Designing Adaptive User Interfaces for Enterprise Resource Planning Systems for Small Enterprises

By

Akash Singh

Submitted in fulfilment of the requirements for the degree of Philosophiae Doctor
in the Faculty of Science at the Nelson Mandela Metropolitan University

December 2011

Promoter: Prof. J.L. Wesson

Declaration

I, *Akash Singh (202 310 175)*, hereby declare that the *thesis* for *PhD (Computer Science)* is my own work and that it has not previously been submitted for assessment or completion of any postgraduate qualification to another University or for another qualification.

Akash Singh

Acknowledgements

I would like to thank my promoter, Prof. Janet Wesson, for her support, advice and encouragement throughout this research. It has been my honour and privilege to learn from her. The enthusiasm and dedication she has for her research was motivational and inspired me to pursue this work with diligence and passion.

The support of SAP Research IA & S Africa and the SAP Meraka Unit for Technology Development (UTD) for this work is hereby acknowledged. Opinions expressed and conclusions arrived at are those of the author, and should not necessarily be attributed to SAP Research or the SAP Meraka UTD.

To Mr. Keith Gibson, thank you for playing a pivotal role in spurring my interest in IT. Your contribution will never be forgotten, and will always inspire me to achieve new heights.

I would like to thank the staff in the Department of Computing Sciences at NMMU for all their technical support, encouragement and motivation during the course of this research.

I would also like to express my gratitude to Mr Danie Venter, from the NMMU Unit for Statistical Consultation, for his assistance in analysing and presenting the results of the evaluations conducted in the course of this research.

To Danie Kok, Jan Eloff, Darelle Van Greunen, and Lucas Venter: thank you for showing me that a PhD does not just consist of a number of pages bound between two covers, but is actually an expression of creative thought (inspired by past and present experiences) that could be used to shape the modern world.

Lastly, I am especially grateful to my friends and family for all their love, and for being that additional pillar of support. This journey would not have been possible without your constant inspiration, encouragement and best wishes.

Summary

It is widely acknowledged that enterprise resource planning (ERP) systems suffer from complex user interfaces. The complexity of these user interfaces negatively affects the usability of these systems. Current research has shown that a need exists to improve the overall usability of ERP systems. This research proposes the use of adaptive user interfaces (AUIs) as a means of improving the overall usability of ERP systems. Research has shown that AUIs are capable of improving system usability by reducing user interface complexity and improving the overall user experience.

The primary objective of this research was to determine how AUIs could be designed to improve the usability of ERP systems. An adaptation taxonomy, ERP system architecture (incorporating an AUI), a set of AUI components and a set of usability heuristics for ERP systems were proposed to support the design, development and evaluation of AUIs for ERP systems.

The proposed adaptation taxonomy provides support for three types of adaptation: content adaptation, presentation adaptation and navigation adaptation. The proposed ERP system architecture is a three-tiered system architecture, consisting of a Presentation Layer (incorporating an AUI), an Application Layer and a Database Layer. The proposed set of AUI components comprise a user model, a task model and a dialog model. The set of proposed usability heuristics aims to identify usability issues of ERP systems within the areas of Navigation, Presentation, Task Support, Learnability and Customisation.

An AUI prototype was developed based on selected adaptive techniques from the proposed adaptation taxonomy and selected components from the proposed system architecture. All of the proposed AUI components were implemented. The AUI prototype was developed for an existing ERP system, namely SAP Business One (SBO). This prototype was designed, in order to resolve the usability issues of SBO identified through the use of the proposed set of heuristics. The development of the AUI prototype was made possible through the use of a software development kit (SDK) provided with SBO. The AUI prototype made use of content adaptation, presentation adaptation and navigation adaptation in order to address the identified usability issues.

An empirical evaluation was conducted on the AUI prototype to determine whether it provided any usability benefits over the standard SBO system. The results from the empirical evaluation revealed that the AUI presented usability benefits with regard to learnability and satisfaction. Users who used the AUI prototype were able to learn how to use the ERP system a lot quicker and were more satisfied than users of the standard SBO system.

The successful implementation of the AUI prototype provided practical evidence that the proposed adaptation taxonomy and the proposed system architecture can be implemented. This research has provided empirical evidence that the use of AUIs can improve the usability of ERP systems. Future research has outlined several possibilities to utilise and enhance the proposed adaptation taxonomy, the ERP system architecture and ERP heuristics, for the purpose of furthering research within the area of AUIs for ERP systems.

Keywords: Adaptive user interfaces, enterprise resource planning systems, small enterprises, system design, user interface design, usability, usability evaluation.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AUI	Adaptive User Interface
CRM	Customer Relationship Management
ERP	Enterprise Resource Planning
HCI	Human-Computer Interaction
ICT	Information and Communications Technologies
IDE	Integrated Development Environment
IDES	Internet Demonstration and Evaluation System
IEEE	Institute of Electrical and Electronics Engineers
MFU	Most Frequently Used
MRU	Most Recently Used
NMMU	Nelson Mandela Metropolitan University
PD-MBUI	Pattern-Driven and Model-Based User Interface
QUIS	Questionnaire for User Interface Satisfaction
SBO	SAP Business One
SDK	Software Development Kit
SUMI	Software Usability Measurement Inventory
UTD	Unit for Technology Development

Chapter 1: Introduction

1.1 Background

Small businesses (enterprises) need to make use of information and communication technologies (ICT) in order to survive (Lefebvre *et al.* 1995; Montazemi 2006; SEDA 2008). The survival rate of small enterprises is extremely low and is a direct result of a highly volatile market and intense competition. ICT can improve the survival rates of small enterprises, by streamlining and automating business processes in order to improve the efficiency of operations as well as the profitability and sustainability of the enterprise (Ndiwalana and Tusubira 2006; SEDA 2008).

An enterprise resource planning (ERP) system is an example of an ICT solution that can be used to standardise business processes in order to improve the operational efficiency, profitability and the sustainability of the enterprise. ERP systems, if implemented correctly, could offer small enterprises an environment where information and related business processes are integrated. This type of an environment provides improvements in terms of efficiency, productivity and service quality, reduction in service costs, automation of business process and the adoption of best practice business models (Light 2005; Nach and Lejeune 2008; Ngai *et al.* 2008; Ragowsky and Gefen 2008).

Existing ERP systems are, however, too rigid and complex for small enterprises. Small enterprises typically operate in a dynamic business environment and need to be flexible in their processes and business models in order to adapt to sudden internal and external changes (Olsen and Sætre 2007). ERP systems are typically rigid and do not provide small enterprises with the level of customisability required in order to adapt to changes in the business landscape.

Another disadvantage is that ERP systems are complex to operate and frustrating to use (Topi *et al.* 2005; ERPwire.com 2008; Matthews 2008). This complexity mainly lies in the “unfriendly” nature of the user interface (Boudreau 2003; Yeh 2006). These characteristics can have an effect on the usability of the ERP system and can impact on the effectiveness and efficiency of task completion (Chew *et al.* 2003; Herbert *et al.* 2006; Matthews 2008).

The usability of ERP systems needs to be improved. The term usability is not often associated with ERP systems and only a few usability studies on ERP systems have been published. Those studies that are published indicate the need for more usable and personalised user interfaces. Adaptive user interfaces (AUIs) could potentially address the usability problems (issues) that occur in most ERP systems. These types of user interfaces attempt to provide interaction that is personalised, easy and effective, to ensure efficient task completion (Dieterich *et al.* 1993; Álvarez-Cortés *et al.* 2007).

The ability of AUIs to support task simplification, error correction, active and intelligent help and improved user satisfaction could assist in reducing the complexity and in improving the overall usability of ERP systems for small enterprises (Dieterich *et al.* 1993).

1.2 Problem Statement

This research aims to address the problem of poor usability inherent in ERP systems designed for small enterprises.

The usability of ERP systems is currently a problem because these systems are too rigid and complex to be used effectively by small enterprises. Small enterprises require software systems that are usable and adaptive, in order to support the dynamically changing business landscape in which they operate.

1.3 Thesis Statement

AUIs can be designed to improve the usability (efficiency, learnability and satisfaction) of ERP systems for small enterprises.

1.4 Research Objectives

The main objective of this research is *to determine how AUIs can be designed to improve the usability of ERP systems for small enterprises*. This is necessary in order to address the problem statement and assess the thesis statement. Several secondary research objectives are derived from the main research objective. These secondary objectives will form the basis of the individual chapters of this thesis. Each chapter will achieve a specific research objective.

These secondary research objectives are:

To select a specific industry sector and ERP system on which to focus this research.

This objective attempts to scope this research by identifying a particular industry sector and an ERP system (used by small enterprises within that sector) for experimentation purposes.

To identify the existing usability issues of ERP systems, and to determine how to evaluate the usability of an ERP system.

This objective attempts to identify the usability issues of ERP systems from existing research, and to identify existing methods and metrics that can be used for evaluating the usability of ERP systems.

To identify the usability issues of the selected ERP system for small enterprises.

The method and metrics identified in the previous objective will be utilised to achieve this objective. A usability evaluation will be conducted on the selected ERP system in order to achieve this objective.

To determine how AUIs could be applied to address the identified usability issues of ERP systems.

AUIs have been proposed as a potential solution to resolve the usability issues of ERP systems. This objective attempts to describe how AUIs could be applied to address the usability issues of ERP systems, more specifically, those usability issues identified in the previous objective.

To design an AUI and develop a prototype to address the usability issues of ERP systems.

This objective builds on the previous objective by designing an AUI to address the usability issues of ERP systems. The development of a prototype through the use of the selected ERP system would assist in demonstrating how an AUI could be developed for an ERP system for small enterprises.

To evaluate the benefits of incorporating an AUI into the selected ERP system.

Once the prototype AUI has been developed, a usability evaluation will be conducted. The results of this usability evaluation will assist in answering the main research question and in determining whether the main research objective has been achieved. These results will also indicate the extent to which AUIs can improve the usability of ERP systems for small enterprises.

To discuss the theoretical and practical contributions of incorporating an AUI into the selected ERP system.

The last objective of this research will be to discuss the contributions made by this study. The contributions made to theory and practice will be discussed.

The research questions derived from these research objectives are discussed in the next section.

1.5 Research Questions

This research will attempt to answer the following research question: *How can AUIs be designed to improve the usability of ERP systems for small enterprises*? Several supporting research questions are listed in Table 1.1. These research questions were derived from the secondary research objectives presented in the previous section and assist in answering the main research question.

No	*Research Question*
1.	Which industry sector and ERP system for small enterprises can this research use for experimental purposes?
2.	What are the existing usability issues of ERP systems, and how can the usability of an ERP system be evaluated?
3.	What are the usability issues of the selected ERP system for small enterprises?
4.	How can AUIs be applied to address the identified usability issues of ERP systems?
5.	How can an AUI be designed and developed to address the usability issues of ERP systems?
6.	What are the benefits of incorporating an AUI into the selected ERP system?
7.	What theoretical and practical contributions have been made by incorporating an AUI into the selected ERP system for small enterprises?

Table 1.1: Supporting Research Questions

The methods that will be used to answer these research questions and achieve the objectives of this research are discussed in the next section.

1.6 Research Methodology

1.6.1 Research Philosophy

This research will adopt a positivist research philosophy. Based on this philosophy, the following assumptions are made with regard to this research (Orlikowski and Baroudi 1991; Weber 2004):

- Reality is separate from the researcher and the object or phenomenon being researched has qualities that exist independently of the researcher;
- Distinct variables will be identified and measured in a quantifiable manner;
- Data (that are a true reflection of reality) will be collected and statistically analysed;
- Hypothesis testing will be done in order to verify or falsify a hypothesis. This hypothesis is presented in the form of the thesis statement for this research; and
- Inferences will be drawn, on the object or phenomenon being researched, from the sample and then made against the stated population.

1.6.2 Research Approach

This research will be conducted using a deductive reasoning approach. The deductive reasoning approach (also known as the hypothetico-deductive approach) attempts to verify or falsify theories presented in the form of hypotheses. Typically, hypothesis testing is conducted in order to verify or falsify the hypotheses. Once the tests are complete, deductions are then made based on the results (Hayes 2000; Hyde 2000; Svensson 2009).

Figure 1.1 illustrates a typical research process conducted in a deductive reasoning approach. The deductive reasoning approach is a clockwise process, which starts with an idea generation phase and is made explicit through the establishment of a research objective. Based on the research objective, several supporting research questions are formulated. Through existing research literature, support is generated in order to qualify and support the idea generated. Further support is obtained through the collection of empirical data.

Theoretical and/or managerial implications are then made on the idea through a process of testing (hypothesis testing). The final stage of the research process involves the drawing of conclusions, highlighting the contributions made, and proposing suggestions for future research (Svensson 2009).

This research will adopt the deductive research process and this will be reflected in the structure of the thesis. The deductive reasoning approach was also selected as it best supports the chosen research philosophy.

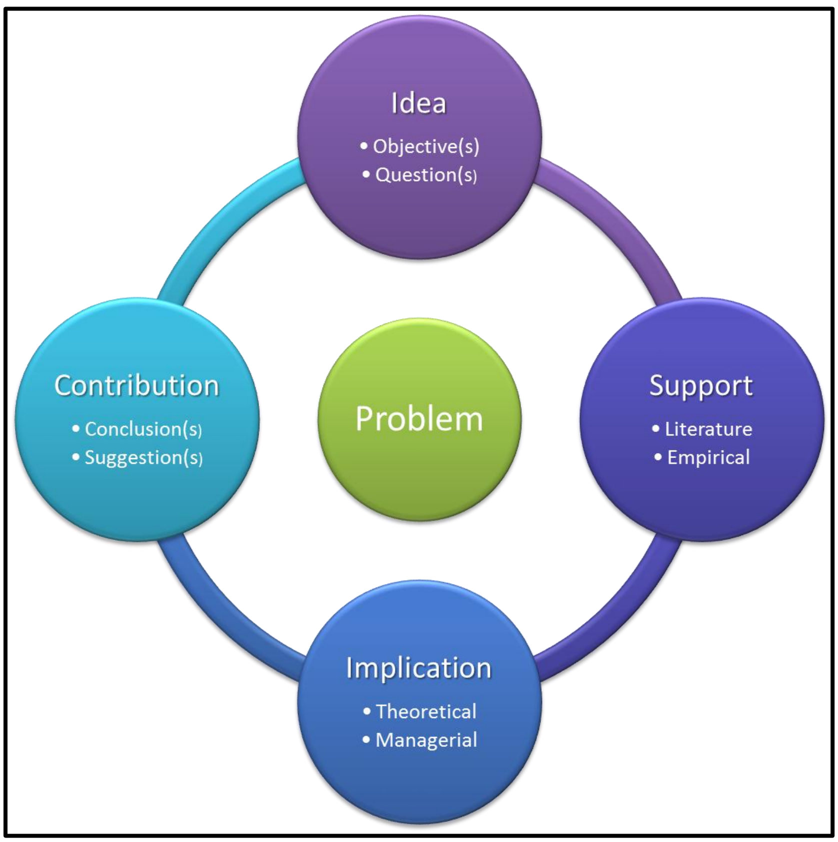


Figure 1.1: Deductive Approach Research Process (Svensson 2009)

1.6.3 Research Methods

This research will make use of several research methods to achieve the research objectives discussed in the previous section. The research methods that will be used include (Olivier 2004; Hofstee 2006; Olivier 2009):

- Extended literature review;
- Prototyping;
- Experimentation; and
- Comparative analysis.

The research methods selected support the selected research philosophy and approach. The extended literature review, prototyping and the experimentation research methods support the *Support* phase (Figure 1.1) of the research process. The use of the comparative analysis research method aligns with the *Implication* phase (Figure 1.1) of the research process.

An extended literature survey of ERP systems for small enterprises, usability and AUIs will be conducted. The purpose of the extended literature survey will be to understand the state-of-the-art with regard to ERP systems for small enterprises, the usability issues associated with ERP systems, existing criteria that could be used to evaluate the usability of ERP systems, and to assist in determining how AUIs could be applied to address the identified usability issues of ERP systems. The extended literature survey will assist in achieving the second and the fourth research objectives.

A preliminary study and an interview study will be conducted. The preliminary study will complement the extended literature survey by identifying an appropriate industry sector and an ERP system that can be used for experimental purposes. This study will contribute to the first research objective. The interview study will be conducted with some of the actual users of the selected ERP system in the identified sector to confirm the usability issues identified in the extended literature survey. The interview study will contribute to achieving the second research objective.

The prototyping phase of this research requires that an AUI for an ERP system be designed and developed. Prototype design and implementation will be used as a proof-of-concept (during the experimental phase of this research) to evaluate whether the proposed AUI addresses the identified usability issues of the selected ERP system. The use of the prototyping research method will assist in achieving the fifth research objective.

The experimentation research method will focus on usability evaluations. Usability evaluations will be performed to identify the usability issues with the selected ERP system. A second usability evaluation will be conducted to determine the extent to which the prototype AUI improves the usability of the selected ERP system. The use of the experimentation research method will assist in achieving the third and the sixth research objectives.

After the experimental phase of this research, a comparative analysis will be conducted in order to compare the results obtained from the adaptive version of the selected ERP system with the results from the non-adaptive version. This will assist in determining whether the incorporation of AUIs does improve the usability of the selected ERP system. The results of the comparative analysis will assist in achieving the last research objective.

The various phases of this research, along with the findings, will be combined and communicated in the form of a research report (thesis).

1.6.4 Research Instruments

Questionnaires will be used during the preliminary and interview studies to guide the interviews. The instruments that will be used to aid the usability evaluations include a test plan, a biographical questionnaire, task lists and post-test satisfaction questionnaires.

1.6.5 Research Data

The data that will be utilised in this research will be primary data that will be gathered from the preliminary and interview studies and the experimentation phases of this research. The data obtained will be both qualitative and quantitative in nature.

1.6.6 Research Analysis

Qualitative data (collected from interviews and questionnaires) will be collated and a thematic analysis will be performed to identify any themes and trends. Simple quantitative analysis will then be conducted in order to determine how the sample reacted to a particular theme or trend.

The quantitative data will be statistically analysed using descriptive and inferential statistics.

1.7 Delineation and Limitations

The scope of this research is restricted to small enterprises currently using ERP systems designed specifically for small enterprises. This will support the identification of usability issues experienced by actual users' of ERP systems for small enterprises. The results from the preliminary study will assist in focusing this research to a specific industry sector and identifying an ERP system that could be used to demonstrate the applicability of AUIs in the domain of ERP systems for small enterprises.

For the purposes of this research, the definition of usability as specified by ISO (1998) will be used. This definition uses the criteria of effectiveness, efficiency and satisfaction as the measures of usability. In addition to these measures, learnability will be included as an additional measure. The inclusion of learnability will assist in determining how difficult it is for a user to learn how to use an ERP system.

1.8 Research Rationale

This research is necessary for the following reasons:

The usability of ERP systems needs to be improved. ERP systems need to be flexible and adaptive in order to be effectively used by small enterprises. The extension of ERP systems into the domain of small enterprises is a fairly recent endeavour and limited research has been published which addresses the usability of ERP systems for small enterprises.

Currently, a large amount of research is dedicated to the individual fields of small enterprises, ERPs, usability and AUIs. However, little research has been published on the combination of these four fields.

Furthermore, limited research is available in identifying the usability issues of ERP systems. Most of this research relates to the usability issues of ERP systems for large enterprises with none pertaining to small enterprises.

Some research has focused on AUIs and their ability to contribute to the field of usability. However, the application of AUIs to the domain of ERP systems for small enterprises is relatively new and needs to be explored and documented.

1.9 Thesis Outline

This thesis will consist of eight chapters. Each chapter will attempt to meet a specific research objective. An adapted version of Figure 1.1 is illustrated in Figure 1.2 to demonstrate how the eight chapters presented in this thesis will support the selected research approach.

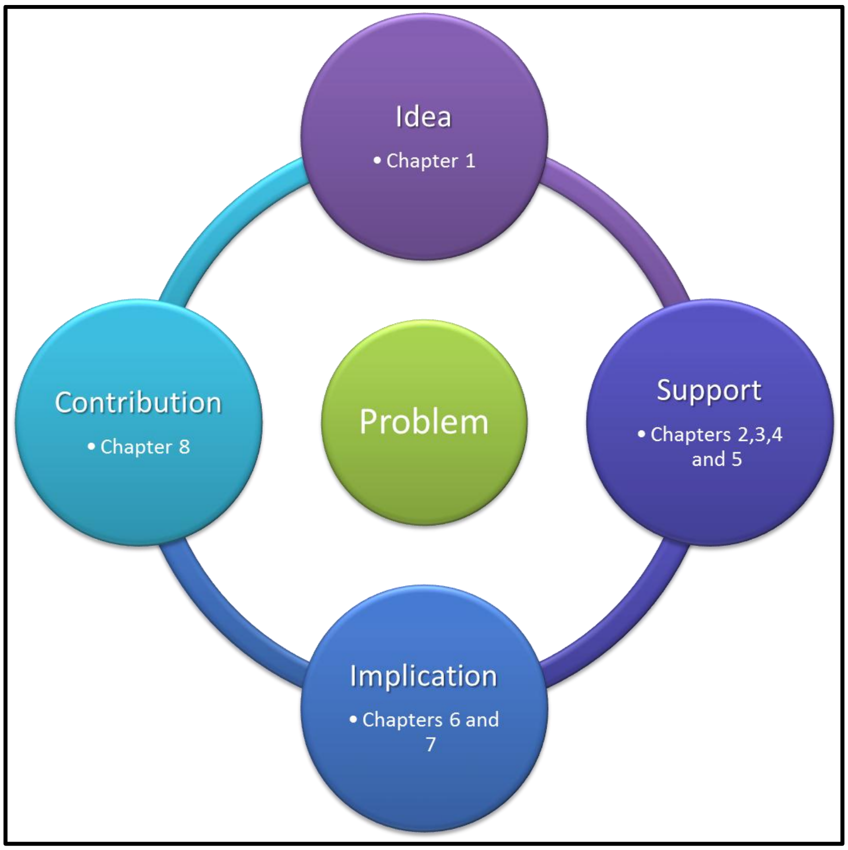


Figure 1.2: Thesis Outline according to the Deductive Reasoning Approach

Chapter 1 (Introduction) has presented a short discussion on the domain of small enterprises, ERP systems, usability and AUIs. Some of the usability issues associated with ERP system for small enterprises and how AUIs could possibly address these problems were briefly discussed. The problem and the thesis statements, the research objectives, the research questions, methodology, delineations and limitations, rationale and outline of this research have also been presented.

Chapter 2 (ERP Systems for Small Enterprises) comprises the first literature study and presents a detailed literature study on the domain of ERP systems. This will be followed by an investigation into ERP systems for small enterprises. The aim of this chapter is to identify an ERP system for small enterprises in a particular industry sector that can be used for experimental purposes for this research. Chapter 2 will use the extended literature review research method to achieve the first research objective of selecting an industry sector and ERP system for small enterprises.

Chapter 3 (Usability of ERP Systems) presents a detailed study on the field of usability and how it is applied to ERP systems. The aim of this chapter is to identify the usability issues of existing ERP systems, and to determine how to evaluate the usability of ERP systems. Chapter 3 will make use of the extended literature review research method to achieve the second research objective – to identify the usability issues of existing ERP systems and to determine how to evaluate the usability of an ERP system.

Chapter 4 (A Usability Evaluation of SAP Business One) makes use of the usability methods and metrics identified in Chapter 3 in order to identify the usability issues of the selected ERP system. This chapter will make use of the experimentation research method in order to achieve the third research objective – to determine the current usability issues of the selected ERP system for small enterprises, namely SAP Business One.

Chapter 5 (Adaptive User Interfaces) presents a detailed discussion of AUIs. The application of AUIs to the domain of ERP systems will also be discussed here. This chapter will describe the most suitable manner in which AUIs can be applied to improve the usability of ERP systems for small enterprises. Chapter 5 will use the extended literature review research method to achieve the fourth research objective – to determine how AUIs can be used to address the usability issues of the selected ERP system.

Chapter 6 (Designing an Adaptive User Interface for SAP Business One) focuses on the design of a prototype AUI for SAP Business One. The design of the AUI will be guided by the current issues impacting on the usability of ERP systems for small enterprises (Chapter 3 and Chapter 4) and the way in which an AUI could be applied to improve the usability of an ERP system for small enterprises (Chapter 5). Chapter 6 will follow the prototyping research design to achieve the fifth research objective – to design and develop an AUI to address the identified usability issues of the selected ERP system.

Chapter 7 (Evaluation) evaluates the prototype AUI implemented in Chapter 6. The usability of the prototype will be evaluated against the attributes identified in Chapter 3. A user study approach will be adopted to aid the usability evaluation. Chapter 7 will make use of the experimentation and comparative analysis research methods to achieve the sixth research objective – to evaluate the benefits of incorporating an AUI into the selected ERP system.

Chapter 8 (Conclusion) presents the conclusions of this research and will make recommendations for future work. Chapter 8 will revisit and examine whether the research objectives set out in this chapter were achieved. This chapter will also address the last research objective by discussing the contributions made by this research. The contributions made by this research will be discussed in the form of theoretical and practical contributions. Several recommendations for future research will be presented and discussed in Chapter 8.

Chapter 2: ERP Systems for Small Enterprises

2.1 Introduction

Several enterprise resource planning (ERP) systems for small enterprises exist and are used in a variety of industry sectors. The main objective of this chapter is to select an industry sector and an ERP system (specific to small enterprises) that would be suitable for experimental purposes.

The remainder of this chapter is split into three main sections – ERP systems, ERP systems for Small Enterprises and ERP selection. The first section provides an overview of the domain of ERP systems and presents a discussion on the importance of usability as a selection criterion when considering the purchase of an ERP system. The second section provides a brief discussion on how small enterprises differ from large enterprises and how this needs to be reflected in their choice of ERP system. Lastly, this chapter presents the results of a preliminary study of ERP vendors in South Africa in order to select an industry sector and an ERP system suitable for experimental purposes.

2.2 ERP Systems

The term ERP (coined by Gartner Inc., in the mid 1990s) was used to describe the third generation of manufacturing resource planning systems (Harwood 2003; Lea *et al.* 2005; Gupta and Kohli 2006). An ERP system is a business information system, which aims to seamlessly integrate all the information flowing in an enterprise. This is accomplished by providing an integrated information system, based on best practices, which can be used by all the key business functions and different departments within an enterprise (das Neves *et al.* 2004; Dillard *et al.* 2005; Yang *et al.* 2007; Seymour and Roode 2008; Wu *et al.* 2008).

An ERP system typically consists of a core set of functional modules. These modules relate to the Financial Management, Human Resources, Sales and Marketing, Manufacturing and Operations, Controlling, and Distribution and Logistics functions of an enterprise (Umble *et al.* 2003; Botta-Genoulaz and Millet 2005, 2006; Van Nieuwenhuyse *et al.* 2007; Pang *et al.* 2008; Microsoft 2009b; ORACLE 2009a; SAP 2009b).

ERP systems are utilised in a variety of industry sectors. These sectors include (Microsoft 2009a; ORACLE 2009b; SAGE 2009; SAP 2009a):

- Financial and Public Services;
- Manufacturing;
- Services;
- Government; and
- Non-Profit Organisations.

A complete list of industry sectors and sub-sectors is provided in Appendix A.

