified as novice and expert CCAs in order to determine the moment when a CCA using the novice UI will receive the expert UI. The pilot study classified participants as expert (n=8) and novice (n=15) users. Both user groups were given two tasks to log via the novice UI. A Mann-Whitney test concluded that there was a significant difference between the novice and the expert users. These differences were analysed and this resulted in the implementation of the transitioning rule.

Implementation of the Analysis Engine consisted of a simple rule, which stated that a user will be provided the expert UI if the user's performance data reaches that of the expert users. The rule implemented by the Analysis Engine was achieved through a pilot study.

The adaptation provided by the proposed AUI model is, however, only successful if the user is satisfied with the adaptation. The next chapter investigates the evaluation of the proposed AUI model to determine its effectiveness.

Chapter 7: Main Evaluation and Results

7.1. Introduction

Evaluation constitutes an important stage in the development of AUIs (Gena, 2005). The benefits of AUIs can only be realised through user testing. Chapter 6 provided a discussion on the implementation of the AdaptiveServiceDesk (ASD) prototype as a proof-of-concept for the adaptive user interface (AUI) model proposed in Chapter 5. Results obtained from the pilot study conducted in Chapter 6 were utilised to provide the adaptation of the ASD prototype. This adaptation, however, is only successful if users are satisfied with it and if it improves the user-system interaction. Usability evaluation becomes necessary to determine the usefulness of the ASD prototype and to investigate the user's interaction with it.

The purpose of this chapter is to discuss the evaluation of the proposed AUI model and the ASD prototype. Results from the pilot study (Chapter 6) will be used to conduct the usability evaluation. The evaluation involves determining the extent to which the proposed AUI model supports the development of an AUI for contact centres (CCs) and the extent to which the proposed AUI model facilitates a novice contact centre agent (CCA), using a graphical user interface (GUI), to become more experienced in logging customer's queries. The novice UI provided for novice CCAs was expected to provide them with the assistance and interface training necessary to gain enough experience to use the expert UI. Expected results from the usability evaluation on the ASD prototype were improvements in users' performances.

This chapter will commence with a discussion on existing AUI evaluation approaches (Section 7.2). There has been a lack of strong theories, models and laws with regards to evaluation of adaptive learning systems (Weibelzahl, 2005); therefore, the discussion on existing AUI evaluation approaches resulted in an evaluation strategy of the proposed AUI model (Section 7.3). Usability testing and a discussion on the key results obtained for this research study are provided in Section 7.4.

7.2. Existing AUI Evaluation Approaches

AUIs are comprised of many elements and these are specified as the AUI components of the proposed AUI model (Section 5.3). The proposed AUI model contains a data component (Knowledge Base and Watcher), a decision-making component (Analysis Engine) and an interface component (Adaptation Effect). Evaluation of AUIs cannot, therefore, be treated as a single process but should rather identify the various components of an AUI and evaluate them separately, where necessary and feasible (Weibelzahl, 2005). Layered evaluation approaches have, therefore, been proposed to evaluate AUIs.

Layered evaluation approaches have been proposed by many researchers (Karagiannidis and Sampson, 2000, Brusilovsky *et al*., 2001, Paramythis *et al*., 2001, Weibelzahl, 2002) and entail separately evaluating the different components (aspects) of AUIs. Layered evaluation uniquely differentiates the evaluation of AUIs from the evaluation of regular systems. The concept of the layered approach was initially proposed by Totterdell and Rautenbach (1990), who suggested that different adaptation metrics should be related to different components of a logical model of AUIs. Totterdell and Rautenbach (1990) identified two basic layers that have to be evaluated separately (Gena, 2005):

1. *The Content Layer*: assessing the accuracy of the inferences made; and
2. *The Interface Layer:* assessing the effectiveness of the interface changes.

Ongoing research on layered evaluation for AUIs ensued, following Totterdell and Rautenbach's (1990) initial proposal. Previous researchers (Karagiannidis and Sampson, 2000, Brusilovsky *et al*., 2001, Paramythis *et al*., 2001, Weibelzahl, 2002) have identified different levels of granularity (layers), and depending on the layers identified, different evaluations and analysis have to be taken into account. In particular, Weibelzahl (2002) identified four evaluation layers:

1. *Evaluation of Input Data*

   AUIs involve making inferences about the user. These inferences can only occur if based on data contained in the User Model. The reliability and validity of the input data is highly important. The empirical test theory is a possible method to test the reliability of the data.

2. *Evaluation of Inference*

   The inference can be based on simple rule algorithms to more complicated Bayesian networks. Evaluation of Inference checks the 'correctness' of the system's assumptions. A method to possibly evaluate the inference would be to compare the system's assumptions about the user with expert ratings.

3. *Evaluation of Adaptation Decision*

   Given the same user properties, adapting the interface results in several possibilities. Evaluation of Adaptation Decision determines whether the chosen decision is the optimal one. A method to possibly evaluate the adaptation decision would be to compare the alternative decisions.

4. *Evaluation of Total Interaction*

   Evaluation of Total Interaction involves conducting a summative evaluation of the whole system. The frequency of adaptation and other dimensions of system behaviour are evaluated. Most importantly, this evaluation layer evaluates user behaviour by assessing both task success and usability.

The High-level Model of Adaptation (Section 5.2.2.1) to the evaluation of AUIs, proposed by Paramythis *et al.* (2001), involved a modular approach: identifying evaluation modules (which comprised one or more of their proposed model's adaptation components) (Section 5.2.2.1), and then evaluating them individually and/or in combination (Paramythis *et al*., 2001).

This section has identified the importance of the layered approach when evaluating AUIs. Different layers have been proposed by various researchers. Paramythis *et al.*, in particular, proposed a modular approach to the evaluation of their proposed AUI model. In view of previous evaluation approaches, the next section will discuss the evaluation of the proposed AUI model.

7.3. Evaluation of Proposed AUI Model

Evaluation of the proposed AUI model (Section 5.3) involves a modular approach which incorporates evaluation layers proposed by Weibelzahl (2002). These evaluation layers can be evaluated individually and in combinations. Figure 7.1 depicts the evaluation layers as they correspond to the AUI components of the proposed AUI model. The evaluation layers of the proposed model are:

- *Data Layer*

  The Data Layer comprises the Knowledge Base and the Watcher AUI components (Figure 7.1). The Data Layer maps directly to Weibelzahl's (2002) *Evaluation of Input Data* layer and involves evaluating the validity of the user's input data.
- *Inference Layer*

  The Interface Layer comprises the Analysis Engine AUI component (Figure 7.1). The Interface Layer maps directly to Weibelzahl's (2002) *Evaluation of Inference* layer and involve evaluating the validity of the inference made.
- *User Interaction Layer*

  The User Interaction Layer comprises the Adaptation Effect AUI component (Figure 7.1). The User Interaction Layer maps directly to Weibelzahl's (2002) *Evaluation of Total Interaction* layer and involve evaluating the usability of the AUI.

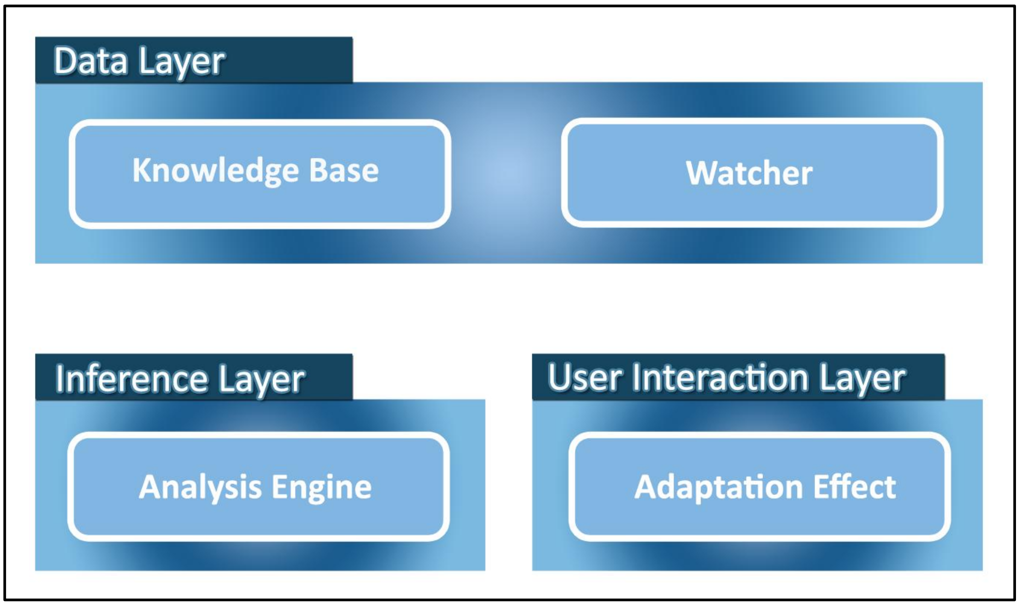


Figure 7.1: The Correspondence between Evaluation Layers and AUI Components of Proposed Model

Evaluation of the Data Layer was performed in the pilot study, which was discussed in Section 6.5. A scenario-based prototype evaluation technique was used to evaluate the Data Layer. Novice and expert users were given tasks to log via the novice UI. This section concluded that the data contained attributes that were reflective of a user's skill level and could thus be inferred. The validity of the data was verified.

Evaluation of the Data Layer serves as a prerequisite for evaluation of the Inference and User Interaction Layer. This Layer was evaluated in combination and required user involvement and a fully functional ASD prototype. Evaluation, therefore, consisted of a usability testing (Section 7.4). Eye-tracking (Section 7.4.6.4) was used to aid the user testing. Instruments used for the evaluation of the Inference and User Interaction Layer were a test plan and a post-test questionnaire.

7.4. Usability Testing

The International Standards Organisation (ISO) defines usability as (ISO 9241-11, 1998):

> *"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use."*

According to this definition, the three most important aspects of usability are effectiveness, efficiency and satisfaction (Tullis and Albert, 2008). Satisfaction is investigated to determine the extent to which the ASD prototype fulfils the user's needs. Efficiency is investigated in terms of learnability, which measures how much efficiency increases over time. Eye-tracking is used to measure effectiveness. Accuracy, which is not formally specified in the definition of usability, is investigated to determine how accurately the adaptation is performed.

7.4.1. Usability Goals and Metrics

The goal of the user testing is specified in terms of the Inference and User Interaction evaluation layers. The goal of the Inference Layer is to determine whether the adaptation timing is accurate. The goal of the User Interaction Layer is to determine whether users are satisfied with the adaptation and to assess their performance.

Accuracy (Section 7.4.6.1), satisfaction (Section 7.4.6.3) and performance are discussed in this evaluation. Performance is discussed in terms of learnability (Section 7.4.6.2) and effectiveness (Section 7.4.6.4). The following section will discuss the usability metrics used in this evaluation to achieve the usability goals. In terms of their corresponding evaluation layers, these were:

Inference Layer

- **Accuracy**
  - *Precision*: Precision is a metric derived from information retrieval and information filtering systems, where relevant contents are selected from a set of contents. For the evaluation of information filtering systems, this metric has been used to measure the accuracy of relevant contents. AUIs have exploited the precision metric to evaluate the accuracy of recommendations and system predictions, etc. (Gena, 2005). Precision in this research study has, therefore, been used to determine the degree of accuracy of the adaptation made in order for a user to obtain the expert UI. Precision was measured as the ratio of relevant adaptations (N_{ra}) to the number of adaptations made (N_a).

$$precision = \frac{N_{ra}}{N_a}$$

The above formula was adopted from Herlocker *et al.* (2004).

User Interaction Layer

- **Learnability**
  - *Number of tasks to learn:* This metric measured the amount of tasks that participants performed on the novice UI until they received the expert UI.
  - *Time to learn*: This metric measured the time that participants spent on the novice UI.
  - *Task completion time:* This metric measured the total time that participants spent performing tasks on the novice UI and the expert UI.

- **Satisfaction**
  - *Self-reported metrics:* Satisfaction was measured by collecting self-reported data from users about their perceptions of the system and their interaction with it. A post-test questionnaire, using a five-point Likert scale, was used to obtain the self-reported data. A rating scale is the most efficient way to capture self-reported data in a usability test (Tullis and Albert, 2008). A typical five-point Likert scale is:

1. Strongly disagree
2. Disagree
3. Neither agree nor disagree
4. Agree
5. Strongly agree

- **Effectiveness**
  - *Eye-tracking metrics:* Effectiveness was measured by analyzing eye-tracking data. The following section (Section 7.4.2) will further discuss the eye-tracking metrics which were used.

7.4.2. Eye-Tracking Metrics

Eye-tracking measures the user's eye movements enabling the researcher to know where the user is looking (Poole and Ball, 2005) and can be extremely useful in a usability test (Tullis and Albert, 2008). They provide an objective source of interface-evaluation data which could help improve the design of interfaces (Poole and Ball, 2005). Eye-tracking data can complement the data obtained in a usability study as it provides more specific information about the user's cognitive process (Pretorius, 2005). Eye-tracking was, therefore, included as part of user testing.

One of the most common analysis methods used in eye-tracking is the area of interest (AOI). The AOI is defined as a part of the interface which is of interest to the research study. A wide variety of eye-tracking metrics are used to analyse identified AOIs (Jacob and Karn, 2003); however, the eye-tracking metrics deemed useful for the purposes of this usability study, as suggested by Pretorius (2005) and Singh (2007), were:

- The fixation duration – time spent looking at an AOI;
- The observation count of an AOI; and
- The time to the first fixation on an AOI.

The fixation duration was used to determine the amount of time participants spent looking at a particular AOI. Longer fixation duration could indicate that the participant experiences difficulty with the identified AOI or that the AOI is of great interest to the participant (Poole and Ball, 2005). Heat maps are commonly used to visualise a series of fixations by multiple participants (Tullis and Albert, 2008). The observation count is indicative of the number of visits of an AOI and could be useful to determine whether participants noticed certain AOIs. The time to the first fixation was used to determine how long it took participants to notice the Visual Step Indicator component of the novice UI.

7.4.3. Instruments

A test plan overview was drawn-up for the evaluation (Appendix E) detailing the goals and overview of the evaluation. This test plan overview also provided the participants with information necessary to log calls. In addition to the test plan overview, two separate test plans were drawn-up, hereafter known as the novice UI test plan and the expert UI test plan. Contrary to the test plan overview, the novice UI test plan and the expert UI test plan were not given to the participants but drawn-up for the sole purposes of the evaluator. The next paragraph will motivate why this method was chosen.

An interview conducted on 23 January 2008 with the manager of NMMU ICT Service Desk, Mrs. R. Vermaak (Vermaak, 2008), revealed that the majority of calls logged by CCAs at the NMMU ICT Service Desk are telephone calls. A simulated CC environment was created whereby the evaluator took the role of the customer and the participant the role of a CCA. The novice UI test plan and the expert UI test plan were, therefore, only intended for the evaluator as they consisted of customer problems which the evaluator reiterated to the participants to log. These customer problems were taken from actual calls logged at the NMMU ICT Service Desk.

The novice UI test plan was intended for logging calls via the novice UI, and comprised eight tasks. All participants were required to log at least four of the eight calls. Tullis and Albert (2008) suggest having at least three or four trials when measuring the learnability of a system. The Adaptation Moment (the moment when the participant transitions from the novice UI to the expert UI) was only possible after the participant had logged at least four calls. The number of calls which resulted in an Adaptation Moment depended on the participant’s performance. If participants did not encounter an Adaptation Moment after logging eight calls they were given the expert UI. The expert UI test plan, as its name suggests, was intended for logging calls via the expert UI, and comprised four tasks. There were either three or four steps involved in logging a customer’s query, depending on to whom the call was assigned (Section 6.2). Every second one of the eight tasks required in the novice UI test plan and the four tasks required in the expert UI test plan entailed four steps (i.e., required a solution), and every other task entailed three steps (i.e., required no solution).

The expert UI additionally contained adaptive lists (Figure 6.23), where the most recent and frequent items were moved to the top of the list; thus, the expert UI test plan consisted of some call categories that matched the call categories presented in the novice UI test plan. On completion of the test, the participant was required to complete a post-test questionnaire (Appendix G) which required both qualitative and quantitative responses. Qualitative responses were in the form of general comments. Quantitative responses were obtained using the Likert scale.

The post-test questionnaire measured:

- The user's overall reaction to the novice UI in terms of design, learnability, navigation and the amount of system feedback given;
- The extent to which task support was provided at the novice UI;
- The user's overall reaction to the expert UI in terms of design, learnability, navigation and the amount of system feedback given;
- The usefulness of the adaptive lists provided in the expert UI;
- The effect of the adaptation (transitioning from the novice UI to the expert UI); and
- The user's UI preference.

The post-test questionnaire was divided into five sections. Section A required participants to rate their biographical details. The Novice UI Section (Section B1) and the Expert UI Section (Section C1) were based on the original QUIS and contained questions relating to the overall perception of the system. The Task Support Section (Section B2) required the participants to rate the amount of task support provided at the novice UI with regards to the Visual Step Indicator and the Title and Task Description section. Questions within this section were based on the Task Support section of the post-test questionnaire designed by Singh (2007). The Adaptive List Section (Section C2) required participants to rate the use of the adaptive lists, and the Adaptivity Section (Section D) required participants to rate the usefulness of the adaptation from the novice UI to the expert UI. The Preference Section (Section E1) required participants to rate their preference of interfaces and Section E2 required general participant comments.

7.4.4. Procedure

User testing was conducted in the same usability laboratory used in the pilot study (Figure 6.29). One participant at a time was tested. Participants were asked to work through the novice UI and then the expert UI test plan (Appendix F) and to complete a post-test questionnaire. The steps for conducting the usability test were adopted from Pretorius (2005) and are shown in Table 7.1.

Step	Step Description
1	Participants were welcomed by the test administrator (the author)
2	Participants were briefed about the following: • The environment (usability lab) • The eye-tracking equipment • The purpose of the evaluation • The evaluation process
3	Participants were asked to complete a consent form (Appendix H).
4	The eye-tracker was calibrated for the participant.
5	The test administrator sat in the observer room whilst the participant sat in the participant room.
6	Participants were asked to commence with the evaluation. All interaction and eye-tracking data was captured. The test administrator monitored the participant at all times.
7	On completing the task at hand, the participant was asked to complete a post-test questionnaire.
8	The participant was debriefed by the test administrator during which the test administrator answered any queries made.
9	The participant was thanked for his/her time.
10	Results (interaction, eye-tracking and post-test questionnaire data) were gathered and analysed.

Table 7.1: Steps for Conducting the Usability Test

7.4.5. Participant Selection

Like the participants selected for a pilot study (Section 6.5.1), the participants selected for a usability study should be representative of the actual users, and for this reason, all participants had a sound knowledge of IT meaning they could assist customers with IT-related queries. The chosen participants were the CCAs at the NMMU ICT Service Desk and staff and IT students at the NMMU Department of CS & IS. The CCAs selected as participants comprised first-level (call takers) and second-level support staff (problem solvers). Staff members from the Department of CS & IS were IT technicians and administration staff.

Thirty participants were selected to take part in the main usability study. A population of six of the 30 participants were CCAs at the NMMU ICT Service Desk and were compensated for their participation. The remaining participants (n=24) were a convenience sample which comprised staff members and students from the NMMU Department of CS & IS who volunteered to be part of the usability study and were not compensated for their time. The demographic profile of the participants is depicted in Table 7.2.

DEMOGRAPHIC PROFILE OF PARTICIPANTS (n=30)					
	Age			Gender	
Total	19-29	30-39	40-50	Female	Male
30	21	4	5	17	13
	Occupation				
Total	CCA	Student	Lecturer	Admin Staff	IT Lab Tech
30	6	15	4	4	1
	Computer Experience (Years)			HEAT Experience	
Total	1-2	3-4	4+	Yes	No
30	2	4	24	8	22
	Service Desk Software Experience (Years)				
Total	0	1-2	3-4	4+	
30	22	2	1	5	

Table 7.2: Demographic Profile of Participants (n=30)

Figure 7.2 (A) illustrates the statistics of the participants' age profile. As depicted in Figure 7.2 (A), 70 percent of the participants were between 19 – 29 years old. The remaining 30 percent were between 30 – 50 years old: 13 percent (n=4) were between 30 – 39 years old and 17 percent (n=5) were between 40 – 50 years old.

Figure 7.2 (B) illustrates the statistics of the participants' gender profile. As depicted in Figure 7.2 (B), 57 percent (n=17) of the participants were female and the remaining 43 percent (n =13) were male.

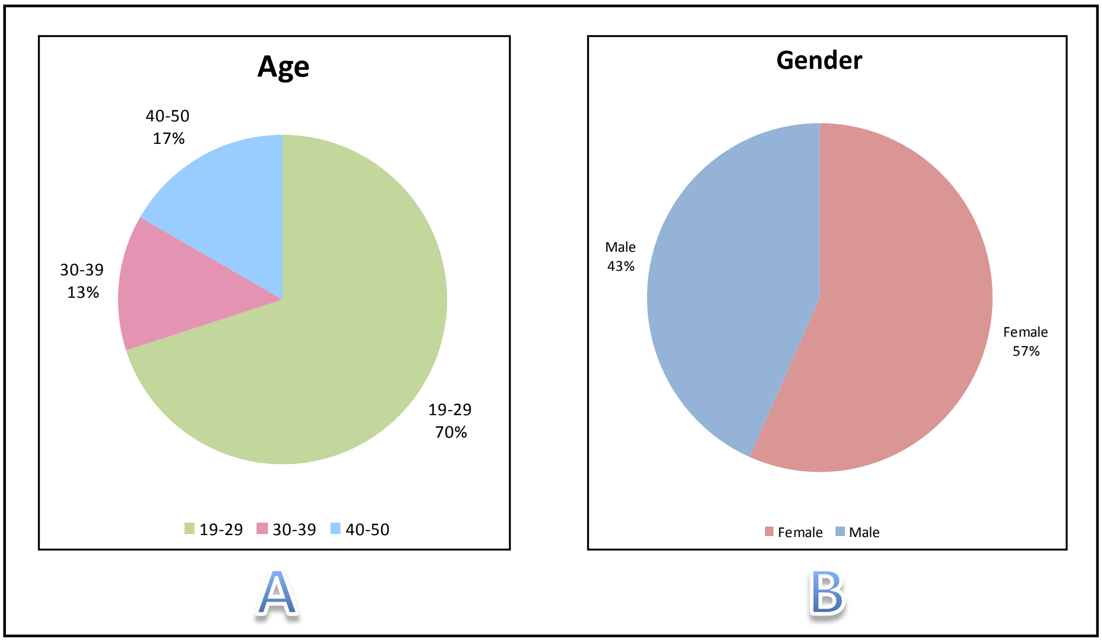


Figure 7.2: Age (A) and Gender (B) Profile of Test Participants (n=30)

Figure 7.3 (A) illustrates the occupation profile of the test participants. As depicted in Figure 7.3 (A), 50 percent (n=15) of the participants were IT students from the NMMU Department of CS & IS. Twenty percent (n=6) of the participants were CCAs at the NMMU ICT Service Desk. The remainder percent comprised staff from the NMMU Department of CS & IS, where 14 percent (n=4) were lecturers, 13 percent were admin staff and 3 percent (n=1) were IT lab technicians.

Figure 7.3 (B) illustrates the computing experience profile of the test participants. As depicted in Figure 7.3 (B), 80 percent (n=24) of the participants had more than four years of computer experience. The remaining 20 percent had one to four years of computer experience, where 13 percent (n=4) had three to four years experience and 7 percent (n=2) had one to two years experience.

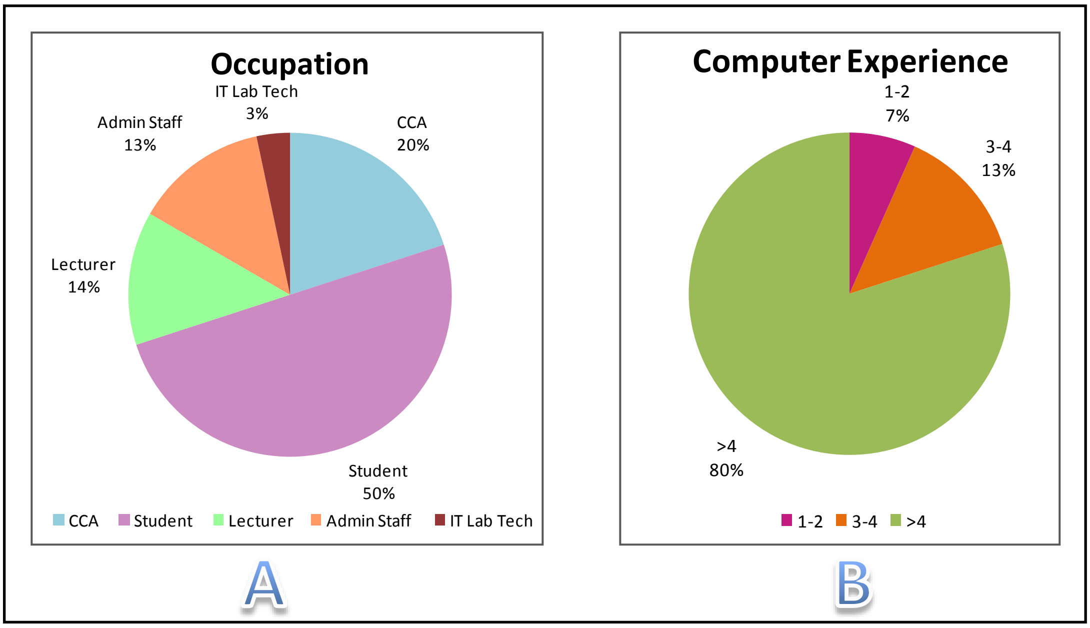


Figure 7.3: Occupation (A) and Computing Experience (B) Profile of Test Participants (n=30)

Figure 7.4 (A) illustrates the HEAT (the service desk software utilised at the NMMU ICT service desk) experience profile of the test participants. As depicted in Figure 7.4 (A), only 20 percent (n=8) of the participants had previously used this software. Six of these eight participants were CCAs and the remaining two were students who had previously come into contact with it. The remaining 80 percent (n=22) had no previous experience of it.

Figure 7.4 (B) illustrates the Service Desk Software experience profile of the test participants. As depicted in Figure 7.4 (B), 73 percent (n=22) of participants had never used service desk software before. Seven percent (n=2) of the participants had one to two years experience of it and they were the same two participants (student) who had come into contact with HEAT software. The remaining 20 percent (n=6) of the participants had more than three years experience of service desk software, and comprised of CCAs, wherein 3 percent (n=1) had three to four years experience with the software and 17 percent (n=5) had more than four years' experience.

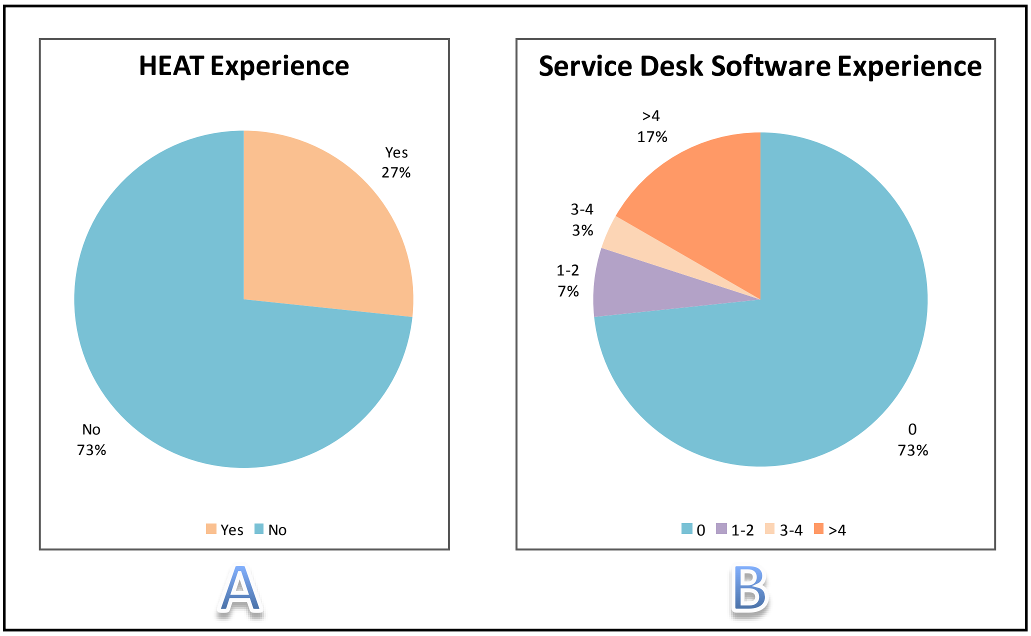


Figure 7.4: HEAT (A) and Service Desk Software Experience (B) Profile of Test Participants (n=30)

7.4.6. Main Usability Testing Results

The results of the usability testing are discussed in terms of the Inference and User Interaction evaluation layers. Analysis of the post-test questionnaire (Appendix G) completed by the test participants (n=30) yielded accuracy (Inference Layer) and satisfaction (User Interaction Layer) results. The post-test questionnaire yielded both quantitative and qualitative results. Analysis of participants' performance data yielded learnability (User Interaction Layer) results. Eye-tracking data was analysed to determine effectiveness (User Interaction Layer).

7.4.6.1. Accuracy Results

Precision was the metric used to measure the accuracy of the Analysis Engine AUI component; hence, it was used for the evaluation of the Inference Layer. The data used to measure precision was gathered from the post-test questionnaire (Appendix G) presented to the participants. Question 5 of Section E2 of the post-test questionnaire required participants to give a binary response (yes or no) to whether they felt the adaptation was an accurate reflection of their expertise. Precision can only be measured if the users' ratings can be transformed into a binary scale (Herlocker *et al.*, 2004). The precision of adaptations made was thus calculated. Results, using the formula presented in Section 7.4.1 are presented in Table 7.3.

PRECISION RESULTS	
Relevant Adaptations (N_{ra})	27
Total Adaptations (N_a)	30
Precision	**0.9**

Table 7.3: Precision Results (n=30)

One adaptation was provided per participant (Section 7.4.3), as each participant was transitioned from a novice UI to an expert UI. As a result, there were a total of 30 adaptations (Table 7.3). Twenty-seven participants felt that they had enough practise using the novice UI when they were transitioned to the expert UI and only three felt that they needed more practise using the novice UI before they were transitioned to the expert UI (Figure 7.5). This meant there were 27 relevant adaptations. The value obtained by calculating the precision indicates the percentage of accurate adaptations (or recommendations) (McLaughlin and Herlocker, 2004). The precision value was 0.9 (Table 7.3) and there was a measured precision of 90 percent. Ninety percent (n=27) of participants felt that the adaptation accurately reflected their expertise level and, therefore, it could be expected that, on average, nine out of every ten adaptations made by the ASD system would be accurate.

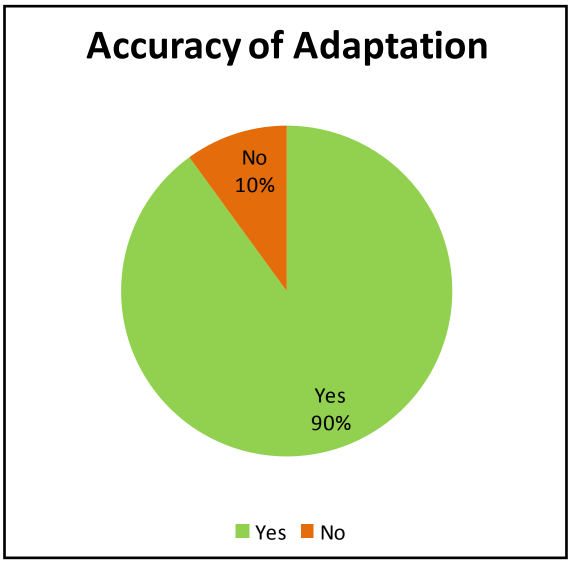


Figure 7.5: Accuracy Results of Participants (n=30)

7.4.6.2. Learnability Results

Learnability was one of the measures used to evaluate the User Interaction Layer. The data used to measure learnability was gathered from the participants' performance data residing within the Knowledge Base. The learnability of the ASD prototype was measured by assessing the time it took participants to learn while using the novice UI, the task completion times of the participants for both novice and expert UIs, and by measuring the amount of tasks that participants performed on the novice UI before they were transitioned to the expert UI.

Figure 7.6 illustrates the number of tasks which were completed by participants, while using the novice UI, before the interface was adapted to the expert UI. Sixty-six percent of participants (n=20) completed the minimum number of tasks (n=4). Section 2.6 stated that novice users become experts very quickly in CC environments, and the fact that most users (n=20) became skilled with using the novice UI after only four tasks further supports this. Ten percent (n=3) completed five tasks before the novice UI was adapted to the expert UI, and 23 percent (n=7) completed more than five tasks, 6.67 percent completed six tasks, 6.67 percent completed seven tasks and ten percent completed eight tasks.

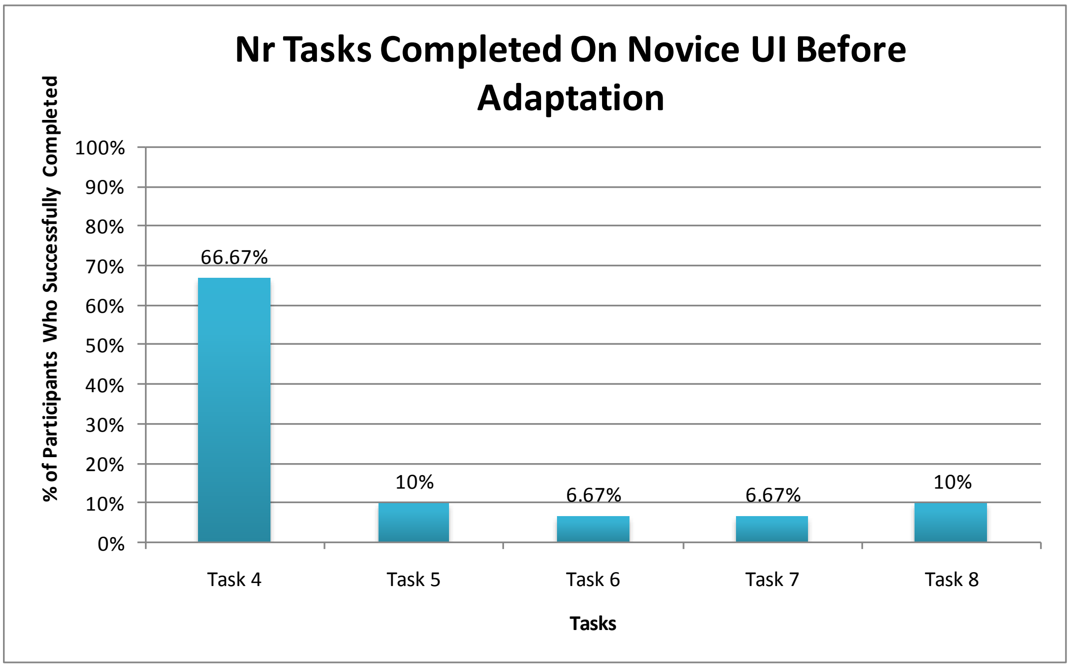


Figure 7.6: The Number of Tasks Completed on Novice UI Before Adaptation

Figure 7.7 illustrates the average time-on-task for tasks 1, 3, 5 and 7 when using the novice UI. The tasks depicted in this figure represent tasks which entailed three call logging steps, and therefore did not require the participant to provide a solution. The learning curve presented in Figure 7.7 does not clearly indicate a learning effect; however, there does appear to be a decrease in the average time-on-task between task 1 and task 7. The average time-on-task for task 1 is 173 seconds and 104.60 seconds for task 7; hence, the ratio shows that participants who required seven tasks (n=5) until their interface was adapted to the expert UI, were initially taking 1.65 times longer. Figure 7.6, however, illustrated that 66.67 percent of participants (n=20) only completed four tasks until they were transitioned to the expert UI. Considering the average time-on-task for task 1 (173 seconds) and task 3 (91.73 seconds), yields the ratio that participants (n=30) were initially taking 1.9 times longer when performing task 1 than they did when performing task 3.

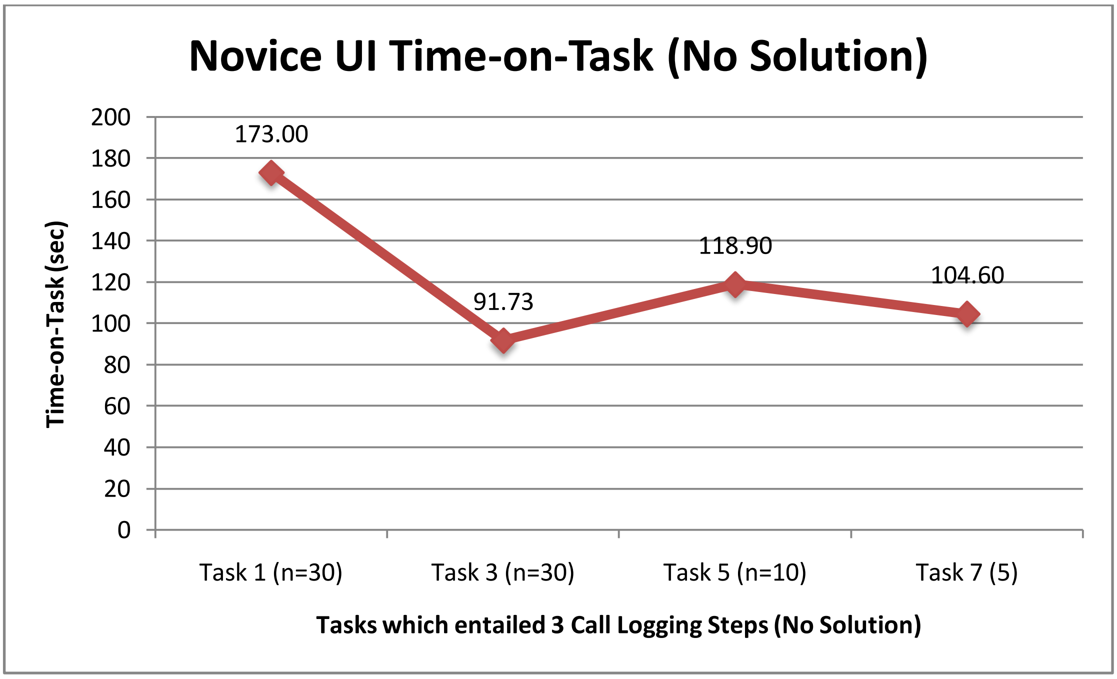


Figure 7.7: Learnability Results of Novice UI (Task 1, 3, 5 and 7)

Figure 7.8 illustrates the average time-on-task for tasks 2, 4, 6 and 8 when using the novice UI. The tasks depicted in Figure 7.8 represent tasks which entailed four call logging steps, and, therefore, required the participant to provide a solution. Contrary to the tasks represented in Figure 7.7, these tasks involved an extra call logging step and, therefore, took a longer period of time. The slope represented in Figure 7.8 appears to be flattening out: there appears to be no significant learning between tasks 2, 4, 6 and 8 on the novice UI.

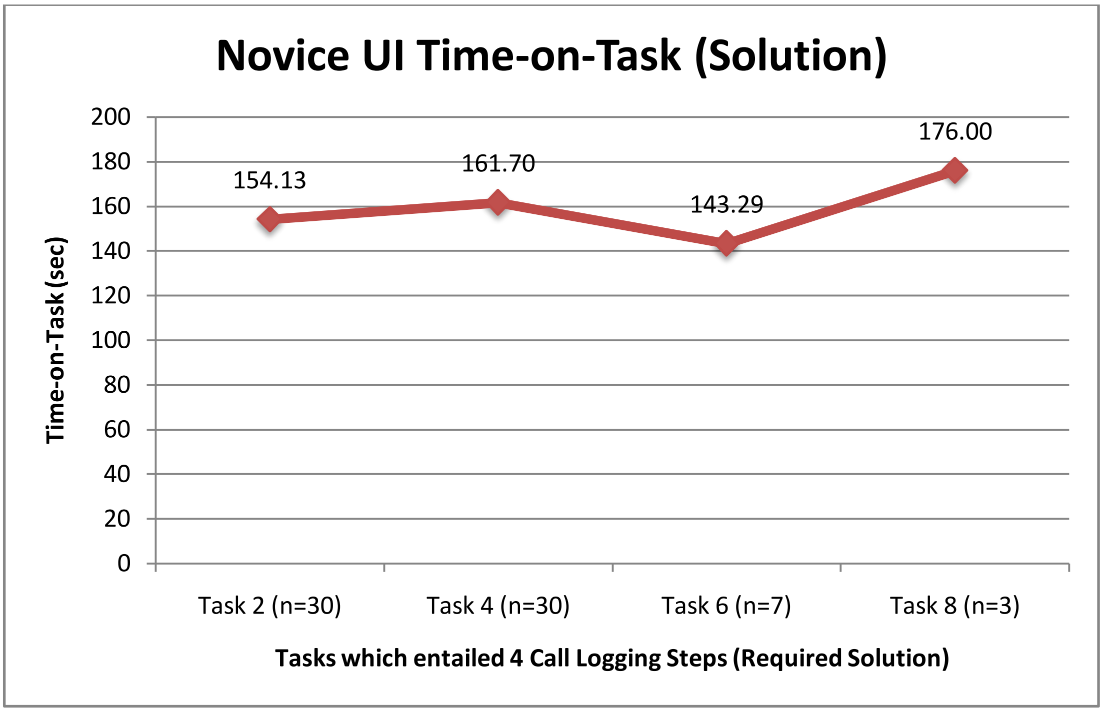


Figure 7.8: Learnability Results of Novice UI (Task 2, 4, 6 and 8)

Figures 7.7 and 7.8 illustrated the average time-on-task for the tasks performed on the novice UI. There appeared to be some learning curve on tasks which did not require the provide solution details call logging step (Figure 7.7) and there did not appear to be any learning on tasks which required the provide solution details call logging step (Figure 7.8). These results are, however, only related to tasks preformed on the novice UI. The differences between the tasks performed on the novice UI and tasks performed on the expert UI have to be investigated to identify whether participants' performances increased when the interface was adapted to the expert UI.

Figure 7.9 illustrates the difference in participants' average time-on-task for tasks 1 to 4, between the novice UI and the expert UI. All participants (n=30) were required to perform at least four tasks on the novice UI and four tasks on the expert UI; therefore, the comparison between the two interfaces involved the first four tasks on the novice UI and the four tasks on the expert UI. Figure 7.9 illustrates a difference between the average time-on-task for the novice UI and the expert UI for all four tasks. The greatest difference between the time-on-task means for the novice UI and the expert UI appear to be task 1 (novice UI mean = 173 sec, expert UI mean = 97.8 sec) and task 4 (novice UI mean = 161.7 sec, expert UI mean = 127.17 sec). Although there appears to be a difference between the time-on-task means for the novice UI and the expert UI, it must be determined whether this difference is significant enough.