2.2.1 ERP Benefits and Shortcomings

ERP systems, in theory, present many benefits. These include (Wang *et al.* 2006; Law and Ngai 2007; Chou and Chang 2008; Ragowsky and Gefen 2008):

- Integrated information flows;
- Automated and integrated business processes;
- Reduced operating costs;
- Improved customer service;
- Improved inventory levels; and
- Improved supplier interaction.

The benefits realised through the use of ERP systems, can be classified in terms of their positive impact on the operational efficiency of the enterprise or their impact on the efficiency of the entire enterprise (Chou and Chang 2008).

Despite the numerous benefits that ERP systems present, there is still a high failure rate of ERP projects. These failure rates range from 40 to 75 percent (Boudreau 2003; Cheng *et al.* 2006; Kwahk 2006; Wang *et al.* 2006; Wu *et al.* 2007; Chou and Chang 2008; Wu *et al.* 2008). The failure to successfully implement (on time and under budget) an ERP system can prevent the realisation of all the benefits, and could present various disadvantages to the enterprise.

Some of the general shortcomings of ERP systems include (Boudreau 2003; Babaian *et al.* 2004; Yeh 2006; Chou and Chang 2008; Ragowsky and Gefen 2008):

- Poor usability and unintuitive user interfaces;
- Complex to implement;
- Lengthy implementation process (years);
- Significant financial investment (hardware, training, implementation and support);
- Incompatibility between legacy systems and ERP system;
- Inadequate fit between business and system requirements;
- Inadequate training; and
- High failure rate.

Selecting the most appropriate ERP system can reduce the risk of implementation failure. The selection of an appropriate ERP system will contribute to minimizing the financial risks, and potentially minimizing the misalignment between the structure, processes and practices of an enterprise and the functionality offered by the ERP system (das Neves *et al.* 2004; Wei *et al.* 2005; Wang *et al.* 2006; Yang *et al.* 2007).

The next section will investigate the importance of usability as a selection criterion, when considering or purchasing an ERP system. Investigating usability as a selection criterion will indicate the value that enterprises place on ensuring the alignment between an ERP system and its users.

2.2.2 Usability Criteria for ERP Systems

Several existing ERP selection criteria exist, which incorporate and propose some form of usability as a selection criterion. Usability, as a selection criterion, is often represented as a non-functional or qualitative criterion (Şen *et al.* 2009). Several terms are commonly used to represent usability(Verville and Halingten 2003; das Neves *et al.* 2004; Wei *et al.* 2005; Law and Ngai 2007; Bueno and Salmeron 2008; Şen *et al.* 2009). These terms are presented in Table 2.1.

Usability Selection Criteria Terms for ERP systems	
User Friendliness	Intuitiveness of ERP system
User Interface	Usability
Ease of Learning	Learnability
Ease of Operation	Operability
Having friendly user interfaces	Understandability
Easy-to-use user interface	Ease of use

Table 2.1: Usability Selection Criteria Terms for ERP systems

The list of terms in Table 2.1 indicates that ERP systems should be usable, and that only those ERP systems which align with the users' goals should be considered by an enterprise for selection. This supports the main objective of this research – by demonstrating the need for usability in ERP systems, specifically those designed for small enterprises. ERP systems designed for small enterprises are discussed in the next section.

2.3 ERP Systems for Small Enterprises

ERP systems were initially designed for large enterprises. Due to an oversaturation of this market, ERP vendors are now focusing their attention on small enterprises (Morabito *et al.* 2005; Naidoo 2005; Vilpola *et al.* 2007).

Several fundamental differences exist between small enterprises and large enterprises. These differences can impact on the software systems used by these enterprises. Some of these differences are (Huin 2004; Liang and Xue 2004; Bruque and Moyano 2007):

- Small enterprises typically centralise decision-making (usually to a single person);
- Small enterprises, typically, do not have any clearly defined standard procedures;
- Small enterprises have limited long-term strategic planning;
- Small enterprises, typically, have a greater dependency on external expertise for software system support;
- Small enterprises face greater risks in software system implementation than large enterprises;
- Small enterprises do not function as a collection of formal departments; and
- Small enterprises differ in terms of their stages of growth.

Small enterprises are heterogeneous in nature, and operate in a dynamic, competitive and highly volatile economy. Constant innovation is needed, in order for these enterprises to survive in their rapidly changing business landscape (Shoniregun 2004; Olsen and Sætre 2007; Trimi 2008). Small enterprises, therefore, have to provide the highest level of operational efficiency with limited access to infrastructure, finance and human resources(Shoniregun 2004; Alexandre *et al.* 2006; Brun and Lanng 2006; Peng *et al.* 2007; Nach and Lejeune 2008; Trimi 2008).

The ability of small enterprises to sustain a competitive advantage through their agile business processes needs to be reflected by their choice of ERP system (Vilpola and Kouri 2006; Trimi 2008). ERP systems designed for small enterprises are different from those designed for large enterprises. The characteristics of ERP systems for small enterprises have shown that these systems are typically easier and quicker to implement and use. They are localised and customised to a particular industry and enterprise, and are adaptable and flexible enough to support the individuality of business processes, amongst small enterprises, in support of the dynamic business landscape of small enterprises (Liang and Xue 2004; Morabito *et al.* 2005; Vilpola and Kouri 2006; Vilpola *et al.* 2007).

ERP systems for small enterprises, as with ERP systems for large enterprises, are used in a variety of industry sectors, and are developed by several ERP vendors. The next section presents the results of a preliminary study conducted with ERP system vendors in South Africa.

2.4 Selection of an ERP System

The main objective this chapter is to select an industry sector and a suitable ERP system that can be used for experimental purposes for this research. This section will focus on the identification of an industry sector – and a specific ERP system for small enterprises for this research.

A survey of small enterprises in South Africa, conducted by World Wide Worx (Goldstuck 2006), established a list of the most widely used ERP/Accounting systems in South Africa. The distribution of these ERP systems, amongst small enterprises in South Africa, is illustrated in Figure 2.1. This distribution is based on the feedback provided by 4676 respondents.

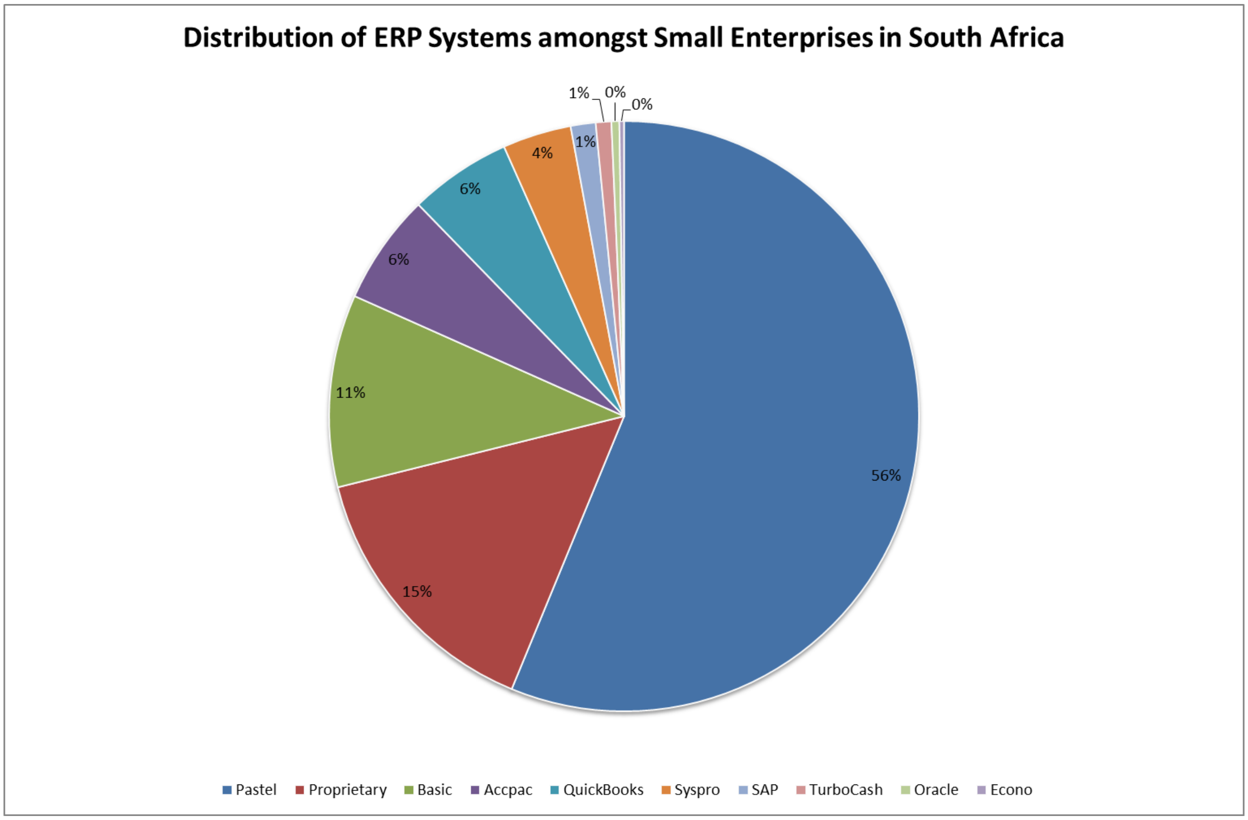


Figure 2.1: Distribution of ERP Systems used by Small Enterprises in South Africa (Goldstuck 2006)

The Proprietary and Basic ERP systems (Figure 2.1) represent a substantial proportion of the ERP systems currently being used in South Africa. These are, however, groupings of several systems; they do not represent a single system, and therefore, cannot be considered for evaluation in terms of the selection of an ERP system. The top five ERP systems currently used (Figure 2.1) are Pastel, Accpac, QuickBooks, Syspro and SAP. A comparison of these systems was performed by means of a preliminary study.

2.4.1 Preliminary Study

2.4.1.1 Goals and Objectives

The objective of the preliminary study was to assist in determining which industry sector and ERP system should be selected for the experimental purposes of this research. In order for a particular industry sector to be selected, it would need to be the most predominant sector addressed by all of the ERP vendors interviewed. A specific ERP system would be considered for selection – only if it supports most of the core activities of an ERP system and if the system were extensible – in terms of development after deployment.

2.4.1.2 Methodology

The preliminary study consisted of structured interviews, which made use of a questionnaire (Appendix B) sent out to the vendors of the Pastel, Accpac, QuickBooks, Syspro and SAP ERP systems in South Africa. Only three out of the five ERP vendors responded to the questionnaire and were willing to be interviewed, namely Pastel, QuickBooks and SAP.

2.4.1.3 Instruments

The questionnaire (Appendix B), which sought to obtain sales information from the ERP vendors on their ERP offerings to small enterprises in South Africa, consisted of six sections. The six sections of the questionnaire aimed to determine the:

- Biographical details of the respondent (Section A);
- Size and sector of small enterprises purchasing the ERP system (Section B);
- Business processes of the small enterprises, which required automation (Section C);
- Modules purchased by the small enterprises for the ERP system (Section D);
- Costs of implementing and supporting the ERP system (Section E); and
- General comments on factors affecting the sale of ERP systems to small enterprises (Section F).

These particular sections were chosen in order to determine:

- The most predominant small enterprise sector purchasing ERP systems (Section B);
- The business processes that small enterprises felt required the most automation (Section C);
- The ERP system modules that small enterprises purchased (Section D) – There might be a need to further delineate this research to a particular module or modules; and
- The amount that small enterprises are willing to spend on purchasing, implementing and supporting such an ERP system (Section E).

The results of the preliminary study will be discussed in the next section.

2.4.2 Results

The responses from the questionnaire revealed that Manufacturing was the most common small enterprise industry which purchased ERP systems. This was followed by the Services industry. The size of small enterprises purchasing ERP systems typically ranged from 21-200 full-time employees, with the exception of those purchasing Pastel, which ranged from 1-200 full-time employees.

The most common reasons why small enterprises purchased an ERP system included:

- Integration of core business processes;
- Increased productivity;
- Automation of certain business processes; and
- Timely access to management information.

Business processes frequently requiring automation through the implementation of an ERP system included:

- Finance and Accounting;
- Sales and Marketing;
- Inventory;
- Operations;
- Procurement; and
- Production and Manufacturing.

The most predominant business process automated is Finance and Accounting, followed by Sales and Marketing, and Inventory.

Administration is the least likely business process to be automated, as it differs between companies, and is not one of the core enterprise requirements. Production and Manufacturing is the business process which requires the most customisation. This is because it is based on products; and there are many different ways of implementing and automating the manufacturing process.

ERP modules commonly purchased by small enterprises included:

- Human Resources;
- Financials;
- Sales;
- Manufacturing;
- Inventory; and
- Reporting.

Customer Relationship Management (CRM) is also featured as a commonly purchased module / application. CRM is a supporting ERP application, and can be purchased as a stand-alone application. Complexity, learnability and satisfaction between the modules differed among vendors, with no commonality.

The costs associated in purchasing an ERP system showed that small enterprises are willing to spend as little R1000 up to more than R10000. This is also dependent on the financial resources of the small enterprise. Initial purchase costs showed a commonality in the region of greater than R10000. Small enterprises were also willing to pay as little – or less than R1000 – to more than R10000 – for their annual support for their ERP system. All of the vendors used a user licensing model to sell their software. This ranged from as little as, or less than R1000, to more than R10000.

Training was either handled by partners (3rd party vendors), or provided by the ERP vendor themselves. These costs ranged between R1000-R7000 per user.

Table 2.2 presents a comparison of the different ERP systems considered for this research. SAP Business One (SBO), the ERP system provided by SAP for small enterprises, was the only ERP system which provided support for extensibility post-deployment. This extensibility is provided through a software development kit (SDK), which provides support for 3rd party vendors and customers to customise the functionality offered by SBO.

QuickBooks does provide an SDK, but this is to support external applications linking to the QuickBooks database and not to enhance the functionality of the ERP system.

ERP System	*Core Modules*	*Extensibility*
SAP Business One	✓	✓
QuickBooks Premier	✓	✗
Pastel Evolution	✓	✗

Table 2.2: ERP System Comparison

2.4.3 Discussion

According to the criteria for selecting an industry sector (Section 2.4.1), the Manufacturing sector was identified as the most predominant industry sector (amongst all three vendors) in which small enterprises purchase ERP systems. The Manufacturing industry sector was, therefore, selected as the most appropriate industry sector for this research.

The results from the preliminary study have shown that small enterprises use traditional desktop direct manipulation-styled ERP systems, which are not browser based. SBO fulfilled all of the selection criteria, in terms of supporting the core activities of an ERP system and providing a means of extensibility post-deployment. Based on this, SBO was selected as the most appropriate ERP system for the purposes of this research.

2.5 Conclusion

This chapter presented a discussion on ERP systems and has shown that ERP systems consist of a core set of modules that, if implemented correctly, can be used to improve the operational efficiency of an enterprise.

The importance of usability as a selection criterion (when selecting an ERP system) has been clearly identified in this chapter. Usability, as a selection criterion, is important and necessary in order to align the functional requirements of the ERP system with the non-functional requirements of its intended users.

This chapter has shown that ERP systems for large enterprises have different requirements when compared with ERP systems for small enterprises. These differences need to be reflected in the requirements of the ERP system in order to support the heterogeneous and dynamic market in which small enterprises operate.

Based on the results of a preliminary study, this chapter has identified that small enterprises in the Manufacturing sector are most likely to purchase an ERP system. The preliminary study further identified that small enterprises that use ERP systems, typically use desktop-based ERP systems with direct manipulation user interfaces. Lastly, SBO was identified as the most extensible ERP system for small enterprises (through the inclusion of an SDK). Because of this level of extensibility (post-deployment), SBO was selected for experimental purposes. Therefore, this research will focus on small enterprises in the Manufacturing sector using SBO for experimental purposes.

The next chapter aims to identify the existing usability issues of ERP systems, and to determine whether any formal metrics or methods exist to evaluate the usability of ERP systems.

Chapter 3: Usability of ERP Systems

3.1 Introduction

The importance of usability with regard to enterprise resource planning (ERP) systems was identified and discussed in the previous chapter (Section 2.2.2). This chapter seeks to develop a further understanding of the relationship between usability and ERP systems by identifying any existing usability issues and determining how the usability of ERP systems can be measured and evaluated.

This chapter consist of three main sections. The first section provides an overview of usability, and how it can be measured and evaluated. This is followed by a discussion, which identifies some of the existing usability issues with regard to ERP systems. Lastly, a set of usability attributes and heuristics for evaluating the usability of ERP systems is proposed.

The proposed set of usability attributes and heuristics were published and presented at the 2009 Annual Conference of the South African Institute of Computer Scientists and Information Technologists (Singh and Wesson 2009).

3.2 Usability

Usability has become a great source of interest for both academia and industry practitioners. Usability is now recognised as a necessity in order to ensure the success of a software system, both commercially and socially (in an enterprise) (Abran *et al.* 2003; Johnson *et al.* 2007; Howarth *et al.* 2009).

3.2.1 Defining Usability

The term usability was introduced in the 1980s as a replacement for the term user-friendliness (Folmer and Bosch 2004; McNamara and Kirakowski 2005). Usability was introduced as an attribute of software quality and an objective of system design (Folmer and Bosch 2004). Several different definitions exist for usability due to the variety of approaches used to describe and measure usability (Abran *et al.* 2003; Folmer and Bosch 2004).

Some of the most widely used definitions of usability are those proposed by the ISO and Jacob Nielsen. ISO 9241-11 defines usability as:

> *"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specific context."* (ISO 1998)

This definition of usability focuses on the characteristics of software quality, as opposed to the definition provided in ISO 9126-1, which looks at usability from a software engineering perspective. ISO 9126-1 defines usability as:

> *"The capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use."*(ISO 2001)

Nielsen (1993) defined usability as "the measure of the quality of the user experience, when interacting with something – whether a web site, a traditional software application, or any other device that the user can operate in some way or the other". Usability, according to Nielsen, can be regarded as a sub-set of system acceptability (Figure 3.1).

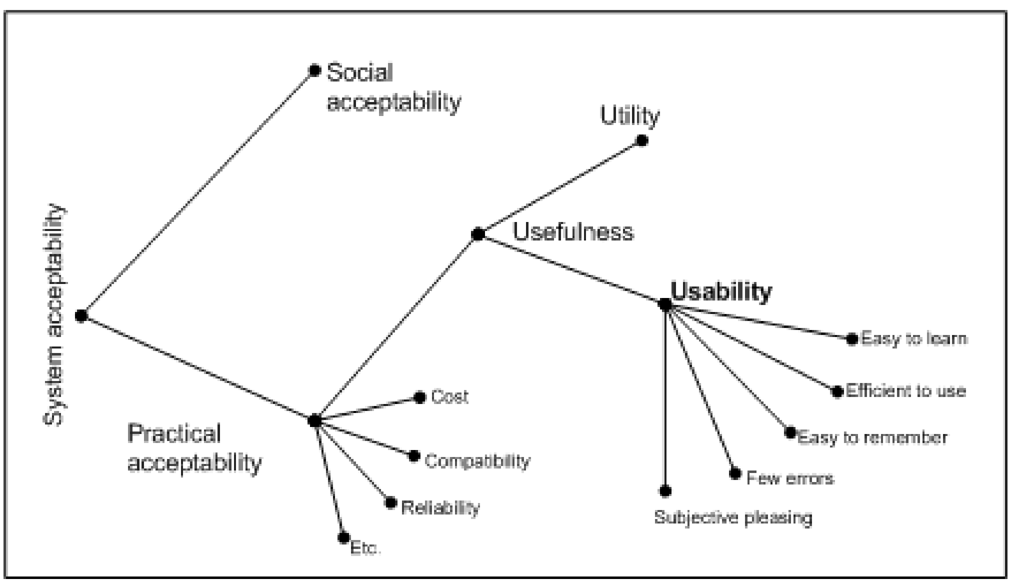


Figure 3.1: Usability as a Sub-Set of Acceptability (Nielsen 1993)

According to Nielsen, usability is a multi-dimensional property consisting of five attributes: learnability, efficiency, memorability, errors and satisfaction (Nielsen 1993). These five attributes are represented in Figure 3.1 under the term usability as: easy to learn (learnability), easy to use (efficiency), easy to remember (memorability), few errors (errors), and subjective pleasing (satisfaction). Nielsen (1993) states that these five attributes have an effect on the usability of a software system; and they can directly impact the usefulness of a system, which then influences the practical acceptability, and in turn, the acceptability of the software system.

Based on the above definitions, we can now state that usability is a multi-dimensional attribute of a software system, which aims to measure the complexity of the interaction between a user and a specific software system for a specific set of tasks (Henry 1998; Microsoft 2000; Lauesen 2005; McNamara and Kirakowski 2005; Juristo *et al.* 2007b, a).

Usability is a core component of this research, as this research focuses on improving the usability of ERP systems for small enterprises. The definition of usability, as defined in ISO 9241-11 (1998) was selected as the most appropriate for this research (Section 1.6). This particular definition was selected, because it is specific in what it attempts to measure in terms of usability.

3.2.2 Benefits of Usability

The aim of usability is to improve the rate at which users' complete a given task with the greatest amount of efficiency, effectiveness and satisfaction. Software systems that are usable present several benefits, as stated below (Microsoft 2000; Barnum 2002; Lauesen 2005; Juristo *et al.* 2007b, a):

- Improved user productivity;
- Improved user acceptance;
- Improved image and reputation of software system;
- Improved task efficiency;
- Improved user satisfaction;
- Improved product differentiation;

- Reduced training and documentation costs; and
- Reduced number of support calls.

Despite the high cost of usability testing, the benefits of a usable software system are far greater than the initial cost justification (Barnum 2002; Juristo *et al.* 2007a).

3.2.3 Usability Attributes

Usability attributes contribute to defining and measuring usability. There are several synonyms for usability attributes, namely usability properties, usability factors, usability characteristics, usability dimensions, usability components and usability scales (ISO 1998; Folmer and Bosch 2004). The term *usability attribute* will be used throughout this thesis.

An attribute may be defined as an internal or external, measurable physical or abstract property of an entity (ISO/IEC 1999). A usability attribute may be defined as a precise measurable component of that abstract concept identified as usability (Folmer *et al.* 2003).

An analysis of existing usability attributes (Appendix C) has resulted in the identification of the most common usability attributes. These usability attributes are: efficiency, learnability, memorability, reliability and satisfaction. These attributes are often defined, in order to suit a specific context or domain. A generalised definition for each attribute is provided below to clearly define the purpose of the attribute and the role it plays in contributing to usability.

3.2.3.1 Efficiency

A software system can be regarded as efficient when the users of that system (in a specific context of use) can complete a task and achieve the intended goals of that task in a specific unit of time – in order to achieve the highest level of productivity (Nielsen 1993; ISO 1998; Folmer *et al.* 2003; Seffah *et al.* 2006).

3.2.3.2 Learnability

Learnability is often defined as the ease with which a software system can be learned by its intended users, so that they can rapidly accomplish their required tasks efficiently and productively (Shackel 1991; Nielsen 1993; ISO 2001; Folmer *et al.* 2003; Folmer and Bosch 2004; Seffah *et al.* 2006).

3.2.3.3 Memorability

Memorability refers to the ease with which the operations of a software system can be remembered. A software system should be easy to remember, so that a user, who has not used the system for a prolonged period of time, should be able to quickly and easily be able to operate the system without any need to re-learn how to use the system(Nielsen 1993).

3.2.3.4 Reliability

Reliability refers to the capability of the software system to possess a low error rate and enable users of the system to recover quickly when errors are made (Nielsen 1993; Folmer *et al.* 2003).

3.2.3.5 Satisfaction

Satisfaction is a subjective usability attribute. This attribute describes the user's level of emotional fulfilment, when using a particular software system (Folmer *et al.* 2003; Seffah *et al.* 2006). In order to provide the highest level of satisfaction, interacting with the software system should be a comforting and pleasurable experience (Nielsen 1993; ISO 1998).

3.2.4 Usability Benefits

The usability attributes discussed in the previous section map directly onto the benefits that can be achieved by supporting usability (Section 3.2.2). This mapping is presented in Table 3.1.

Usability Attribute	Usability Benefit
Efficiency	Improves task efficiency
	Improves user productivity
Learnability	Reduces training and documentation costs
Memorability	Improves user productivity
Reliability	Reduces number of support calls
Satisfaction	Improves user experience
	Improves user adoption

Table 3.1: Usability Attributes and Benefit Mapping

Table 3.1 shows that providing a software system that is efficient to use can improve user productivity and task efficiency. Additional training and document costs are reduced when the system is easy to learn. Providing a system with operations that contribute towards memorability assists in improving user productivity. Costs can further be reduced by providing a system that efficiently handles errors and provides effective support for recovering from errors. Greater levels of adoption and satisfaction can be achieved by providing a system, which creates a comforting, pleasurable and positive experience.

This section has defined and discussed the most common usability attributes. These attributes need to be converted into usability metrics, in order to be measured effectively.

3.2.5 Usability Metrics

Usability cannot be measured directly, since it requires the use of usability measures (metrics) (Hornbæk 2006). Usability metrics are the measurable form of usability attributes (Dix *et al.* 2004). Metrics for evaluating usability assist in indicating the quality-in-use of software systems (Constantine and Lockwood 1999; Hornbæk and Law 2007). According to the Institute of Electrical and Electronics Engineers (IEEE), a software metric is that which has software data for input and a single numerical value for output.

This single value, when interpreted, indicates the degree to which the software system possesses a given attribute, which could affects its quality (ISO 2001; Seffah *et al.* 2001; Winter *et al.* 2007).

Usability metrics can be divided into objective and subjective metrics (ISO 1998). Objective metrics include effectiveness and efficiency. Effectiveness indicates the degree to which the users' goals are achieved in terms of accuracy and completeness. Metrics for efficiency provide an indication of the level of effectiveness that has been achieved. Satisfaction is regarded as a subjective measure. Satisfaction metrics are those metrics, which aim to measure the user's perception of the user interface or the user's interaction with the software system (ISO 1998; Hornbæk 2006).

Examples of usability metrics for the relevant usability attributes (effectiveness, efficiency and satisfaction), are presented in Table 3.2.

Usability Objectives	*Effectiveness Metrics*	*Efficiency Metrics*	*Satisfaction Metrics*
Overall usability	Percentage of goals achieved; Percentage of users successfully completing tasks; Average accuracy of completed tasks;	Time to complete a task; Tasks completed per unit time; Monetary costs of performing the test;	Rating scale for satisfaction; Frequency of discretionary use; Frequency of complaints;
Learnability	Number of functions learned; Percentage of users who manage to learn the criterion;	Time to learn criterion; Time to re-learn criterion; Relative efficiency while learning;	Rating scale for ease of learning;
Error Tolerance	Percentage of errors corrected or reported by the system; Number of user errors tolerated;	Time spent correcting errors;	Rating scale for error handling;

Table 3.2: Examples of Usability Metrics Adapted from ISO 9241-11 (1998)

Usability metrics can also be classified into three broad categories, namely performance metrics, preference metrics and predictive metrics (Constantine and Lockwood 1999).

3.2.5.1 Performance Metrics

Usability metrics which intend to measure the actual performance of the user, when completing a task in a certain context may be classified as performance metrics. Performance metrics provide an indication of the user's actual interaction with the system, such as the following (Shneiderman 1998; Constantine and Lockwood 1999; Seffah *et al.* 2001):

- Time to complete a task;
- Time to learn the software product;
- Number of user-induced input errors corrected;
- Number of functions successfully tested;
- Rate of errors by the user; and
- Frequency of help requests.