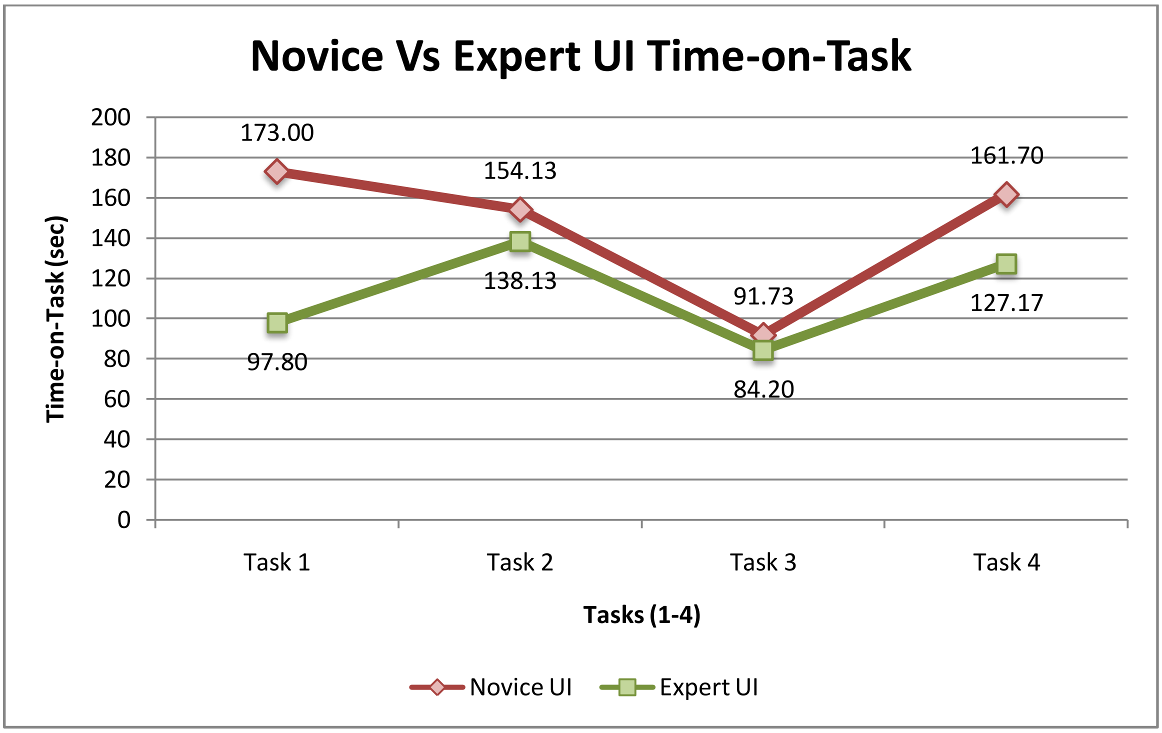


Figure 7.9: Learnability Results Between Novice UI and Expert UI (n=30)

Consultation with the NMMU statistician, Mr. D. Venter (Venter, 2008) suggested the use of a paired samples *t*-test, to determine whether there is a significant difference in participants' performance on the two interfaces (novice UI and expert UI). A paired samples *t*-test is utilised when the design is a within-subjects design. A within-subjects design is used to compare the means for the same group of participants on two different products or designs, which is ideal for this research study (Tullis and Albert, 2008).

In order to determine whether there was a significant difference between the time-on-task means for the novice UI and the expert UI, the following null hypothesis was tested using a paired samples *t*-test:

> *H_o: There is no difference in participants' performances on the novice and expert UIs.*

A paired samples *t*-test was conducted in the data analysis software package, STATISTICA V8.0. A significance level of 0.05 (5 percent) indicates that with a 95 percent confidence level, a hypothesis that is actually correct will not be rejected. The null hypothesis will be rejected if the p-value falls within the lower 5 percent of possible values. The p-value (number between 0 and 1) reflects the strength of the data used to to evaluate the null hypothesis. A small p-value indicates strong evidence against the null hypothesis (Rumsey, 2003).

Detailed results of the *t*-test can be viewed in Appendix I. The results of the paired samples *t*-test indicate that there is a significant difference between participants' performances on the novice and expert UIs ($t = 7.81$, $df = 29$, $p < 0.0005$, $d = 1.43 > 0.8$). Participants' performances improved significantly over time; thus, the null hypothesis can be rejected with a 95 percent confidence level. Therefore, it can be concluded that:

> *There is a significant difference in participants' performances on the novice and expert UIs.*

Learnability Contributions

This sub-section discussed the learnability results of the ASD prototype and some key findings have been identified. Firstly, 66.67 percent of participants (n=20), performed the minimum number of tasks on the novice UI (n=4) before the Adaptation Moment occurred. This is significant, as it indicates that the novice UI provided enough learning before participants received the expert UI and that this learning occurred quickly. The fact that the accuracy of the adaptation (Section 7.4.6.1) had a precision result of 90 percent further indicates the high percentage of participants who were satisfied with the adaptation timing. Another key finding identified in this sub-section was that a significant difference was found between participants' performance on the novice UI and their performance on the expert one. This finding indicated that participants performed substantially faster on the expert UI and is significant because an objective of this research was to improve CCAs' productivity (Section 1.3.1). The next section will discuss satisfaction results to determine the user's experience of the ASD prototype.

7.4.6.3. Satisfaction Results

Besides learnability (Section 7.4.6.2), user satisfaction was the other measure, used to evaluate the User Interaction Layer. The data used to measure this, was both qualitative and quantitative. Qualitative data (Section 7.4.6.3.2) consisted of participants' responses to open-ended questions (Section E2 of the post-test questionnaire in Appendix G). Quantitative data consisted of participants' ratings obtained from the post-test questionnaire. The results of the quantitative data will be discussed further.

7.4.6.3.1. Quantitative Analysis

Ratings were given on a five-point Likert scale, where a rating of three represented a neutral rating. Ratings above three indicated that the participant was satisfied and there was no cause for concern; however, ratings below three indicated a cause for concern and required an explanation. The median, mean and standard deviations were calculated to obtain overall ratings and these values can be seen in Appendix I. The level of satisfaction achieved by the test participants is indicated by the median and mean values, and the measure of variability is indicated by the standard deviation values.

Figure 7.10 illustrates the user satisfaction ratings for the novice UI. Section B1 of the post-test questionnaire measured the user satisfaction ratings for the novice UI. Participants were highly satisfied with the novice UI, as all attributes, with the exception of Phrasing of Error Messages, had a mean and median between four and five. Participants (n=30) especially found the novice UI very easy to navigate and learn and were highly satisfied with its screen design. Participants were, however, still somewhat satisfied with the Phrasing of Error Messages as its mean was above 3 (3.64) and it had a median of 4. Participants ratings for the Phrasing of Error Messages only applied to 11 participants, however, as most participants (n=19) made no mistakes and thus received no error messages. The standard deviations ranges from 0.61 to 1.21 and there appeared to be a general consensus among participants.

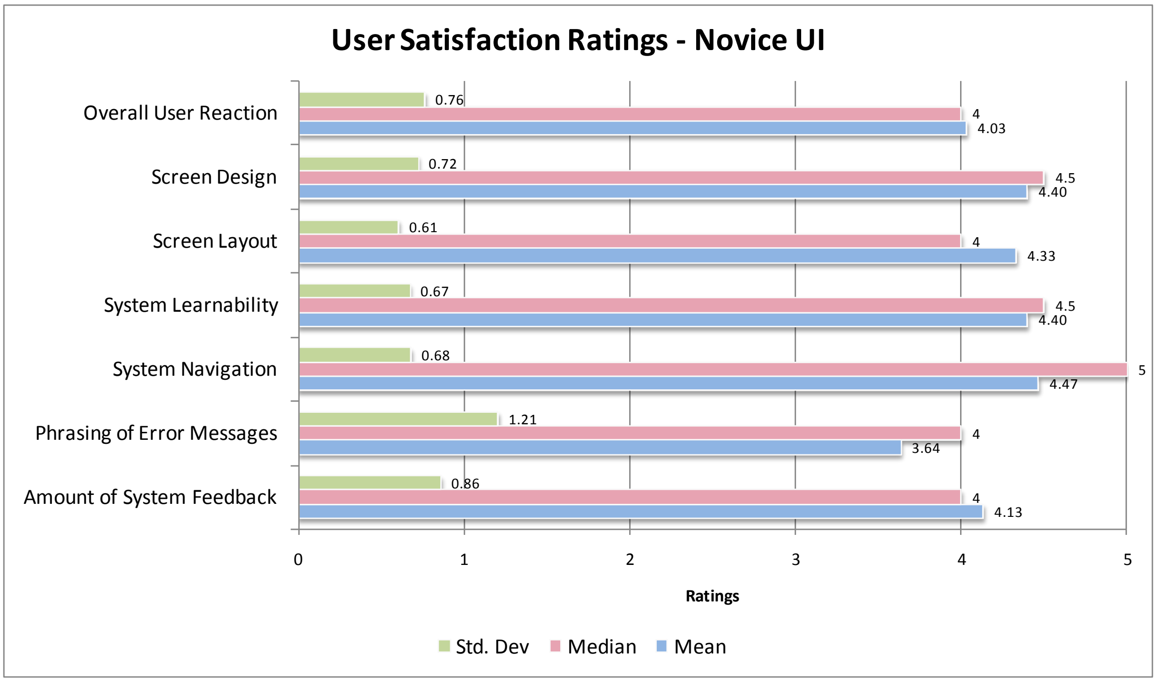


Figure 7.10: User Satisfaction Results for Novice UI (n=30, Except Phrasing of Error Messages Attribute Where n=11)

Figure 7.11 illustrates the user satisfaction ratings for the expert UI. Section C1 of the post-test questionnaire measured the user satisfaction ratings for this. Participants were highly satisfied with the expert UI, as all attributes had a mean and median between four and five. Similarly to the novice UI, participants especially found the expert UI very easy to navigate and learn and were highly satisfied with its screen design. Participants were also highly satisfied with the layout of the expert UI. Participants ratings for the Phrasing of Error Messages for the expert UI only applied to 11 participants, as most participants (n=19) made no mistakes and received no error messages. The mean and median values for the expert UI also appear to be somewhat higher than those of the novice UI. A possible reason for this is that participants were more familiar with the expert UI than the novice UI. They had practised on the novice UI and, therefore, had a more enjoyable experience with the expert UI. The standard deviations range from 0.49 to 0.94, therefore, there appeared to be general consensus.

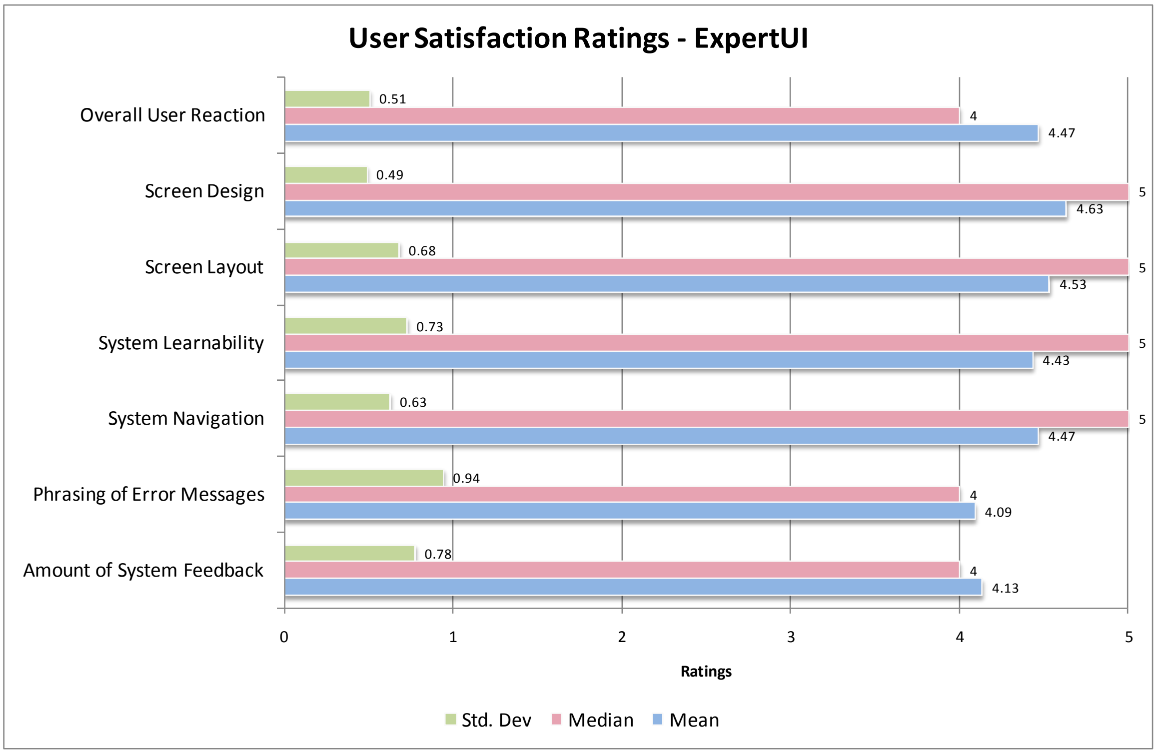


Figure 7.11: User Satisfaction Results for Expert UI (n=30, Except Phrasing of Error Messages Attribute Where n=11)

The Task Support section (Section B2) of the post-test questionnaire attempted to determine how satisfied participants were of the level of task support provided at the novice UI. The novice UI provided more task support than the expert UI one because novice users required more guidance. Task support was essentially provided in the form of the Visual Step Indicator (Section 6.4.1) and task based information displayed at the Title and Task Description section UI (Section 6.4).

Figure 7.12 illustrates the ratings for task support provided at the novice UI. Participants were generally satisfied with the overall level of task support it provided. In particular, participants were highly satisfied with the Visual Step Indicator, the indication of progress provided and support provided on the novice UI to facilitate call logging, as these attributes had a mean and median between four and five. Even though participants were satisfied with the level of task support from the Visual Step Indicator, participants' ratings on its helpfulness were not that high as its mean was below four. Participants were additionally not very satisfied with the usefulness of the task-based information provided, as this mean was 3.77. There is, however, no cause for concern as the medians for the Helpfulness of Visual Step Indicator and the usefulness of Task-Based

Information were fairly high (median=4). Standard deviations ranged from 0.67 to 1.17; therefore, there appears to be general consensus.

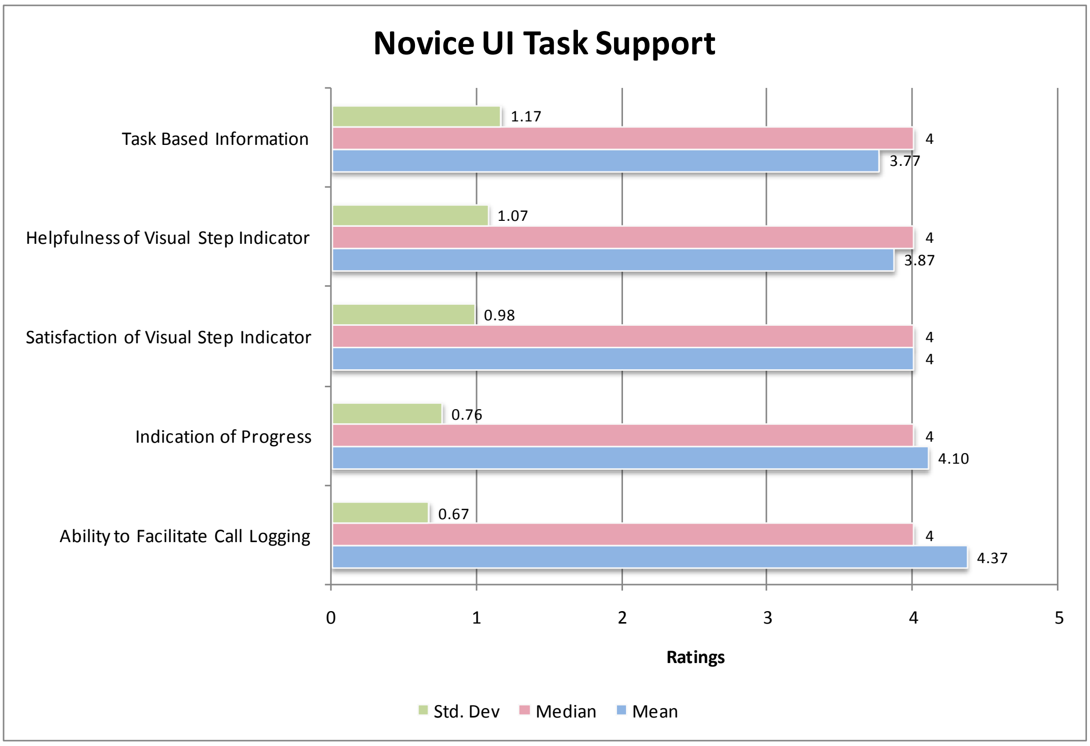


Figure 7.12: Task Support Results for Novice UI (n=30)

The Adaptive List section (Section C2) of the post-test questionnaire attempted to determine how satisfied participants were with the adaptive lists provided on the expert UI. The expert UI contained adaptive lists, which provided expert users with a more efficient UI (Section 6.4.2), as opposed to the static lists provided on the novice UI.

Figure 7.13 illustrates the user ratings for the adaptive lists provided on the expert UI. Participants (n=30) were highly satisfied with the adaptive lists as the mean and median values ranged between four and five. Participants rated the ordering of the adaptive lists as satisfying, logical, very useful and noticeable. Standard deviations ranged from 0.63 to 0.81; therefore, there appeared to be general consensus.

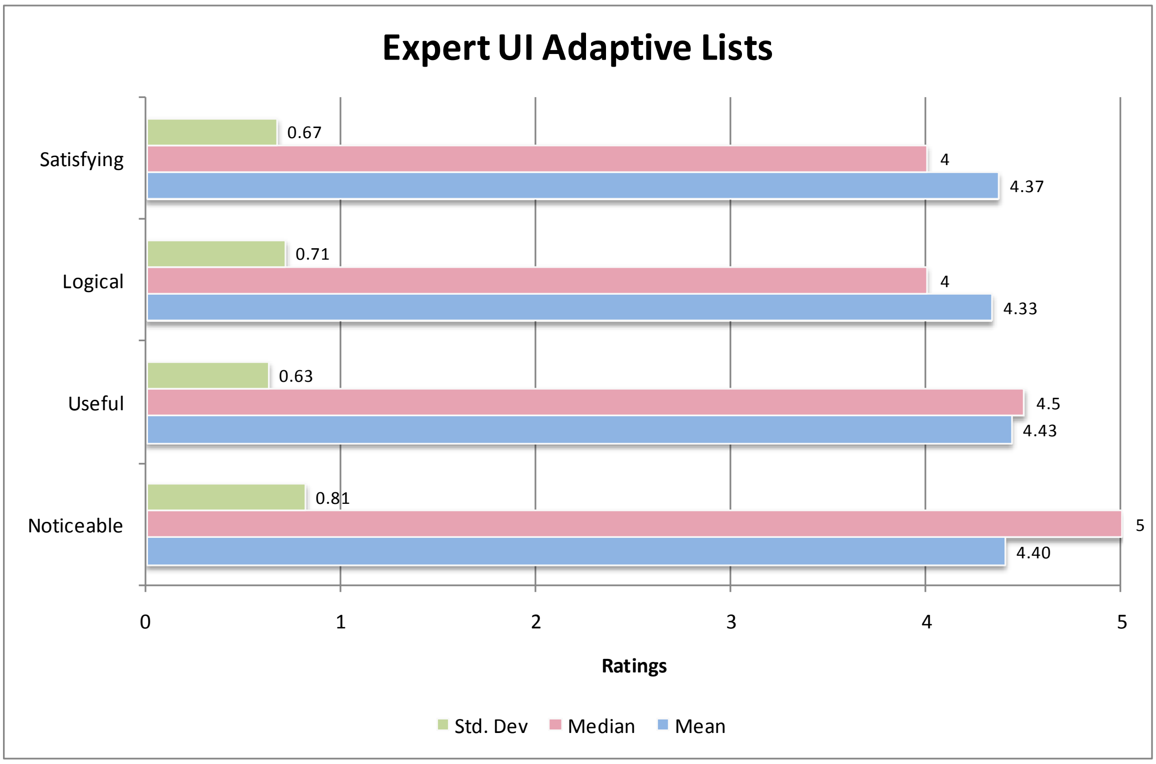


Figure 7.13: Adaptive List Results for Expert UI (n=30)

Section C2 of the post-test questionnaire also required participants to specify which list (static on novice UI or adaptive on expert UI) they preferred. Figure 7.14 illustrates that most users (87 percent) preferred the adaptive lists. Only 13 percent participants (n=4) preferred the static lists.

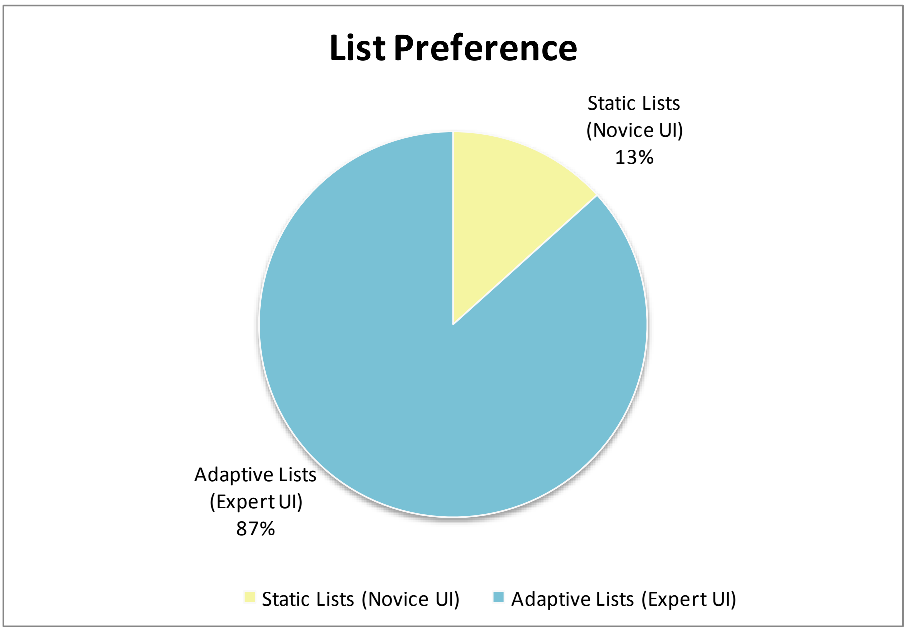


Figure 7.14: List Preference Results (n=30)

The Adaptivity section (Section D) of the post-test questionnaire attempted to determine how satisfied participants were with the adaptation of the novice UI to the expert one. Figure 7.15 illustrates the user ratings. Participants (n=30) were highly satisfied as all attributes had a mean and median between four and five. Participants rated the adaptation of the novice UI to the expert UI as clear, pleasant and useful. Standard deviations ranged from 0.75 to 0.86; therefore, there appeared to be general consensus.