3.2.5.2 Preference Metrics

Preference metrics are those metrics, which attempt to quantify the subjective evaluation and the preference of the users for any particular software system. Preference metrics can be measured using satisfaction-based questionnaires, such as the Software Usability Measurement Inventory (SUMI) questionnaire, or the Questionnaire for User Interface Satisfaction (QUIS). Examples of preference metrics include (Constantine and Lockwood 1999; Seffah *et al.* 2001):

- How easy the software system is to explore and master (Learnability);
- How well the software system supports productive use (Efficiency);
- How easily the interface of the software system can be configured (Customisation);
- How easy it is to correct system errors (Operability); and
- How easily the functions of the software system can be found (Understandability).

3.2.5.3 Predictive Metrics

Usability metrics, which assess the design quality of prototypes, are referred to as predictive metrics (Constantine and Lockwood 1999; Seffah *et al.* 2006). There are five essential usability metrics, which are classified as predictive metrics. These include (Constantine and Lockwood 1999; Seffah *et al.* 2006):

- Essential Efficiency – measures the degree to which a specific user interface design matches the actual interactions required by the user;
- Task Co-ordination – measures the degree to which the frequency of tasks match their expected level of difficulty;
- Task Visibility – measures the fit between the visibility of features and the capabilities needed to complete a given task;
- Layout Uniformity – measures the spatial arrangement of the elements on the user interface; and
- Visual Coherence – measures the degree to which the user interface keeps related things together.

The values obtained through the use of predictive metrics indicate an estimate or prediction of usability (Seffah *et al.* 2006).

3.2.5.4 Discussion

This section has shown how usability attributes can be measured, by converting them into metrics (either objective or subjective). Categories of metrics in the form of preference, performance and predictive metrics have been discussed. These metrics extend the objective and subjective classification, by also including metrics which can be used to evaluate the design of the user interface. The next step is to determine how to utilise these metrics by discussing the various methods of evaluating usability.

3.2.6 Usability Evaluations

Usability evaluations assess the quality of a software system (Costabile and Matera 2001; Kjeldskov *et al.* 2010). The process of evaluating usability is experimental in nature, and involves collecting data, based on the users' interactions with the software system (Daniels *et al.* 2007).

There are two types of usability evaluations, namely formative and summative. Formative evaluations are conducted during the development of a system, in order to form or influence design decisions. Summative evaluations are conducted after the product is finished, to ensure that the product meets certain standards or satisfies certain requirements set by the sponsors or other agencies (Barnum 2002; Te'eni *et al.* 2006).

Other types of usability evaluations include use and impact evaluations, and longitudinal evaluations. Use and impact evaluations are conducted during the actual use of the product by real users in a real context. Longitudinal evaluations involve the repeated observation or examination of a set of subjects over time – with regard to one or more evaluation variables (Te'eni *et al.* 2006).

Methods for evaluating usability can generally be classified as: analytical or empirical. These two primary classifications could be in the form of expert-based methods, user testing methods, exploratory methods and analytical methods (Tselios *et al.* 2008). Figure 3.2 illustrates this classification.

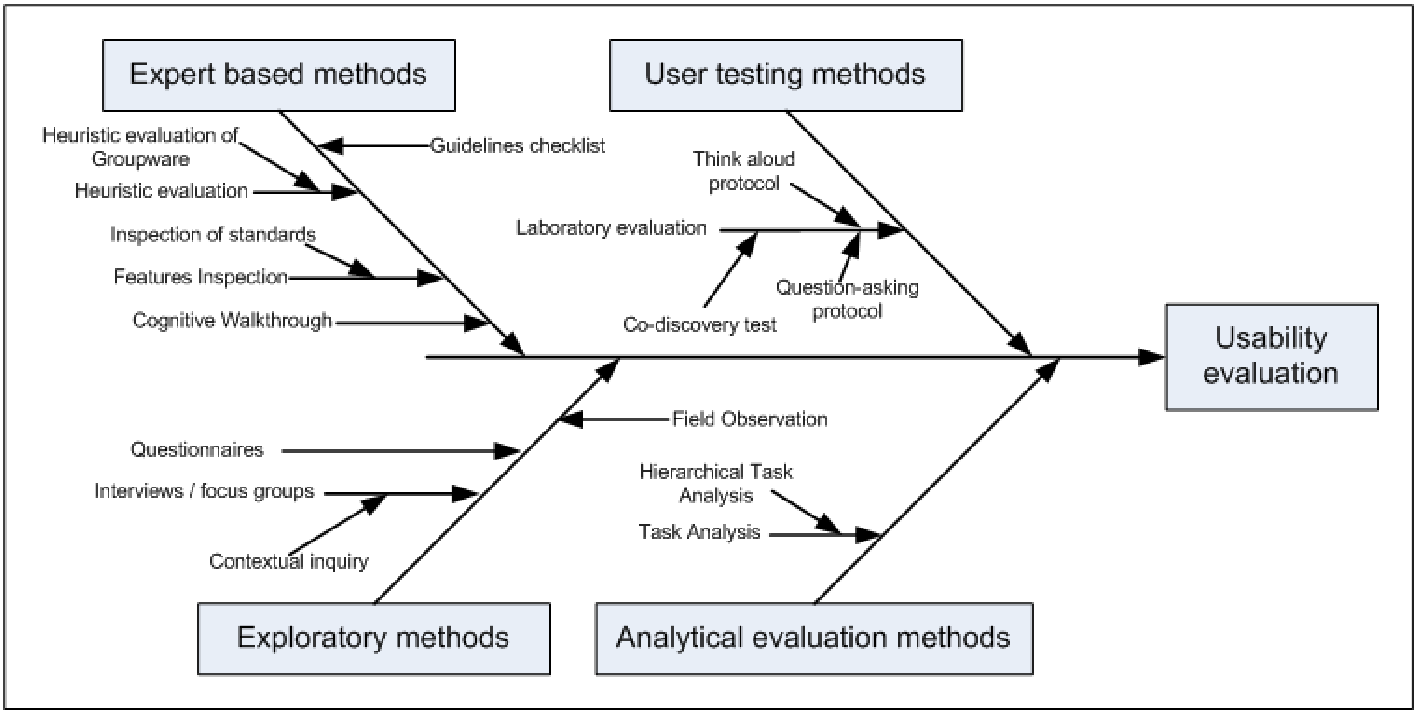


Figure 3.2: Taxonomy of Usability Evaluation Methods (Tselios *et al.* 2008)

3.2.6.1 Analytical Methods

Analytical evaluation methods are typically conducted in a very structured manner, by usability experts or designers. Often regarded as inspection methods, these methods aim to inspect the user interface for potential usability issues. Analytical methods assist in identifying potential usability issues by having the evaluators (usability experts) inspect a user interface with a set of guidelines or heuristics (Jaspers 2009).

Some of the most common analytical usability evaluation methods are: heuristic evaluations, guideline review, cognitive walkthrough, pluralistic walkthrough, framework-based inspection and user-model based analysis (Te'eni *et al.* 2006). Other examples of analytical methods can be found in Figure 3.2 under analytical evaluation methods and expert based methods. The most common analytical method and the most widely adopted usability evaluation method is the heuristic evaluation (Edwards *et al.* 2008; Jaspers 2009).

3.2.6.2 Empirical Methods

Empirical methods of conducting a usability evaluation involve the actual users of the software system. These methods collect facts about the users' interactions with the actual software system. The data collected from an empirical method can either be qualitative or quantitative (Te'eni *et al.* 2006). Common empirical usability evaluation methods include: surveys or questionnaires, interviews including focus groups, lab-controlled experiments and field studies (Barnum 2002; Te'eni *et al.* 2006; Jaspers 2009). Other examples of empirical methods can be found in Figure 3.2 under user testing methods and exploratory methods.

3.2.6.3 Comparison of Evaluation Methods

Selecting a particular method of testing is often a difficult task, as each method has its own merits and trade-offs. A key difference between the empirical and analytical methods of evaluating usability is that analytical methods are cost effective, but do not involve the actual users of the software system (Hollingsed and Novick 2007). Empirical methods are traditionally costly but involve the actual users of the software system. Analytical evaluations are also only capable of identifying potential usability issues. Actual usability issues can only be identified using empirical evaluation methods. The most common analytical method used is heuristic evaluation and the most common empirical method used is the lab-controlled experiment.

3.3 Usability Issues of Existing ERP Systems

Currently, little work has been done in the area of usability of ERP systems. Most of the studies that have been conducted are discussed in the following paragraphs. A study conducted by Yeh (2006) on evaluating the performance of ERP systems from a user's perspective, revealed that the ERP systems suffer from poor usability and are difficult to use. The poor usability of ERP systems is the result of these systems being complex, inflexible and difficult to use (Yeh 2006). Users from this study stated that interacting with the ERP system (used in the study) left them with a sense of information overload.

A study conducted by Topi (2005) aimed to narrow this gap, by publishing a paper, which indicated the usability issues identified of an ERP system. These issues included:

- Navigation, in terms of finding functionality and information efficiently, was a complex and tedious process;
- Guidance from the ERP system, to ensure accurate navigation and task completion, was limited;
- The capability to adapt to support users' actions, and to ensure task completion was lacking;
- Users were unable to retrieve frequently accessed data efficiently;
- Any presentation of the output was difficult to understand and interpret; and
- The user interfaces were complex and intimidating to novice users.

The complexity of the user interface of ERP systems, and the difficulty with navigation were also identified in a study conducted by Matthews (2008). This study further identified that some parts of the ERP system required different commands and different types of interaction. Users who participated in this study mentioned that it was difficult to transfer information from one section of the ERP system to another. The difficulty in finding information and understanding how to navigate, was identified as the biggest usability issue of ERP systems (Matthews 2008).

A need exists to identify and understand the usability issues of ERP systems. Evaluating the usability of ERP systems requires that specific usability attributes and metrics for ERP systems be identified. Existing usability attributes and metrics to measure ERP systems will be discussed in the next section.

3.4 Measuring ERP Usability

There is currently no standard way to determine the usability of an ERP system. Several usability evaluations of ERP systems were reviewed and analysed to determine the most common usability attributes and metrics.

3.4.1 Existing ERP Usability Attributes and Metrics

Several existing usability studies on ERP systems were identified and analysed. These studies were conducted by the following authors: Matthews (2008), Keystone (2007), Herbert *et al.*(2006), Topi *et al.* (2005), and Calisir and Calisir (2004). The usability evaluations conducted in these studies evaluated the usability of ERP systems by using an heuristic evaluation approach. The review found that despite the consistent use of Nielsen's ten heuristics, other non-standard usability attributes and metrics were added.

The additional attributes and metrics used to evaluate the usability of the various ERP systems, varied between the different evaluations, with only some commonalities. The commonality amongst the various attributes and metrics was analysed (Appendix D), in order to propose a set of usability attributes and metrics that would be capable of supporting the identification of usability issues with regard to ERP systems.

The analysis revealed that the most common heuristics used, which are specific to ERP systems, are those which relate to: Navigation, Presentation (layout and output), Task Support, Learnability, and Customisation. These common attributes correspond to the discussion on usability issues of ERP systems (Section 3.3).

3.4.2 Proposed ERP Heuristics

Nielsen's ten heuristics assist in assessing the general usability of a software system. These heuristics, however, do not support the evaluation of ERP systems in terms of Navigation, Presentation, Task Support, Learnability and Customisation. These five attributes (Figure 3.3) are used in this section – as a basis for proposing a set of usability heuristics specific to ERP systems.

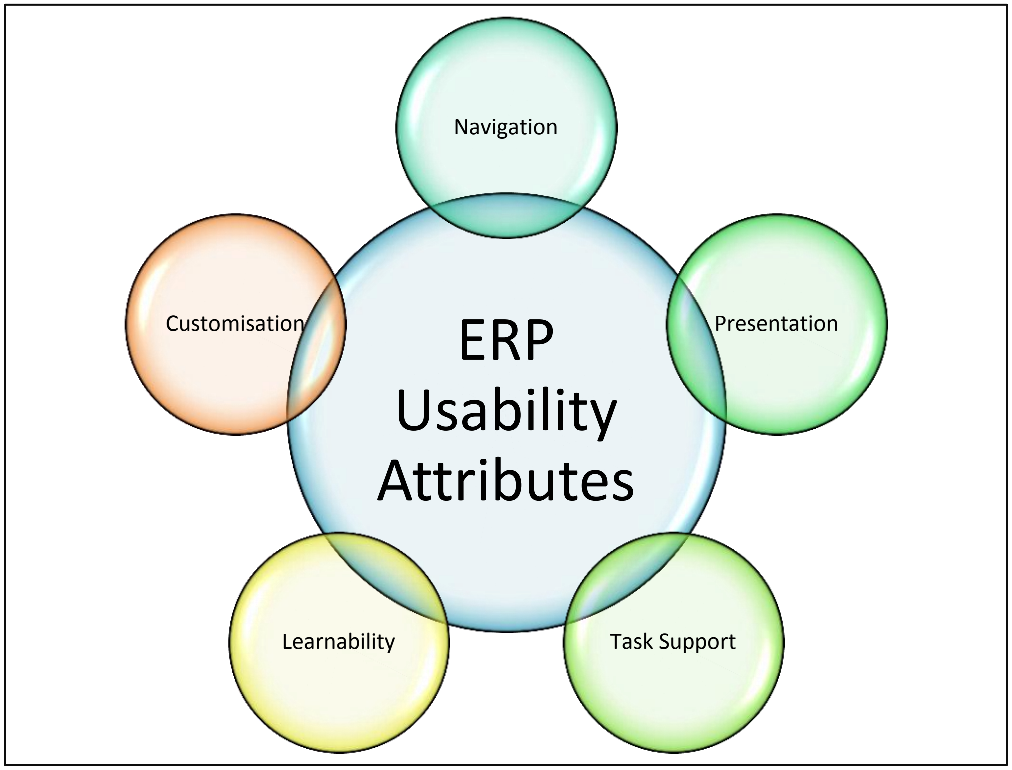


Figure 3.3: Proposed ERP Usability Attributes

The proposed ERP heuristics could be combined with Nielsen's ten heuristics, to identify potential usability issues with an ERP system. Another reason for proposing a set of ERP heuristics is that current heuristics used to evaluate the usability of ERP systems are inconsistent. The use of inconsistent heuristics makes it difficult to compare the results of heuristic evaluations of ERP systems. A standardised set of heuristics would eliminate this inconsistency, and allow for the results of heuristic evaluations performed on ERP systems to be directly comparable.

Using the five attributes (Navigation, Presentation, Task Support, Learnability, and Customisation) to assess the usability of an ERP system requires that these attributes be converted into usability metrics or heuristics. This would enable these new heuristics to be used in a heuristic evaluation. Table 3.3 contains a list of the proposed attributes and the expanded heuristic form.

Usability Attributes	*Heuristics*
Navigation	Navigation and access to information and functionality
Presentation	Presentation of screen and output
Task Support	Appropriateness of task support
Learnability	Intuitive nature of system
Customisation	Ability to customise

Table 3.3: Proposed ERP Usability Attributes and Heuristics

3.4.2.1 Navigation

Navigation is a major design issue for most ERP systems (Topi *et al.* 2005; Matthews 2008). To overcome this issue, the heuristic, *Navigation and access to information and functionality*, is proposed. This aims to determine the ability to identify and access appropriate information, system functionality, menus, reports, options and elements accurately and effectively. This heuristic aims to determine whether:

- Information can be easily accessed;
- Functionality can be found quickly and easily;
- Guidance through the correct sequence of transactions to complete a business process is available;
- The existing user interface design supports efficient and accurate navigation;
- Functionality to search for information is available;
- Searched items and the required information are correlated;
- The different interaction styles of the various users are supported;
- Alternative navigation techniques are supported;
- Guidance-type information is provided; and
- The next sequence of transactions or steps is clearly indicated.

3.4.2.2 Presentation

One of the other major issues identified in the usability studies involving ERP systems was that the user interface was considered complex, and that output from the ERP system was difficult to understand and interpret (Topi *et al.* 2005; Matthews 2008). This is due to the large amount of data required from the users, and the lack of effective and logical groupings of fields. Other reported issues include inaccurate representation of information in the form of reports.

In order to overcome this, the heuristic, *Presentation of screen and output* is proposed. This heuristic aims to determine the appropriateness of the layout of menus, dialog boxes, controls, and information on the screen for data entry and output generation. This heuristic is used to determine whether:

- Visual layout is well designed;
- Information provided by the ERP system is timely, accurate, complete and understandable;
- Output is well structured and easy to understand and interpret;
- The information presented supports informed decision-making;
- The output provides clear visibility into the various other departments; and
- The user interfaces are intuitive.

3.4.2.3 Task Support

The misalignment between the ERP system and the business processes of an enterprise often creates the greatest amount of complexity and resistance to using the ERP system (Topi *et al.* 2005; Matthews 2008). To address this issue, the heuristic, *Appropriateness of task support*, is proposed. This heuristic aims to establish whether there is an accurate alignment between the ERP system and the real world. This is necessary, in order to ensure effective task support and efficient task completion. Evaluating this heuristic is done by determining whether:

- Terminology used by the ERP system is consistent with the terminology of the user;

- Information provided by the ERP system is in real time;
- Responses from the ERP system are quick and efficient;
- Tasks can be completed efficiently;
- User productivity is improved;
- Routine and redundant tasks are automated;
- Ease-of-use of the ERP system; and
- Information flow between the various organizational departments is improved.

3.4.2.4 Learnability

ERP systems are traditionally regarded as being complex to learn and use (Topi *et al.* 2005; Matthews 2008). To evaluate this aspect, the heuristic, *Intuitive nature of system* is proposed. The aim of this heuristic is to determine the degree of effort required to learn how to use the ERP system efficiently. This heuristic is evaluated by determining whether:

- A user can learn how to use the ERP system without a long introduction;
- The various functions of the ERP system can be identified by exploration;
- There is sufficient online help to support the learning process;
- It is easy to become skilful in using the ERP system within a short amount of time; and
- The ERP system is intimidating and complex to learn and use.

3.4.2.5 Customisation

A key aspect of any ERP system is that it should be customisable (Topi *et al.* 2005; Matthews 2008). Customisability is measured in terms of whether the ERP system:

- Meets the needs of the enterprise and its processes;
- Meets the individual requirements of the users; and
- Supports the users' business processes.

The heuristic, *Ability to customise*, is proposed to support these needs. The aim of this heuristic is to evaluate the ease of customising the ERP system, in order to ensure accurate alignment between the ERP system and the business processes, the system and its users, and the ERP system's involvement with its user and the business process. Evaluating this heuristic requires that the following be determined:

- The ease with which the ERP system can be configured to a particular industry type;
- The capability of the ERP system to support user-level customisation;
- The capability of the ERP system to support customisation for the user at a transaction level;
- The alignment between the ERP system and the business processes of the enterprise;
- The ability of the ERP system to update existing business processes, and (or) to include new ones;
- The ability of the ERP system to be re-configured over a period of time; and
- The ability of the user interface to be configured without affecting the underlying business logic of the ERP system.

3.4.3 Discussion

Current usability evaluations of ERP systems use different attributes and metrics in order to measure usability. This section has reviewed existing usability evaluations of ERP systems; and it has identified the most common attributes and metrics. The five common attributes identified include: Navigation, Presentation, Task Support, Learnability, and Customisation. These attributes were expanded into heuristics to support the identification of usability issues of ERP systems. The heuristics proposed could be used together with existing general usability heuristics (i.e. Nielsen's ten heuristics) – in order to form a comprehensive set of heuristics that evaluates both the general and the ERP-specific aspects of ERP systems. The use of these heuristics could also provide a common basis for comparing the results of usability evaluations of ERP systems.

3.5 Conclusion

This chapter has presented a discussion on the usability of ERP systems. Furthermore, it has ascertained that usability is required to ensure the success of a software system. The five key attributes of usability were identified, namely learnability, memorability, reliability and satisfaction. Evaluating usability requires the use of usability metrics. This chapter has identified three major categories of usability metrics, namely performance metrics, preference metrics and predictive metrics.

This chapter has further identified and compared the two key usability evaluation approaches, namely analytical evaluation and empirical evaluation.

This chapter has revealed that ERP systems typically suffer from usability issues relating to: Navigation, Presentation, Task Support, Learnability and Customisation. Usability evaluations of ERP systems are usually in the format of an heuristic evaluation (analytical evaluation). These evaluations have shown that inconsistent attributes and heuristics are used to evaluate the usability of ERP systems. The need for a consistent set of usability attributes and heuristics specific to ERP systems was identified.

This chapter has made a novel contribution by proposing a set of usability attributes and heuristics for ERP systems. These attributes were identified through an extensive literature review of usability evaluations of ERP systems. The proposed attributes are aligned with the major areas in which ERP systems experience usability issues, namely Navigation, Presentation, Task Support, Learnability and Customisation. These attributes were expanded into heuristics to support the identification of usability issues of ERP systems.

The next chapter will make use of the proposed ERP heuristics in a usability evaluation of SBO. The aim of the usability evaluation is to identify the usability issues of SBO. SBO was selected as the most suitable ERP system (for small enterprises) for experimental purposes in Chapter 2.

Chapter 4: A Usability Evaluation of SAP Business One

4.1 Introduction

The aim of this chapter is to identify the usability issues of SAP Business One (SBO). SBO was selected in Chapter 2 as an example of a typical enterprise resource planning (ERP) system designed for small enterprises and will be used for the usability evaluation. The usability evaluation will be conducted in the form of an interview study and an heuristic evaluation.

The remainder of this chapter is structured as follows: a discussion is presented on the interview study that was conducted. This is followed by a discussion on the results of the heuristic evaluation. The heuristic evaluation used the proposed ERP heuristics (Chapter 3) together with Nielsen's ten heuristics. In order to validate the proposed ERP heuristics, a discussion is presented comparing the usability issues identified by Nielsen's ten heuristics with the usability issues identified by the proposed ERP heuristics. This is followed by a set of recommendations on how to improve the overall usability of SBO.

The results of the heuristic evaluation were published and presented at the 2009 Annual Conference of the South African Institute of Computer Scientists and Information Technologists (Singh and Wesson 2009).

4.2 Interview Study

An interview study was conducted of small enterprises in the Manufacturing sector that used SBO. The aim of the interview study was to identify any possible usability issues that the users experience with SBO. A secondary objective of the interview study was to identify the most common tasks that users perform with SBO.

4.2.1 Research Design

The interview study comprised structured interviews and observations. The observations were used to complement the interviews, so that the users could explain by demonstrating using SBO, where problems were experienced.

A questionnaire (Appendix E) was used as a guide for the interview study. The questionnaire consisted of two set of questions – one set, which was aimed at the technology driver (the person who initiated the SBO initiative in the small enterprise), while the second set aimed at the business user (the user of SBO). The questions in the technology driver section related to the overall usage of SBO within the enterprise. The business user section contained questions relating to Navigation, Presentation, Task Support, Learnability and Customisation. These specific questions were asked, as they were based on the ERP-specific attributes, as proposed in Chapter 3.

The interview study revealed that the technology drivers were also users of SBO; and therefore, the business user section of the questionnaire also applied to the technology drivers.

4.2.1.1 Participant Selection

Fifty small enterprises in the Manufacturing sector (using SBO) in South Africa were contacted regarding their willingness to participate in the interview study. Only 12% (six) of the small enterprises contacted were willing to participate in the interview study. Many of the enterprises that use SBO in South Africa were not willing to participate in this study, as they felt that it could impact on their operations and data security. Eighteen users in total were interviewed and observed. These eighteen users were distributed across the small enterprises that were willing to participate.

4.2.2 Results

Feedback obtained from the interview study was in the form of qualitative data. These data were obtained from a sample of six small enterprises and eighteen users. An analysis of the qualitative data was done, using a thematic analysis; and the results were categorised in terms of the proposed ERP criteria (Chapter 3). Once the data were categorised, a quantitative analysis of the data (against the sample) was conducted, in order to identify the segmentation within the data. The results are discussed below in terms of their categorisation.

4.2.2.1 Business Process Alignment

The alignment of SBO with the business processes of the small enterprises showed that SBO was either completely aligned (100%); or, it provided a 50% alignment; or, it provided very little or no alignment. Seventeen percent of the small enterprises interviewed indicated that SBO provided a 100% alignment with their business processes, whilst 33% indicated that there was a 50% alignment with their business processes. Fifty percent of the small enterprises interviewed indicated that there was very little or no alignment of SBO with their business processes. Figure 4.1 illustrates this distribution.

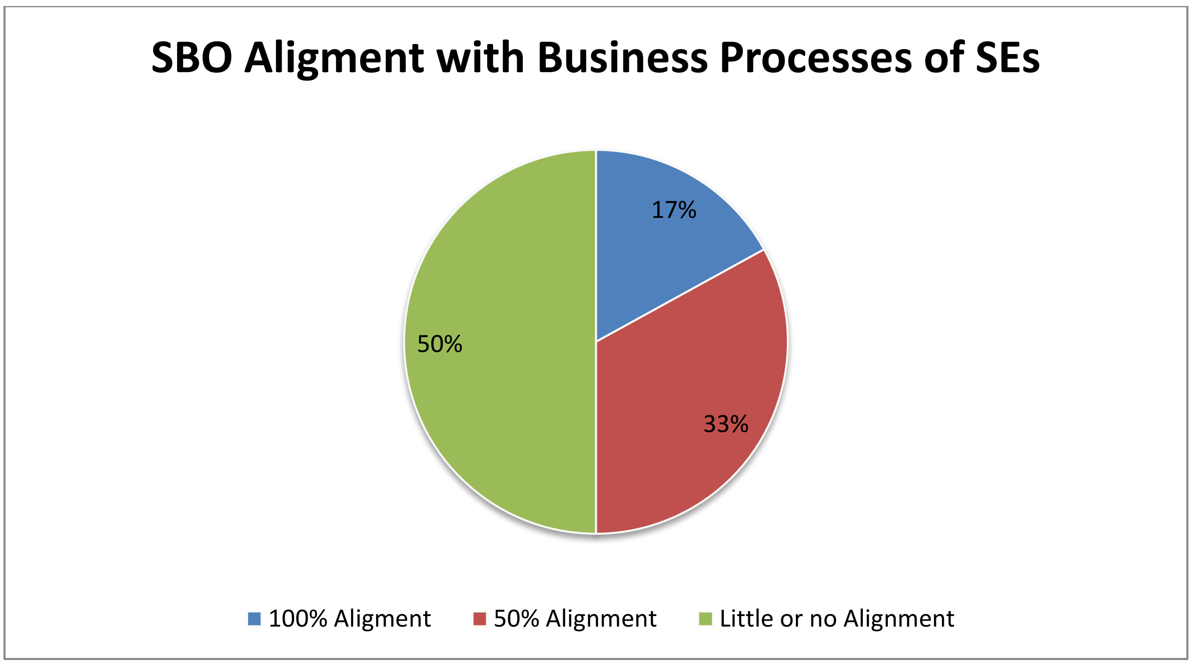


Figure 4.1: Alignment of SBO with Business Processes of Small Enterprises (n=6)

The small enterprises, which experienced a complete (100%) or a partial alignment (50%) to their business processes, appreciated the fact that SBO introduced some form of standardisation to their business processes. Those small enterprises, which experienced little or no alignment, were struggling to overcome this challenge; and they found that SBO introduced greater complexity and costs into their traditional business processes.