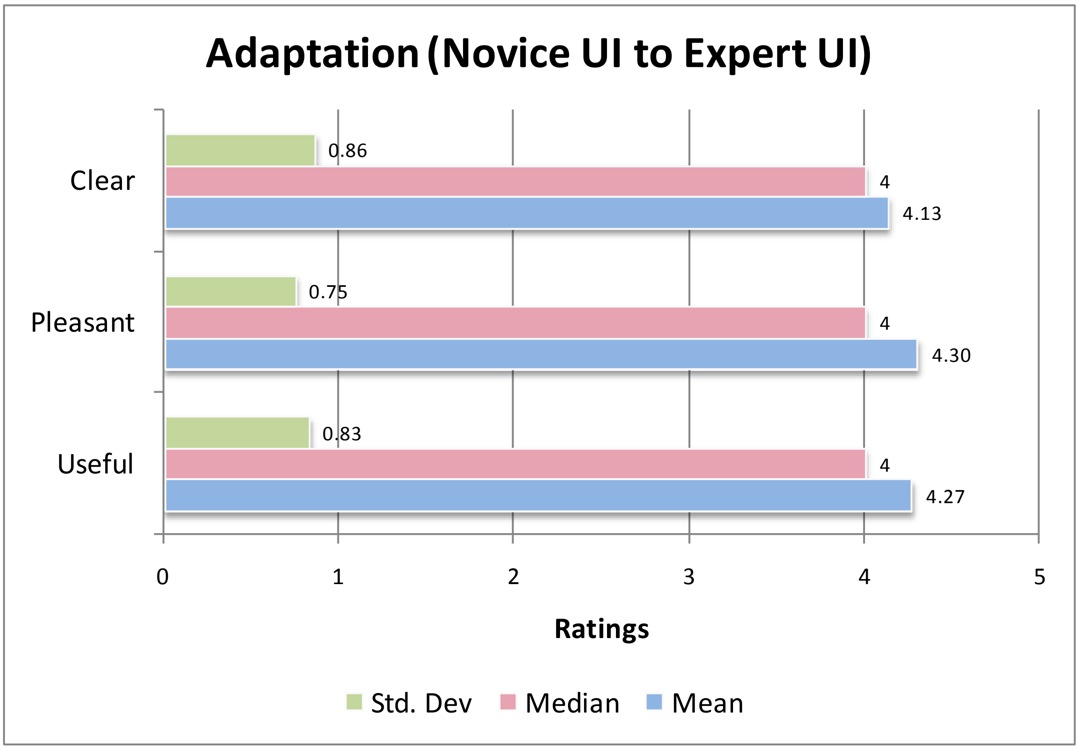


Figure 7.15: Adaptation Results (n=30)

The Preference section (Section E1) of the post-test questionnaire required users to rate their preference for the novice UI, the expert UI, and the adaptation of the novice UI to the expert UI, ranging from one (best) to three (worst). Figure 7.16 illustrates the user ratings of the preferences. Thirty-six percent of participants (n=11) rated having the adaptation of the novice UI to the expert UI as the best scenario and 50 percent of participants (n=15) preferred only having the expert UI. The 50 percent of participants who preferred the expert UI did, however, indicate that the reason they preferred it was because of training conducted on the novice UI (Section 7.4.6.3.2). Only 13.33 percent of participants (n=4) preferred only having the novice UI. Forty-three percent (n=13) rated the adaptation as their second preferred choice and 33.33 percent of participants (n=20) preferred the expert UI as their second choice. More than eighty percent of participants rated the adaptation (n=24) and the expert UI (n=25) as their first and second choice. Those who rated the expert UI as either first or second indicated that their ratings were affected by the training provided by the novice UI; therefore, it can be argued that 80 percent of participants preferred the adaptation of the novice UI to the expert UI. Sixty-three percent of participants (n=19) rated the novice UI as the worst choice; however, it can be argued that this rating was due to their unfamiliarity with the UI, as they received the novice UI first.

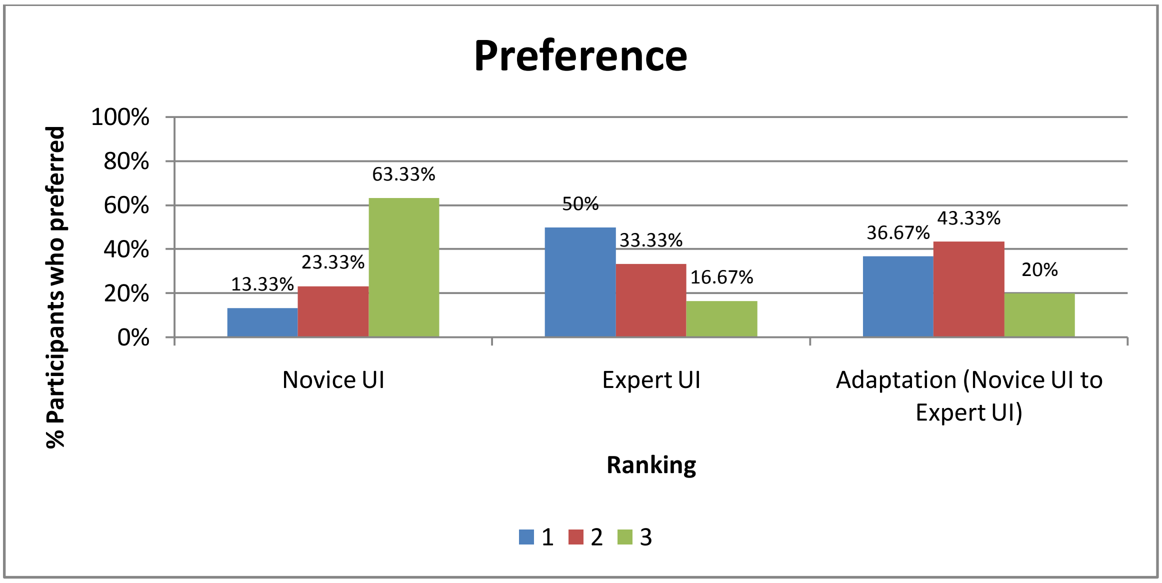


Figure 7.16: Preference Results (n=30)

Satisfaction Contributions

This sub-section reported on the quantitative satisfaction results of the ASD prototype and some key findings were identified. Firstly, the overall user satisfaction results for both the novice UI and the expert UI were rated incredibly high (mean between four and five). This indicates that participants were very pleased with the design of the interfaces. Another key finding was the high satisfaction results with regards to the task support provided at the novice UI. This is significant because it indicates that user's positively utilised the task support provided. The adaptive lists provided at the expert UI were also highly rated. It can be concluded that, generally, participants were highly satisfied with novice and expert UIs as well as the adaptation of the novice UI to the expert UI. This research finding is significant because an objective of this research was to enhance the CCAs' interaction with the UIs (Section 1.3.1). The following section reports on the qualitative analysis of participants responses.

7.4.6.3.2. Qualitative Analysis

The qualitative data was obtained from open-ended questions found in Section E2 of the post-test questionnaire. Participants were asked to provide both positive and negative feedback on the adaptation from the novice UI to the expert UI. A report on participants' negative and positive comments will follow.

Positive Comments

All participants (n=30) were satisfied with the level of training brought about by the novice UI and found the novice UI extremely useful. Some of the positive comments supporting this were:

- *"The novice UI introduces the system to those who are inexperienced*";
- *"The novice UI helped a new user understand the fundamental business logic of the system"*;
- *"Learning the concepts of logging a call was easier on the novice UI"*;
- *"The expert UI is easier to use after having used the novice UI"*; and
- *"The novice UI facilitates learning of even minor aspects of the service desk"*.

A small number of participants stated that the adaptive lists provided at the expert UI enabled them to log calls more quickly. Some positive comments indicated that participants were pleased that the interface adapted according to their skill level and the following support this:

- *"Nobody wants to stay in one place all the time"*;
- *"Very useful because the novice UI taught me the basic, and when it started getting frustrated, it switched to the more efficient expert UI"*;
- *"Very good concept for a novice CCA"*; and
- *"Allows the user to become a power user"*.

Negative Comments

More than half of the participants (53 percent) provided no negative comments. The negative comments provided were with regards to the sudden change to the expert UI. Participants who provided negative comments generally felt that the difference between the novice UI and the expert UI was a slight learning curve. Some negative comments provided by participants with regards to the adaptation of the novice UI to the expert UI were:

- *"Difference between the interface is quite dramatic";*
- *"User might become confused when presented with a new interface";*
- *"Changing style of the UI (many screens to one) might cause frustration";* and
- *"User might not be ready for transition to expert UI"*.

Thus, future work could include an intermediary UI whereby the interface is adapted from the novice UI to an intermediate UI to the expert UI. The adaptation would be more gradual and possibly reduce any confusion brought about by the sudden change from the novice UI to the expert one. Participants had no negative comments with regards to the novice UI; however, there were a couple of negative comments with regards to the expert UI:

- *"..had to remember what task to do in what section of the screen unlike support provided with novice UI"; and*
- *"No tooltips on expert UI"*.

The expert UI did not provide step-by-step guidance (Section 6.4.2) as the novice UI did and, therefore, one participant, in particular, stated having some difficulties with remembering the ordering of tasks. No extra tooltips were provided on the expert UI as they were on the novice UI and a single participant preferred the extra information provided by the tooltips. Future work could include tooltips on the expert UI.

Forty-seven percent of the participants provided negative comments with regard to the differences between the novice and the expert UIs. Positive comments, however, indicated that all users were highly satisfied with the training provided by the novice UI. The fact that most participants (80 percent) rated the expert UI as either their first or second choice (Section 7.4.6.3.1) suggest that they were generally satisfied by the adaptation from the novice UI to the expert UI.

7.4.6.4. Effectiveness Results

Eye-tracking was used to measure the effectiveness of the design of both the novice and expert UIs. The design of the novice and expert UIs entailed two sections; namely, the Title and Task Description section and the User Input section, which were adopted from Singh (2007). The novice UI additionally entailed a Visual Step Indicator, and the expert UI provided the more efficient adaptive lists. Eye-tracking, therefore, in particular, was used to firstly measure the effectiveness of the two interface sections of both the novice and expert UI. Eye-tracking was additionally used to measure the effectiveness of the Visual Step Indicator and the adaptive lists. Five callibration points were used and the fixation radius (the smallest distance that can separate fixations) was 35 pixels. The AOIs defined for both novice and expert UIs were the Title and Task Description section and the User Input section. These sections were illustrated in Figure 6.7. The novice UI contained additional AOIs to evaluate the effectiveness of the Visual Step Indicator.

Figure 7.17 illustrates three AOIs: Visual Step Indicator 1.1, 1.2 and 1.3, superimposed on the novice UI. These three AOIs were analysed to determine how long it took participants to notice the Visual Step Indicator on the novice UI when performing the first step of the first task.

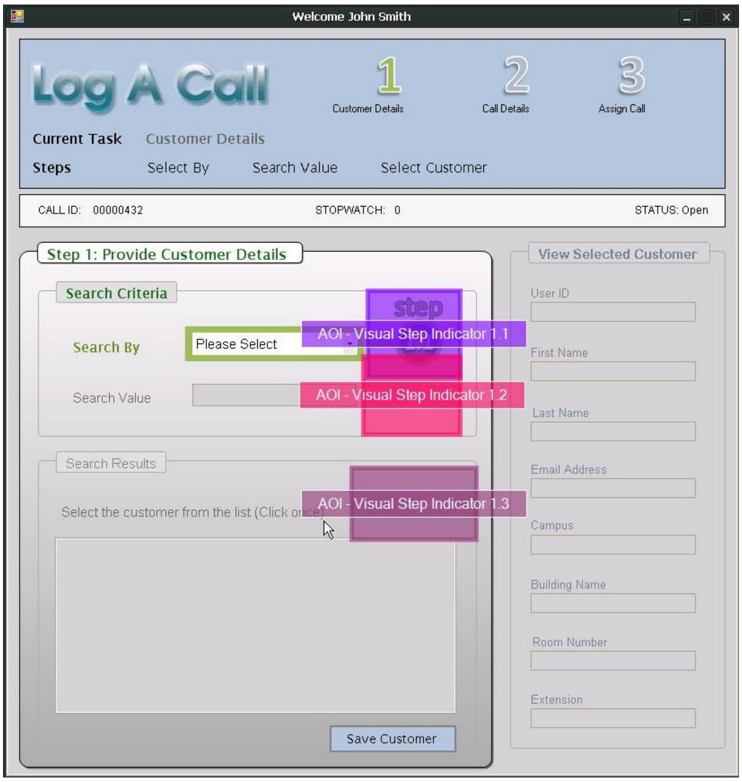


Figure 7.17: Visual Step Indicators Represented as AOIs

Figure 7.18 illustrates the results of the eye-tracking analysis of the Visual Step Indicators depicted in Figure 7.17. Results indicate that, on average, it took participants less than five seconds (mean and median = 4.59 seconds, n=25) to notice the first Visual Step Indicator and less than 10 seconds (mean and median = 7.67 seconds, n=22) to first notice the second one. These results illustrate the effectiveness of the Visual Step Indicator. The time taken to first notice the third Visual Step Indicator for the first task was, however, extremely long (mean and medium = 40.26 seconds, n=15) and this could possibly have affected the time participants took to perform the *provide customer details* step of task 1. It must also be noted than 16.67 percent of participants (n=5) did not notice the first Visual Step Indicator, 26.67 percent of participants (n=8) did not notice the second one and half the participants (n=15) did not notice the third one. This indicates that the Visual Step Indicator should be better highlighted. These results, however, only apply to the circle section of the Visual Step Indicator (Figure 6.11); therefore, it can be argued that participants who did not notice the circle section of the Visual Step Indicator were provided with enough assistance by the green border section of the Visual Step Indicator (Figure 6.11).

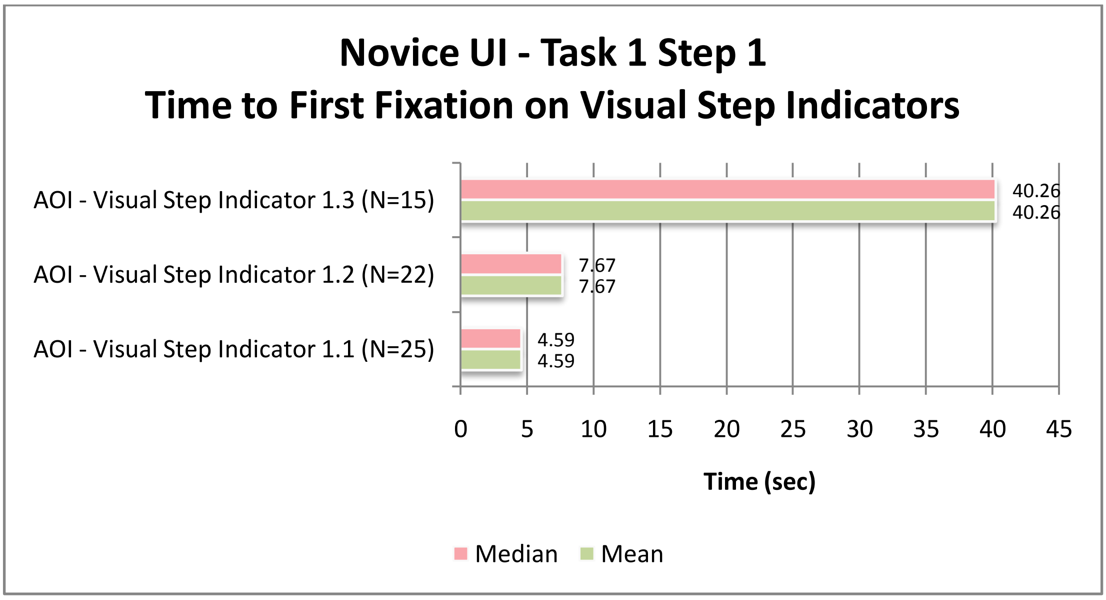


Figure 7.18: Time to First Fixation Results for Novice UI Task 1 Step 1 (n=30)

Fixation duration was the metric used to measure the time participants spent looking at the Title and Task Description and the User Input sections. Heat maps were used to visualise the fixation duration of all participants. Heat maps consist of a screenshot of the ASD system and a heat map mask superimposed on top of the screenshot. The heat map mask highlights the areas where participants have been looking. The heat map mask style used to visualise the fixation duration was the absolute duration, which shows how long participants looked at the different areas in the image. The fixation duration is represented by the use of colour: red indicates longer fixation duration and green indicates shorter fixation duration.

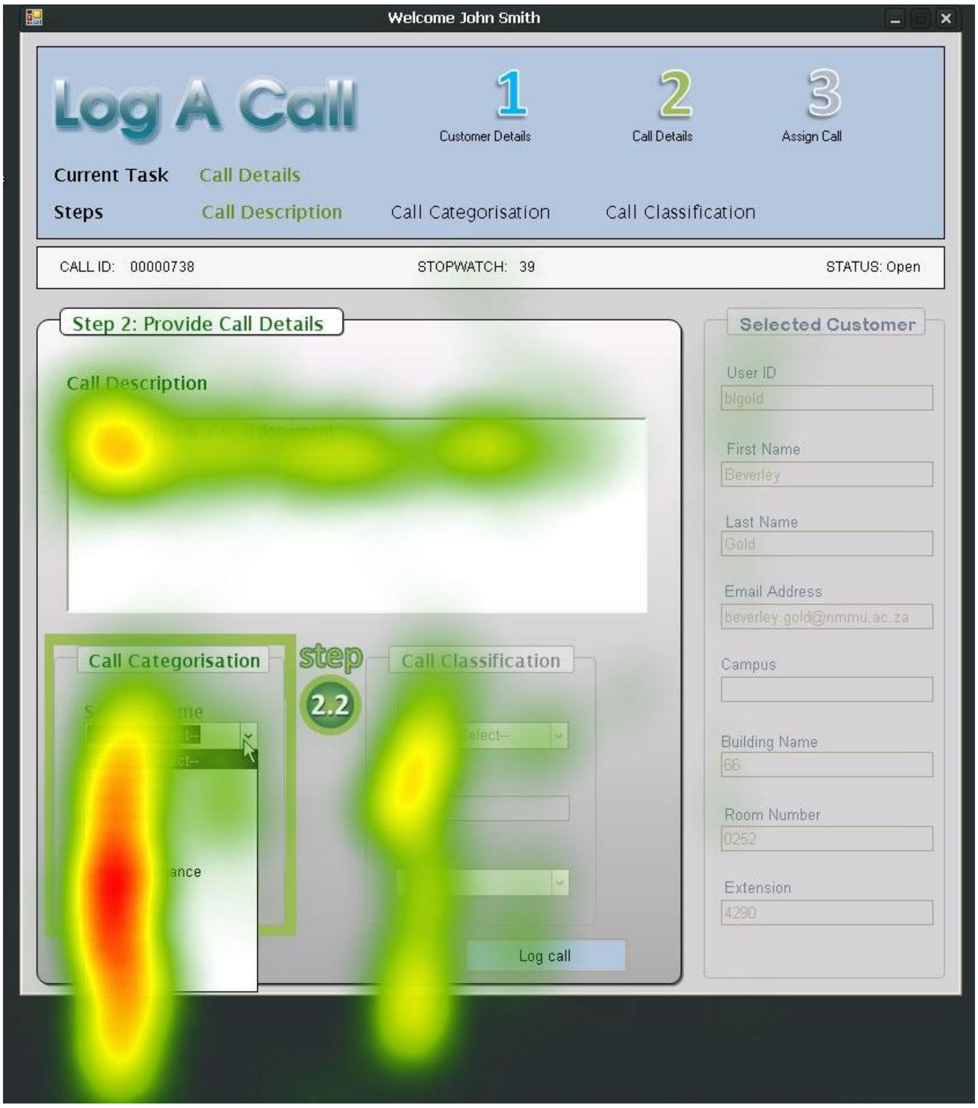


Figure 7.19: Novice UI Heat Map - Fixation Duration of Task 1 Step 2

(Max Absolute Duration = 95.12 sec, n =30)

The heat map, represented in Figure 7.19, visualises the fixation duration of all the participants (n=30) for the second step (provide call details) of the first task performed on the novice UI. Figure 7.19 reveals that participants spent no time fixating on the Title and Task Description section but rather focussed more on the User Input section. Participants, in particular, appeared to have fixated on specific areas of the User Input section of the novice UI (*Call Description*, *Call Categorisation* and *Call Classification* in Figure 7.19). These specific areas which participants appear to have fixated on represent the three steps of the *provide call details* task and this, therefore, further proves the effectiveness of the guidance brought about by the Visual Step Indicator.