Eighty-three percent of the technology drivers (the people responsible for the introduction of SBO in the enterprise) expressed concerns with the reporting capabilities of SBO. The lack of effective and efficient reporting functionality in SBO required the small enterprises to purchase and use additional reporting tools that interfaced with Microsoft Excel (Excel). Currently, data are exported from SBO to Excel, where it is manipulated and managed "a lot more easily" than with SBO.

Eighty-three percent of the small enterprises stated that reports are currently generated with Excel. Only one of the small enterprises was comfortable with reports using SBO, but this small enterprise still used Excel for some of its reports.

4.2.2.2 Navigation

The ability to access information and functionality effectively and efficiently was generally disliked. Seventy-two percent of the users regarded the process of finding appropriate information and functionality as generally difficult. The remaining 28% of users regarded the ability to access appropriate information and functionality as good. Based on the responses of the users, there was a request for improved guidance, to overcome the challenge of finding appropriate information and functionality.

4.2.2.3 Presentation

The presentation aspect of SBO was split in two categories. The first category was presentation, in terms of the layout of the user interface, while the second category related to the output provided (reports).

User Interface Layout

Seventeen percent of users felt that SBO provided a very structured request for data. These users further mentioned that they did not use a lot of the fields that were displayed on the user interface. SBO does provide the functionality to hide certain fields that are not mandatory. Users who were aware of this functionality made use of it. Those who were not aware of this functionality ignored the unused fields.

Fifty percent of the users found that the layout of the user interface did not affect their daily tasks, whilst 22% indicated that it was "very user friendly". The remaining 28% of the users did not like the user interface in terms of layout; and they felt that it did not contribute to the successful and efficient completion of their daily tasks. Figure 4.2 illustrates this distribution.

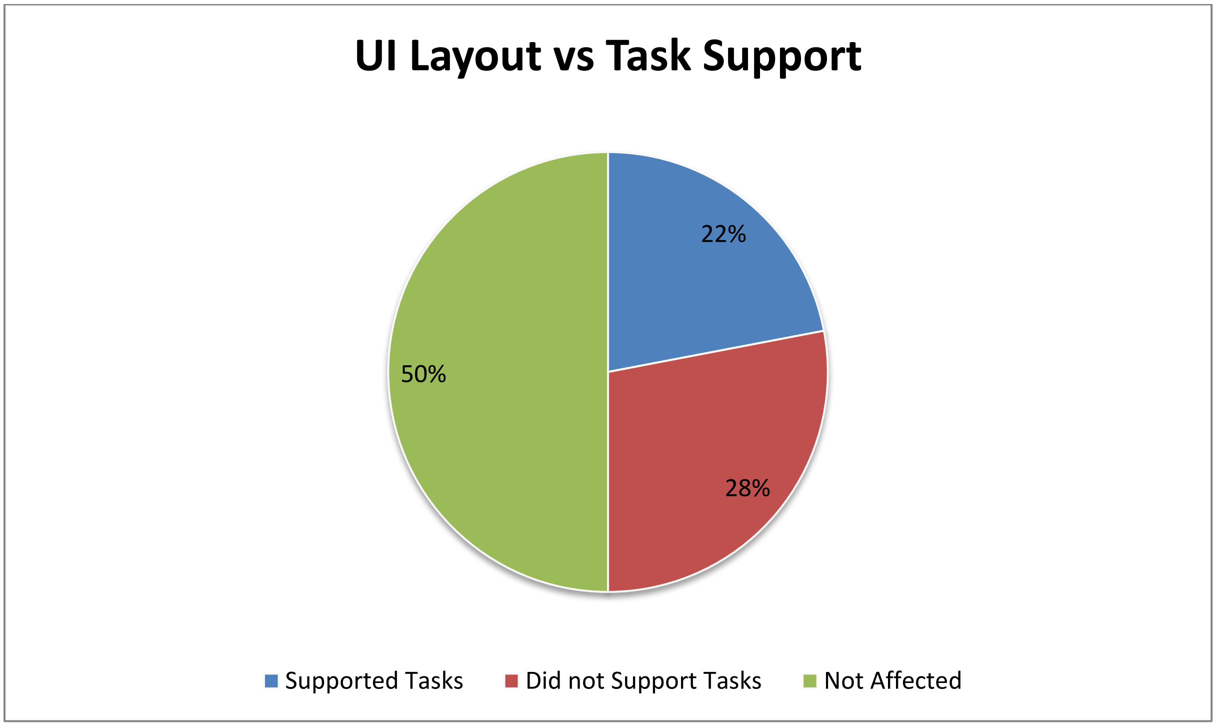


Figure 4.2: Presentation of SBO in terms of Task Support (n=18)

Reports

Reports were the most problematic issue with regard to SBO. Only 11% of the users interviewed were satisfied with the reporting functionality of SBO. The remaining 89% of the users found the reports uninformative and inaccurate. This made the process of making informed decisions from reports an ongoing challenge. Reports currently take too long to generate, resulting in the correction of values and the re-generation of reports, which is very time-consuming. All of the small enterprises interviewed use Excel as their choice of reporting engine. The decision to use Excel was taken – in order to eliminate the above-mentioned challenges when using the reporting function of SBO.

Experienced users indicated that more customisation was required, in order for reports to meet their own preferences (i.e. graphs instead of tables).

4.2.2.4 Task Support

Seventy-two percent of the users viewed the alignment between SBO and their daily tasks as accurate. The remaining 28% felt that there was no alignment between SBO and their daily tasks. This made working with SBO very difficult and frustrating. Figure 4.3 illustrates this distribution.

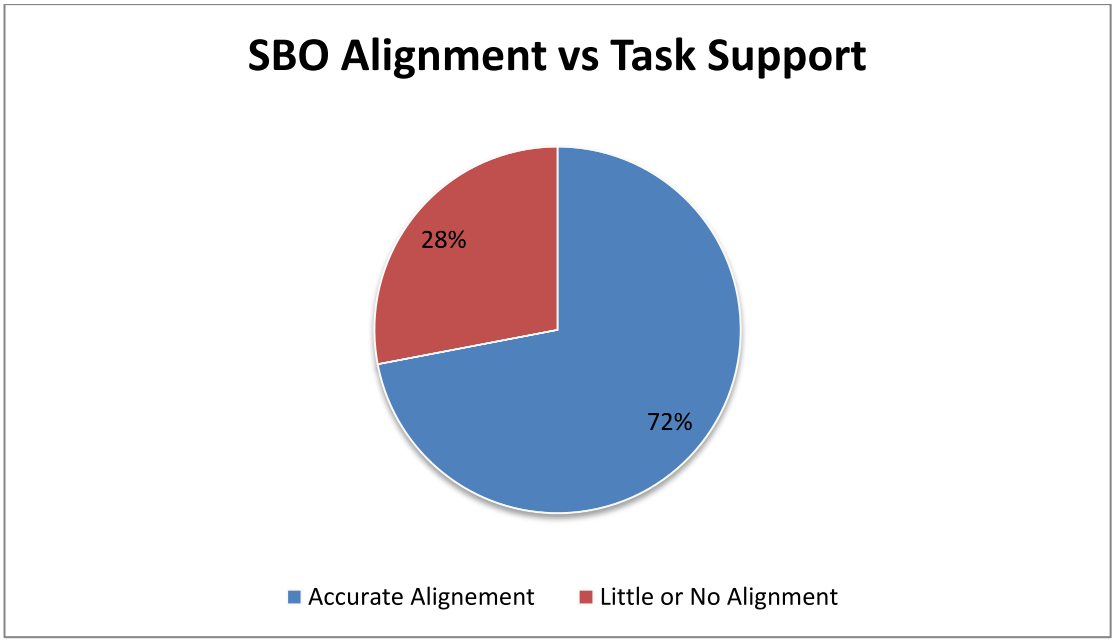


Figure 4.3: Alignment of SBO with Daily Tasks of User (n=18)

Thirty-nine percent of the users felt that too many steps were required to complete a task. This made completing tasks a long and tedious activity.

4.2.2.5 Learnability

Learnability of SBO was considered to be an initial challenge. New users (those with little or no ERP experience) found SBO to be unintuitive, confusing and difficult to use. Sixty-one percent of the users indicated that SBO was easy to learn. The remaining 39% felt that learning SBO was a challenge. Initially, these users had to make a note of all the various steps that needed to be completed, in order to complete a task. These users felt that learning something new would be a challenge, as a whole new sequence of steps would need to be remembered. Another factor which influenced the learnability of SBO was the terminology used, because it is different from the terms to which the users are exposed on a daily basis.

The Help functionality within SBO was either: not installed; not used; or if it was used, it was regarded as unintuitive and uninformative. Figure 4.4 illustrates these categorisations. Fifty-five percent of the users interviewed did not use the Help functionality, whilst the 28% that did use it found that it was not useful in assisting them to complete their tasks or to find functionality. Seventeen percent of the users interviewed commented that the Help functionality of SBO was not installed.

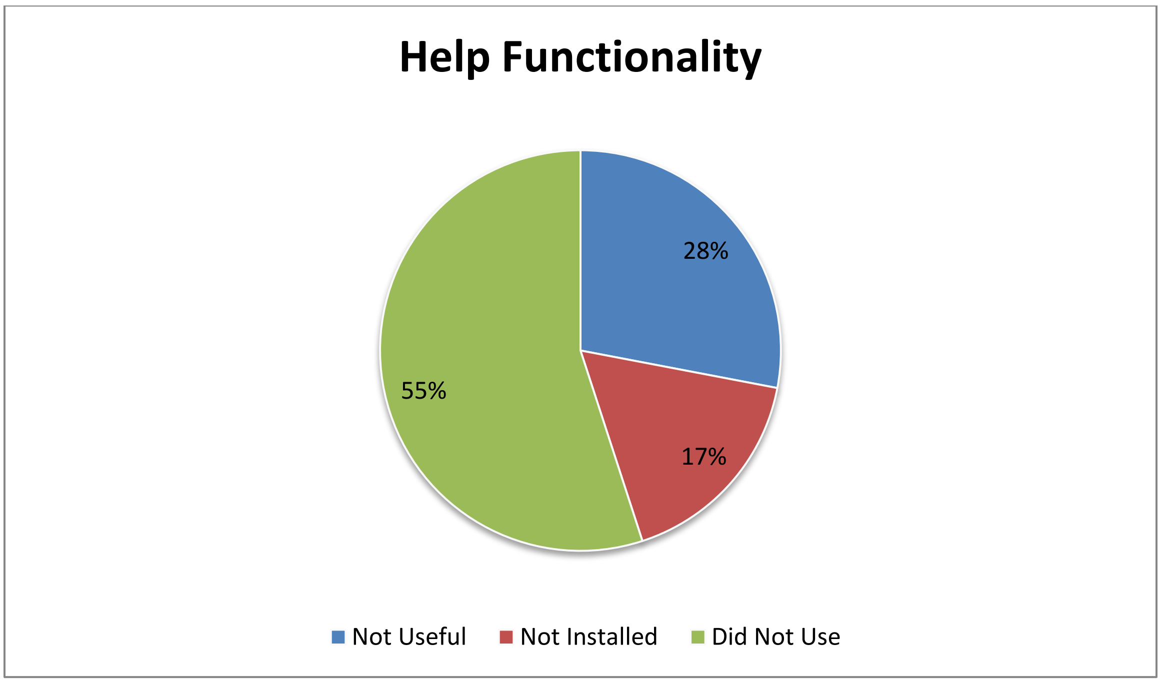


Figure 4.4: Usage of Help Functionality within SBO (n=18)

Several users indicated that solving a problem independently was a challenge. A need for guidance was expressed that could assist in accessing appropriate information and functionality, and could also assist in improving the learning process. Providing guidance could further eliminate the need for costly training and could up-skill the user within a short amount of time. Experienced users of SBO found that newer users still experience the same initial learning challenges with SBO that they had experienced.

4.2.2.6 Customisation

Seventy-eight percent of the users interviewed felt that SBO was not customisable. These users also indicated that SBO was very rigid to use. An example of this was the lack of a wizard-style user interface for report generation. Only 22% of the users were aware of the user-level customisation capabilities of SBO. Figure 4.5 illustrates the awareness level of customisation in SBO amongst the users interviewed.

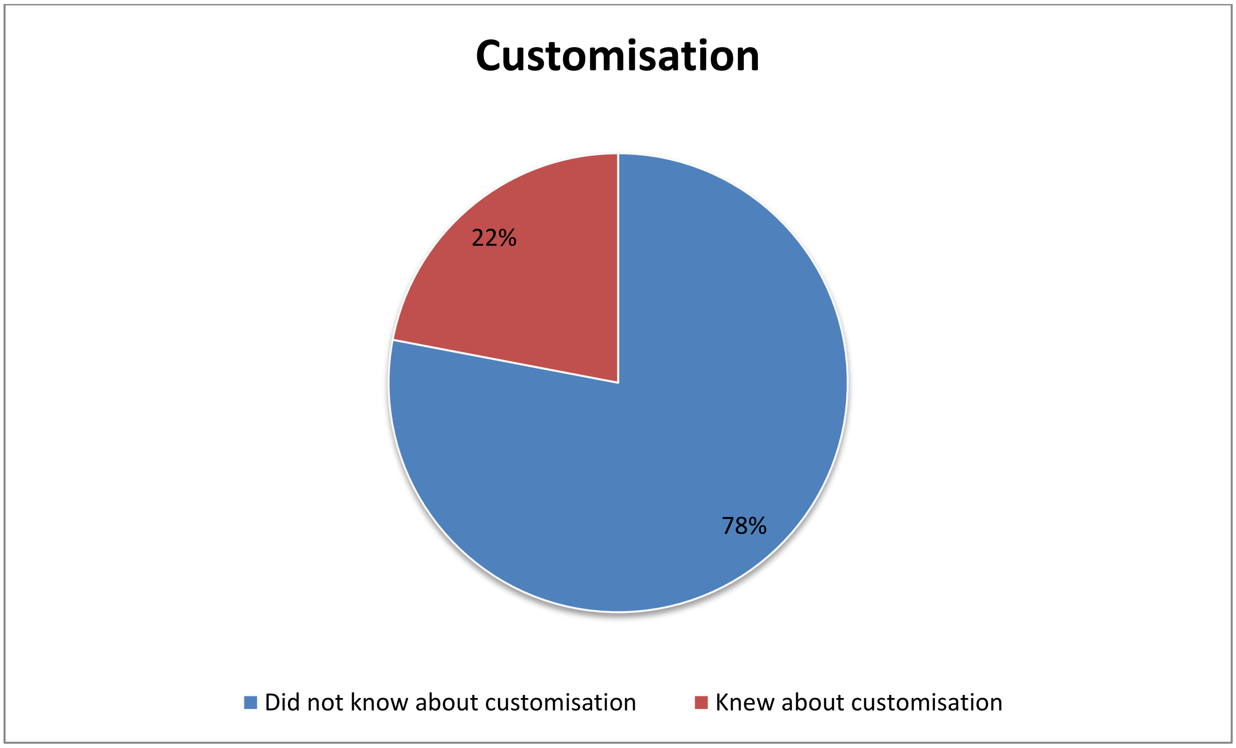


Figure 4.5: Awareness of Customisation Capabilities of SBO (n=18)

More experienced users felt that SBO should be customisable to the extent where routine manual tasks could be automated. User-level customisation of user interfaces and reports that could be pre-configured based on usage by a particular user, were also lacking. Some of the users mentioned that this type of customisation would be "nice", and that SBO should be capable of "thinking for the user".

4.2.3 Discussion

One of the common aspects of SBO that was reported by the small enterprises was that it was simple and easy to use. The users often referred to it as being “very user friendly”. Learnability was the other area, which received the greatest praise amongst the users.

A common dislike was the reporting capabilities of SBO, as it is presently considered cumbersome and inaccurate. Other general dislikes included the fact that SBO was not very intuitive and flexible. Some of the small enterprises felt that it introduced more complexity into the working environment, making existing business processes more labour intensive. These small enterprises further mentioned that not everything was transparent, and that SBO was costly to maintain – to the extent where operating profit was spent on configuring SBO.

Another major dislike was that there was no functionality to support any undoing of user actions. This was frustrating, as it often doubled the amount of work that needed to be done. In order to reverse erroneous entries, users needed to enter contra-transactions.

The interview study also identified that the two most predominant tasks performed by users are *processing a purchase order* and *creating a customer*. These two tasks are used in the next section, in which a heuristic evaluation was conducted on SBO.

4.3 Heuristic Evaluation

An analytical evaluation was performed in the form of an heuristic evaluation. This analytical method was chosen, because it is cost-effective and capable of detecting the most number of usability issues − without the use of a usability laboratory.

The heuristic evaluation was used to complement the interview study, by determining whether the usability problems identified in the interview study could be detected and identified in the heuristic evaluation. The heuristic evaluation was also used as a means of validating the proposed ERP heuristics (Section 3.4.2).

4.3.1 Research Design

The aim of the heuristic evaluation was to identify the potential usability issues with SBO. Two sets of usability heuristics were used in the evaluation process. The first set of heuristics was Nielsen's ten heuristics, while the second set comprised the five ERP heuristics proposed in Chapter 3. A scenario (Appendix F) was given to the experts, as a guide to evaluating the usability of SBO. The two tasks were provided as part of the scenario:

- Process a purchase order; and
- Create a customer.

These two tasks were identified in the interview study as the most frequently performed tasks by users of SBO. Figure 4.6 is the user interface for processing a purchase order in SBO.

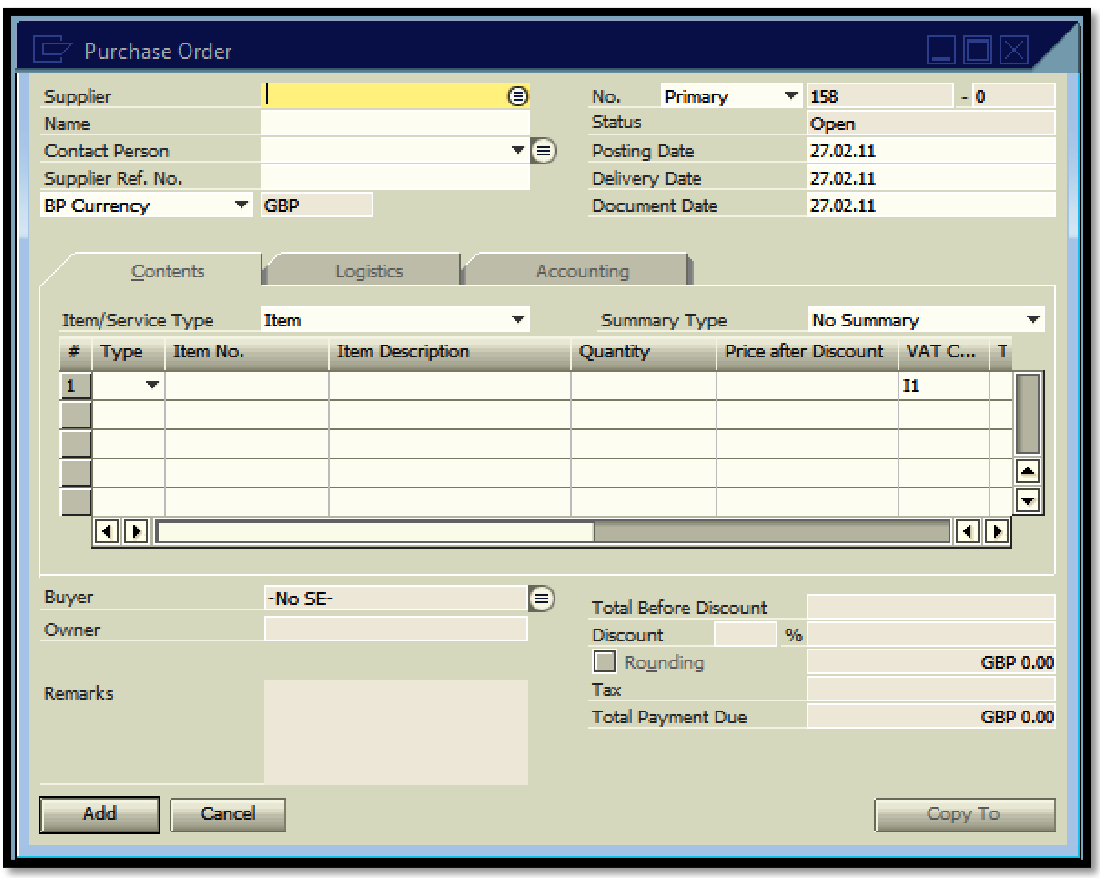


Figure 4.6: User Interface for Processing a Purchase Order in SBO

The process of completing a purchase order typically involves:

- Selecting the appropriate supplier;
- Selecting the appropriate Posting Date, Delivery Date and Document Date;
- Selecting the appropriate item(s);
- Entering the required item quantity;
- Selecting the appropriate buyer;
- Entering the appropriate order discount (if necessary);
- Verifying delivery details;
- Verifying payment details; and
- Processing the purchase.

The second most frequently performed task identified in the interview study was the task of creating a customer. Figure 4.7 illustrates the SBO user interface for creating a customer.

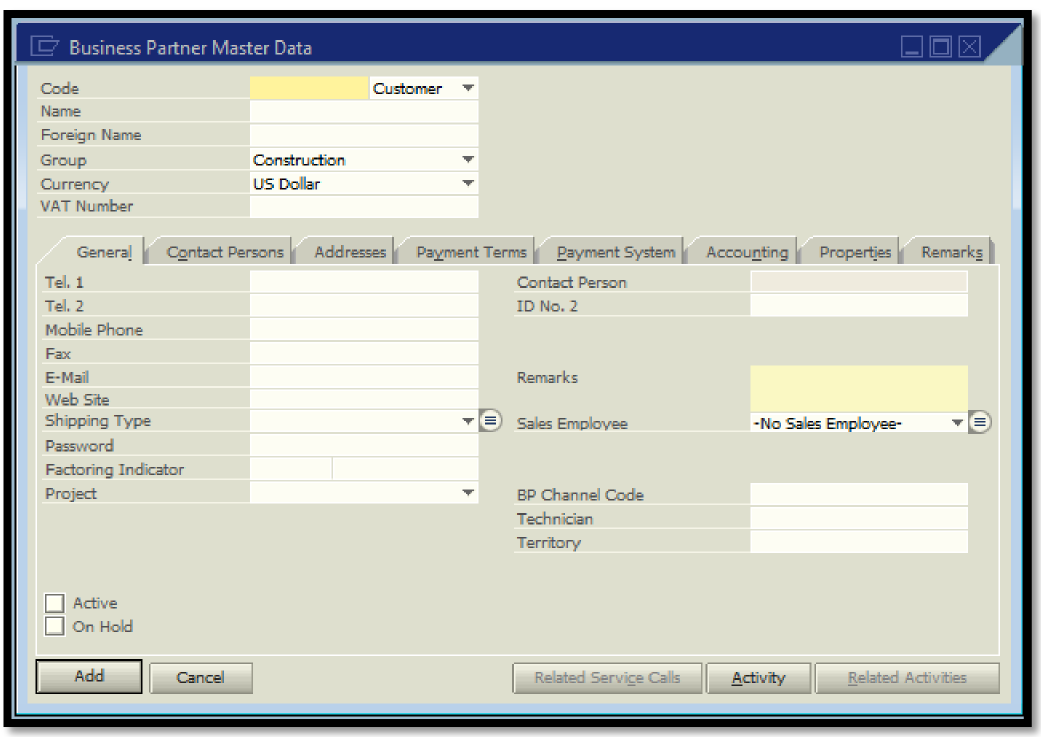


Figure 4.7: User Interface for Creating a Customer (Business Partner) in SBO

The process of creating a customer typically involves:

- Entering an appropriate and unique business partner code;
- Selecting the appropriate business partner type (i.e. customer, supplier or lead);
- Selecting an appropriate currency type for the customer;
- Selecting the preferred sales person;
- Entering a contact person for the customer;
- Entering billing and shipping details;
- Entering payment details; and
- Specifying whether the customer is an active customer or "on hold".

In order to evaluate the usability of the SBO, usability experts (n=3) were provided with two sets of heuristics (Nielsen's ten heuristics and the proposed ERP heuristics). The two sets of heuristics were combined, and then provided to the experts in a combined checklist (Appendix G). The checklist used a five-point severity rating, in order to assess the severity of the usability issues. Experts were asked to determine whether a usability issue was:

0) Not a usability problem;
1) A cosmetic problem that would not affect the usability;
2) A minor usability problem that users can work around;
3) A major usability problem that users cannot work around; or
4) A catastrophic usability problem that is not allowing the users to perform their daily activities.

Space for additional comments, in order to justify the experts' selection was provided. The results obtained from the analytical evaluation comprised quantitative and qualitative results.

4.3.1.1 Participant Selection

Three usability experts were asked to rate the usability of SBO against two sets of heuristics. The three usability experts are research associates at the SAP Meraka Unit for Technology Development (UTD) in South Africa. One of the experts is the author of this thesis. These particular experts were selected, because of their usability experience, which ranged from three to six years. Their experience in working with ERP systems (ranging from three to four years) also contributed to their selection.

4.3.2 Results

The data collected from the heuristic evaluation were quantitative and qualitative in nature. The quantitative data referred to a numerical rating of a specific usability issue and the qualitative data was provided by the usability experts as subjective responses. The numerical data were statistically analysed, in order to determine the median of the ratings from the usability experts. Based on these results, inferences were made as to the severity of the usability issue. An analysis of the qualitative data involved a thematic analysis of the subjective responses provided by the usability experts.

The remainder of this section discusses the results of the heuristic evaluation in terms of the inferences made on the quantitative data and the themes which emerged from the qualitative analysis.

4.3.2.1 Quantitative Results

The results for the quantitative analysis were split into results which originated from data pertaining to the use of the general usability heuristics, and those results which originated from the ERP heuristics. These heuristics were combined into a single heuristic checklist (Appendix G), but the discussion of these results is done separately, in order to determine whether the proposed ERP heuristics identified significantly different usability issues from the general usability heuristics.

General Usability Results

The analysis of the general usability inspection data was done by collecting the responses from the three experts, and then allocating these responses to the appropriate heuristic. The mode of these individual responses was then calculated per heuristic for each usability expert. This was necessary, in order to determine the most frequently occurring rating per usability issue for each expert. The median of the mode values was then calculated, in order to obtain the best central value that could represent the severity of the usability issue.

Potential usability issues with SBO were identified by using Nielsen's ten heuristics. Figure 4.8 illustrates the general usability heuristics and their median severity ratings.

Further analysis was conducted on the results, to gain a better understanding as to what the specific usability issues were for each heuristic. The analysis was done by taking the response for each question, per heuristic, for all three experts and determining the median value of those responses. Only those heuristics with median values that were equal to or greater than two were considered as a potential usability issue. Those median values that were less than two were not considered – because these issues were regarded as either cosmetic or not applicable (according to the five-point scale).

Potential usability issues relating to the visibility of the system status and the match between SBO and the real world were identified as cosmetic usability issues (median = 2). The ability of SBO to help users recognise, diagnose and recover from errors, as well as the flexibility and efficiency of using SBO, were identified as minor usability issues (median = 3).

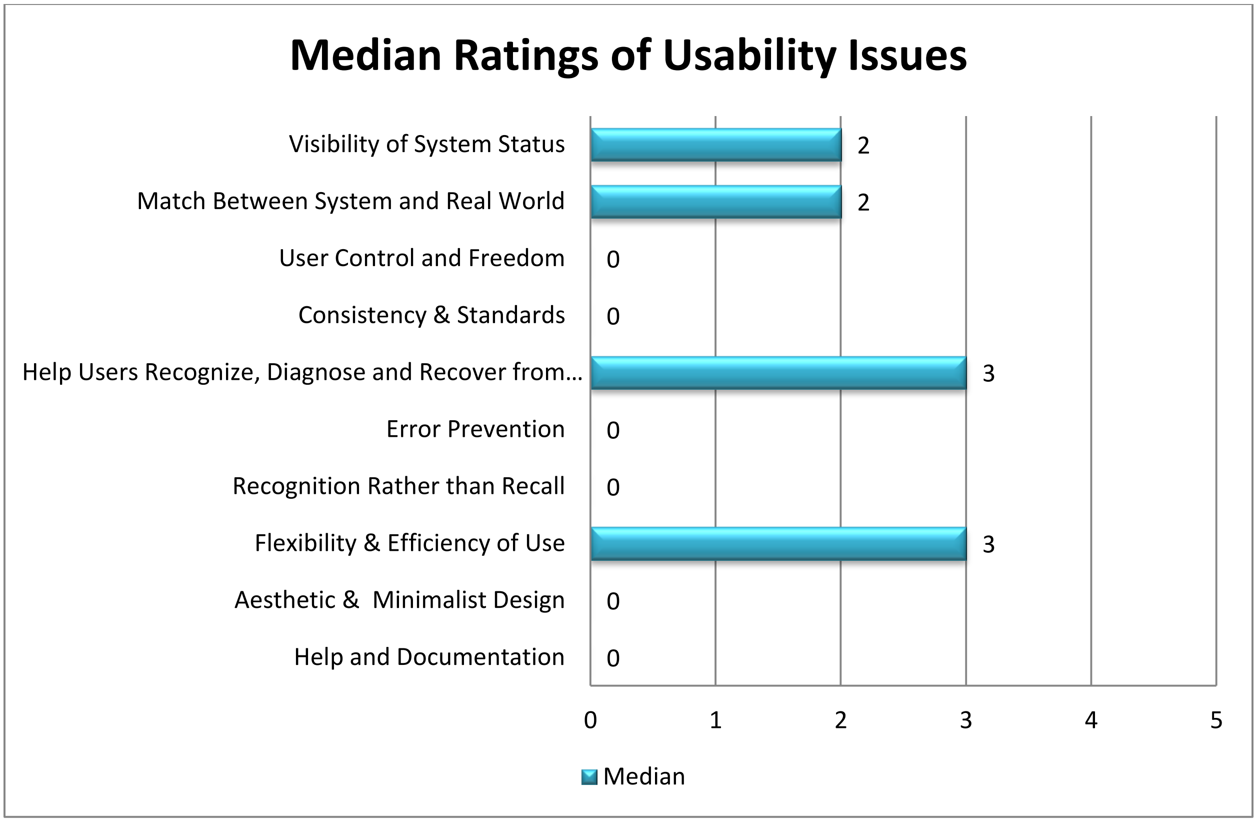


Figure 4.8: Potential Usability Issues identified using Nielsen's Ten Heuristics (n=3)

Table 4.1 tabulates the potential usability issues for the *Visibility of system status* heuristic and lists the median severity ratings of the potential usability issue. The aim of this heuristic was to determine whether SBO always kept the user informed about what was going on, through appropriate feedback in a reasonable amount of time.

The potential usability issues in Table 4.1 indicate that SBO provides insufficient feedback and does not keep the user informed about what is going on in an appropriate and timely manner. The identification of these potential usability issues shows that SBO does not provide enough visual indication in terms of selection, navigation and feedback.

Potential Usability Issue	*Median Severity Rating*
Pop-up windows do not allow users to see in which field an error has occurred	2
There is some form of feedback from SBO for every user action	2
There is visual feedback in menus or dialog boxes about which choice the cursor is on now	2
The current status of an icon is clearly indicated	2
SBO provides visibility: that is, by looking, the user can tell the state of the system and the alternatives for action	2
GUI menus make obvious which item has been selected	2
GUI menus make obvious whether de-selection is possible	2
SBO makes use of context labels, menus, maps and place makers as navigational aids if the user has to navigate between multiple screens	2
There is visual feedback in menus or dialog boxes about which choices are selectable	3
There is visual feedback when objects are selected or moved	3

Table 4.1: Visibility of System Status and Potential Usability Issues (n=3)

Potential usability issues identified for the *Match between system and real world* heuristic are tabulated in Table 4.2. The purpose of the Match between system and real world heuristic was to determine whether SBO used the user's language, with words, phrases and concepts familiar to the user, rather than terms that were specific to SBO. The heuristic also measured whether SBO followed real-world conventions in terms of making information appear in a natural and logical order.

Potential Usability Issue	*Median Severity Rating*
Related and interdependent fields appear on the same screen	2
Tasks are described in terminology that is familiar to users on data entry screens	2
Field-level prompts are provided for data entry screens	2
Menu choices fit logically into categories that have readily understood meanings	2
Input data codes are meaningful	2

Table 4.2: Match between System and Real World and Potential Usability Issues (n=3)

The potential usability issues in Table 4.2 are all cosmetic usability issues. These potential usability issues show that SBO does not provide an accurate alignment between itself and the real world – in terms of speaking the user's language and making information appear in a natural and logical order.

Table 4.3 tabulates the potential usability issues that were identified with respect to the *Helping users recognise, diagnose and recover from errors* heuristic. The purpose of this heuristic was to evaluate whether error messages are expressed in plain language, in order to precisely indicate the problem and to constructively suggest a solution.

Potential Usability Issue	*Median Severity Rating*
Prompts are stated constructively without overt or implied criticism of the user	2
Error messages suggest the cause of the problem	2
Error messages inform the user of the error's severity	2
Prompts imply that the user is in control	3
Prompts are brief and unambiguous	3
Error messages are worded so that the system, not the user, takes the blame	3
Error messages are grammatically correct	3
Error messages place users in control of the system	3
Multiple levels of error messages are available with respect to novice and expert users	3
Errors detected in a data entry field are highlighted by the system	4
Error messages provide appropriate semantic information	4
Error messages indicate what action the user needs to take to correct the error	4

Table 4.3: Help Users Recognise, Diagnose and Recover from Errors and Potential Usability Issues (n=3)

Twenty-five percent of the potential usability issues identified for this heuristic were cosmetic usability issues (median severity rating = 2). Fifty percent were minor usability issues (median severity rating = 3), while the remaining twenty-five percent were major usability issues (median severity rating = 4).

The number of potential usability issues with regard to helping users recognise, diagnose and recover from errors indicates that SBO does not express error messages in plain language – in order to clearly articulate the problem and to constructively suggest a possible solution. The minor potential usability issues identified show that the cause and severity of error messages are not adequately conveyed to the user. Major usability problems identified suggest that error messages are not worded appropriately, but in such a way that they are ambiguous and sometimes fail to place the user in control of the situation.

Flexibility and efficiency of use was the last heuristic which was identified as a potentially minor usability issue. This heuristic aimed to evaluate whether SBO offered functionality that could demonstrate its ability to cater for both the novice and the expert user. This heuristic also determined whether SBO allowed users to tailor frequently performed actions, and was capable of providing the advanced user with an alternative means of accessing and operating the system. The potential usability issues identified for Flexibility and efficiency of use are tabulated in Table 4.4.

Potential Usability Issue	*Median Severity Rating*
Menu items have mnemonic codes which support the system's type-ahead strategy	2
SBO offers "find next" and "find previous" shortcuts for searches	3
Users can save a partially filled screen for data entry screens with many fields	3
SBO provides function keys for high-frequency commands	3
Expert users have the option for entering multiple commands in a single string	3
SBO allows novice users to enter the simplest, most common form of each command, and expert users to add parameters	3
Users can define their own synonyms for commands	3

Table 4.4: Flexibility and Efficiency of Use and Potential Usability Issues (n=3)

The potential usability issues in Table 4.4 show that SBO is not flexible enough to support different types of users. SBO does not support the provision of additional functionality (mnemonic codes, alternative commands, saving partially completed screens), which would enable different levels of users to interact and work more efficiently.

ERP Usability Results

The analysis of the ERP usability inspection data was done by collecting the responses from the three experts and allocating them to the appropriate usability issue. The mode of these individual responses was then calculated per usability issue for each usability expert. This was necessary, in order to determine the most frequent rating per usability issue for each expert. The median of the mode values was then calculated to obtain the best central value that could represent the severity of the usability issue in terms of it being a potential usability issue.

The potential usability issues of SBO were identified with regard to the *Navigation and access to information and functionality, Presentation of screen and output* (reports), *Appropriateness of task support, Intuitive nature of SBO and the Ability to customise SBO*. Figure 4.9 illustrates the ERP usability heuristics and their severity ratings.

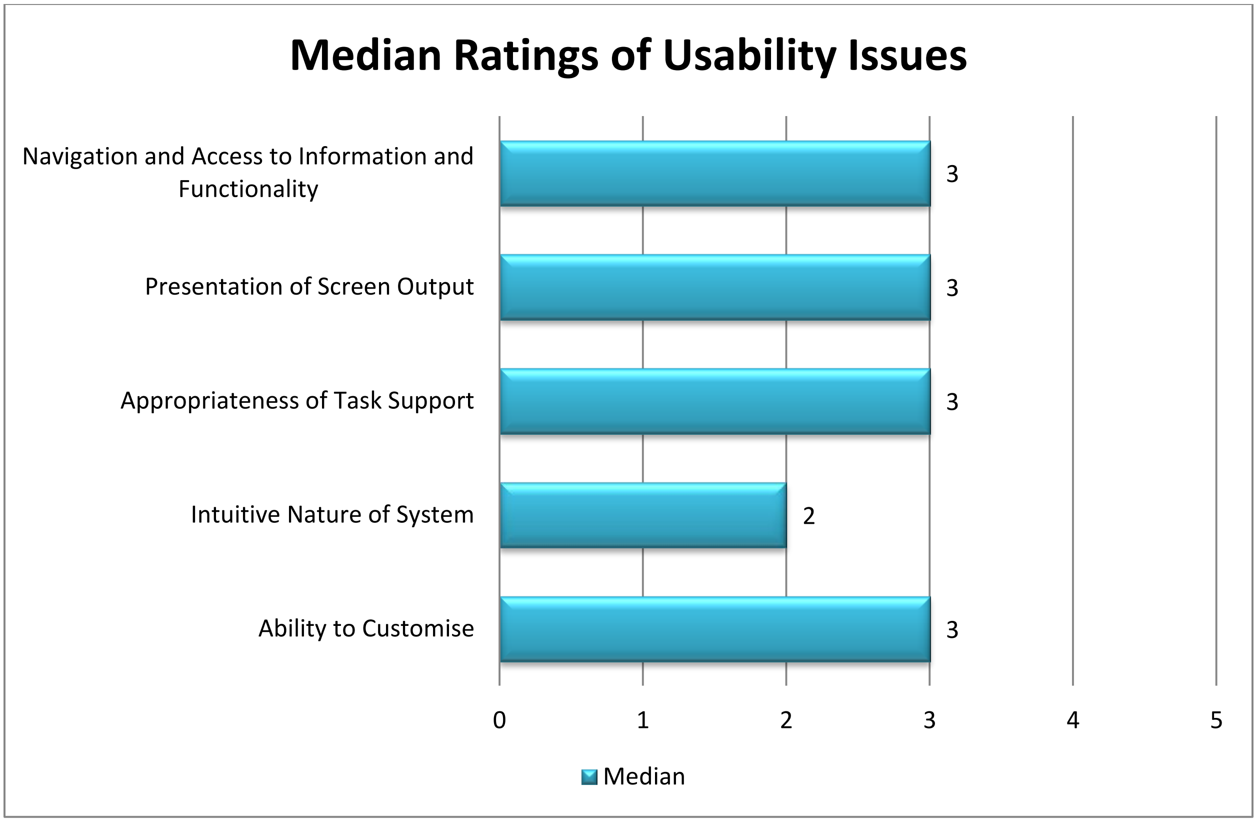


Figure 4.9: Potential Usability Issues identified using ERP heuristics (n=3)

All of the heuristics evaluated were identified as either cosmetic or minor potential usability issues. The intuitive nature of SBO was regarded as a cosmetic usability issue (median severity rating = 2). Navigation and access to information, Presentation of screen and output, Appropriateness of task support, and Ability to customise were identified as minor usability issues (median severity rating = 3).

These heuristics were further analysed to gain insight as to what the specific usability issues were for each heuristic. The analysis was done by taking the response for each question per heuristic for all three experts and determining the median value of those responses.

Only those median values that were equal to or greater than three were considered as potential usability issues. Those values that were less than two were not considered, because those issues were regarded as being either cosmetic or not applicable.

Table 4.5 tabulates the identified potential usability issues in terms of the *Navigation and access to information and functionality* heuristic. This heuristic aimed to determine the ability to identify and access appropriate information, system functionality, menus, reports, options and elements – accurately and effectively.

Potential Usability Issue	*Median Severity Rating*
Information is easy to find	3
Sufficient help is provided for finding the correct functionality, information and screens	3
SBO can guide the user through the correct sequence of transactions to complete a business process	3
Evidence of the next sequence of transactions or steps are provided by SBO	3
There is a search functionality	3
The search functionality supports finding information	3
The information found from the search functionality matches the information required	3
Navigation suits different interaction styles of different users	3

Table 4.5: Navigation and Access to Information and Functionality and Potential Usability Issues (n=3)

The potential usability issues identified, revealed the need for better navigation and access to appropriate information, system functionality, menus, reports, options and elements accurately and efficiently. The identified potential usability issues with regard to the navigation and access to information and functionality heuristic can be split into three categories: navigation, guidance and the finding of information and functionality.

Potential usability issues with regard to navigation show that there are only a limited number of ways in which to navigate in SBO. This is further limited with regard to the different navigation styles of individual users. Guidance was also regarded by the experts as a minor usability issue. Currently, SBO does not provide guidance in terms of guiding users through the correct sequence of transactions to complete a business process.

Guidance with completing transactions is also lacking. SBO does not indicate the current and the next sequence of steps to successfully complete a transaction. The ability to find information and functionality in SBO is a minor usability problem. Information is currently difficult to find and it is not adequately supported by the available search functionality. This is also true when searching for the appropriate functionality.

The potential usability issues with regard to the *Presentation of screen and output* are tabulated in Table 4.6. This heuristic attempted to measure the appropriateness of the layout of menus, dialog boxes, controls and the information on the screen – for the purposes of data entry and output (reports) generation.

Potential Usability Issue	*Median Severity Rating*
The visual layout is well designed	3
The output (reports) is easy to understand and interpret	3
The output (reports) is current and comprehensive	3
Information from the output (reports) presented supports informed decision making	3
Output (reports) provided by the system provides clear visibility into the various other business units and departments	3

Table 4.6: Presentation of Screen Output and Potential Usability Issues (n=3)

The identified potential usability issues mostly related to the output provided by SBO in the form of reports. The reports provided by SBO were regarded as a minor usability issue as they were not easy to interpret in order to support informed decision making. Reports were also not transparent in terms of providing clear visibility to the other business units and departments.

The visual layout of SBO was not well designed and posed a minor usability issue, as it was not intuitive and also required data to be exported in order to be manipulated in other software systems. Further evidence to support the poor design of the visual layout is presented is Table 4.5 which highlights the difficulties in terms of finding information and functionality.

The potential usability issues identified with regard to the *Appropriateness of task support* heuristic are presented in Table 4.7. The aim of this heuristic was to measure the accuracy of the alignment of the users tasks between SBO and the real world to ensure effective task support and efficient task completion.

Potential Usability Issue	*Median Severity Rating*
The terminology used in SBO is consistent with the terminology of the user	3
SBO enables efficient completion of user tasks	3
SBO improves user productivity	3
SBO automates routine and redundant tasks and data	3

Table 4.7: Appropriateness of Task Support and Potential Usability Issues (n=3)

The potential usability issues in Table 4.7 show that SBO does not accurately support the alignment of users' tasks and the real world, thereby not ensuring effective task support and efficient task completion. This is evident in the minor usability issues which show that SBO does not enable the efficient completion of tasks. This negatively impacts on the user's productivity. The terminology used in SBO is not aligned with the terminology used by users and impacts on the efficient use of SBO.

The usability issues with regard to the *Intuitive nature of SBO* are tabulated in Table 4.8. This heuristic evaluated the ease of learning of SBO.

Potential Usability Issue	*Median Severity Rating*
On-line help is sufficient to support the learning process	3
It is easy to become skilful at using SBO within a short amount of time	3

Table 4.8: Intuitive Nature of SBO and Potential Usability Issues (n=3)

The usability issues in Table 4.8 show that the ability to learn and become proficient with SBO is a long process and is currently not supported by the online help.

The usability issues with regard to the customisation abilities of SBO are tabulated in Table 4.9. This heuristic aimed to evaluate the ease in which SBO could be customised to ensure accurate alignment between itself, the business processes of the small enterprise and the user. The usability issues identified are tabulated in Table 4.9. These usability issues relate to two types of customisations: customisation for individual users and customisation with regard to the small enterprise.

The potential usability issues in Table 4.9 show that SBO provides limited customisation abilities for the user and the small enterprise. Customisation in terms of the user is limited, as it does not support the different ways in which multiple users could complete a specific task. This represents a minor usability issue, as it could hamper the effectiveness, efficiency of completing the task, as well as the overall satisfaction achieved after completing the task. The lack of customisation in terms of supporting user preferences also supports this argument. The results in Table 4.9 further indicate that SBO is very rigid in terms of customisation to the business processes and transactions of small enterprises. These potential usability issues were identified as mostly being minor usability issues (severity rating of three) and could result in a misalignment between SBO and the small enterprise.

Potential Usability Issue	*Median Severity Rating*
SBO supports customisation in terms of the user preferences	3
SBO can be configured to update existing business processes and (or) to include new business process	3
A particular module of SBO can be customised	3
SBO supports customisation to promote business agility	3
SBO is easy to change and re-configure over a period of time without making the system more complicated	3
The GUI of SBO can be configured without affecting the underlying business logic	3

Table 4.9: Ability to Customise and Potential Usability Issues (n=3)

Comparison of Results

The use of the general heuristics revealed the following usability issues:

- SBO does not provide sufficient visual indication in terms of selection, navigation and feedback;
- SBO does not express error messages in plain language, in order to clearly describe the problem and constructively suggest a possible resolution;
- SBO was not flexible enough to support different levels of users; and
- SBO does not support the provision of additional functionality, which would enable different levels of users to work more efficiently.

Use of the ERP heuristics revealed the following potential usability issues:

- SBO does not provide an effective and efficient means of searching for information and functionality;
- SBO does not provide guidance, in order to guide the user through the correct sequence of steps, in order to complete a transaction or business process;
- SBO does not provide an effective reporting function that is capable of producing reports that are intuitive and could support informed decision-making;
- SBO does not have a visual layout that is intuitive in terms of easily finding information and functionality;
- SBO does not accurately support the alignment between the system and the real world;
- SBO does not support customisation on a multiple user level to the extent where different users can configure navigation styles, layouts and interaction styles; and
- SBO does not support customisation to the level of a particular business process or transaction for a specific small enterprise.

The general usability heuristics identified the following areas with usability issues: *Visibility of system status, match between SBO and the real world, ability of SBO to help users recognise, diagnose and recover from errors, as well as flexibility and efficiency of use*. An analysis of the ERP heuristics identified the following usability issues: *navigation and access to information and functionality, presentation of screen and output, appropriateness of task support, intuitive nature of SBO and the ability to customise SBO*.

Only four of the ten general heuristics identified usability issues, as compared with the ERP heuristics whereby all of the heuristics identified usability issues. This shows that when used individually, these heuristics are capable of identifying significantly different usability issues with regard to an ERP system. The combination of these two sets of heuristics could, therefore, contribute towards identifying general usability issues and ERP specific usability issues.

Usability issues that were identified in both sets of heuristics showed insufficient alignment between SBO, the small enterprise and the user. Another common usability issue that was identified was the fact that SBO lacks flexibility in terms of supporting different levels and types of users.

4.3.2.2 Qualitative Results

Qualitative feedback from the heuristic evaluation was analysed thematically using thematic analysis. The analysis revealed that the feedback could be grouped into the following categories: general comments, terminology, visual indication, errors and functionality.

General Comments

The usability experts felt that SBO was quick and responsive. A key concern expressed was that guidance was lacking – especially for novice users. One of the experts suggested that more colour should be added, in order to effectively aid the user. Help, in terms of providing guidance, was viewed as being difficult to use. The results obtained from the Help functionality were regarded as being inaccurate.

Terminology

The terminology used in SBO was not obvious and was often confusing to the experts, as it did not relate to the given scenario. Menus were only regarded as easy to use, once the terminology was understood.

Visual Indication

SBO does not provide a clear visual indication of when the data was saved or added (e.g. adding a customer). Mandatory and optional input fields are not clearly marked. An “orange” arrow constantly appeared next to several fields, but its function was not obvious. The user interface contains many fields, with little or no groupings to indicate relevancy or a relationship between fields. There was also no visual indication of any mechanism that could assist with guidance.

Errors

Error messages appeared at the bottom of the screen. This was regarded by the experts as unconventional. The description of the error messages did not clearly explain what the problem was, or how it could be solved.

Functionality

The experts found that the functionality of SBO could be found relatively easily, once the user had become familiar with SBO. It was noted that SBO lacked a global search function to assist with guidance. SBO does not support the reversal (“undo”) of actions; and this was regarded as a major usability issue. The experts frequently noted that there was a lack of a “next button”, which would indicate the next action to be performed.

4.4 Discussion

The usability evaluation supported the identification of 25 usability issues. These usability issues were classified according to the usability attributes proposed for ERP systems (Section 3.4.2). The classification revealed the following groupings:

- Task Support had four usability issues;
- Navigation had eight usability issues;
- Presentation had five usability issues;
- Learnability had two usability issues; and
- Customisation had six usability issues.

The usability issues were further analysed to determine which issues were a result of the user interface being complex, as opposed to those usability issues that were a direct result of the functionality of the ERP system (which could not be configured). Only 11 of the 25 usability issues related to the complexity of the user interface. These 11 issues are tabulated in Table 4.10.

No.	*Usability Issue*	*Category*
1.	Too many steps required to complete a task	Task Support
2.	SBO does not enable efficient completion of tasks	Task Support
3.	SBO does not improve user productivity	Task Support
4.	SBO does not automate routine tasks	Task Support
5.	Finding information and functionality is difficult	Navigation
6.	SBO cannot guide the user through the correct sequence of tasks	Navigation
7.	Evidence of the next sequence of steps to complete a task is not provided	Navigation
8.	There are too many unused fields on the screen	Presentation
9.	Layout of the user interface does not contribute to the efficient completion of tasks	Presentation
10.	Steps need to be manually recorded the first time in order to be remembered for future use	Learnability
11.	Personalisation of UIs is not possible	Customisation

Table 4.10: Identified Usability Issues of SBO

The information in Table 4.10 shows that the usability issues for SBO occurred in all five of the proposed usability attributes for ERP systems. These attributes are:

- Navigation;
- Presentation;
- Task Support;
- Learnability; and
- Customisation.

The next section discusses how the identified usability issues and areas impact on the design of SBO. Suggestions are provided on how these usability issues could be resolved.

4.5 Design Implications

The previous section identified five areas that could be improved to improve the overall usability of SBO. These areas are Navigation, Presentation, Task Support, Learnability and Customisation. This section provides an overview of the improvements that could be made, and their implications on the design and usability of SBO.

4.5.1 Navigation

Improving the navigation and guidance capability of SBO could improve the usability and support the process of discovering information and functionality. This process could be provided through the design of system-based guidance that could reduce initial learning times and improve efficiency and productivity.

The design of system-based guidance for SBO needs to include:

- Guidance to support alternative navigation styles;
- Guidance, which supports the effective completion of transaction and business processes – guiding the user through the correct sequence of steps;

- Guidance, which shows users where errors are, what caused them and how they can be resolved; and
- Search functionality, which enables accurate information and functionality to be found effectively and easily.

4.5.2 Presentation

Improving the presentation of the user interface could have a direct impact on the usability of SBO. Several of the users reported that the visual layout and presentation of the user interface is too complex and confusing. This was mostly because of the large amounts of unused fields that were present. The heuristic evaluation also identified that the lack of intuitiveness of the visual layout of the user interface negatively affected the ability to find information and functionality effectively and efficiently. The lack of intuitiveness and the inherent complexity of the user interfaces further resulted in the completion of tasks being negatively affected.

Another area of presentation that could be addressed is the design of the appropriate reports. This, however, would need to be done by the SAP Partners and is out of the scope of this research.

4.5.3 Task Support

Task support with regard to the design of SBO could be improved by improving the alignment between SBO and the users' tasks. This would impact on usability, by improving the user's ability to complete tasks more efficiently and in turn would result in improved user productivity. Improving task support could also be extended to automate the routine tasks of the user, and therefore reduce the number of steps required to complete a task. These improvements would need to be implemented on a functional level by an SAP Partner.

4.5.4 Learnability

Learnability is a secondary area that could improve if improvements were made to the other four areas. There are no specific design recommendations that can be made to improve learnability that would be significantly different in any way from those made to the other four areas.

4.5.5 Customisation

Customisation with regard to SBO needs to be discussed in terms of the needs of the small enterprise and the user. The customisation of SBO for the small enterprise is typically done by external companies (SAP Partners) who are trained to handle customisation at a business process level. These partners are responsible for customising SBO, so that it aligns with the business processes of a small enterprise to ensure accurate results. This type of customisation is not the focus of this research and will not be discussed in any further detail.

In order to improve the customisation capabilities of SBO, the following design issues need to be considered:

- SBO needs to be customisable to support different levels and types of users;
- SBO needs to support user customisation, to the extent where users can change the way in which transactions are executed and completed, in order to complete a business process; and
- SBO needs to support customisation in terms of user preferences for the visual layout and presentation of the user interface.

Customisation at the user level would allow for the user interface to be adaptable, and this could improve the efficiency of task support.

4.6 Conclusion

This chapter presented the results of an interview study and an heuristic evaluation of SBO. An interview study was conducted with small enterprises in the Manufacturing sector, using SBO. The results from the interview study revealed that small enterprises felt that SBO was not flexible and intuitive. The results further revealed the need for improved: guidance (navigation), better layout of the user interface (presentation), improve task support (task support), better support to improve learnability (learnability) and user-level customisation to improve efficiency (customisation). The interview study further revealed that the two most frequently performed tasks by small enterprises in SBO are processing a purchase order and creating a customer.

A heuristic evaluation was conducted using a set of general heuristics and the ERP heuristics, as proposed in Chapter 3. The results from the heuristic evaluation showed that the proposed ERP heuristics supported the usability issues identified with the general heuristics, but were also capable of identifying significantly different usability issues of SBO. This assisted in validating the ability of the proposed ERP heuristics to identify the usability issues of ERP systems. Two consistent issues were identified by both sets of heuristics in the heuristic evaluation. The first usability issue related to the need for an improved alignment between SBO. This is necessary in order to provide an improved level of task support and could be achieved through more accurate customisation. The second usability issue that was identified related to the need to provide support for different levels of users. This is necessary in order to provide support learnability and guidance.