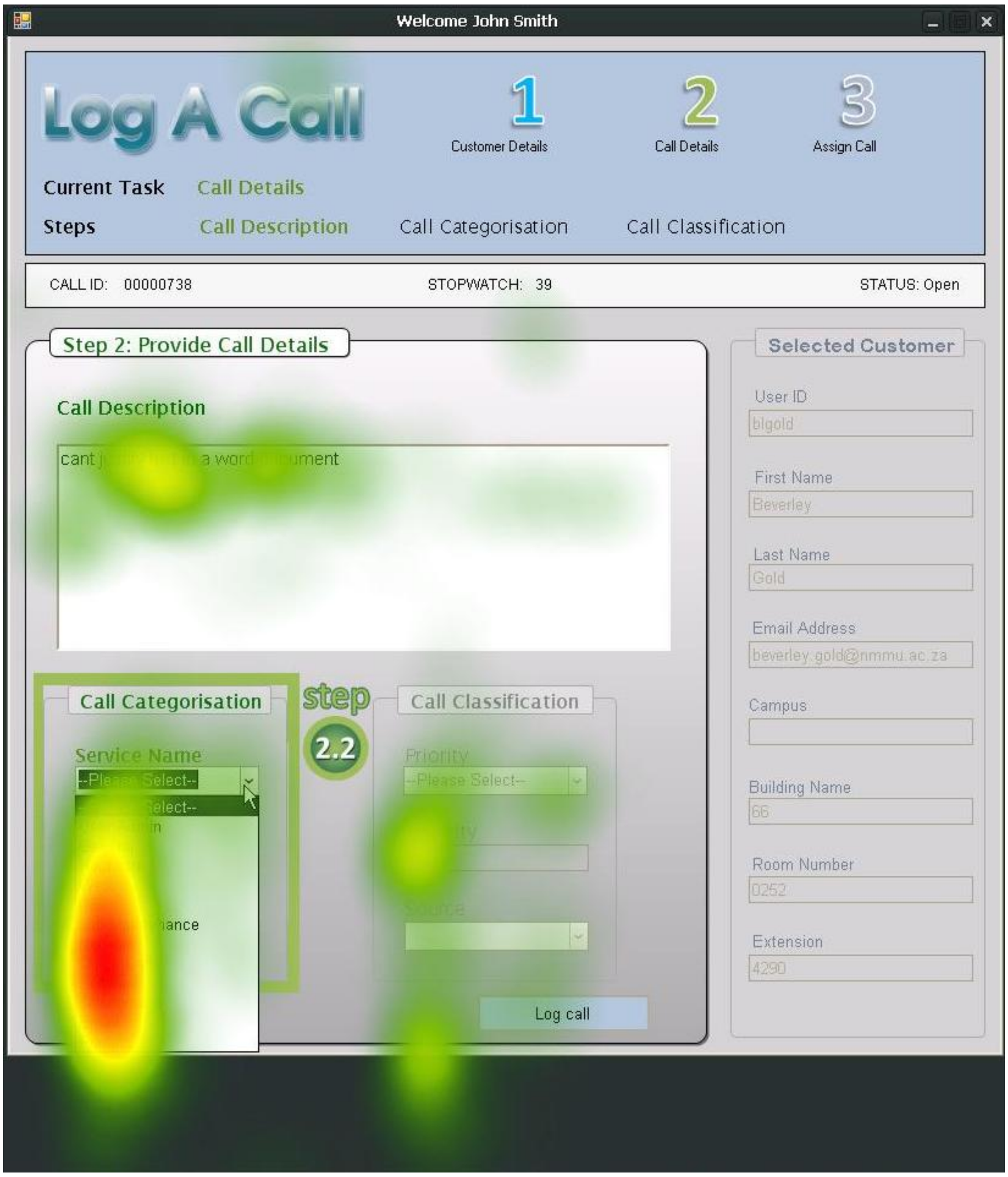


Figure 7.20: Novice UI Heat Map - Fixation Duration of Task 4 step 2

(Max Absolute Duration = 75.24 sec, N=30)

The heat map, represented in Figure 7.20, visualises the fixation duration of all the participants (n=30) for the second step (*provide call details*) of the fourth task performed on the novice UI. Figure 7.20 reveals that when performing task 4, unlike task 1 (Figure 7.19), participants spent some time fixating on the Title and Task Description section.

Participants, however, still focussed more on the User Input section and, in particular, appeared to have fixated on the same specific areas that they fixated on in task 1 (Figure 7.19). There did, however, appear to be a shorter fixation duration in Figure 7.20 than there was in Figure 7.19 (see intensity of heat mask), indicating that participants had become more familiar with the novice UI. These assumptions are further supported by comparing the fixation duration times (Figure 7.21) and the observation count (Figure 7.22) of tasks 1 and 4 of the novice UI.

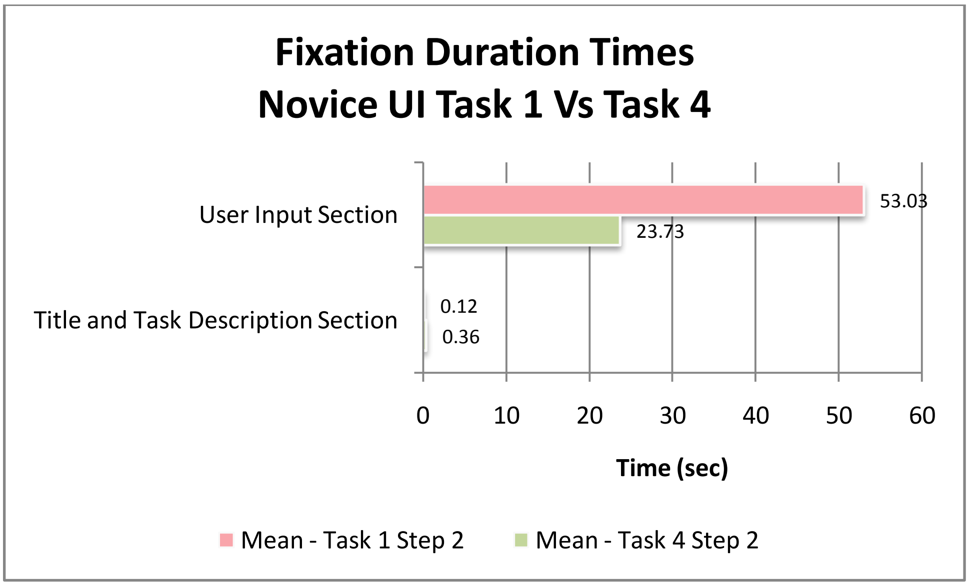


Figure 7.21: Fixation Duration Times for Novice UI Task 1 and 4 Step 2 (n=30)

The fixation duration times on the novice UI for tasks 1 and 4 of the second call logging step (Figure 7.21), indicated that participants initially spent more time in the User Input section of the first task (mean = 53.03 sec, median = 52.4 sec) than they did in the User Input section of the fourth task (mean = 23.73 sec, median = 24.24). A comparison of the mean values in Figure 7.24 between the first and fourth tasks performed on the novice UI revealed a 45 percent reduction in time in which participants fixated on the User Input Section. This implies that the test participants were more familiar with the novice UI when they performed the fourth task than they were when they performed the first task. The fixation duration times on the novice UI for tasks 1 and 4 of the second call logging step (Figure 7.21) also indicated that the test participants hardly noticed the Title and Task Description section of the UI (task 1 mean = 0.12 sec, task 4 mean = 0.36 sec).

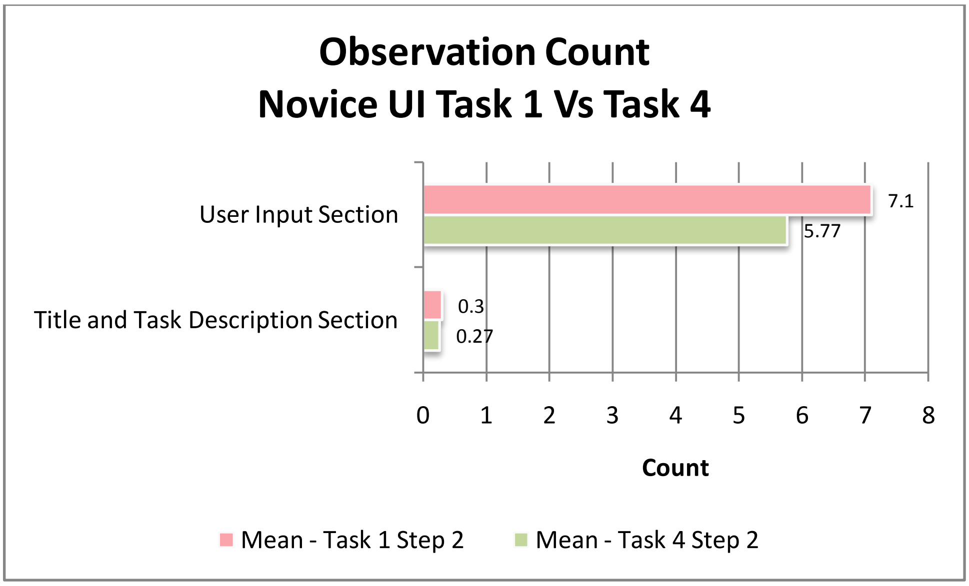


Figure 7.22: Fixation Duration Times for Novice UI Task 1 and 4 Step 2 (n=30)

The observation count data of the two identified AOIs for both tasks 1 and 4 (Figure 7.22) were compared. The observation count data provided an indication of how many times the test participants looked at a particular AOI. Participants (n=30) visited the User Input section (task 1 mean = 7.1 visits, task 4 mean = 5.77 visits) more than they did the Title and Task Desrcription section (task 1 mean = 0.3 visits, task 4 mean = 0.27 visits) and this supports the previous fixation duration results, further proving that the Title and Task Description section was of little use to the participants. The observation count data also indicated that the test participants paid fewer visits to the User Input section when performing task 4 (mean = 5.77 visits, median = 5 visits) than they did when performing task 1 (mean = 7.1 visits, median = 6.5 visits), further supporting the previous fixation duration results.

The heat map, represented in Figure 7.23, visualises the fixation duration of all the participants (n=30) for the first (*identify the customer*) and second step (*provide call details*) of the fourth task performed on the expert UI. Figure 7.23 reveals that once again, participants (n=30) spent no time fixating on the Title and Task Description section, but rather focussed more on the User Input section. Participants, in particular, appeared to have fixated on specific areas on the User Input section of the expert UI and in comparison with tasks 1 (Figure 7.19) and 4 (Figure 7.20) of the novice UI, the expert UI entailed more intense heat masks. The expert UI entailed adaptive lists, and, as depicted in Figure 7.23, the intensity of the heat mask superimposing the adaptive list is greatest towards the top section of the adaptive list. The static lists depicted in Figure 7.19 and 7.20 visualised the entire list as highly intense; therefore, the fact that only the top section of the adaptive list was visualised as being highly intense implies that the participants fully utilised the most commonly used items located in the top section of the adaptive list.

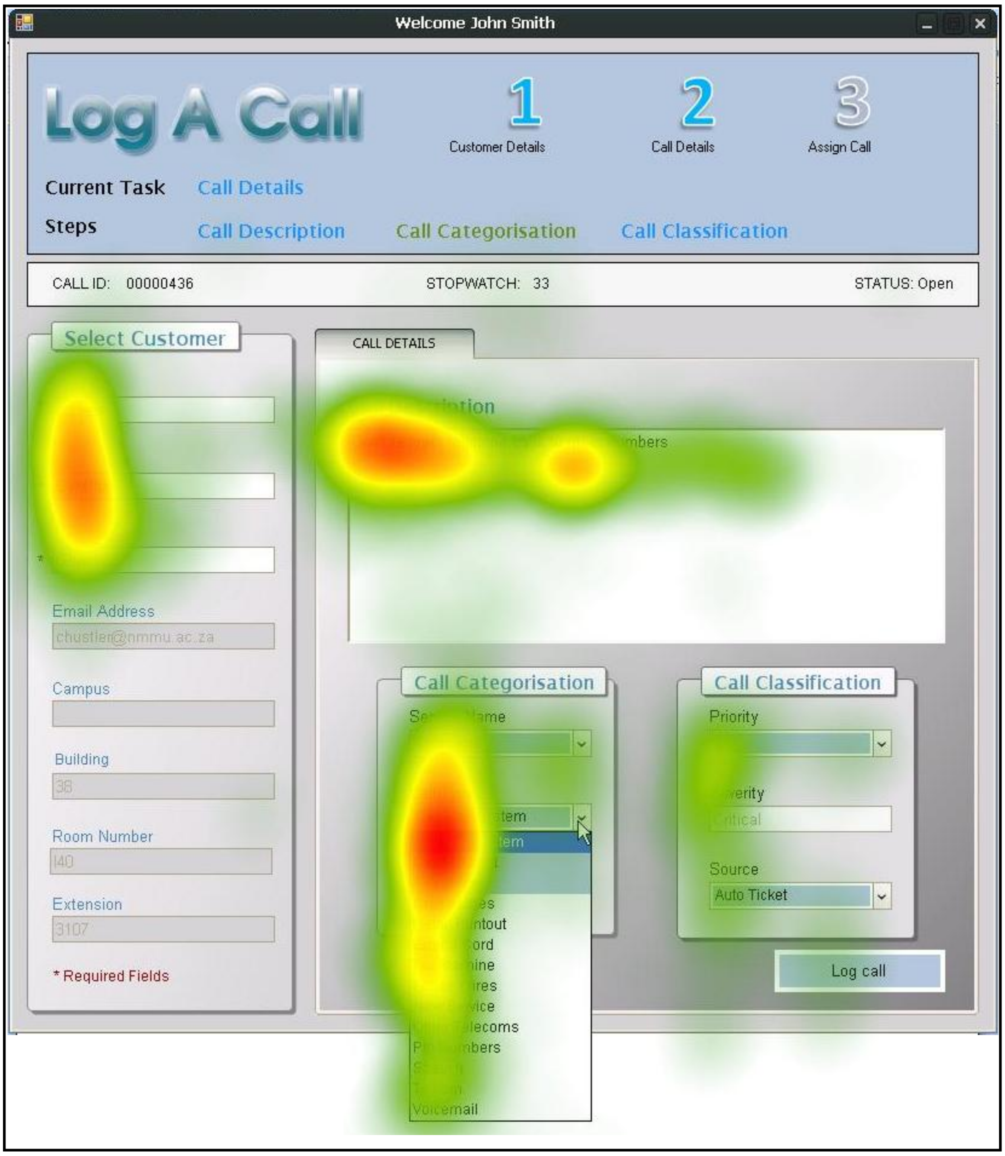


Figure 7.23: Expert UI Heat Map – Fixation Duration of Task 4 Step 2
(Max Absolute Duration = 59.01 sec, N=30)

Eye-Tracking Contribution

This sub-section analysed eye-tracking results to determine the effectiveness of the adaptation from the novice UI to the expert UI. Some key eye-tracking findings have been identified. Firstly, the time to first fixation results indicated that participants noticed the Visual Step Indicator immediately; however, results did suggest a need for some improvement. The support provided by the Visual Step Indicator was, however, extremely evident in the heat maps of the novice and expert UI. All heat maps consisted of an intense heat map mask at designated areas of the interface, and this is a clear indication that the support provided by the Visual Step Indicator was highly effective. Another key eye-tracking finding indicated the clear evidence of the learning provided at the novice UI by the reduction of fixation times on the novice UI. This reduction in fixation times was significant because it indicates that participants' familiarity with the novice UI had improved. The eye-tracking results also showed the effectiveness of the adaptive lists provided on the expert UI. A key eye-tracking finding with regards to the interface element of the proposed AUI model showed that all participants identified the User Input section AOI but hardly noticed the Title and Task Description section AOI; thus, this section could be deemed unhelpful, which strongly motivates the need for modification of the proposed AUI model.

7.5. Conclusions

The evaluation outlined and discussed in this chapter showed that through the ASD prototype implemented in Chapter 5, the proposed AUI model (Chapter 6) could assist with the development of an AUI for CC operations and successfully assist a novice CCA to become more experienced.

Existing AUI evaluation approaches suggest evaluating an AUI in layers (Section 7.2). Different researchers have proposed different levels of granularity (layers) and, depending on the layers identified, different evaluations needed to be considered. An evaluation strategy for the proposed AUI model was identified. The proposed AUI model will, therefore, follow a layered evaluation approach where it will consist of three evaluation layers: the Data Layer, the Inference Layer and the User Interaction Layer (Section 7.3).

The Data Layer was evaluated in the pilot study, conducted in Chapter 6, and this evaluation confirmed the validity of the data. The Inference and the User Interaction Layers were evaluated in this chapter through usability testing.

The objectives of the usability study were specified in terms of the Inference and User Interaction evaluation layers. The objective of the Inference Layer was to determine the accuracy of the adaptation timing. The goal of the User Interaction Layer was to measure the user's experience in terms of satisfaction, learnability and effectiveness. Eye-tracking was a complimentary evaluation technique used to measure effectiveness.

User testing involved simulating a CC environment and giving participants CC calls to log. A novice and expert UI test plan (Appendix F) was constructed in order to log calls via the novice and expert UIs respectively. All participants were given a mandatory amount of four calls to log via both the novice and expert UI, where the maximum amount of calls given to participants to log via the novice UI was eight. Data was obtained from the post-test questionnare as well as by obtaining participants' performance data stored in the Knowledge Base.

Participants selected for the usability study (Section 7.4.5) comprised CCAs, both first- and second-level support, and students and staff within the CS & IS Department at NMMU. All participants were required to have a sound backgound in IT.

Results of user testing in terms of the Inference evaluation layer, showed that the test participants felt that the adaptation accurately reflected their expertise level (Section 7.4.6.1). The Inference Layer results, in particular, showed high accuracy results (precision=90 percent), and this strongly indicates that the proposed AUI model is highly accurate when inferring a CCA's expertise level. Results of user testing in terms of the User Interaction evaluation layer were with regard to learnability (Section 7.4.6.2) and satisfaction (Section 7.4.6.3).

Learnability results were significant. Major findings indicated that the learning which occured on the novice UI was quick: the majority of participants encountered an Adaptation Moment after logging only four tasks on the novice UI. Major findings also indicated a significant difference in participants' performance times from using the novice UI to using the expert UI. A *t*-test was used to confirm the difference. Participants spent less time on the expert UI and this implies that there was significant learning performed on the novice UI.

Satisfaction results were both quantitative (Section 7.4.6.3.1) and qualitative (Section 7.4.6.3.2). Quantitative satisfaction results ranged between 4 and 5 using a five-point Likert scale. Quantitative satisfaction results produced major findings, showing that participants were highly satsified with the design of the novice and expert UIs. Participants were also highly satisfied with the adaptation from the novice UI to the expert UI. The support provided by the Visual Step Indicator was highly rated, and the aid that adaptive lists provided at the expert UI also produced significant satisfaction results. Qualitative satisfaction results further supported the quantitative satisfaction results. These significant results highlight the CCAs' interaction with the UI was highly satisfactory.

Effectiveness results were obtained through eye-tracking and this was conducted to support previous results. Eye-tracking results further showed the effectivenes of the adaptation from the novice UI to the expert UI. Major findings indicated that the support provided by the Visual Step Indicator on the novice UI was highly effective. A reduction in fixation times further supported the amount of learning done on the novice UI.

The next section concludes this dissertation, summarising its contribution and achievements, as well as outlining possibilities for future research.

Chapter 8: Recommendations and Conclusions

8.1. Introduction

The aim of this dissertation was to develop an AUI model that could be used to accomodate the varying UI skill levels of CCAs to improve CC operations. In order to demonstrate its effectiveness, this model was implemented as a prototype and was named the AdaptiveServiceDesk (ASD). The prototype was then evaluated to determine its efficiency and usability. This chapter will reflect on the research contributions and provide recommendations for future research.

8.2. Thesis Statement Revisited

The thesis statement of this research study, as stated in Section 1.3.2 was:

> *An AUI model that can be adapted according to a CCA's UI skill level can improve that CCA's productivity and enhances the CCA's interaction with the UI.*

An AUI model for CCs was designed (Chapter 5) and implemented (Chapter 6). Significant findings of the main evaluation conducted in Chapter 7 of this dissertation, revealed that:

- There is a significant difference in participants' performances on the novice and expert UIs; and
- Participants were highly satisfied with their interaction with the novice UI and the expert UI, and found the adaptation from the novice UI to the expert UI extremely useful.

It can, therefore, be concluded, that, based on the research conducted in this dissertation, the thesis statement, which was put forward, has been successfully justified.

8.3. Research Objectives Revisited

The following research objectives were identified in Section 1.3.4:

- An investigation into CCs (Chapter 2);

- An investigation into the different user expertise levels and UIs which facilitate these levels (Chapter 3);
- An investigation into AUIs (Chapter 4);
- An investigation into existing AUI models (Chapter 5);
- Designing an AUI model for CC operations (Chapter 5);
- The implementation of a prototype to evaluate the proposed AUI model (Chapter 6); and
- The evaluation of the proposed AUI model (Chapter 7).

These objectives were organised into research questions (Table 1.1) and each of these were then analysed and discussed in the preceding chapters. By meeting these research objectives, this study has made both a theoretical and a practical contribution. These theoretical contributions are identified in Section 8.4.1 and practical ones in Section 8.4.2. Recommendations for future research are provided in Section 8.6.

8.4. Research Contributions

This section examines the theoretical and practical contributions that this research can make to the field of training CCAs to perform their tasks more effectively and efficiently.

8.4.1. Theoretical Contributions

The theoretical contributions of this research are twofold. The primary theoretical contribution is the proposed AUI model designed especially for CC operations, discussed in Section 8.4.1.1. The second is the evaluation strategy of the proposed AUI model, discussed in Section 8.4.1.2.

8.4.1.1. AUI Model Design

The main theoretical contribution provided by this dissertation is the proposed AUI model which came about as a result of an extensive literature study. The theory of model-design was first examined before an investigation into existing AUI models could take place. Existing AUI models were investigated on the basis of criteria established according to the literauture study conducted in Chapter 4. Due to the investigation into model-design, it was concluded that existing AUI models cannot be viewed as complete models. Existing AUI models are also not specifically designed to the domain of CCs and did not satisfy all criteria stipulated. An IUI model, however, had been proposed and successfully implemented in the CC domain, and this contributed to the proposal of an AUI model for CCs. The proposed AUI model comprised three components:

1. An architectural element;
2. A component-level element; and
3. An interface element.

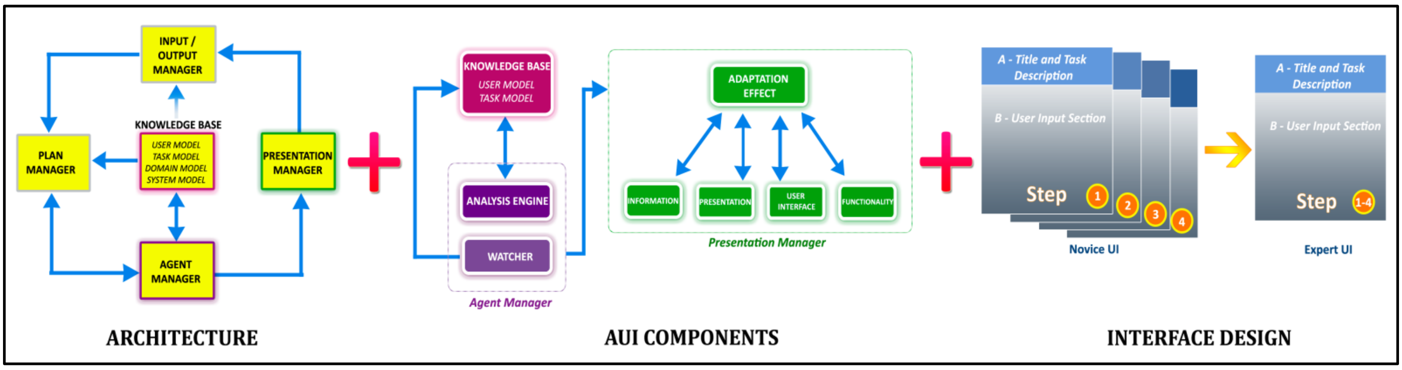


Figure 8.1: Proposed AUI Model

The architectural element of the proposed AUI model, as depicted in Figure 8.1, entailed the same one presented in Singh's (2007) IUI model. Further contributions to the proposed AUI model comprised the component-level and interface elements.