The results from the heuristic evaluation supported the identification of 25 usability issues, of which only eleven related to the complexity of the user interface. These eleven usability issues were categorised, according to the ERP usability attributes proposed in Chapter 3, namely Task Support, Navigation, Presentation, Learnability and Customisation.

This chapter has presented a discussion on how the identified usability issues of SBO could be addressed to improve the overall design of SBO. This discussion highlighted several recommendations that could be made to improve overall level of Task Support, Navigation, Presentation, Learnability, and Customisation of SBO. These recommendations were proposed in order to address the identified usability issues and potentially improve the usability of SBO.

The next chapter will present a discussion on AUIs and will investigate how AUIs could be applied to address the usability issues of SBO identified in this chapter.

Chapter 5: Adaptive User Interfaces

5.1 Introduction

Adaptive user interfaces (AUIs) aim to improve the exchange of information between a user and a software system. The ability of AUIs to reduce the complexity of a software system and to enhance usability is achieved by means of task simplification, active and intelligent help, error correction and improved user satisfaction (Chapter 1).

The primary objective of this chapter is to determine how AUIs can be applied to address the usability issues identified with enterprise resource planning (ERP) systems in Chapters 3 and 4. These issues related to the Navigation, Presentation, Task Support, Learnability and Customisation of the ERP system. A secondary objective of this chapter is to select an AUI architecture that could be specialised for the domain of ERP systems.

Achieving the above objectives requires an understanding of AUIs, identifying how AUIs can improve usability and determining how AUIs could be used to potentially improve the usability of ERP systems. These sub-objectives are discussed in this chapter.

5.2 Overview

Research on AUIs emerged in the early 1980s and has been an active research area for the past three decades. AUIs were introduced with the intention of providing the individual user with an optimal user interface, for a particular situation, based on the characteristics of that user (Kühme 1993; Gajos *et al.* 2006).

AUIs continue to be an active research area in the field of HCI and have become the preferred solution for developing personalised user interfaces for complex software systems (Tsandilas and Schraefel 2005; Zudilova-Seinstra 2007; Findlater and McGrenere 2008; Letsu-Dake and Ntuen 2009). AUIs support the improvement of usability through the delivery of personalised user interfaces. This is achieved by reducing the complexity of the user interface and making it easy, effective and efficient to learn and use, whilst improving the overall user satisfaction (Dieterich *et al.* 1993; López-Jaquero *et al.* 2005; Álvarez-Cortés *et al*. 2007).

Several definitions of AUIs exist. These definitions suggest that the objective of an AUI is to dynamically and autonomously adjust the user interface of a software system to better suit the needs of each individual user, based on the individual user's characteristics, preferences and behaviour (Dieterich *et al.* 1993; Kühme and Schneider-Hufschmidt 1993; Langley 1999; McGrenere *et al.* 2002; Findlater and McGrenere 2004; Feng *et al.* 2006; Leung *et al.* 2006; te Brake *et al*. 2006; Yen and Acay 2009).

Literature does not clearly distinguish between an adaptive system and an AUI. Benyon *et al.*(1987) stated that an adaptive system is a software system that is capable of altering aspects of its structure, functionality, or interface to accommodate the different needs of individuals or groups of users and their constantly changing needs. Based on this definition and the one above, we can infer that an AUI is a part of an adaptive system. For the purposes of this research, the definition of an AUI, as proposed by Haas and Hettinger (2001) will be used:

> *"An adaptive interface can be defined as a set of displays, controls, a human operator, and an underlying software system that is capable of modifying the portrayal of information, the affordance of control, and the allocation of tasks to be performed, as a function of the state of the operator, the state of the system, and the environment in which both the operator and the system are immersed" (Haas and Hettinger 2001).*

Zou, Lerner, Leung, Morisson and Wringe (2008) proposed an AUI architecture in an attempt to simplify the user interface for the Eclipse development environment (IDE). The AUI was developed as a plug-in and collected statistics on how the individual users interacted with the Eclipse menu system. Adaptations to the menu were dynamically executed by hiding irrelevant and infrequently used menu items. Figures 5.1 and 5.2 illustrate the difference between the original non-adaptive menu and the adapted menu (when the plug-in is active).

Figure 5.2 illustrates the example whereby the user continuously creates new projects by selecting the "New" item on the File menu. The AUI plug-in "dynamically" exposed the "New" item and hid the other, less frequently used items. The plug-in also added the "Expand Menu" item to the adapted menu to provide the user with the necessary functionality to revert back to the original menu - in which no items were hidden.

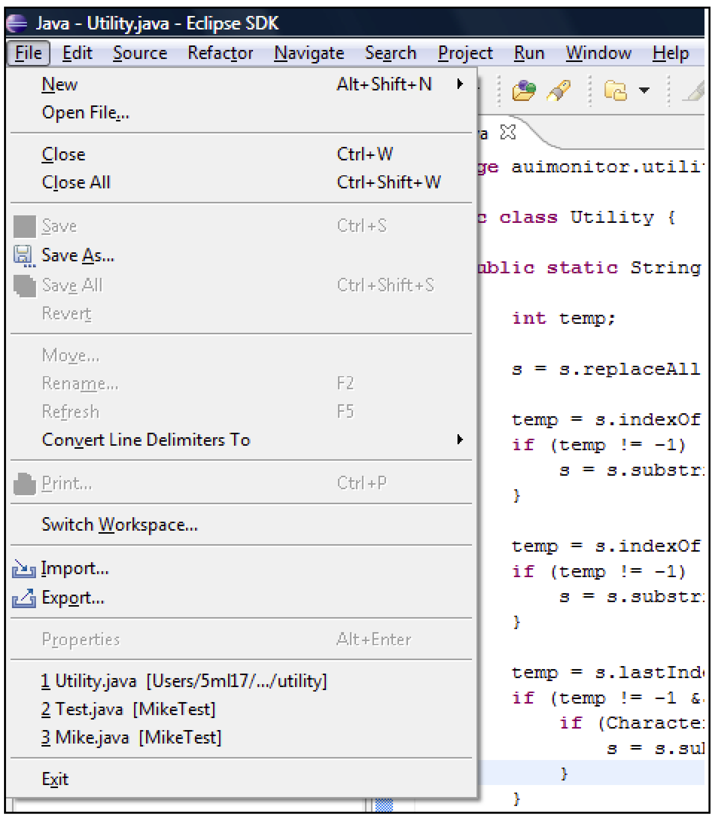


Figure 5.1: Non-Adaptive Menu (Zou *et al.* 2008)

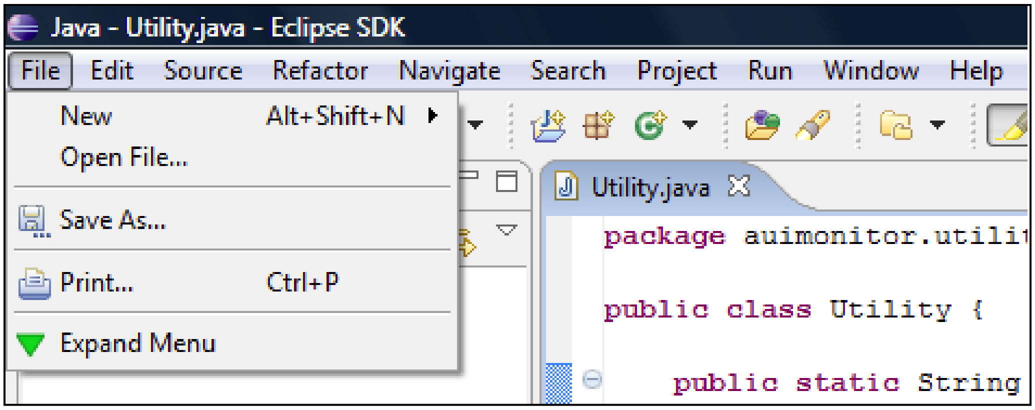


Figure 5.2: Adaptive Menu (Zou *et al.* 2008)

5.3 Types of Adaptation

There are four main types of user interface adaptation, namely content adaptation, presentation adaptation, navigation adaptation and device adaptation (Reichenbacher 2004; Peng and Silver 2007; Zarikas 2007). These different types of adaptations are influenced by the user, the task currently being executed, the device on which the user is executing the task and the current context in which the task is being executed by a specific user on a particular device.

5.3.1 Content Adaptation

Content or information adaptation refers to the process of adapting and delivering personalised information for a specific user or device for a specific task in the current context of use (Reichenbacher 2004; te Brake *et al.* 2006; Zarikas 2007). This is often accomplished through adaptive selection and prioritisation of information, which is based on the data gathered on the user's goals and the user's interaction with the software system over a period of time (Vasilyeva *et al.* 2005; Peng and Silver 2007; Yen and Acay 2009).

5.3.2 Presentation Adaptation

The purpose of presentation or visualisation adaptation is to provide the most appropriate visualisation of the user interface and its contents with regard to a specific user, the current task and the device on which the current task is being executed (Reichenbacher 2004; Vasilyeva *et al.* 2005; Peng and Silver 2007; Zarikas 2007; Ramachandran 2009). Presentation adaptation is performed with knowledge of the characteristics of the user or device.

5.3.3 Navigation Adaptation

Navigation or interface adaptation refers to the process of providing additional guidance to assist the user in accomplishing a specific task (Ramachandran 2009). This is typically implemented by means of managing and adapting (hiding, sorting, annotating, removing or adding) navigational links, based on the knowledge of the user and the specific task that needs to be completed (Reichenbacher 2004; Vasilyeva *et al.* 2005; Ramachandran 2009). An example of navigation adaptation is shown in Figure 5.2, where an adaptive menu was used to display the most frequently used menu items.

5.3.4 Device Adaptation

Device adaptation refers to the process of adapting the user interface to a variety of devices. A typical example of this would be accessing an email client from a desktop computer versus accessing the same email client from a mobile web browser using a cellular phone (Peng and Silver 2007).

Device adaptation often makes use of content adaptation, presentation adaptation and navigation adaptation to deliver a user interface to a particular user, whilst considering the physical attributes and processing capabilities of a particular device.

This particular type of adaptation will not be considered in this research, as this study only focuses on ERP systems that are accessed when using a desktop computer.

5.3.5 Discussion

Chapters 3 and 4 have highlighted the fact that ERP systems are currently difficult to use, due to their lack of appropriate navigation and guidance, user interface presentation, task support, and user-level personalisation. Table 5.1 shows how the different adaptation types, described in this chapter, can be used to address the usability issues identified for SBO (Chapter 4).

No.	Usability Issue	Adaptation Type
1.	Too many steps required to complete a task	Presentation
2.	SBO does not enable efficient completion of tasks	Content
3.	SBO does not improve user productivity	Content and Presentation
4.	SBO does not automate routine tasks	Content
5.	Finding information and functionality is difficult	Navigation
6.	SBO cannot guide the user through the correct sequence of tasks	Navigation
7.	Evidence of the next sequence of steps to complete a task is not provided	Navigation
8.	There are too many unused fields on the screen	Presentation
9.	Layout of the user interface does not contribute to the efficient completion of tasks	Presentation
10.	Steps need to be manually recorded the first time in order to be remembered for future use	Presentation
11.	Personalisation of UIs is not possible	Content and Presentation

Table 5.1: Application of AUIs to ERP systems

Content adaptation can be used to improve the usability by adapting and delivering personalised information for a specific user and task, thereby making tasks easier to complete and improving user productivity. Presentation adaptation could assist by visually aiding users with the completion of tasks to improve user productivity. Presentation adaptation can also assist with usability issues relating to learnability and customisation. This is made possible by adapting the presentation of the user interface to support individual personalisation for users, and to improve learnability by visually aiding the user through the completion of a task.

Navigation adaptation can be applied to usability issues which relate to navigation. Navigation adaptation can assist by adapting the user interface to provide better guidance. Improving the guidance would enable more efficient ways of finding information and functionality, whilst guiding users through the correct sequence of steps needed to complete a task.

5.4 AUI Components

The process of adapting the user interface typically involves the use of several components. These components are represented in the form of declarative models (López-Jaquero *et al.* 2005). An investigation was conducted into the most widely utilised declarative models. The purpose of the investigation was to identify the most frequently used models in the design and implementation of AUIs. These models were: the user model, the task model, the domain model, the dialog model and the presentation model (Puerta 1997; Keeble *et al.* 2000; Ghédira *et al.* 2002; Menkhaus and Pree 2002; Jameson 2003; Liu *et al.* 2003; Gajos and Weld 2004; Reichenbacher 2004; López-Jaquero *et al.* 2005; Vasilyeva *et al.* 2005; Zarikas 2007; van Tonder and Wesson 2008).

These models are used to assist in building the user interface, by providing information that is acquired from the users when completing tasks in a specific context. The context comprises the user characteristics (stored in the user model), the current application domain (stored in the domain model), and the tasks typically performed by the user (stored in the task model) (Tran *et al.* 2009).

Other declarative models that are also used include: the system model, the application model, the adaptation model, the context model, the interaction model, the platform model and the device model (Puerta 1993; Keeble *et al.* 2000; Menkhaus and Pree 2002; Gajos and Weld 2004; Reichenbacher 2004; Feng *et al.* 2006; Zarikas 2007; van Tonder and Wesson 2008).

5.4.1 User Model

A user model is a declarative model containing data based on the user's interaction with a software system. These data represent the static and dynamic characteristics of the user, such as the user's preferences for different tasks and the user's cognitive characteristics when completing a task (Keeble *et al.* 2000; Ghédira *et al.* 2002; Jameson 2003; Ahmed and Ashraf 2007; Zarikas 2007; van Tonder and Wesson 2008).

The process of user modelling involves the construction of the user model and requires the software system to learn and record how the user interacts with the system. This is necessary in order for the system to make inferences (about the user), which will lead to the construction of the user model and the AUI (Puerta 1993; Jameson 2003).

The data stored within the user model are used to support the construction and adaptation of the user interface at run-time. The use of a user model can be explained by referring to Figures 5.1 and 5.2. The usage data of the user's interactions with the *File Menu* were stored in a user model. Upon consulting with the user model, the system was able to determine how to adapt the *File Menu* by determining which menu items were used and should be displayed. A typical example involving the use of a user model could occur when a software system requires information on how to adapt the format of the content for a particular task that would best suit the needs and preferences of a particular user (Ahmed and Ashraf 2007; Tran *et al.* 2009).

The user model is used in conjunction with other models during the adaptation process. Other models that are linked to the user model during the adaptation process include the task model, the domain model and the presentation model (Ghédira *et al.* 2002; Ahmed and Ashraf 2007; Tran *et al.* 2009).

5.4.2 Task Model

A task model may be defined as a hierarchical representation of the tasks and sub-tasks performed by the users of a particular software system, which are completed, in order to achieve a particular set of goals (Paternò 1999; Menkhaus and Pree 2002; Menkhaus and Fischmeister 2003; Reichenbacher 2004; van Tonder and Wesson 2008; Montabert *et al.* 2009; Tran *et al.* 2009; Vidani and Chittaro 2009).

Each task (complex task) and sub-task (atomic task) stored in the task model contains a trigger for the task, the steps required in order to successfully complete the task and the pre- and post-conditions for that task. The task model also contains the semantics between the tasks and sub-tasks, as well as the relationship between the task and the context in which the task is executed (Menkhaus and Pree 2002; Menkhaus and Fischmeister 2003; Jiang *et al.* 2008; van Tonder and Wesson 2008; Gonzalez-Calleros *et al.* 2009).

Constructing the task model is often done in conjunction with the user model. This is necessary in order to understand the functional roles played by users when accomplishing tasks, and to record their individual perception of these tasks (Ahmed and Ashraf 2007; Jiang *et al.* 2008).

The task model can be used to generate and adapt user interfaces for multiple modalities, platforms and devices. Through the use of a task model, the transitional state of a task can be predicted based on the user actions in the current state. Adaptations performed by the software system which result from these predictions include: providing task assistance, navigation adaptation and dynamic presentation of the information and the user interfaces (Ohigashi and Omori 2006; van Tonder and Wesson 2008).

5.4.3 Domain Model

The domain model is responsible for storing knowledge on the application domain (Benyon 1996; Reichenbacher 2004). This knowledge is expressed in terms of the relationship between the objects in a given domain (Puerta *et al.* 1994; Puerta 1997). The task model and the user model are typically derived from the domain model (Ahmed and Ashraf 2007).

Domain models can support the process of adapting user interfaces by means of automating the user interface design process. This is possible through the explicit declaration of domain characteristic objects and functionalities (Puerta *et al.* 1994; Ahmed and Ashraf 2007). The domain model cannot be used separately from other models when assisting with the adaptation and generation of user interfaces. The relationship which exists between the domain model and the task model associates the various tasks and sub-tasks with objects and functionalities from the domain model. This association supports the process of executing tasks or sub-tasks. A relationship must exist between the domain model and the associated task model, because each task or sub-task needs to be connected to objects and functionalities in the domain model, which support the execution of the task or sub-task (Brusilovsky and Cooper 2002; Ahmed and Ashraf 2007).

5.4.4 Dialog Model

A dialog model complements the task model by describing and organising the various task activities that need to be performed on the user interface (Puerta 1997; Van den Bergh and Coninx 2004; Clerckx *et al.* 2005). Dialog models are hierarchically organised in a similar way to that of task models (Menkhaus and Fischmeister 2003). Defining a dialog model involves the specification of when users can invoke the functions, interaction media, select or specify inputs, and when the software system can query and present information by using the various user interface elements (Puerta 1997; Ahmed and Ashraf 2007).

5.4.5 Presentation Model

The presentation model is used to describe the visual appearance of the user interface and to accommodate for the different devices from which a user may access a software system (Menkhaus and Fischmeister 2003; Ahmed and Ashraf 2007).

There are two different types of presentation models: abstract and concrete presentation models. Abstract presentation models typically present a high-level view of the user interface in terms of presentation mapping (between forms and between pages). Concrete or detailed presentation models include information on the spatial layout and platform specific design details (da Silva *et al.* 2005; Ahmed and Ashraf 2007).

Presentation models, similar to the dialog models, are derived from the task model. Due to their similarity, most developers often combine these two models (the presentation and the dialog models). The interaction between the user and the software system is described in the dialog model, based on the user interface elements (controls) defined in the presentation model (Puerta 1997; Nóbrega *et al.* 2005).

Presentation models are involved in the process of user interface adaptation and generation in terms of automatically constructing and adapting the presentation of a user interface to a particular task (Molina and Flores 2008).

5.5 Benefits and Shortcomings of AUIs

Contemporary research into adaptive systems has shown the need for increased flexibility in user interfaces, whilst preserving the usability of the software system (Keeble *et al.* 2000). This is necessary in order for software systems to be able to adapt to a variety of users, platforms and devices, whilst being able to be used effectively (Keeble *et al.* 2000; López-Jaquero *et al.* 2003). AUIs are only considered useful if it can be demonstrated that the inclusion of adaptation in the user interface can result in the generation of user interfaces that are easier and more effective to use (Keeble *et al.* 2000; López-Jaquero *et al.* 2003).

Factors which influence the usability of AUIs include (Strachan *et al.* 2000):

- User Involvement:
    - Adaptations must reflect and address the concerns that users have with an existing non-adaptive system. This can only be done by studying the users' needs and identifying their issues.
- Quality of Adaptations:
    - Adaptations need to appear seamless and must match the look and feel of the original system.
- User Feedback:
    - Conducting user testing to obtain feedback on the adaptive system supports the identification of those usability issues which could then be addressed.

AUIs present several benefits and shortcomings which could impact on the usability of a software system. The impact of AUIs on the usability of software systems is discussed in the next two sections.

5.5.1 Benefits

AUIs have been proposed as a means of presenting user interfaces that are easy, efficient and effective to use, whilst being flexible enough to respond to unplanned system events. The ability of AUIs to meet the challenge of improving the usability (through the use of various types of adaptive techniques) of a software system is widely accepted (Stephanidis *et al.* 1998; Leung *et al.* 2006; Letsu-Dake and Ntuen 2009).

According to Letsu-Dake and Ntuen (2009), AUIs should support the users in achieving usability goals, such as meeting changing user requirements whilst retaining the following characteristics:

- Easy, efficient, and effective to use;
- Make complex systems usable;
- Recognise and intervene in system failures;
- Improve task efficiency and reduce computational resources utilised; and

- Accommodate heterogeneous user groups and profiles.

Determining what makes some AUIs usable and others frustrating to use, to the extent where task completion is hampered, needs to be considered. Gajos et al. (2006) demonstrated that the frequency with which the user interacts with the AUI and the perceived complexity of the task both contribute greatly to the usability of the AUI. In a later study, Gajos et al. (2008) showed that both the predictability and the accuracy of the AUI had an influence on the user's satisfaction levels.

AUIs present several advantages over traditional non-adaptive user interfaces, in terms of (Álvarez-Cortés *et al.* 2007):

- Provision of user interfaces that are personalised to a particular user. This improves usability by understanding the user's behavioural patterns when interacting with the software system and generating user interfaces based on this understanding.
- Automation of routine tasks and activities based on the user's interaction with the system. This improves usability in terms of efficiency by allowing the user to focus on other tasks and activities.
- Reducing information overflow in terms of finding information in large and complex software systems. This improves usability in terms of filtering out irrelevant information and only delivering what the user requires.
- Provision for help with new and complex software systems. This improves usability in terms of detecting and correcting user misconceptions and errors, and by providing information on how to rectify these issues. Providing help based on the user's interaction can also be used as a means to simplify tasks and improve task completion times.

Based on the discussion above, we may conclude that AUIs have the potential to support user-level personalisation and presentation, to improve task support, to improve system navigation and guidance and to improve the overall learnability of a software system. AUIs do, however, have several disadvantages, as will be discussed in the next section.

5.5.2 Shortcomings

Research on AUIs has shown that AUIs can violate some usability principles. The main issue with regard to these violations is that these usability principles were established primarily for direct manipulation user interfaces (Höök 2000; Tsandilas and Schraefel 2004). SBO makes use of typical direct manipulation user interfaces. Therefore, it is noted that any future AUI designs for SBO should not violate the usability principles of direct manipulation user interfaces.

Earlier work on AUIs has shown that AUIs are capable of hindering good HCI design. These shortcomings often meant that AUIs did not meet the expectations of its intended user population. This was mainly due to a lack of thorough usability evaluation. The adaptations contained in these studies were very simple and were based on limited knowledge of the user or on simple user actions, rather than trying to infer the user's actions from complex user models (Höök 2000; Tsandilas and Schraefel 2004; Peng and Silver 2007).

The results from earlier research have shown the need for software systems that can filter information, make suggestions, guide complex tasks or provide other forms of assistance that could reduce the cognitive overhead and workload of the users. These capabilities could be provided through the inclusion of an AUI in the software system.

The main issues with regard to AUIs and usability are the following (Höök 2000; Paymans *et al.* 2004; Tsandilas and Schraefel 2004; Peng and Silver 2007):

- Uncertainty;
- Transparency and Predictability;
- Privacy and Trust; and
- Controllability.

These issues are discussed in more detail in the following sub-sections.

5.5.2.1 Uncertainty

User goals cannot be predicted with absolute certainty and consequently need to be estimated with some sense of probability (Tsandilas and Schraefel 2004).

5.5.2.2 Transparency and Predictability

Adaptive systems are often not transparent and do not provide users with an understanding of the inner workings of the AUI (Höök 2000; Tsandilas and Schraefel 2004).

Predictability is a challenge with adaptive systems in terms of ensuring consistency with regard to the relationship between the inputs provided and the response generated (Höök 2000; Tsandilas and Schraefel 2004).

5.5.2.3 Privacy and Trust

Adaptive systems containing a user model hold some representation of the user. Depending on the anonymity of the user, this information could or could not be shared with a user community leading to privacy issues (Höök 2000).

5.5.2.4 Controllability

Controllability is one of the biggest usability issues with regard to adaptive systems and AUIs; and is also one of the main reasons why successful AUIs are hard to find in practice (Paymans *et al.* 2004). The issue of controllability is a result of the AUI failing to provide the user with a feeling of control. According to Paymans et al. (2004), this could be a result of the user being unable to build an adequate mental model of the system.

The issue of controllability is further discussed by Peng et al. (2007). These authors state that the optimal means of ensuring controllability is for the AUI to provide the user with control over the adaptations. Providing the user with control over the adaptations then leads to another debate as to whether the system is truly adaptive or adaptable, or whether it would support a mixed-mode adaptation. According to Tsandilas and Schraefel (2004), an adaptive system may incorporate adaptable characteristics (allowing the user some level of control).

Another challenge with regard to AUIs, is that there are only a limited number of published empirical usability evaluations (Álvarez-Cortés *et al.* 2007). According to Höök (1998), it is very important to distinguish between the adaptive features of the software system and the general usability of the entire system. This is also the reason why most usability evaluations of AUIs typically involve the adaptive and the non-adaptive user interfaces (Höök 1998; Tsandilas and Schraefel 2005; Álvarez-Cortés *et al.* 2007). Determining what to measure and how to measure usability is also a challenge with regard to AUIs. Several traditional usability evaluations methods have been used in the evaluation of AUIs, such as: interviews, questionnaires and think-aloud protocols, but their applicability to evaluating AUIs is considered limited (Álvarez-Cortés *et al.* 2007).

5.6 AUI Architectures and Frameworks

An investigation of several AUI architectures and frameworks was conducted in order to select the most appropriate AUI architecture or framework that could be specialised to the domain of ERP systems.

Several architectures and frameworks for AUIs exist. However, only those which met certain criteria were considered as candidates for this research. These selection criteria were:

1) The architecture or framework must supports at least two of the required adaptation types;
2) The architecture or framework must support at least a user model; and
3) The architecture or framework must clearly distinguish between the application logic and the interface logic.

Criterion one was stipulated to determine whether an architecture or framework was capable of performing the various types of adaptation required to address the identified usability issues. Furthermore, to support the process of adaptation, literature suggests that a user model must be present to support and guide the adaptation (Section 5.4.1) and hence, the need for criterion two. The last selection criterion was stipulated to identify an architecture or framework that would clearly separate the application logic from the adaptation logic. This is necessary, in order to separate the AUI logic from the business logic of the ERP system. Supporting this functionality, allows for the core functionality of the ERP system to operate independently of the AUI.

5.6.1 Simplified Adaptive System Architecture

The Simplified Adaptive System Architecture (Figure 5.3), as proposed by Ramachandran (2009), was selected because it met all of the stipulated selection criteria. The proposed architecture supports two types of adaptation (presentation and navigation adaptation), it makes use of a user model (user profile) to achieve the adaptation and it keeps the business logic separate from the interface logic. This is achieved through the use of the Business logic interpreter in the Adaptation engine component.

This architecture was originally proposed to demonstrate how adaptive presentation and adaptive navigation could be modelled and applied to the domain of health care. The architecture (Figure 5.3) comprises three main components:

1) The *Database* which is used to store the various rules associated with the adaptation types, as well as a profile of the user.

The *Adaptation engine* which consists of the adaptive navigation engine, the adaptive presentation engine and the business logic interpreter. The business logic interpreter ensures that the adaptation types support the business rules of the system.

The *Adaptive user interface* which consists of various combinations of presentation and navigational elements for different users. The AUI is generated by the adaption engine.