The component-level element of the proposed AUI model comprised AUI components (Figure 8.1) and was derived from the literature study presented in Chapter 4 and the investigation into existing AUI models. The component-level element details three architectural components: the Knowledge Base, Agent Manager and Presentation Manager.

The Knowledge Base AUI component utilised a User Model and Task Model which store the necessary user information required for adaptation. The Analysis Engine and Watcher AUI components represented the Agent Manager architectural components. The Watcher AUI component is responsible for acquiring implicit information from the user and storing this information in the Knowledge Base. The Analysis Engine contained the rules required for the Adaptation Effect AUI component. The rules within the Analysis Engine were derived from statistical results obtained from a pilot study. The Adaptation Effect AUI component represented the Presentation Manager architectural component and provided an adaptation effect to the User Interface, which provided an adaptation from a UI designed for novice users to one designed for expert users.

The Adaptation Effect provided a User Interface adaptation and the design of the interface was provided by the interface design component of the proposed AUI model. The interface design formally specified the design of two interfaces; namely, the novice UI and the expert UI. The structure of the novice and expert UI were derived from the literature study provided in Chapter 3. The interfaces of both the novice and expert UI were divided into two sections: the Title and Task Description section and the User Input section, and these two sections were adapted from the interface element of the IUI model proposed by Singh (2007).

8.4.1.2. Evaluation Strategy

The secondary theoretical contribution entailed an evaluation strategy for the proposed AUI model (Figure 8.2). An investigation into evaluation methods of AUIs revealed that they need to be evaluated in layers. Various evaluation layers were identified in literature, and three evaluation layers adopted from Weibelzahl (2002) were deemed as most appropriate for this research study. These were the Evaluation of Input Data, Evaluation of Inference and Evaluation of Total Interaction. These were renamed in terms of the layers they represented for the proposed AUI model. Evaluation of Input Data layer was renamed the Data Layer, Evaluation of Inference was renamed the Inference Layer and Evaluation of Total Interaction was renamed the User Interaction Layer.

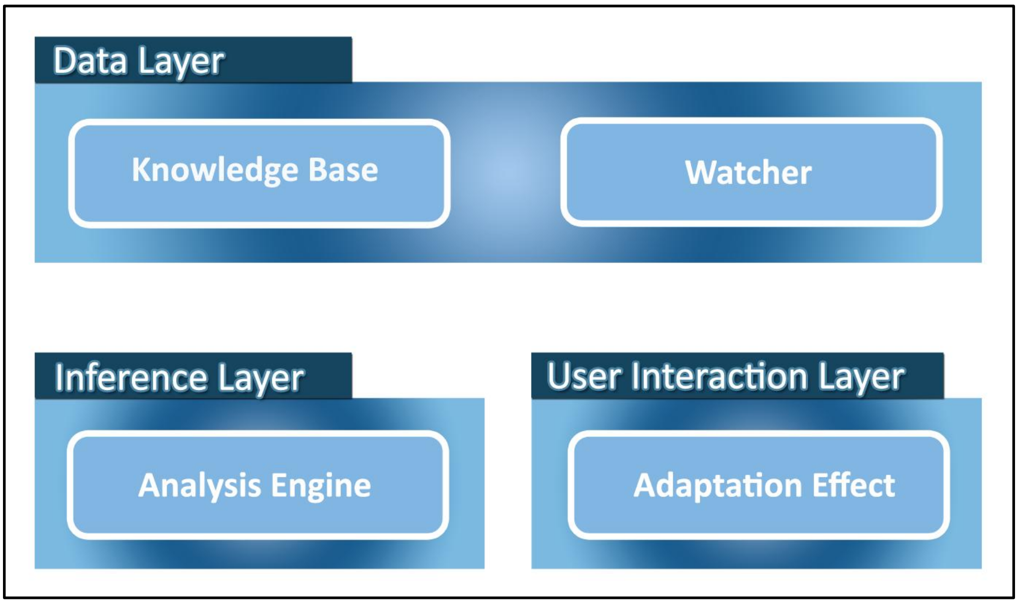


Figure 8.2: The Correspondence between Evaluation Layers and AUI Components of Proposed Model

The Data Layer of the proposed AUI model was responsible for validating the data which was obtained from the Watcher and stored in the Knowledge Base. The Data Layer, therefore, utlised the Knowledge Base and Watcher AUI components of the proposed AUI model. The pilot study conducted in Section 6.5 was responsible for evaluating the Data Layer.

The Inference Layer of the proposed AUI model was responsible for validating the inference made by the Analysis Engine. The Inference Layer, therefore, utilised the Analysis Engine AUI component of the proposed model. Usability testing conducted in Chapter 7 was responsible for evaluating the Inference Layer by measuring the accuracy of adaptations made.

The User Interaction Layer of the proposed AUI model was responsible for evaluating the user experience of the Adaptation Effect; therefore it utilised the Adaptation Effect AUI component. Usability testing, conducted in Chapter 7 was responsible for evaluating the User Interaction Layer by measuring the learnability, satisfaction and effectiveness of adaptations made.

8.4.2. Practical Contributions

The primary practical contribution provided by this dissertation entailed the implementation of a prototype, the ASD, as proof-of-concept for the proposed AUI model. The secondary practical contribution provided by this dissertation entailed the practical evaluation of the ASD prototype. The primary and secondary practical contributions are further discussed in Sections 8.4.2.1 and 8.4.2.2 respectively.

8.4.2.1. Practical Implementation of Prototype

Implementation of the ASD prototype was done to prove that the proposed AUI model could be used to develop AUIs for CC operations. The implementation of the ASD prototype comprised the AUI components and the interface design element of the proposed AUI model. Implementation of the Knowledge Base components, namely the User and Task Models, were conducted in XML. Implementation of the User Model, in particular, entailed storing the informative moments (IM). Implementation of the Watcher involved updating the User and Task Models. Implementation of the interface design element consisted of two types of interfaces; the novice and expert UI. These UIs were implemented according to the design presented in Section 5.3.2. The Title and Task Description section was implemented to provide task support and the User Input section comprised the direct manipulation part of the interface.

The novice UI provided additional task support in the form a Visual Step Indicator, which guided the user step-by-step through the process of logging a customer's query. The step-by-step guidance was implemented based on the literature study conducted in Chapter 3. Implementation of the expert UI included accelerators in the form of pop-up and adaptive lists. These accelerators were implementated based on the literature study conducted in Chapters 3 and 4.

Implementation of the Analysis Engine was directly affected by the practical evaluation of the prototype. Results of the practical evaluation of the Data Layer provided the rules required to adapt the interface from the novice UI to the expert UI. A discussion on the practical evaluation of the ASD prototype follows.

8.4.2.2. Practical Evaluation of Prototype

The evaluation strategy (Section 7.3) identified three layers: the Data, Inference and User Interaction layers which required evaluation. A pilot study, discussed in Section 6.5, evaluated the Data Layer and confirmed the validity of the data. The Inference and User Interaction evaluation layers were evaluated in Chapter 7 through user testing.

The user testing was conducted in a controlled environment to measure the usability of the ASD prototype. The Inference Layer measured the accuracy of the inference made by the Analysis Engine. Results of the user testing indicated that users felt that the inference (adaptation from novice UI to expert UI) was an accurate reflection of their expertise. The User Interaction Layer measured the users' experiences with regards to their satisfaction, the learnability of the ASD prototype and its effectiveness. Quantitative satisfaction results of the user testing indicated that users were highly satisfied with the ASD prototype. Qualitative satisfaction results further showed that users were highly satisfied with the learning provided by the novice UI. Learnability results of the user testing indicated a significant increase in users' performances when they were transitioned to the expert UI. Effectiveness results, through eye-tracking, confirmed the usefulness of the Visual Step Inidicator provided at the novice UI and further supplemented the satisfaction and learnability results.

8.5. Limitations of Research

The research conducted in this dissertation was not without its limitations. Recruiting test participants considered to be novice users is normally much easier than recruiting ones considered to be expert users and this was evident in this research study. The pilot study (evaluation of the Data Layer) conducted in Chapter 6 required the participation of both novice and expert users so that a transitioning period could be determined, based on the overall performance data of both kinds of users. The scope of this research was limited to the NMMU ICT Service Desk and the CCAs working there were considered to be expert users. Telkom SA Ltd. and Dimension Data CCs were initially investigated to be used in this research, however, due to the confidentiality of these CCs' data, the NMMU ICT Service Desk was selected. This resulted in a limited number of expert users available for selection. All Eight participants were eventually selected as expert users for the pilot study. A bigger sample size would have encouraged a more detailed statistical analysis; therefore, more significant results could have been obtained.

During the pilot study, the transitioning period was determined based on tasks given to both the novice and expert users. The number of these tasks was limited to two per test participant. This was because participation in the pilot study took CCAs an hour away from their busy working environments. More tasks could possibly have provided a more accurate transitioning period. It must, however, be noted that usability testing conducted in Chapter 7, which used the data obtained from the pilot study, resulted in 90 percent of participants acknowledging that the transitioning period identified was an accurate reflection of their expertise.

8.6. Recommendations

The research conducted in the dissertation has enabled various recommendations with regards to theory (Section 8.7.1), practice (Section 8.7.2) and future research (Section 8.7.3) to be made.

8.6.1. Recommendations for Theory

The evaluation of the User Interaction Layer revealed that participants generally did not utilise the task support provided by the Title and Task Description section of the interface. Therefore, the interface design element of the proposed model could possibly be modified to exclude the Title and Task Description section from both novice and expert UIs.

The evaluation of the User Interaction Layer revealed that some participants felt the change from a novice UI to a expert UI was too sudden. The interface design element of the proposed model could, therefore, be possibly modified to include an intermediary UI positioned between the novice and expert UI. This modification enables a more gradual transitioning for CCAs.

The User Model stored in the Knowledge Base contains data about the Informative Moments (IMs) and the potentially Predictive Features associated with them. If more tasks were used to build the User Model, and more expert users acquired, more detailed statistical analsys could be performed to determine whether certain IMs, or even PFs, could be excluded from the User Model.

8.6.2. Recommendations for Practice

This research has proposed an AUI model for CC operations. The model comprised a architectural, component-level and interface design element. These components are quite generic and could be used as specifications to build AUI applications within other domains.

8.6.3. Recommendations for Future Research

The research tracked the transitions of CCAs from using the novice UI to using the expert UI once they had reached a certain skill level. This research only provides adaptation to the CCA’s UIs if they had improved their skill level and does not provide adaptation to the CCA’s UIs if they were performing worse at their specified skill level. Future research could cater for this scenario.

This research study only utilised the User Interface element of the Adaptation Effect AUI component. Future work could entail adaptation to presentation, information and functionality; hence, utilising the Presentation, Information and Functionality elements of the Adaptation Effect AUI component. The presentation of the CC information could possibly be adapted to the CCAs' skill levels where novice CCAs could, perhaps, be provided with a larger size text. The CC information, itself, could be adapted to the CCAs' skill levels so that novice CCAs receive more detailed information than expert CCAs, and functionality could be adapted, possibly by incorporating plan recognition.

The scope of this research was limited to a relatively small CC; therefore, a limited amount of CCAs was available to be utilised to build the User Model. Future work could involve a research study on a larger-sized CC so that a more comprehensive User Model could be built.

The functionality that CC applications provide can become quite extensive. This research was limited to the logging of customer queries. Future research could entail a detailed task analysis on CC applications in order to identify possible areas which can benefit from including adaptation at the interface.

This research employed stereotypes as the user modelling technique used to build the User Model providing the adaptation. The stereotype user-modelling technique that was used employed a simple, rule-based approach. AUIs have utilised other user modelling techniques (Section 4.5.2.2). Future work could, therefore, investigate building the User Model using machine learning or predictive statistical model techniques

8.7. Summary

The goal of this research was to develop an AUI model for CCs to improve CCAs' productivity and usability. The goal was successfully achieved, both theoretically, with the proposal of the AUI model, and practically, with the implementation of a prototype serving as proof-of-concept for the proposed AUI model. There have been limitations while conducting this research; therefore, future research ideas have outlined possibilities available to provide improvements to the AUI model and prototype.

Bibliography

ALBRECHT, D. W. & ZUKERMAN, I. (2001) Predictive Statistical Models for User Modeling. *User Modeling and User-Adapted Interaction 11*, pp. 5-18.

ALVAREZ-CORTES, V., ZAYAS-PEREZ, B. E., ZARATE-SILVA, V. H. & URESTI, J. A. R. (2007) Current Trends in Adaptive User Interfaces: Challenges and Applications. *Electronics, Robotics and Automotive Mechanics Conference*. pp. 312-317.

AUSTRALIAN NATIONAL AUDIT OFFICE (1996) Telephone Call Centres [online]. Available at: http://www.anao.gov.au/uploads/documents/Telephone_Call_Centres.pdf (Last Accessed: 16 October 2008).

AVAYA (2005) Next-Generation Customer Contact: Viewpoints from the Avaya Customer Contact Council [online]. Available at: http://www.avaya.com/gcm/master-usa/en-us/wowtopics/callctrvscontactctr/callctrvscontactctr.htm (Last Accessed: April 2007).

BALDWIN, J. F., MARTIN, T. P. & TZANAVARI, A. (2000) User Modelling Using Conceptual Graphs for Intelligent Agents. *Proceedings of the Linguistic on Conceptual Structures: Logical Linguistic, and Computational Issues.*

BENYON, D. & MURRAY, D. (1993) Applying user modeling to human-computer interaction design. *Artificial Intelligence Review,* vol. 7**,** pp. 199-225.

BON, J. V., JONG, A. D. & KOLTHOF, A. (2007) Functions and Processes in Service Operation. *Foundations of IT Service Management Based on ITIL V3.* pp. 303.

BOURIE, P. Q., DRESCH, J. & CHAPMAN, R. H. (1997) Usability Evaluation of an On-line Nursing Assessment. *In Proceedings of AMIA Symposium*, pp. 914.

BROWN, C. M. (1998) Human-Computer Interface Design Guidelines. Intellect Books.4, pp. 65-78.

BRUSILOVSKY, P. & COOPER, D. W. (2002) Domain, Task, and User Models for an Adaptive Hypermedia Performance Support System. *IUI '02: Proceedings of the 7th international conference on Intelligent user interfaces.* San Francisco, California, USA, ACM Press, pp. 23-30.

BRUSILOVSKY, P., KARAGIANNIDIS, C. & SAMPSON, D. (2001) The benefits of layered evaluation of adaptive applications and services. *Empirical Evaluation of Adaptive Systems. Proceedings of workshop at the Eighth International Conference on User Modeling, UM2001*. pp. 1-8.

BUXTON, W. A. S., KURTENBACH, G. P. & SELLEN, A. J. (1993) An empirical evaluation of some articulatory and cognitive aspects of "marking menus". *Human Computer Interaction,* vol. 8 (1)**,** pp. 1-23.

CALVERT, N. (2004) *Gower Handbook of Call and Contact Centre Management.* pp. 1-340.

CARROLL, J. M. & CARRITHERS, C. (1984) Training wheels in a user interface. *Commun. ACM,* vol. 27**,** pp. 800-806.

CENA, F., CONSOLE, L., GENA, C., GOY , A., LEVI, G., MODEO, S. & TORRE, I. (2006) Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide. *AI Communications,* vol. 19**,** pp. 369-384.

CLARK, B. & MATTHEWS, J. (2005) Deciding Layers: Adaptive Composition of Layers in a Multi-Layer User Interface. *Proceedings of 11th International Conference on Human-Computer Interaction,* vol. 7.

CROW, D. & SMITH, B. (1993) The role of built-in knowledge in adaptive interface systems. *Proceedings of the 1st International Conference on Intelligent User Interfaces*. ACM. pp. 97-104.

DIETERICH, H., KÜHME, T., MALINOWSKI, U. & SCHNEIDER-HUFSCHMIDT, M. (1993) State of the Art in Adaptive User Interfaces. *Adaptive User Interfaces.* Amsterdam, North-Holland, Elsevier Science Publishers B.V., pp. 11-48.

DILLON, A. & SONG, M. (1997) An empirical comparison of the usability for novice and expert searchers of a textual and a graphic interface to an art-resource database. *Journal of Digital Information,* vol. 1 (1).

DIMENSION DATA (2006) *Merchants Global Contact Centre Benchmarking Report 2006.* Dimension Data. pp. 1-197.

DOMIK, G. O. & GUTKAUF, B. (1994) User Modeling for Adaptive Visualization Systems. *VIS '94: Proceedings of the conference on Visualization '94.* Washinton, D.C., USA, IEEE Computer Society Press, pp. 217-223.

EGAIN (2005) 7 Habits of Highly Effective Contact Centers and Help Desks [online]. Available at: http://i.i.com.com/cnwk.1d/html/itp/7habits_contact_centers_help_desks.pdf (Last Accessed: 2 October 2008).

EHLERT, P. A. M. (2003) Intelligent user interfaces: introduction and survey. Mediamatics / Data and Knowledge Systems group, Delft University of Technology. SYSTEMS, D. O. I. T. A. Research Report DKS03-01 / ICE 01. Available at: http://www.kbs.twi.tudelft.nl/docs/report/DKS03-01.pdf (Last Accessed: 20 October 2008).

FINDLATER, L. & MCGRENERE, J. (2004) A comparison of static, adaptive, and adaptable menus. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM New York, NY, USA. pp. 89-96.

FINDLATER, L. & MCGRENERE, J. (2008) Impact of Screen Size on Performance, Awareness, and User Satisfaction with Adaptive Graphical User Interfaces. *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. ACM. pp. 1247-1256.

FISCHER, G. (2001) User Modeling in Human-Computer Interaction. *User Modeling and User-Adapted Interaction (UMUAI),* vol. 11**,** pp. 65-86.

FLUSS, D. (2005) The Real-Time Contact Center: Strategies, Tactics, and Technologies for Building a Profitable Service and Sales Operation. AMACOM, pp. 14-18.

FRANTZ, M. G. (2006) The Evolution of the IT Help Desk to the Service Desk [online]. Available at: http://www.thinkhdi.com/hdi2006/files/CompuComServiceDeskWhitePaper-01-06.pdf (Last Accessed: 17 October 2008).

FRONTRANGE.COM (2008) [online]. Available at: www.frontrange.com (Last Accessed: 13 December 2008).

GAJOS, K., CZERWINSKI, M., TAN, D. & WELD, D. (2006) Exploring the Design Space for Adaptive Graphical User Interfaces. *Proceedings of Advanced Visual Interfaces (AVI)*.

GAJOS, K., EVERITT, K., WELD, D. S., TAN, D. S. & CZERWINSKI, M. (2008) Predictability and Accuracy in Adaptive User Interfaces. *CHI 2008 Proceedings,* pp. 1271-1274.

GALITZ, W. O. (2007) *The Essential Guide to User Interface Design.* Third ed., Wiley Publishing Inc., pp. 1-90.

GANS, N., KOOLE, G. & MANDELBAUM, A. (2003) Telephone Call Centers: Tutorial, Review, and Research Prospects. *MANUFACTURING SERVICE OPERATIONS MANAGEMENT,* vol. 5**,** pp. 79-141.

GANS, N. & ZHOU, Y.-P. (2002) Managing Learning and Turnover in Employee Staffing. *Oper. Res.,* vol. 50**,** pp. 991-1006.

GENA, C. (2005) Methods and techniques for the evaluation of user-adaptive systems. *Knowledge Engineering Review,* vol. 20**,** pp. 1-37.

GHORBANI, A. A. & ZHANG, J. (2007) GUMSAWS: A Generic User Modeling Server for Adaptive Web Systems. *Proceedings of the Conference on Communication Networks & Services Research (CNSR 2007)*. Fredericton, Canada.

GLINER, J. A. & MORGAN, G. A. (2000) *Research Methods in Applied Settings.* Lawrence Erlbaum Associates. pp. 1-465.

GREAT BRITAIN TREASURY, CENTRAL COMPUTER AND TELECOMMUNICATIONS AGENCY, GREAT BRITAIN OFFICE OF GOVERNMENT COMMERCE & OFFICE OF GOVERNMENT COMMERCE (2000) Implementing the Service Desk infrastructure. *Best Practice for Service Support.* pp. 37-39.

GREENBERG, S. & WITTEN, I. H. (1985) Adaptive personalized interfaces: A question of viability. *Behaviour & Information Technology,* vol. 4**,** pp. 31-45.

HEATHCOTE, P. M. (2003) Training. *A-Level Information and Communication Technology*.pp. 265.

HERLOCKER, J. L., KONSTAN, J. A., TERVEEN, L. G. & RIEDL, J. T. (2004) Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.,* vol. 22**,** pp. 5-53.

HOFSTEE, E. (2006) *Constructing a Good Dissertation: A Practical Guide to Finishing a Masters, MBA or PhD on Schedule.* Johannesburg, South Africa, EPE.

HOOK, K. (2000) Steps to take before intelligent user interfaces become real. *Interacting with Computers,* vol. 12**,** pp. 409-426.

HORGEN, S. A. (2001) Essay on Intelligent User Interfaces and adaptivity. *Division of Intelligent Systems, NTNU*.

HORVITZ, E. (1999) Uncertainty, Action, and Interaction: In Pursuit of Mixed-Initiative Computing. *IEEE Intelligent Systems,* vol. 14**,** pp. 17-20.

HURST, A., HUDSON, S. E. & MANKOFF, J. (2007) Dynamic detection of novice vs. skilled use without a task model. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

INSTRANET, I. (2003) The Top 10 Information Challenges in today's Call Centre. Research Report Available at: http://www.inst-informatica.pt/servicos/informacao-e-documentacao/biblioteca-digital/areas-aplicacionais/crm/CRM_The%20Top%2010%20Information%20Challenges%20in%20todays%20call%20center.pdf (Last Accessed: 13 December 2008).

INTERNATIONAL STANDARDS ORGANISATION - ISO9241-11 (1998) Ergonomic Requirements for office work with visual display terminals (VDTs), Part 11: Guidance on usability.

JACOB, R. J. K. & KARN, K. S. (2003) Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises. *The Mind's eye: Cognitive The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research,* pp. 603-573.

JAMESON, A. (2002) Adaptive Interfaces and Agents. *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications.* Lawrence Erlbaum Associates, Inc, pp. 305-330.

JETTMAR, E. (2000) Adaptive Interfaces: Effects on User Performance [online]. Available at: http://www.danger-island.com/true/papers/AdaptationJettmar.html (Last Accessed: April 2007).

KARAGIANNIDIS, C., KOUMPIS, A. & STEPHANIDIS, C. (1995) Supporting Adaptivity in Intelligent User Interfaces: The case of Media and Modalities Allocation. *1st ERCIM Workshop on User Interfaces for All (UI4ALL 95).* Heraklion, Greece.

KARAGIANNIDIS, C. & SAMPSON, D. G. (2000) Layered Evaluation of Adaptive Applications and Services. *Adaptive Hypermedia and Adaptive Web-Based Systems: International Conference, Ah 2000, Trento, Italy, August 28-30, 2000: Proceedings*. Springer.

KAY, J. (1993) Pragmatic User Modelling for Adaptive Interfaces. *HUMAN FACTORS IN INFORMATION TECHNOLOGY*, vol. 10**,** pp. 129-129.

KELLOGG, W. A. & BREEN, T. J. (1987) Evaluating user and system models: applying scaling techniques to problems in human-computer interaction. *SIGCHI Bull.*, vol. 17**,** pp. 303-308.

KOBSA, A. (1993) User Modeling: Recent work, prospects and hazards. IN MALINOWSKI, U., KUHME, T. & SCHNEIDER-HUFSCHMIDT, M. (Eds.) *Adaptive User Interfaces: Principles and Practice.* North-Holland.

KOBSA, A. (1995) Supporting User Interfaces for All Through User Modeling. *Proceedings HCI International '95,* pp. 155-157.

KOELLE, D. (2004) Intelligent User Interfaces [online]. Available at: http://www.davekoelle.com/intint-outline.jsp (Last Accessed: 17 August 2007).

KOOLE, G. & MANDELBAUM, A. (2001) Queueing Models of Call Centers An Introduction [online]. Available at: http://www.cs.vu.nl/obp/callcenters (Last Accessed: April 2007).

KROGSAETER, M. & THOMAS, C. G. (1994) Adaptivity: System-Initiated Individualization. IN OPPERMANN, R. (Ed.) *Adaptive User Support : Ergonomic Design of Manually and Automatically Adaptable Software.* New Jersey, USA, Lawrence Erlbaum Associates Inc., pp. 67-96.

KULES, B. (2000) User Modeling for Adaptive and Adaptable Software Systems [online]. Available at: http://www.otal.umd.edu/UUGuide/wmk/ (Last Accessed: 19 May 2007).

KUMAR, R. R. (2005) User Interface. *Human Computer Interaction.* Firewall Media.2, pp. 52-53.

LANGLEY, P. (1999) User Modeling in Adaptive Interfaces. *Proceedings of the Seventh International Conference on User Modeling.* Springer-Verlag, pp. 357-370.

LIVETIME.COM (2008) [online]. Available at: http://www.livetime.com/webservicedesk/app (Last Accessed: 19 October 2008).

LOPEZ-JAQUERO, V., MONTERO, F., MOLINA, J. P., FERNANDEZ-CABALLERO, A. & GONZALEZ, P. (2003) Model-Based Design of Adaptive User Interfaces through Connectors. *LECTURE NOTES IN COMPUTER SCIENCE*, pp. 245-257.

MANDELBAUM, A. (2004) Call centers (centres): Research bibliography with abstracts. *Version*, vol. 3.

MAYBURY, M. & WAHLSTER, W. (1998) Intelligent User Interfaces: An Introduction. *Readings in Intelligent User Interfaces.* San Francisco, Kaufmann, M, pp. 1-13.

MCLAUGHLIN, M. R. & HERLOCKER, J. L. (2004) A collaborative filtering algorithm and evaluation metric that accurately model the user experience. *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. pp. 329-336.

MCTEAR, M. F. (2000) Intelligent interface technology: from theory to reality? *Interacting with Computers,* vol. 12**,** pp. 323-336.

MICROSOFT (2002) Service Management Functions: Service Desk [online]. Available at: http://www.microsoft.com/technet/solutionaccelerators/cits/mo/smf/smfsvcdk.mspx (Last Accessed: 17 October 2008).

MITCHELL, J. & SHNEIDERMAN, B. (1989) Dynamic versus static menus: an exploratory comparison. *SIGCHI Bull.,* vol. 20**,** pp. 33-37.

MOHAGHEGHI, P. & AAGEDAL, J. (2007) Evaluating Quality in Model-Driven Engineering. *Proceedings of the International Workshop on Modeling in Software Engineering*. IEEE Computer Society. pp. 6.

NIELSEN, J. (1993) What is usability? *Usability Engineering.* San Francisco, Morgan Kaufmann, pp. 36-46.

OKA, M. & NAGATA, M. (1999) A graphical user interface shifting from novice to expert. *Proceedings of HCI International (the 8th International Conference on Human-Computer Interaction) on Human-Computer Interaction: Ergonomics and User Interfaces-Volume I - Volume I*. Lawrence Erlbaum Associates, Inc. pp. 341-345.

OLIVIER, M. S. (2004) *Information Technology Research - A Practical Guide for Computer Science and Informatics. Second Edition.* Pretoria, South Africa, Van Schaik Publishers.

OPPERMANN, R. (1994a) *Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software.* New Jersey, USA, Lawrence Erlbaum Associates Inc.

OPPERMANN, R. (1994b) Adaptively supported adaptability. *International Journal of Human-Computer Studies,* vol. 40**,** pp. 455-472.

PADILLA, M. (2003) Strike a balance: User's expertise on interface design [online]. Available at: http://www.ibm.com/developerworks/web/library/wa-ui/ (Last Accessed: 20 October 2008).

PARAMYTHIS, A., STEPHANIDIS, C. & TOTTER, A. (2001) A modular approach to the evaluation of Adaptive User Interfaces. IN CHIN, D., WEIBELZAHL, S. & WEBER, G. (Eds.) *Adaptive evolutionary information systems.* Sonthofen, Germany, pp. 9-24.

PARBHOO, S. (2002) CCMA National Call Centre [online]. Available at: http://www.ccma.org.za/DisplayNews2.asp?ID=8 (Last Accessed: March 2006).

PAZZANI, M., MURAMATSU, J. & BILLSUS, D. (1996) Syskill & Webert: Identifying interesting web sites. *In Proceedings of the Thirteenth National Conference on Artificial Intelligence,* pp. 54-61.

PECHENIZKIY, M., PUURONEN, S. & VASILYEVA, E. (2005) Towards the Framework of Adaptive User Interfaces for eHealth. *CBMS '05: Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems.* Washington, DC, USA, IEEE Computer Society, pp. 139-144.

PETT, M. A. (1997) *Nonparametric Statistics for Health Care Research.* SAGE. pp. 1-307.

PIETERSEN , J. (2008) Personal Communication. Senior lecturer and statistical consultant at the Nelson Mandela Metropolitan University.

POOLE, A. & BALL, L. J. (2005) Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects. *Encyclopedia of Human-Computer Interaction. Pennsylvania: Idea Group, Inc,* vol. 780.

PRESSMAN, R. S. (2005) *Software Engineering-A Practitioner's Approach. Sixth Edition.* McGraw-Hill.

PRETORIUS, M. (2005): The Added Value of Eye Tracking in the Usability Evaluation of a Network Management Tool. MSc Dissertation, Department of Computer Science and Information Systems, Nelson Mandela Metropolitan University, Port Elizabeth.

PRUMPER, J., FRESE, M., ZAPF, D. & BRODBECK, F. C. (1991) Errors in computerized office work: differences between novice and expert users. *SIGCHI Bull.*, vol. 23**,** pp. 63-66.

RATHOD, M. (2001) Customer Service: Now just a call away. *The Indian Edition: Network Magazine - Solutions for the competitive edge.*

REICHENBACHER, T. (2003) Adaptive Methods for Mobile Cartography. *Proceedings of the 21st International Cartographic Conference*. Durban, South Africa. pp. 1311-1323.

RICH, E. (1979): Building and Exploiting User Models. PhD thesis, Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pennsylvania.

RICH, E. (1998) User Modeling via Stereotypes. *Readings in intelligent user interfaces.* San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., pp. 329 - 342.

ROBERTSON, J. (2002) Knowledge management for call centres [online]. Available at: http://www.steptwo.com.au/files/kmc_callcentre.pdf (Last Accessed: 16 October 2008).

ROBERTSON, J. (2003) Knowledge management for front-line staff [online]. Available at: http://www.steptwo.com.au/files/kmc_frontline.pdf (Last Accessed: 16 October 2008).

ROBERTSON, J. (2005) Two-types of in-bound call centres [online]. Available at: http://www.steptwo.com.au/files/cmb_twocallcentres.pdf (Last Accessed: 16 October 2008).

ROSS, E. (2000) Intelligent User Interfaces: Survey and Research Directions. University of Bristol. Research Report CSTR-00-004. Available at: http://www.cs.bris.ac.uk/Publications/Papers/1000447.pdf (Last Accessed: 13 December 2008).

RUMSEY, D. J. (2003) *Statistics for Dummies.* Wiley Publishing, Inc. pp. 1-355.

SANDERSON, S. (2003) Organisation and People. *Introduction to Help Desk Concepts and Skills.* McGraw-Hill Osborne. pp. 27-53.

SAXENA, K. B. C. (1993) Computer-Aided Decision Supprt Engineering. IN HOLSAPPLE, C. W. & WHINSTON, A. B. (Eds.) *Recent Developments in Decision Support Systems.* Springer, pp. 325-326.

SCHUMACHER, R. (2004) Designing for Contact Centers. User Centric, Inc. Available at: http://www.technologyexecutivesclub.com/PDFs/ArticlePDFS/DesigningforContacCentersWhitePaper.pdf (Last Accessed: 13 December 2008).

SHARP, D. E. (2003) Introduction to Call Centres. *Call Center Operation : Design, Operation and Maintenance.*pp. 2-3.

SHNEIDERMAN, B. (2003) Promoting universal usability with multi-layer interface design. *Proceedings of the 2003 conference on Universal usability*. Canada. ACM. pp. 1-8.

SHNEIDERMAN, B. & MAES, P. (1997) Direct Manipulation vs. Interface Agents. *Interactions*, vol. 4**,** pp. 42-61.

SINGH, A. (2007): An Intelligent User Interface Model for Contact Centre Operations. MSc Dissertation, Department of Computer Science and Information Systems, Nelson Mandela Metropolitan University, Port Elizabeth.

SITEHELPDESK.COM (2008) [online]. Available at: http://www.sitehelpdesk.com/IT/Service-Management.htm (Last Accessed: 19 October 2008).

STEEL, A. (2003) Understanding and Enhancing Call Centre Computer- Human-Human Interaction. *CHI '03 extended abstracts on Human factors in computing systems*. pp. 690-691.

STEEL, A., JONES, M., APPERLEY, M. & JEHAN, T. (2002) The use of auditory feecback in call centre CHHI. *CHI '02 extended abstracts on Human factors in computing systems*. ACM. pp. 768-769.

SYSAID.COM (2008) [online]. Available at: http://www.sysaid.com/ (Last Accessed: 19 October 2008).

TIMBRELL, G., KOLLER, S. & LINDSTAEDT, S. N. (2005) A Knowledge Infrastructure Hierarchy Model for Call Centre Processes. *Call Centre Processes, in: J.UCS - Journal of Universal Computer Sciences*.

TORRE, I. (2002) Users Modeling for Adaptive Call Centers. *Adaptive Hypermedia and Adaptive Web-Based Systems.* pp. 603-607.

TOTTERDELL, P. & RAUTENBACH, P. (1990) Adaptation as a Problem of Design. IN BROWNE, D., TOTTERDELL, P. & NORMAN, M. (Eds.) *Adaptive User Interfaces.* pp. 79-83.

TSANDILAS, T. & SCHRAEFEL, M. C. (2005) An empirical assessment of adaptation techniques. *CHI '05 extended abstracts on Human factors in computing systems*. USA. ACM. pp. 2009-2012.

TULLIS, T. & ALBERT, B. (2008) *Measuring the User Experience: Collecting, Analysing, and Presenting Usability Metrics.* Morgan Kaufmann. pp. 1-317.

TYLER, S. W., COOK, L. K., GARGAN JR., R. A. & SCHLOSSBERG, J. L. (1991) An Intelligent Interface Architecture for Adaptive Interaction. IN TYLER, S. W. & SULLIVAN, J. W. (Eds.) *Intelligent User Interfaces.* New York, ACM Press, pp. 85-108.

VAN GREUNEN, D. (2002): Formal Usability Testing of Interactive Educational Software: A Case Study Approach. Masters Dissertation, University of Port Elizabeth, Port Elizabeth, South Africa.

VENTER, D. (2008) Personal Communication. Senior lecturer and statistical consultant at the Nelson Mandela Metropolitan University.

VERMAAK, R. (2008) Personal Communication. Manager at the Nelson Mandela Metropolitan University.

WAERN, A. (1997) What is an Intelligent Interface? [online]. Available at: http://www.sics.se/~annika/papers/intint.html (Last Accessed: 20 October 2008).

WEIBELZAHL, S. (2002): Evaluation of Adaptive Systems. PhD Thesis, University of Trier, Germany.

WEIBELZAHL, S. (2005) Problems and Pitfalls in Evaluating Adaptive Systems. *Proceedings of the Fourth Workshop on the Evaluation of Adaptive Systems, held in conjunction with the 10th International Conference on User Modeling (UM'05), Edinburgh, UK.* pp. 57-66.

WRIGHT, P., MOSSER-WOOLEY, D. & WOOLEY, B. (1997) Techniques & Tools for Using Color in Computer Interface Design. *Crossroads*, vol. 3**,** pp. 3-6.

WU, J. (2000) Accommodating both Experts and Novices in One Interface [online]. Available at: http://www.otal.umd.edu/UUGuide/jingwu/ (Last Accessed: 15 May 2008).

ZUDILOVA-SEINSTRA, E. (2007) On the role of individual human abilities in the design of adaptive user interfaces for scientific problem solving environments. *Knowledge and Information Systems*, vol. 13**,** pp. 243-270.

Appendix A: Pilot Study Background Questionnaire

The Department of Computer Science & Information Systems

Usability Evaluation

Background Questionnaire

Nelson Mandela Metropolitan University
for tomorrow

This questionnaire is aimed at determining your biographical details, level of computer literacy and experience with the HEAT application. All information provided will be held in the strictest confidence and will be used for research purposes only. Please respond as honestly as possible.

SECTION A: Biographical Information

Name & Surname	
Contact Tel. No.:	
Gender:	
Home Language	
Age:	
E-mail Address:	
Qualification:	

SECTION B: Computer Literacy

HEAT Product Suite				
1.1	How many years experience do you have working with Helpdesk Software	0	1-4	5+
1.2	Approximately how many days of the week do you use the application?	0	1-3	4-5

Computer Experience				
2.1	How often do you work on a computer (days)?	1-2	3-4	5
2.2	How many years of computer experience do you have?	<5	6-10	10+

Service Desk Experience				
3.1	How many years service desk /help desk software have you used?	0	1-3	4+
3.2	How many years of experience do you have working with service desk /help desk software?	0	1-3	4+

Thank you for your participation

Appendix B: Pilot Study Test Plan Overview

The Department of Computer Science & Information Systems

Usability Evaluation

Test Plan Overview

Nelson Mandela Metropolitan University *for tomorrow*

Introduction

The main goal of this evaluation is to capture your low-level interaction data whilst using the system and to evaluate how easy it is to log calls using the novice user interfaces.

It will be required that a number of calls (user queries) be logged. The steps involved in logging a call are:

1. A customer is identified.
2. The call's (query) details are captured.
3. The call is assigned to a CCA/technician.
4. If the call is assigned to the CCA receiving the call (you), the call's solution details are provided.

Information necessary to log calls

Step 2: Call Details Information

No *Critical* calls (priority of **1**) will be logged for the purpose of this evaluation. If the call is urgent to the customer, it will have a priority of **2** (***Urgent***), else it will just be **Standard** calls (priority of **3**).

Step 3: Assignment Information

Note: If the call involves a **Telecoms** type problem, it should be assigned to:

Quinten - Quinten Booysen (**North** Campus)

Note: If the call involves assistance with software (Ms Office, Web Browsers etc) , it should be assigned to you, where your details are:

CCA-User1 situated on **South** Campus

Step 4: Solution Details Information

If the call is assigned to you, you need to provide the call’s solution details i.e. the call’s solution description, the cause of the problem and where you resolved the call.

Cause: If the customer requires quick assistance, the calls are logged as **requests**

Resolved: This refers to how the call was resolved

Appendix C: Pilot Study Test Plan

The Department of Computer Science & Information Systems

Novice UI Usability Evaluation

Test Plan

Nelson Mandela Metropolitan University *for tomorrow*

Task 1

You have just received an email. You open it, hoping it's not another chain letter. It reads:

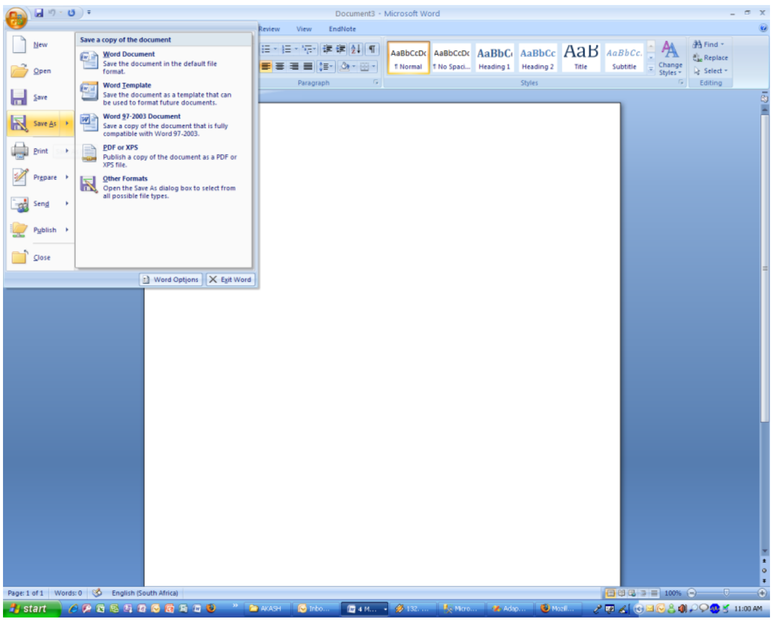
> *Morning my name is David Jenkings (my username is davej) and I would like to log a job card for a telephone. The following extensions can't make calls: 041 550 1234 and 041 550 5678.*
>
> *Thanks*
>
> *David*

You log in to the AdaptiveHelpdesk Software and start logging the call.

Task 2

That wasn't that bad. You are quite pleased with yourself. You just logged your first call for the day. You are about to check your email for new mail, but suddenly the phone rings.....

Hint: While talking to the customer, you decide to open up the application to assist you with providing the customer with an accurate solution. Please see the figure below for a screenshot of the opened application.



The Department of Computer Science & Information Systems

Expert UI Usability Evaluation

Test Plan

Nelson Mandela Metropolitan University

for tomorrow

Task 1

You have just received an email. You open it, hoping it's not another chain letter. It reads:

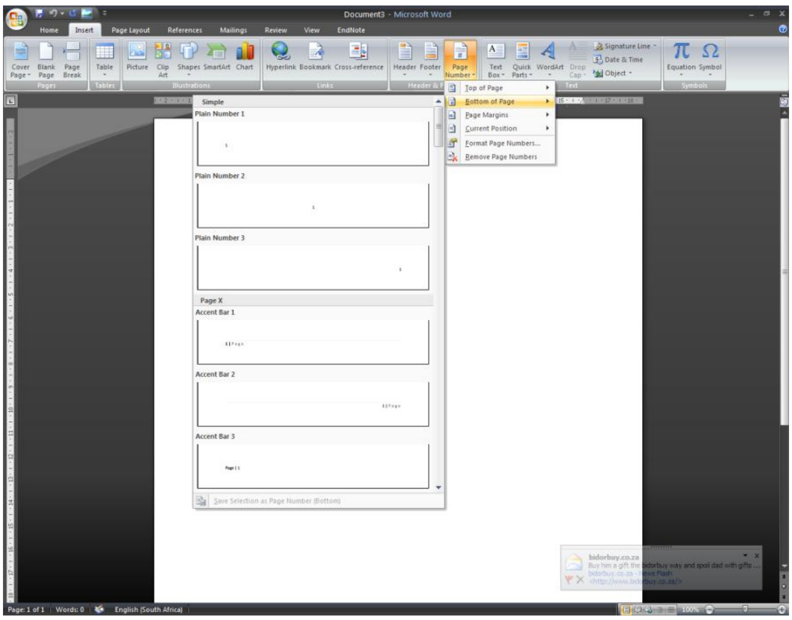
> *Morning my name is Charles Sheppard (my username is cjsheppard) and I would like to log a job card. My telephone keeps dialling the wrong number.*
>
> *Thanks*
>
> *Charles*

You log in to the AdaptiveHelpdesk Software and start logging the call.

Task 2

You are about to check your email for new mail, but suddenly the phone rings…..

Hint: While talking to the customer, you decide to open up the application to assist you with providing the customer with an accurate solution. Please see the figure below for a screenshot of the opened application.