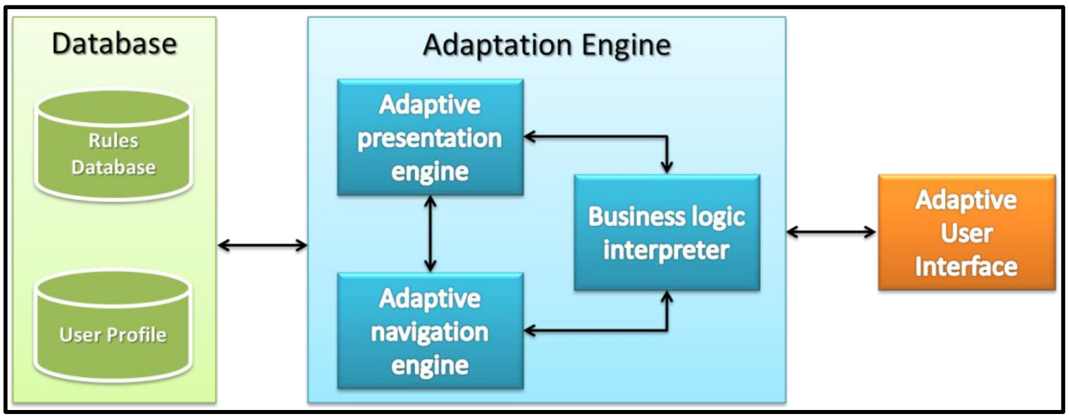


Figure 5.3: Simplified Adaptive System Architecture (Ramachandran 2009)

Figure 5.3 is a very generic architecture which not provide a greater level of granularity in terms of specifying which AUI components are used, apart from the user model (profile), to achieve the various types of adaptation. An advantage of this architecture is that it could be extended to include the ability to support content adaptation. Furthermore, a second level of abstraction could be added to the architecture, to indicate which AUI components are used for each particular adaptation engine, as well as the interaction between these components and how they contribute to the process of generating and updating the AUI.

5.6.2 PD-MBUI Framework

The pattern-driven and model-based user interface (PD-MBUI) (Figure 5.4), as proposed by Ahmed *et al*. (2007), was selected because it satisfied selection criterion two. This particular framework makes use of all of the declarative models (Section 5.4) required to achieve adaptation. A limitation with this framework is that it does not explicitly satisfy selection criteria one and three. However, Ahmed and Ashraf (2007) have stated that adaptation could be achieved by means of adapting a pattern to a particular context for a specific user.

This particular framework makes use of HCI patterns and various declarative models to automatically generate and render user interfaces. Patterns used for the dialog model assist with the grouping of tasks and suggest the sequence between the various dialog views. The presentation patterns assist with the mapping of complex tasks onto interaction elements that are defined in the presentation model. The PD-MBUI framework makes use of a fourth model called the layout model. This model supports the process of generating and rendering user interfaces by specifying a particular layout.

This framework suffers from several shortcomings. Firstly, it does not specify which types of adaptation are supported and therefore, it does not provide evidence of its capability to support adaptation. Secondly, it does not describe how the application logic is separated from the interface logic. This framework would need to be extended and possibly redesigned to account for these shortcomings, and for it to be suitable for the domain of ERP systems.

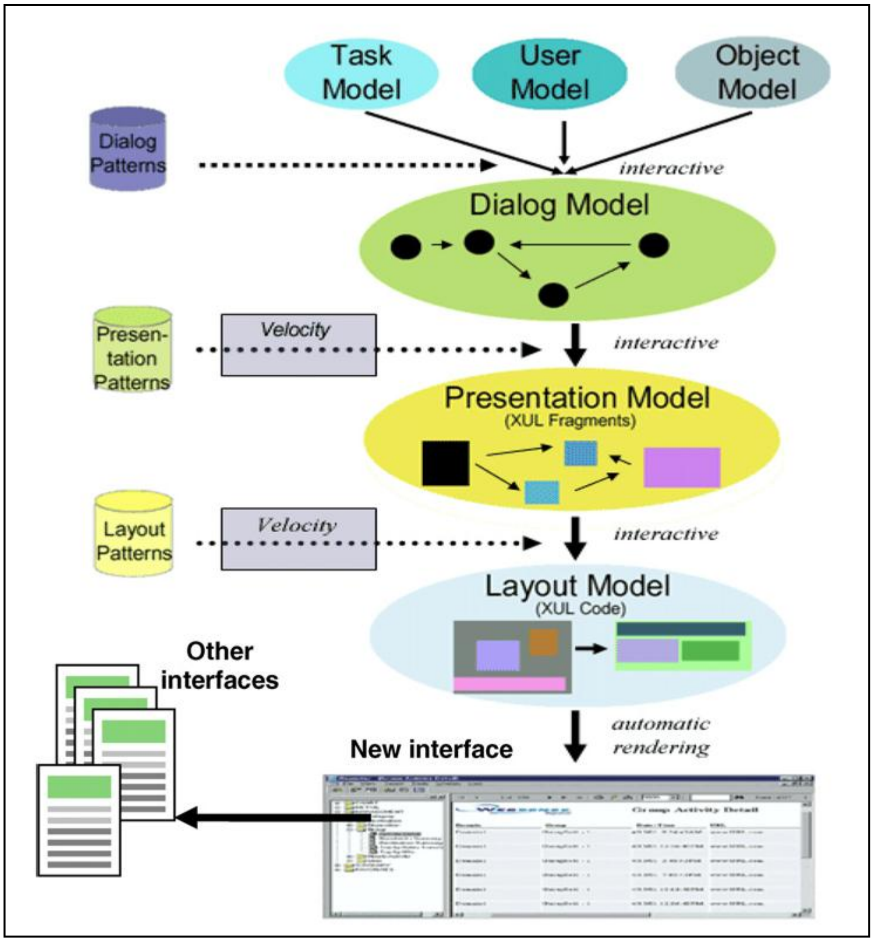


Figure 5.4: PD-BMUI Framework (Ahmed and Ashraf 2007)

5.6.3 Adaptive eHealth Framework

The Adaptive eHealth framework (Vasilyeva *et al.* 2005) (Figure 5.5) was selected because it satisfied two of the three selection criteria. The proposed framework satisfied selection criterion one by supporting all three types of adaptation (content, presentation, and navigation). Selection criterion two was satisfied in terms of the proposed framework: making use of a user model, task model and environment or domain models (in the adaptation engine and data repository) to achieve the adaptation.

This framework was originally proposed by Vasilya et al. (2005) to emphasise the main components of an AUI specific to eHealth. The framework shows the interaction of three components (the Participants, the Adaptation Engine and the Data Repository) in terms of information flows that are necessary to generate AUIs specific to eHealth. The framework makes use of various engines, of which the most important engine contributing to the process of generating AUIs is the Adaptation Engine. The Adaptation Engine consists of a Knowledge base, a model generator and an adaptation effect provider. The model generator is responsible for the generation of user, task and domain models. The adaptation effect provider supports navigation adaptation, presentation adaptation and content adaptation.

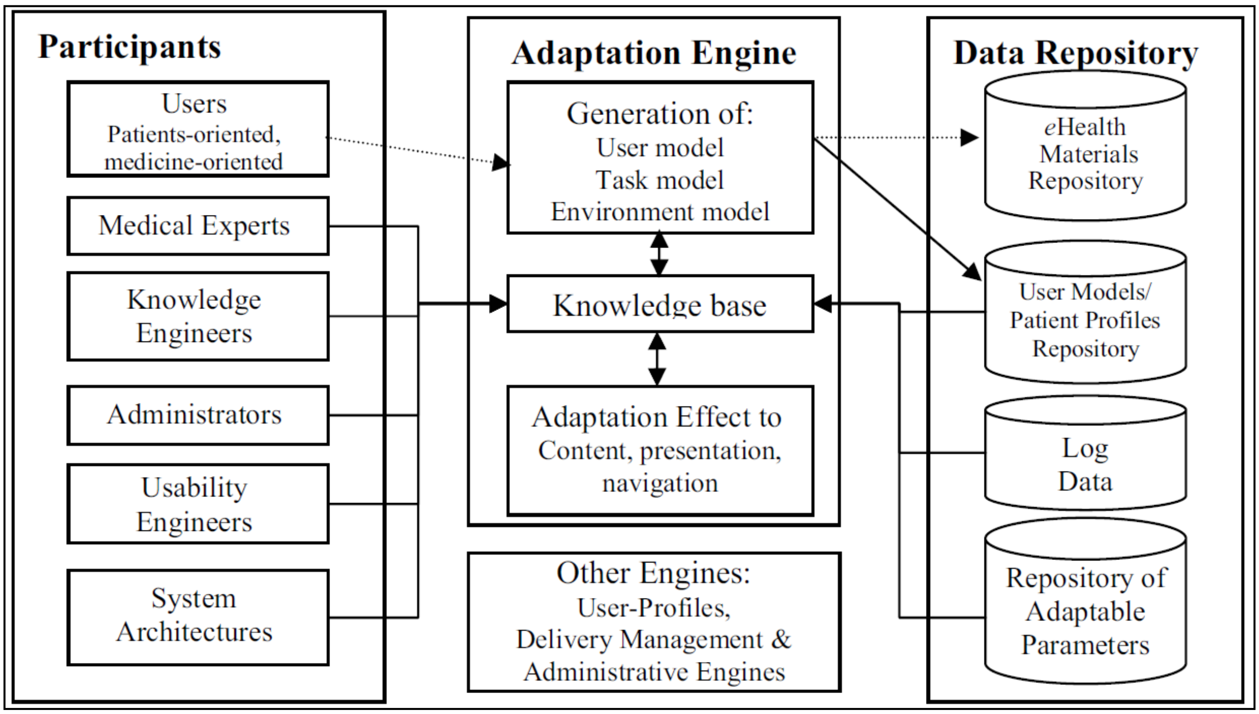


Figure 5.5: General Framework of an Adaptive eHealth System (Vasilyeva *et al.* 2005)

The proposed framework is specialised to the domain of eHealth. This is evident in the Participant component (medical experts), the Adaptation Engine component (Knowledge base), and the Data Repository component (Patient Profiles). This framework would need to be generalised in order for it to be suitable for the domain of ERP systems.

Adapting this framework to the domain of ERP systems could be achieved by replacing the health-specific aspects of the framework with ERP-specific aspects. Secondly, this framework does not distinguish between the application logic and the interface logic and hence, does not satisfy selection criterion three. Therefore, if this framework were to be considered for this research, all of the identified shortcomings would then need to be addressed.

5.6.4 Comparison of AUI Architectures and Frameworks

The AUI architecture and frameworks presented in the previous section were identified as the most suitable candidates to support the design of an AUI for ERP systems. Table 5.2 presents a comparison between the architecture and frameworks against the stipulated selection criteria.

		Simplified Adaptive System Architecture	*PD-MBUI Framework*	*Adaptive eHealth Framework*
Selection Criteria	Supports at least two of the required adaptation types	✓	✗	✓
	Supports at least a user model	✓	✓	✓
	Clearly distinguishes between the application logic and the interface logic	✓	✗	✗

Table 5.2: Comparison of Existing AUI Architectures and Frameworks

Based on the comparison in Table 5.2, only the Simplified Adaptive System Architecture meets all of the stipulated selection criteria. Despite being proposed for the domain of health, this architecture is generic enough to be applied to any other domain. This architecture could easily be adapted to the domain of ERP systems and could incorporate the third adaptation type (content adaptation) that is currently not supported.

Another advantage with the Simplified Adaptive System Architecture is that it does not suffer the complexities of the other frameworks. These complexities include:

- Lack of support for adaptation (PD-MBUI Framework);
- Lack of clear separation between the application logic and the interface logic (PD-MBUI Framework and Adaptive eHealth Framework); and
- Domain specialisation (Adaptive eHealth Framework).

The generic nature of the Simplified Adaptive System Architecture and the ease with which it could be specialised for the domain of ERP systems, makes it the most suitable architecture to be used for this research.

5.7 Conclusion

This chapter presented a discussion on AUIs and has identified the potential of AUIs to address the usability issues of complex software systems.

Three main types of adaptation were identified and discussed in this chapter, namely content adaptation, presentation adaptation and navigation adaptation. This chapter has shown how these three types of adaptation could potentially be used to address the usability issues of SBO.

The most frequently used AUI components necessary to adapt a user interface were identified in this chapter. These components are: the user model, the task model, the domain model, the dialog model, and the presentation model.

This chapter has shown that AUIs are capable of enhancing efficiency and improving overall satisfaction. Several disadvantages were also identified and discussed, including the ability of AUIs to violate some of the usability principles established for direct manipulation user interfaces. SBO makes use of a direct manipulation user interface and any future AUI designs for SBO should support the following standardized usability principles: uncertainty, transparency and predictability, privacy and trust, and controllability.

Several AUI architectures and frameworks exist, which could support the development of an AUI. However, none of these architectures and frameworks are specific to the domain of ERP systems. This chapter has revealed that the most suitable AUI architecture to be specialised to the domain of ERP systems is the Simplified Adaptive System architecture, as proposed by Ramachandran (2009). This particular architecture was selected as it supported at least two adaptation types, a user model and it clearly distinguished between the application logic and the user interface logic.

The next chapter presents a discussion on the design of an AUI for SBO. This design will be based on the application of the different adaptation types to address the usability issues of SBO, and the specialisation of the selected AUI system architecture to the domain of ERP systems.

Chapter 6: Designing an Adaptive User Interface for SAP Business One

6.1 Introduction

Chapter 5 has established that adaptive user interfaces (AUIs) could potentially address the usability issues of enterprise resource planning (ERP) systems. However, applying AUIs to the domain of ERP systems is currently constrained by the lack of guidance in terms of design and implementation.

This chapter aims to address the above limitation by discussing how AUIs could be designed for ERP systems. This objective is achieved by proposing adaptation requirements that could be used to address the identified usability issues of ERP systems and by proposing an AUI design for ERP systems incorporating: an adaptation taxonomy, a system architecture and AUI components. Lastly, the design is validated through the implementation of an AUI for SAP Business One (SBO).

The proposed AUI design incorporating the adaptation taxonomy, system architecture and AUI components were published and presented at the 13th International Conference on Enterprise Information Systems (Singh and Wesson (2011).

6.2 Adaptation Requirements

Adaptation requirements are necessary in order to determine how the various adaptation types can be applied to address the usability issues of SBO (identified in Chapter 4). Further analysis of the identified usability issues revealed that most of these issues could be grouped as either efficiency or learnability usability issues (Table 6.1).

No.	Usability Issue	Adaptation Type	Usability Benefit
1.	Too many steps required to complete a task	Presentation	Efficiency
2.	SBO does not enable efficient completion of tasks	Content	Efficiency
3.	SBO does not improve user productivity	Content and Presentation	Efficiency
4.	SBO does not automate routine tasks	Content	Efficiency
5.	Finding information and functionality is difficult	Navigation	Efficiency
6.	SBO cannot guide the user through the correct sequence of tasks	Navigation	Learnability
7.	Evidence of the next sequence of steps to complete a task is not provided	Navigation	Learnability
8.	There are too many unused fields on the screen	Presentation	Efficiency
9.	Layout of the user interface does not contribute to the efficient completion of tasks	Presentation	Efficiency
10.	Steps need to be manually recorded the first time in order to be remembered for future use	Presentation	Learnability
11.	Personalisation of UIs is not possible	Content and Presentation	Efficiency

Table 6.1: Classification of Usability Issues for SBO

Table 6.1 shows that none of the identified usability issues relate to effectiveness. This is because the issue of effectiveness is typically handled by the ERP consultant who configures the ERP system for a particular enterprise.

The aim of this section is to propose adaptation requirements that could assist in resolving the identified usability issues of SBO. These adaptation requirements would attempt to improve the efficiency and learnability of SBO (and overall satisfaction) through the application of content, presentation and navigation adaptation.

Six major questions need to be answered in order for adaptation to successfully occur. These six questions are (Knutov *et al.* 2009):

1) What can we adapt? (What?)
2) What can we adapt to? (What to?)
3) Why do we need adaptation? (Why?)

4) Where can we apply the adaptation? (Where?)

5) When can we apply the adaptation? (When?)

6) How can we apply the adaptation? (How?)

Answering these six questions forms part of an adaptation process (Figure 6.1) that was originally proposed by Brusilovsky (1996). This process is an iterative process that starts with the intended goals of the user and ends with the system adapting itself to meet these intended user goals. The answers to the first five questions will be discussed in this section, while the answer to the last question will be discussed in the subsequent sections (6.3 and 6.4).

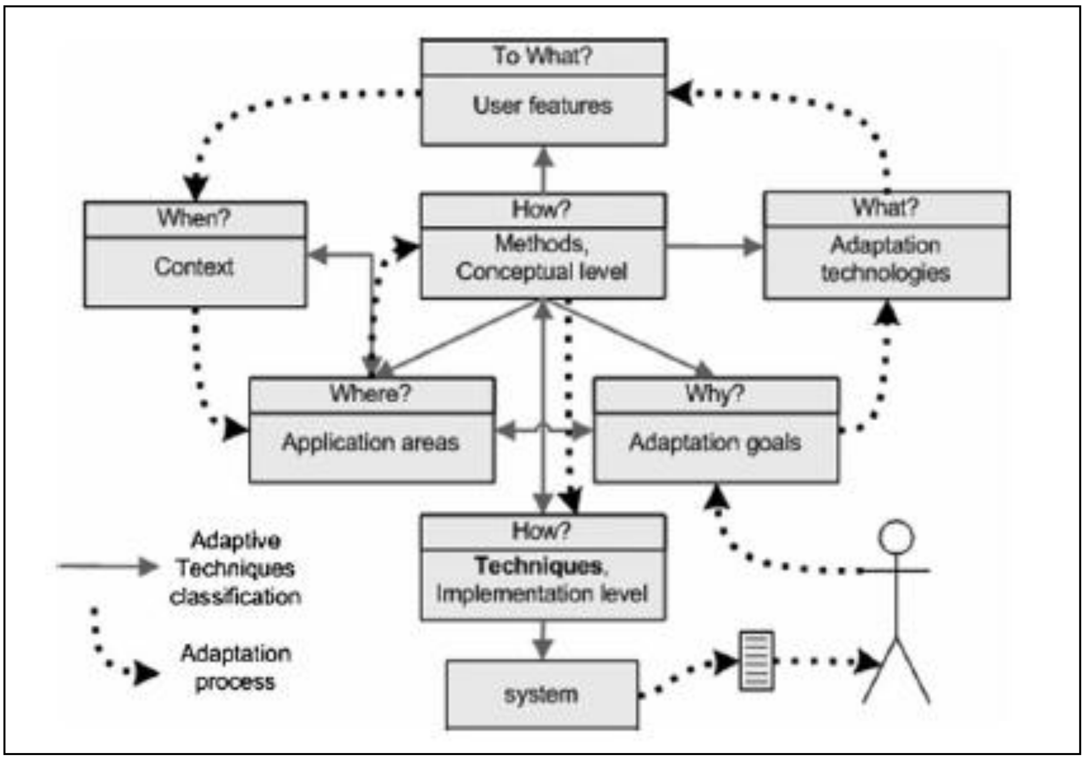


Figure 6.1: The Adaptation process (Brusilovsky 1996; Knutov *et al.* 2009)

6.2.1 Content Adaptation

Question 1: What can we adapt?

The delivery of information to the user can be adapted in terms of automation. This would enable more efficient task completion and improved user productivity (Section 4.5.3).

Question 2: What can we adapt to?

Content adaptation has been found to be most effective using the technique list-based adaptation (Findlater and McGrenere 2004). Using list-based adaptation, the most recently used (MRU) item is moved to the top of the list and is followed by the two most frequently used (MFU) items. These items are determined based on the usage information in the user model. According to the keystroke-level model (Card *et al.* 1983), list-based adaptation could provide the greatest level of efficiency, as it would theoretically result in a reduction in the time it takes to scan through a list and mentally evaluate each option.

Question 3: Why do we need adaptation?

The usability issues identified in Chapter 4 revealed that users spend a lot of time finding the relevant information needed to complete a task, as opposed to merely focusing on completing the task.

Question 4: Where can we apply the adaptation?

Using the list-based adaptation techniques, this type of adaptation can be applied to any user interface element (control) that presents a list to the user for selection purposes.

Question 5: When can we apply adaptation?

This type of content adaptation can be applied at run-time to any control that makes use of lists.

6.2.2 Presentation Adaptation

Question 1: What can we adapt?

Presentation adaptation could be applied to the controls of the forms of SBO – in an attempt to improve efficiency and learnability.

Question 2: What can we adapt to?

Presentation adaptation techniques would need to improve efficiency and learnability by positively impacting on task support. The following is proposed to address the identified usability issues:

- Controls that are infrequently used should be hidden over a period of time; and
- Layouts of user interfaces should be adapted to match the mental model of the user for that specific task.

Although re-arranging the controls would address one of the identified usability issues in SBO (Table 6.1), the keystroke-level analysis revealed that visually re-ordering these controls would not make any significant impact on the efficiency or the learnability of SBO. The results showed that, theoretically, it would take the same amount of time to complete the required form.

Question 3: Why do we need adaptation?

Presentation adaptation is proposed in order to potentially improve the efficiency and learnability of SBO (Chapter 5). Currently, the complex layout of the user interfaces and the large amount of unused controls negatively affects usability by making it inefficient in completing existing tasks and a tedious exercise to learn new tasks.

Question 4: Where can we apply adaptation?

This type of presentation adaptation can be applied to the user interface of any form with which the user interacts.

Question 5: When can we apply adaptation?

The suggestions made for presentation adaptation could be applied at run-time to the controls of any user interface with which the user interacts.

6.2.3 Navigation Adaptation

Question 1: What can we adapt?

Several suggestions are made in Section 4.5.1 on how the guidance and learnability of SBO could be improved. Essentially, the manner in which users are guided through a task needs to be improved. This could have an impact on both the efficiency and the learnability of SBO.

Question 2: What can we adapt to?

Improved methods of task support with regard to guidance are required in order for SBO to be capable of guiding users through the correct sequence of steps for a particular task.

Question 3: Why do we need adaptation?

Navigation adaptation was proposed in order to resolve the identified learnability issues of SBO (Chapter 5). The identified usability issues revealed that more needs to be done in order to improve guidance with regard to task support.

Question 4: Where can we apply adaptation?

This type of navigation adaptation can be applied to controls for all the user interfaces used by a particular user.

Question 5: When can we apply adaptation?

The appropriate navigation adaptation techniques should be applied at run-time when the user requests a certain form (in order to complete a business process).

6.3 Adaptation Design

Currently, no adaptation techniques, system architectures or AUI components are available that are specific to the domain of ERP systems. This places a constraint on the ability to design and implement an AUI for an ERP system. To overcome this limitation, this section proposes an adaptation taxonomy, a system architecture and a set of AUI components that could be used to support the design and implementation of AUIs for ERP systems.

6.3.1 Proposed Adaptation Taxonomy

Adaptation techniques are required to successfully implement the three types of adaptation (content, presentation, and navigation) discussed in the previous chapter. An adaptation taxonomy was proposed by Knutov *et al.* (2009) to answer the question on how adaptation can be implemented for adaptive hypermedia systems. The taxonomy proposed by Knutov *et al.* (2009) is illustrated in Figure 6.2. It depicts the different methods, techniques and approaches for hypermedia systems. This taxonomy was intended to offer a wider range of techniques best suited to the diverse range of users, and to provide better adaptation results. As a result, the taxonomy was proposed to capture the latest trends and technologies to support users in achieving more accurate and richer adaptations.

A detailed literature study revealed that such a taxonomy does not exist for ERP systems. This section proposes an adaptation taxonomy for ERP systems that could be used to guide and support the design and implementation of AUIs for ERP systems (Figure 6.3).

The proposed adaptation taxonomy (Figure 6.3) presents several modifications of the original taxonomy (Figure 6.2). The components of the original taxonomy are illustrated in blue, and the new proposed components are illustrated in orange. Only those adaptation techniques that could be used to address the usability issues identified in Chapters 4 were selected from the original taxonomy. These adaptation techniques are (highlighted in blue in Figure 6.3):

- Inserting/Removing Fragments (content adaptation);
- Altering Fragments (content adaptation);
- Sorting Fragments (presentation adaptation); and
- Guidance (navigation adaptation).