Appendix D: Pilot Study Results

Background Questionnaire Results			
	Mean	**Median**	**Std. Dev**
HEAT Experience	1.57	1	0.8
Computer Experience	2.61	2.5	0.4
Service Desk Experience	1.46	1	0.67
Overall Expertise Value	1.88	1.67	0.48

Table D.1: Pilot Study Background Questionnaire Results (n=23)

Exp Level	IM1Na.1	IM2Na.1	IM3Na.1	IM4Na.1	IM5Na.1	IM6Na.1	IM7Na.1	TTTT-Na1	T-Na1
E01	55.73	59.55	56.25	56.58	54.53	47.39	58.96	53.00	55.14
E02	53.62	58.04	56.28	29.91	45.33	49.64	51.32	57.14	54.48
E03	52.99	58.64	54.86	40.25	52.98	55.00	47.32	58.29	55.52
E04	52.24	58.83	60.76	52.29	48.27	51.98	58.39	51.76	55.11
E05	43.46	56.12	57.39	51.21	25.40	48.68	52.84	52.82	52.46
E06	51.75	59.95	49.37	52.49	51.71	55.34	49.77	58.38	55.38
E07	53.73	59.08	50.85	52.75	50.82	55.62	52.94	56.88	55.26
E08	52.73	58.10	53.75	59.41	31.44	53.66	56.02	58.11	55.67
N01	47.02	48.26	46.43	42.86	58.60	53.24	51.25	32.08	42.70
N02	52.79	54.02	56.55	50.35	56.87	54.04	50.12	57.50	55.37
N03	46.36	43.33	35.93	43.16	56.85	38.52	44.62	35.97	39.65
N04	42.62	45.05	48.85	50.37	58.01	46.70	25.07	30.23	39.57
N05	52.96	41.78	53.55	52.69	50.04	53.52	55.98	53.00	51.18
N06	53.64	47.69	50.04	59.96	58.04	55.36	47.78	53.35	51.90
N07	53.12	41.40	51.41	54.73	51.63	51.42	54.64	47.79	48.71
N08	47.87	39.80	50.07	52.23	54.46	51.44	48.29	47.08	47.22
N09	53.95	50.16	56.05	40.60	50.39	31.92	48.11	46.82	48.33
N10	55.87	31.51	53.23	31.94	50.57	51.06	56.68	55.29	49.27
N11	54.39	47.54	40.75	54.62	58.04	51.87	52.74	58.73	52.07
N12	43.34	40.18	28.77	58.87	39.46	52.40	50.35	30.67	37.13
N13	35.77	52.07	48.51	51.35	42.20	52.46	54.15	60.59	53.58
N14	52.13	57.47	45.54	55.83	55.17	54.89	51.60	58.73	54.74
N15	41.94	41.44	44.82	55.52	49.21	33.85	31.07	35.79	39.56

Table D.2: Pilot Study IM Scores for Task 1 (n=23)

Exp Level	IM1Na.1	IM2Na.1	IM3Na.1	IM4Na.1	IM5Na.1	IM6Na.1	IM7Na.1	IM8Na.1	IM9Na.1	TTTT-Na1	T-Na1
E01	55.73	59.55	56.25	56.58	54.53	47.39	58.96	59.57	54.99	53.00	55.14
E02	53.62	58.04	56.28	29.91	45.33	49.64	51.32	61.24	57.65	57.14	54.48
E03	52.99	58.64	54.86	40.25	52.98	55.00	47.32	56.68	42.60	58.29	55.52
E04	52.24	58.83	60.76	52.29	48.27	51.98	58.39	53.37	59.41	51.76	55.11
E05	43.46	56.12	57.39	51.21	25.40	48.68	52.84	63.58	54.48	52.82	52.46
E06	51.75	59.95	49.37	52.49	51.71	55.34	49.77	55.61	60.52	58.38	55.38
E07	53.73	59.08	50.85	52.75	50.82	55.62	52.94	57.69	56.92	56.88	55.26
E08	52.73	58.10	53.75	59.41	31.44	53.66	56.02	54.79	57.33	58.11	55.67
N01	47.02	48.26	46.43	42.86	58.60	53.24	51.25	48.00	52.99	32.08	42.70
N02	52.79	54.02	56.55	50.35	56.87	54.04	50.12	49.72	44.21	57.50	55.37
N03	46.36	43.33	35.93	43.16	56.85	38.52	44.62	49.52	42.72	35.97	39.65
N04	42.62	45.05	48.85	50.37	58.01	46.70	25.07	41.26	45.11	30.23	39.57
N05	52.96	41.78	53.55	52.69	50.04	53.52	55.98	45.49	23.42	53.00	51.18
N06	53.64	47.69	50.04	59.96	58.04	55.36	47.78	59.00	56.92	53.35	51.90
N07	53.12	41.40	51.41	54.73	51.63	51.42	54.64	40.01	49.43	47.79	48.71
N08	47.87	39.80	50.07	52.23	54.46	51.44	48.29	44.99	54.15	47.08	47.22
N09	53.95	50.16	56.05	40.60	50.39	31.92	48.11	31.78	49.05	46.82	48.33
N10	55.87	31.51	53.23	31.94	50.57	51.06	56.68	51.47	53.97	55.29	49.27
N11	54.39	47.54	40.75	54.62	58.04	51.87	52.74	47.55	50.45	58.73	52.07
N12	43.34	40.18	28.77	58.87	39.46	52.40	50.35	33.98	46.57	30.67	37.13
N13	35.77	52.07	48.51	51.35	42.20	52.46	54.15	44.49	53.00	60.59	53.58
N14	52.13	57.47	45.54	55.83	55.17	54.89	51.60	42.20	47.43	58.73	54.74
N15	41.94	41.44	44.82	55.52	49.21	33.85	31.07	57.99	36.68	35.79	39.56

Table D.3: Pilot Study IM Scores for Task 1 (n=23)

	Expert Users	Novice Users		
	Rank Sum	Rank Sum	U	p-value
IM1Na.1	110.0	166.0	46.0	.392
IM2Na.1	155.0	121.0	1.0	.000
IM3Na.1	138.0	138.0	18.0	.005
IM4Na.1	95.0	181.0	59.0	.975
IM5Na.1	68.0	208.0	32.0	.076
IM6Na.1	111.0	165.0	45.0	.357
IM7Na.1	120.0	156.0	36.0	.131
TTTT-Na1	118.5	157.5	37.5	.149
T-Na1	**148.0**	**128.0**	**8.0**	**.000**
IM1Na.2	76.5	199.5	40.5	.213
IM2Na.2	123.0	153.0	33.0	.087
IM3Na.2	121.0	155.0	35.0	.115
IM4Na.2	116.0	160.0	40.0	.213
IM5Na.2	103.0	173.0	53.0	.681
IM6Na.2	124.0	152.0	32.0	.076
IM7Na.2	122.0	154.0	34.0	.101
IM8Na.2	146.0	130.0	10.0	.001
IM9Na.2	140.5	135.5	15.5	.002
TTTT-Na2	144.5	131.5	11.5	.001
T-Na2	**150.0**	**126.0**	**6.0**	**.000**

Table D.4: Pilot Study Mann-Whitney Test Results (n=23)

Appendix E: Main Study Test Plan Overview

Nelson Mandela Metropolitan University
Department of Computer Science
and Information Systems

This questionnaire is part of research towards a MSc in Computer Science and Information Systems

Test Plan Overview: AdaptiveServiceDesk

Introduction

The goal of this evaluation is to evaluate how quickly novice users can become experts, with regard to using the Adaptive Service Desk's (ASD) user interface (UI) and if this transition improves their performance. During the evaluation the usefulness of the application will be tested.

It will be required that a number of calls (user queries) be logged. The steps involved in logging a call are:

5. A customer is identified.
6. The call's (query) details are captured.
7. The call is assigned to a CCA/technician.
8. If the call is assigned to the CCA receiving the call (you), the call's solution details are provided.

ASD consists of 2 kinds of interfaces. The first kind is designed for novice users and is thus regarded as the novice user interface (UI). The novice UI displays a separate screen for each one of the call logging steps (Figure 1).

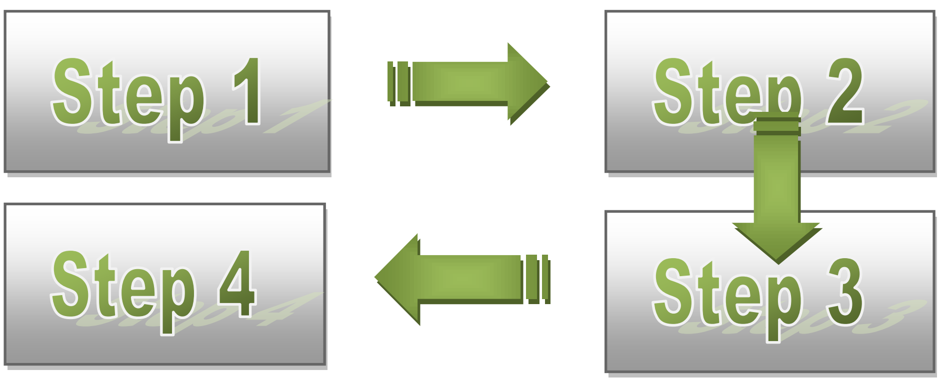


Figure 1: Novice UI screens

The second kind is designed for expert users and is thus regarded as the expert UI. The expert UI displays all the call logging steps tabbed in one interface (Figure 2).

Figure 2: Expert UI screen

The expert UI additionally contains adaptive lists (Figure 3) which lists the top 3 most recent and frequent items. The top 3 items are visually represented by having a different background colour than the other items in the list.

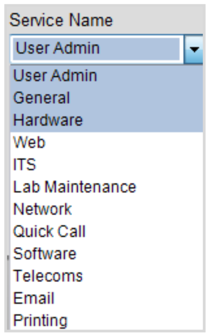


Figure 3: Adaptive list

Both the novice and expert UI contain a section dedicated to the delivery of task-based information (Figure 4).

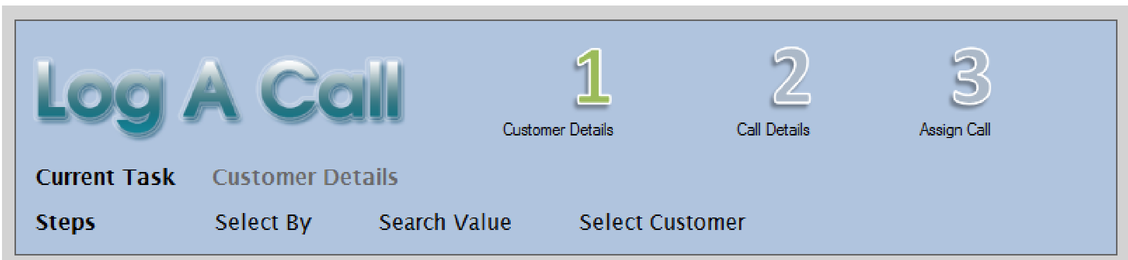


Figure 4: Task-based section

The colours for the Task-based information are either:

- Grey – Indicates that the step has not yet been attempted
- **Blue** – Indicates that the step has been completed
- **Green** – Indicates the step you are currently busy with

Information necessary to log calls

Step 1: Customer Details Information

All students' calls are logged under the userID/ name/ surname, **student**, but their student nr should always be entered in the call description.

Step 2: Call Details Information

No Critical calls (priority of **1**) will be logged for the purpose of this evaluation. If the call is urgent to the customer, it will have a priority of **2** (**Urgent**), else it will just be **Standard** calls (priority of **3**).

Step 3: Assignment Information

Note: If the call involves:

1. Software installations
2. Telecoms
3. Printers

it should be assigned to one of the following technicians/CCAs:

Quinten - Quinten Booysen (**North** Campus) – **Telecom** (telephone etc) type problems

MarkK – MarrK Kerspay (**2nd avenue**) – **Printer** type problems

AHoulie – Anisa Houlie (**South** Campus) – **Software** Installations

Note: If the call requires:

1. Resetting students passwords (which is a quick call)
2. assistance with software (Ms Office, Web Browsers etc)

the call should be assigned to you, where your details are:

CCA-User1 situated on **South** Campus

Step 4: Solution Details Information

If the call is assigned to you, you need to provide the call's solution details i.e. the call's solution description, the cause of the problem and where you resolved the call.

1. Cause: If customer requires quick assistance, the calls are logged as **requests**
2. Resolved: This refers to how the call was resolved

Appendix F: Main Study Test Plan

Nelson Mandela Metropolitan University
Department of Computer Science
and Information Systems

This questionnaire is part of research towards a MSc in Computer Science and Information Systems

Test Plan: Novice UI

TELEPHONE CALLS

Task 1

Hi my name is Brian Gallant (bgallant). I need you to urgently move the telephone extension from present store room to the new store room but I need to keep the same telephone no.

Task 2

Hi I am a student and I forgot my computer password. Can you please reset it for me? student nr is 203123456

Task 3

Hi, my name is David Levey (dwlevey) and I cannot make outgoing calls from my office phone.

Task 4

Hey. This is Beverly Gold (blgold). I'm typing in a word document and I'm would like my paragraph justified but i don't know how. Can you please help me?

Task 5

Hi I am a research assistant and student. I urgently need to have MS Excel installed on my pc.

Task 6

Hi. Could you please assist me with editing the header & footer in a Word document. Anette Knight (akknight)

Task 7

Hey this is Dedre. The computer in 011006 must be linked to a printer in 011005a. Dedre Erasmus (dderasmus).

Task 8

Hey this is Adrian Konik. Can u please help me? I want to change my homepage to NMMU portal.

Nelson Mandela Metropolitan University
Department of Computer Science
and Information Systems

This questionnaire is part of research towards a MSc in Computer Science and Information Systems

Test Plan: Expert UI

TELEPHONE CALLS

Task 1

Hey I'm not able to redirect the telephone call to another number. Cynthia Hustler (chustler).

Task 2

Hi. This is Daniel O Connor (daoconnor). I would like a signature in my e-mails. Do you know how to set this up?

Task 3

The computer in 011006 must be mapped to printer in 011005a. This is Assim Alpaslan (ahalpaslan).

Task 4

Hi I'm Colleen Hopgood (CHOPGOOD). I'm working on PowerPoint and I need to print multiple slides on a page but i have no idea how to do this. Could you please help me?

Appendix G: Main Study Post Test Questionnaire

Nelson Mandela Metropolitan University
Department of Computer Science

and Information Systems

This questionnaire is part of research towards a MSc in Computer Science and Information Systems

Post Test Questionnaire: *AdaptiveServiceDesk*

	Section A: Biographical Details (mark with X where appropriate)					
1	Gender:	Male	Female			
2	Age:					
3	Occupation:					
4	Computer experience (years)	1	1 - 2	3 - 4	4 +	
5	Service desk software experience (years)	0	1 - 2	3 - 4	4 +	N/A
6	Have you previously used the HEAT application?	No		Yes		
	Section B1: Novice UI					
1	Overall reaction to the system	Very frustrating				Very satisfying
		1	2	3	4	5
2	Screen design	Very frustrating				Very satisfying
		1	2	3	4	5
3	The layouts of the screens	Very confusing				Very clear
		1	2	3	4	5
4	Learnability of the system	Very difficult				Very easy
		1	2	3	4	5
5	Navigation of the system	Very difficult				Very easy
		1	2	3	4	5
6	Phrasing of error messages	Very unpleasant				Very pleasant
		1	2	3	4	5
7	Amount of System Feedback	Very insufficient				Very sufficient
		1	2	3	4	5
	Section B2: Task Support					
1	Task based information	Not useful				Very useful
		1	2	3	4	5
2	Use of Visual Step Indicator	Very unhelpful				Very Helpful
		1	2	3	4	5
3	Use of Visual Step Indicator	Very frustrating				Very satisfying
		1	2	3	4	5
4	Indication of progress	Very ambiguous				Very clear
		1	2	3	4	5
5	Ability to facilitate call logging	Not at all				Very Much
		1	2	3	4	5
	Section C1: Expert UI					
1	Overall reaction to the system	Very frustrating				Very satisfying
		1	2	3	4	5
2	Screen design	Very frustrating				Very satisfying

		1	2	3	4	5
3	The layouts of the screens	Very confusing				Very clear
		1	2	3	4	5
4	Learnability of the system	Very difficult				Very easy
		1	2	3	4	5
5	Navigation of the system	Very difficult				Very easy
		1	2	3	4	5
6	Phrasing of error messages	Very unpleasant				Very pleasant
		1	2	3	4	5
7	Amount of System Feedback	Very insufficient				Very sufficient
		1	2	3	4	5
	Section C2: Adaptive List					
1	Ordering of lists of options (according to recency and frequency of use)	Not Noticeable				Very noticeable
		1	2	3	4	5
2	Ordering of lists of options (according to recency and frequency of use)	Not useful				Very useful
		1	2	3	4	5
3	Ordering of lists of options (according to recency and frequency of use)	Very confusing				Very logical
		1	2	3	4	5
4	Ordering of lists of options (according to recency and frequency of use)	Very frustrating				Very satisfying
		1	2	3	4	5
5	Preference	Static lists (lists on novice UI)				Adaptive lists (lists on expert UI)
	Section D: Adaptivity					
	Transition (Novice UI to Expert UI)	Not useful				Very useful
		1	2	3	4	5
		Very unpleasant				Very pleasant
		1	2	3	4	5
		Very confusing				Very clear
		1	2	3	4	5

	Section E1: Preference	**Rank (1=Best) (3=Worst)**
	Novice UI	
	Expert UI	
	Transition (Novice UI to Expert UI)	

	Section E2
1	With which user interface (Novice or Expert) did you feel you were the most efficient (quick)? Please explain.
2	With which user interface (Novice or Expert) did you feel the most frustrated? Please explain.
3	Describe positive aspects of transitioning the interface from a novice to expert user interface.

4	Describe negative aspects of transitioning the interface from a novice to expert user interface.
5	Do you feel the switch to the expert interface was an accurate reflection of your expertise? Please explain.
6	Please provide any general comments or suggestions for improvement below

Appendix H: Consent Form

Nelson Mandela Metropolitan University
Department of Computer Science
and Information Systems

This questionnaire is part of research towards a MSc in Computer Science and Information Systems

Consent Form

You have been selected as a research participant for the evaluation of the AdaptiveHelpdesk. This evaluation is being conducted by Bronwin Jason (Bronwin.Jason@nmmu.ac.za). Please do not hesitate to ask if you have any questions about the evaluation. As a participant you have certain rights, which are listed below. You will be asked to perform various tasks with this software. The purpose of this evaluation is to capture your low-level interaction data as well as performance details. A secondary goal entails rating the usability of the software. We expect the session to last about 60 minutes. An embedded video will record your interaction and your eye movements and your comments will be recorded. This data will be used only for research purposes and will not be distributed nor viewed by anyone not associated with this evaluation process. Your name will not be associated with any data that are collected during this evaluation session. There are no known risks associated with this evaluation. You will be asked to complete a feedback questionnaire, containing questions relevant to this evaluation.

Your rights as a participant are as follow:

1. You have the right to withdraw from the session at any time for any reason.
2. At the conclusion of the session, you may see your data if you so desire. If you decide to withdraw your data, please inform the evaluators immediately.
3. Finally, we greatly appreciate your time and effort for participating in this evaluation. Remember, you cannot fail any part of this session, and there are no right or wrong answers. Your signature below indicates that you have read this consent form in its entirely and that you voluntarily agree to participate.

Name & Surname:		Contact Tel. No.:	
Signature:		Date:	

Appendix I: Main Study Results

	MeanNov	MeanExp	DiffN-E
P1	127.5	104.5	23
P2	116.25	92	24.25
P3	213	153.25	59.75
P4	221	178.5	42.5
P5	159.5	103.75	55.75
P6	117.5	101.25	16.25
P7	126	110	16
P8	180	131.5	48.5
P9	157.75	152.25	5.5
P10	234	123.5	110.5
P11	188.5	155.5	33
P12	162.75	125.25	37.5
P13	110.25	88.75	21.5
P14	118.25	104.25	14
P15	105.5	85.25	20.25
P16	126.5	101.25	25.25
P17	132.75	114.5	18.25
P18	203	125.25	77.75
P19	141.25	91.5	49.75
P20	108.75	80	28.75
P21	111	87.5	23.5
P22	158.75	105.5	53.25
P23	156	134.5	21.5
P24	121.5	91.75	29.75
P25	83.75	82.25	1.5
P26	136.75	106.75	30
P27	153.75	101.25	52.5
P28	122	88.75	33.25
P29	137.75	106.5	31.25
P30	123	128	-5
Mean	145.1417	111.825	33.31667

Table I.1: t-test Learnability Results (n=30)

Overall Novice UI Satisfaction Results			
	Mean	**Median**	**Std. Dev**
Overall User Reaction	4.03	4	0.76
Screen Design	4.40	4.5	0.72
Screen Layout	4.33	4	0.61
System Learnability	4.4	4.5	0.67
System Navigation	4.47	5	0.68
Phrasing of Error Messages	3.64	4	1.21
Amount of System Feedback	4.13	4	0.86

Table I.2: Overall Novice UI Satisfaction Results (n=30)

Overall Expert UI Satisfaction Results			
	Mean	**Median**	**Std. Dev**
Overall User Reaction	4.47	4	0.51
Screen Design	4.63	5	0.49
Screen Layout	4.53	5	0.68
System Learnability	4.43	5	0.73
System Navigation	4.47	5	0.63
Phrasing of Error Messages	4.09	4	0.94
Amount of System Feedback	4.13	4	0.78

Table I.3: Overall Expert UI Satisfaction Results (n=30)

Novice UI Task Support Results			
	Mean	**Median**	**Std. Dev**
Task Based Information	3.77	4	1.17
Helpfulness of Visual Step Indicator	3.87	4	1.07
Satisfaction of Visual Step Indicator	4	4	0.98
Indication of Progress	4.1	4	0.76
Ability to Facilitate Call Logging	4.37	4	0.67

Table I.4: Novice UI Task Support Results (n=30)

Expert UI Adaptive List Results			
	Mean	**Median**	**Std. Dev**
Noticeable	4.4	5	0.81
Useful	4.43	4.5	0.63
Logical	4.33	4	0.71
Satisfying	4.37	4	0.67

Table I.5: Expert UI Adaptive List Results (n-30)

Adaptivity Results			
	Mean	**Median**	**Std. Dev**
Useful	4.27	4	0.83
Pleasant	4.3	4	0.75
Clear	4.13	4	0.86

Table I.6: Adaptivity Results (n=30)

Preference Results			
	First	**Second**	**Third**
Novice UI	4	7	19
Expert UI	15	10	5
Adaptation	11	13	6

Table I.7: Preference Results (n=30)