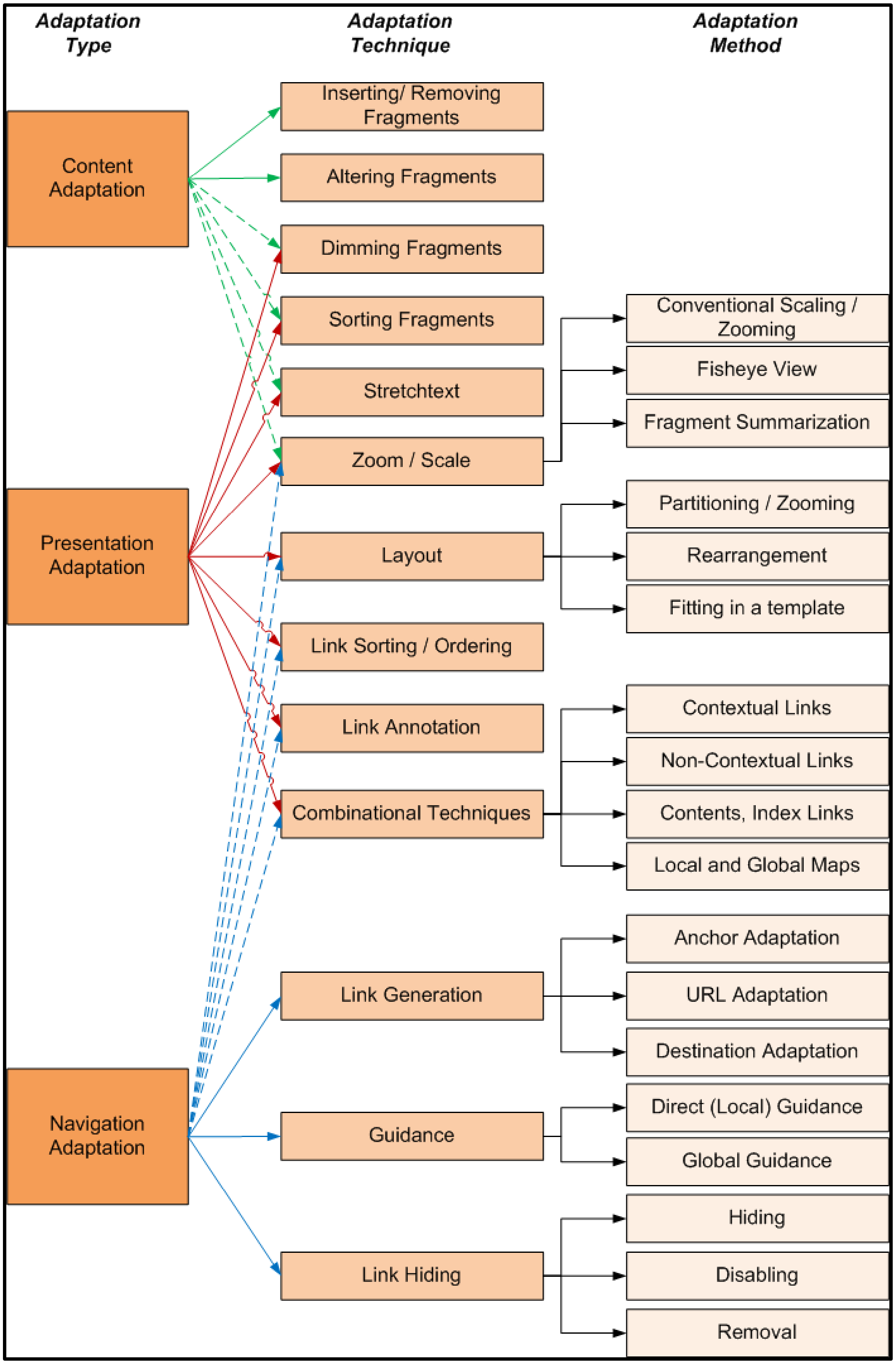


Figure 6.2: Original Taxonomy of Adaptation Techniques (Knutov *et al.* 2009)

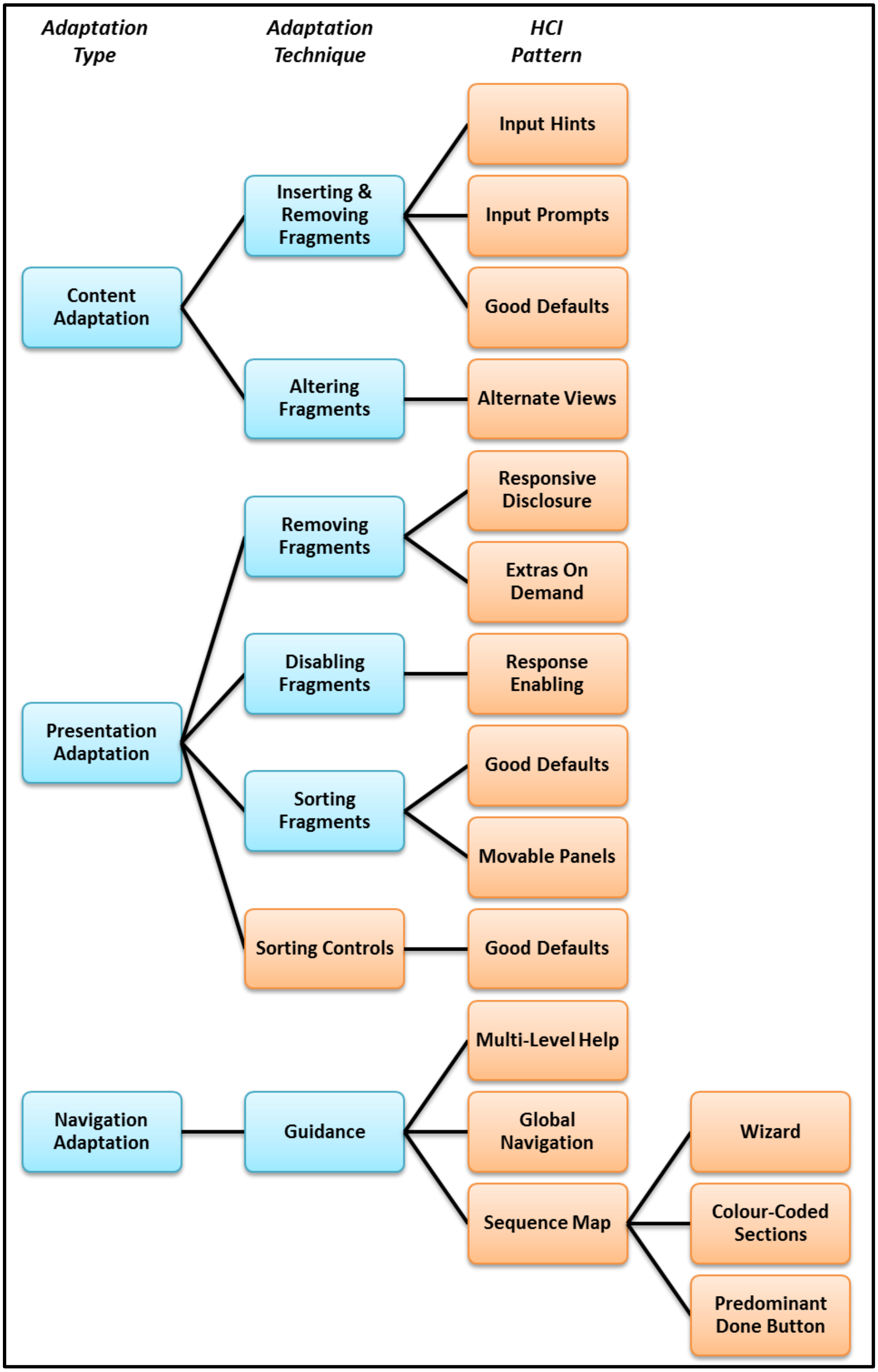


Figure 6.3: Proposed Adaptation Taxonomy

As this research focuses on traditional desktop-based ERP systems (Section 1.7), the approaches and techniques originally proposed for hypermedia systems were replaced by existing human-computer interaction (HCI) design patterns for desktop applications. Existing HCI patterns were selected to support the design of usable ERP systems. These patterns could assist in solving common user interface design problems in a given design domain (Tidwell 2011). HCI design patterns provide an abstract and re-usable form of successful and usable design solutions (Seffah 2010).

Several existing catalogues of HCI patterns exist for both desktop and web-based systems. The aim of these catalogues is to improve system usability by providing patterns to support user interface and interaction design. The most comprehensive, recent and widely referenced catalogues include those of Tidwell (2011) and van Welie (2008). The HCI patterns proposed by Tidwell (2011) could be applied to both desktop and web-based applications, whilst the patterns proposed by van Welie (2008) mainly apply to web-based systems.

The HCI patterns proposed by Tidwell (2011) were selected on the basis of their comprehensiveness and applicability to desktop systems. The following sections discuss how the proposed HCI patterns could be used to implement the various types of adaptation.

6.3.1.1 Content Adaptation

Content adaptation was identified as a means of addressing the usability issues with regard to task support (Table 6.1). Based on the proposed adaptation taxonomy, content adaptation can be implemented by inserting, removing, or altering fragments of information.

The purpose of inserting, removing or altering fragments of information is to ensure efficiency, and to improve user productivity. This can be accomplished by means of providing hints as to what is required from a particular input control, providing prompts in the control (to provide the user with an indication of what is required) and providing good defaults (which are suggested values based on the user's interaction over a period of time).

An example of the Good Defaults HCI pattern, as proposed by Tidwell (2011), is presented in Table 6.2 and illustrated in Figure 6.4. This HCI pattern could be used to successfully meet the proposed adaptation requirements for content adaptation (Section 6.2.1).

Definition:	Pre-fill form controls with best guess values that the user might require.
Use When:	A reduction in the amount of time to complete a form is needed. The user has provided enough contextual information for the user interface to make accurate guesses.
Why:	The user is more efficient in completing a task. The user is spared the effort of thinking about a possible value and of typing or selecting the value.
How:	Pre-populate the appropriate controls when the form loads with the default values. This should only be done when the default values can be predicted or determined with a certain amount of accuracy, in order to ensure that the user will most likely not change the default value.

Table 6.2: Example of Good Defaults Pattern (Tidwell 2011)

The application of the Good Defaults pattern to ERP systems is illustrated in Figure 6.4. The sales employee *David Batra* has been pre-selected by the ERP system, as he is the employee who is MFU. This would save the user some time, as opposed to first viewing the entire list of sales employees and then selecting the appropriate sales employee (Figure 6.5).

Figure 6.4: Application of Good Defaults Pattern

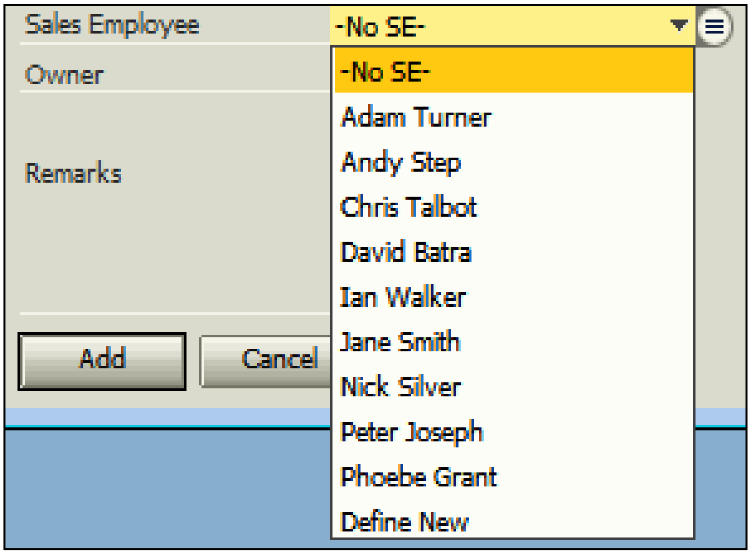


Figure 6.5: Complete List of Sales Employees

6.3.1.2 Presentation Adaptation

Presentation adaptation was proposed to address the usability issues that were identified with regard to efficiency and learnability (Table 6.1). Several adaptation techniques are proposed in Figure 6.3 to address these issues.

Removing fragments uses the HCI patterns of Responsive Disclosure and Extras on Demand. These two HCI patterns aim to reduce the amount of controls on the user interface by displaying only that which is necessary at a particular point in time, whilst providing the option to display more or everything (Extras on Demand). Responsive Disclosure gradually builds the user interface as the user progresses through the task, and eventually reaches the complete and final state of the user interface.

The technique of Disabling Fragments uses the HCI pattern of Response Enabling. This HCI pattern aims to address the issue of guiding the user through a task by enabling task activities in an ordered manner. Responsive enabling could potentially address the adaptation requirements proposed for presentation adaptation. An example of Response Enabling is presented in Table 6.3 and illustrated in Figure 6.6.

Definition:	Response enabling guides a user through a sequence of steps by progressively enabling more from the user interface, as each step (activity) is completed.
Use When:	User interface is complex enough to guide the user through the task step-by-step.
Why:	Users are guided through the user interface interactively in order to create the correct mental model. Response enabling restricts the user from performing hazardous applications, as the user interface has "locked out" those actions by disabling them. This eliminates unnecessary messages and errors.
How:	Only those actions relevant to the user's first steps are enabled and all of the other actions are disabled. As the user makes choices and performs more actions, more of the disabled actions are enabled.

Table 6.3: Example of Response Enabling Pattern (Tidwell 2011)

Figure 6.6 illustrates the different task activities for the Purchase Order user interface in SBO. Based on the Response Enabling adaptation HCI pattern, only once Section one is complete, will Section two be enabled. The same applies for Section three.

Sorting Fragments uses two HCI patterns. These are Good Defaults and Movable Panels. These HCI patterns address the usability issue of personalisation with regard to the user interface, and will re-order the fragments of the user interface based on the user's previous interactions (Good Defaults). It could also allow the user to re-order the fragments based on his/her own preference (Movable Panels).

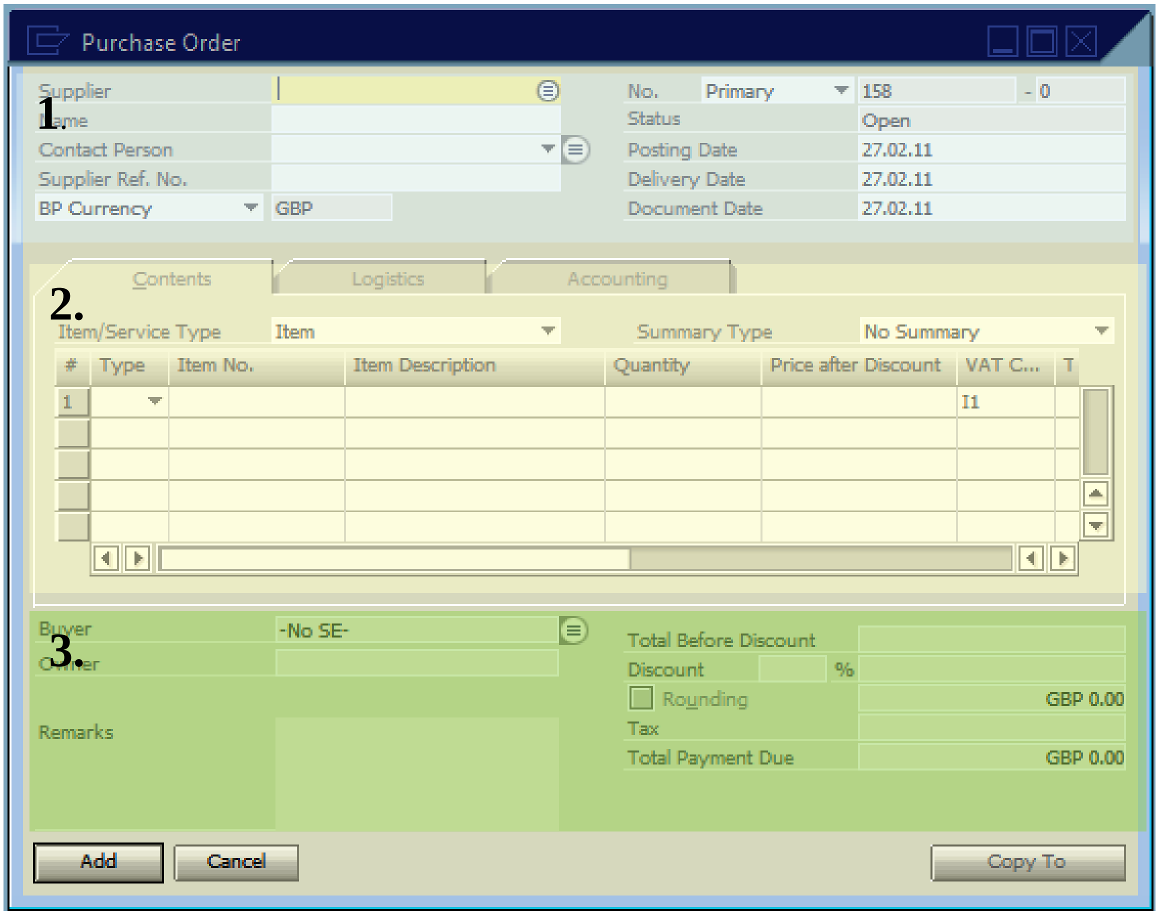


Figure 6.6: Task Segmentation for SBO Purchase Order User Interface

The last presentation adaptation technique is Sorting Controls. This technique is similar to Sorting Fragments, but uses only the Good Defaults HCI pattern. This technique sorts the order of the controls based on the interaction order of the various controls for a particular task or activity for a particular user.

6.3.1.3 Navigation Adaptation

Most of the identified usability issues relating to navigation suggest the need for improved guidance (Table 6.1). For this reason the Guidance technique was selected (from the original taxonomy) and the following HCI patterns are proposed:

- Multi-level Help (providing help at various places and levels within the user interface);
- Global Navigation;

- Sequence Map (which breaks a task into various activities, and also makes use of the Wizard, Colour-Coded Sections and Predominant Done Button HCI patterns).

These techniques aim to ensure that users are guided through the correct sequence of tasks (Sequence Map), evidence of the next sequence of steps is provided (Sequence Map and Colour-Coded Sections) and finding information and functionality is facilitated (Multiple-Level Help). The implementation of these techniques would assist in realising the proposed adaptation requirements for navigation adaptation (Section 6.2.3). An example of the Colour-Coded Sections pattern is presented in Table 6.4 and illustrated in Figure 6.7.

Definition:	The colour-coded sections pattern makes use of colour to identify different sections for a part of the whole user interface.
Use When:	Illustrate the current status of a control (e.g. disabled, enabled, active, complete, or error).
Why:	Provide the user with a visual indication on the state of the control and the action required to be performed on that control.
How:	Select a colour for each of the different states of the control. Apply the relevant colour to the control once its state has changed.

Table 6.4: Example of Colour-Coded Sections Pattern (Tidwell 2011)

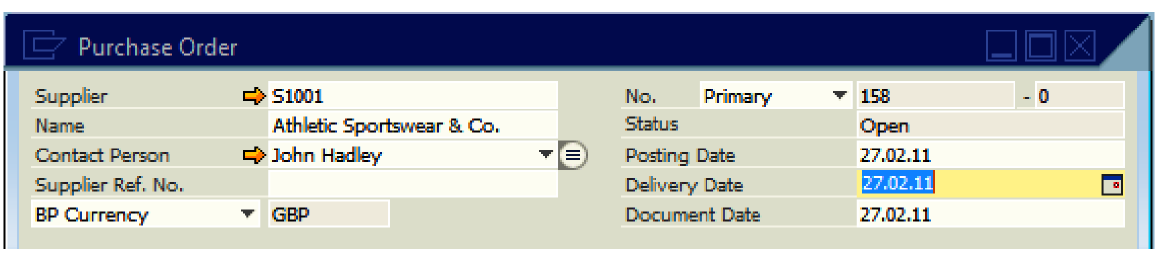


Figure 6.7: Extract of SBO Purchase Order User Interface

Figure 6.7 illustrates the use of colour with regard to the state of a control in SBO. A control can be in one of three states at any given time. The association between the state of a control and its respective colour is presented in Table 6.5.

State	Colour
Disabled	Grey
Enabled	White
Active / Selected	Orange

Table 6.5: Control State vs Colour

The proposed adaptation taxonomy was intended as a means of supporting the design and implementation of AUIs for ERP systems. The next section builds on this by proposing a system architecture for ERP systems incorporating an AUI.

6.3.2 Proposed System Architecture

An AUI system architecture was selected in Chapter 5 that could be specialised for the domain of ERP systems (Section 5.6.3). A specialised version of the selected architecture, specific to ERP systems, is illustrated in Figure 6.8.

Some of the original components were combined with new components to make up the proposed system architecture. Components from the original architecture are illustrated in blue, while the new components that were included are illustrated in orange in Figure 6.8.

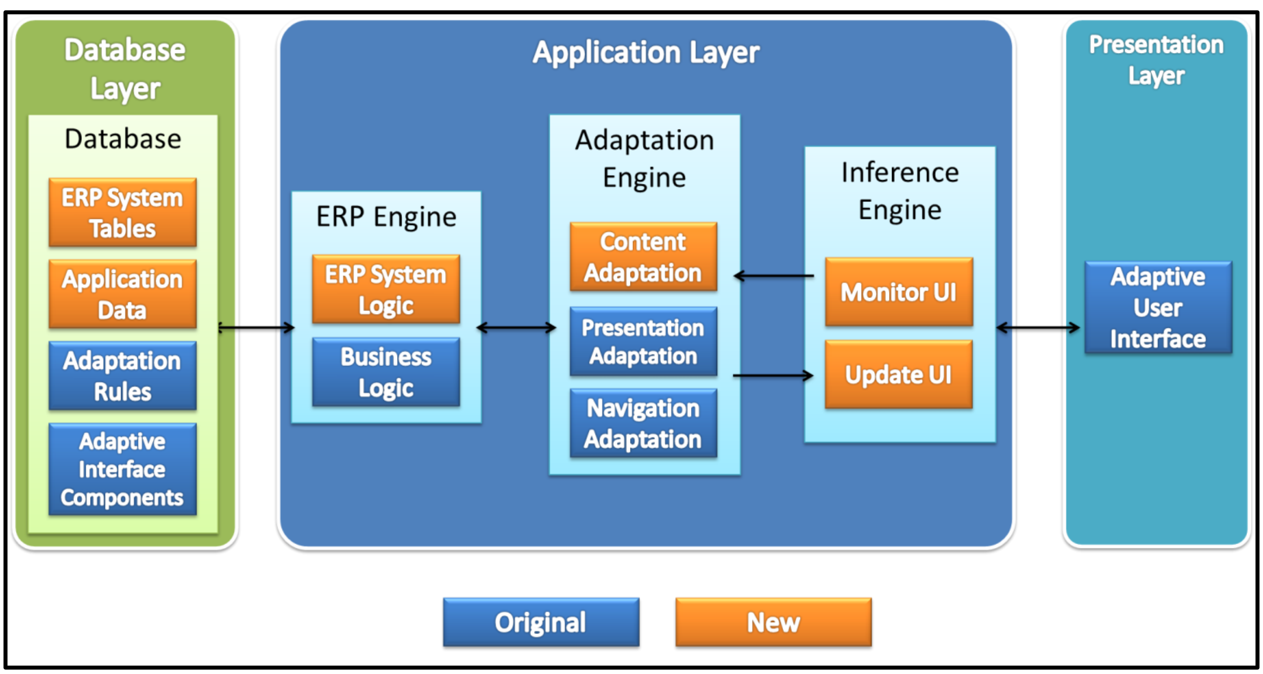


Figure 6.8: Proposed System Architecture

The proposed system architecture differs from the original architecture through the addition of the following components:

- An Inference Engine, which is responsible for monitoring the user interactions with the AUI and for updating the AUI;
- Support for content adaptation – this type of adaptation was missing from the original architecture;
- An ERP Engine which contains both the business logic and the ERP system logic; and
- Support for the ERP system database tables and the application data tables in the database.

These additions are implemented across three different layers of the proposed system architecture.

The architecture represents a typical three-tier distributed architecture, having a data layer (Database), an application layer (ERP Engine, Adaptation Engine and Inference Engine), and a presentation layer (Adaptive User Interface) (Sommerville 2007). These layers and their respective functions are discussed in the following sections.

6.3.2.1 Database Layer

The database layer contains the central database. This database is accessed by the ERP system and is responsible for storing the default ERP system tables, any customised tables specific to a particular enterprise, application data, adaptation rules and the AUI components (user model, task model and dialog model). The database in the original architecture (Ramachandran 2009) did not make provision for the ERP system tables, as well as for any application data that might be generated through the use of the system.

The inclusion of the adaptive interface components is also a new addition. The original database stored the user profile (model), but did not make provision for any of the other adaptive components (task model and dialog model).

6.3.2.2 Application Layer

The application layer is responsible for the functionality of the ERP system and is also responsible for the adaptation and generation of the user interface. The application layer makes use of an inference engine, an adaptation engine and an ERP engine. The inference engine is responsible for monitoring any interaction that the user has with the AUI, and for generating and updating the user interface. The inference engine was not part of the original system architecture. AUIs are required to constantly monitor the user interactions with the system and to update the user interface (Knutov *et al.* 2009).

6.3.2.3 Presentation Layer

A log file is proposed (Figure 6.9) in order for the inference engine to successfully monitor and capture the user's interaction with the user interface. The log file will need to exist for each user of the ERP system and is responsible for capturing the following four main pieces of data:

- The ID of the ERP system form – captured when the form load event is triggered;
- Start time – the date and time when the task was started (when the ERP system form was loaded);
- Control sequence – the sequence in which the controls were selected, the value that was entered or selected, and the date and time when the control was selected; and
- End time – the date and time when the task was ended (when the ERP system form was closed).

The log file will assist with content adaptation and presentation adaptation. In terms of content adaptation, it will provide the list of values selected or entered for a particular control. This will assist in applying list-based adaptation (Section 6.2.1). With regard to presentation adaptation, the log file will assist by providing the list of controls with which a particular user interacted on a particular form. This would assist in terms of identifying unused or infrequently used controls (which could be hidden) to simplify the user interface (Section 6.2.2).

```
<FormID></FormID>
<StartTime></StartTime>
<ControlSequences>
                  <Control>
                     <ID></ID>
                     <Value></Value>
                     <DateTime></DateTime>
                  </Control>
</ControlSequences>
<EndTime></EndTime>
```

Figure 6.9: Proposed Log File Schema

Furthermore, having this information could also assist in determining the user's mental model with regard to completing a particular task (based on the form ID), and spatially laying out the MFU controls, according to that mental model.

The Adaptation Engine takes the input from the Inference Engine and determines what type of adaptation is required to perform on the user interface in order to improve user productivity. The proposed system architecture is capable of supporting content, presentation and navigation adaptation, whilst the original architecture only supported presentation and navigation adaptation. The ERP Engine is the existing functional ERP system that controls all of the business logic and ERP functionality.

The Adaptive Engine will interact with the ERP Engine in order to maintain the integrity of the system and the data. The ERP engine was not part of the original architecture (Ramachandran 2009), but was included in order to specialise the proposed architecture for ERP systems. Inclusion of the ERP Engine also indicates how the Adaptive Engine interacts with the ERP system.

6.3.2.4 Presentation Layer

The presentation layer contains the AUI. The AUI will be generated based on one or more of the required adaptation types from the Adaptation Engine, and delivered through the Inference Engine. Any interaction performed by the user will be monitored by the Inference Engine. Determining the validity of the data and saving the data to the database will be done by the ERP Engine.

6.3.3 Adaptive Interface Components

The different types of adaptation (content adaptation, presentation adaptation, and navigation adaptation) discussed in Chapter 5 make use of several adaptive interface components in order to implement the adaptation. These adaptive interface components are represented in the form of declarative models. An investigation into the most widely utilised declarative models for AUIs was performed. The models identified were the user model, the task model, the domain model, the dialog model and the presentation model.

These models assist in the building of AUIs by providing information that is acquired from the users when completing their tasks within a specific context (Ahmed and Ashraf 2007; Zarikas 2007; van Tonder and Wesson 2008). For the purposes of this research, only the user model (Section 5.4.1), the task model (Section 5.4.2), the dialog model (Section 5.4.4) and the presentation model (Section 5.4.5) will be used. It is not necessary to include a domain model (Section 5.4.3) for ERP systems, as the functionality of an ERP system is configured in terms of a specific enterprise (this role is traditionally performed by an ERP consultant).

The remaining adaptive components discussed in this section would be stored in the database (adaptive interface component) of the proposed system architecture (Figure 6.8).

6.3.3.1 User Model

User models are a critical component of any AUI, as they provide the information necessary to adapt the user model to the different users (Ahmed and Ashraf 2007; Tran *et al.* 2009). The proposed user model (Figure 6.10) will exist for each user and would support the proposed log file (Figure 6.9) by containing a summary of the key elements that the ERP system would require (in order to adapt the user interface). These elements will be grouped according to the various tasks (ERP system form IDs) performed by the user.

The data stored in the user model would assist in the decisions made by the adaptation engine (Figure 6.8) in terms of supporting content adaptation and presentation adaptation. All of the data stored in the user model would be continuously updated based on the user's interaction with the ERP system. These data would be refreshed once a task has been completed. This differs from the log file, which maintains a complete history of the user's interaction with the ERP system. The data stored in the user model include:

- Form ID – The ID of an ERP system form, which is captured when the FORM_LOAD event handler is called in the ERP system.
- Control sequences – Stores the sequence in which the controls were selected for a particular task. For each control in this sequence, the ID of that control and the MRU item and two MFU items are stored.

```
<usermodel>
    <formID>
        <ControlSequences>
            <Control>
                <ID></ID>
                <Values>
                    <MRU></MRU>
                    <MFUVs>
                        <MFU1></MFU1>
                        <MFU2></MFU2>
                    </MFUVs>
                </Values>
            </Control>
        </ControlSequences>
    </formID>
</usermodel>
```

Figure 6.10: Proposed User Model Schema

The user model is typically called when a form is loaded to update the controls on that form. Capturing the sequence in which the controls were selected could assist with presentation adaptation – in terms of supporting the effective task completion by the re-ordering of the controls on the user interface. This would also support presentation adaptation in terms of only displaying those fragments and controls that are used by a particular user (Section 6.2.2).

Storing the MRU values and MFU values could assist with content adaptation, in terms of supporting the Good Defaults HCI pattern (Figure 6.3). The user model (through the adaptation process) could assist in resolving the following identified usability issues (Table 6.1):

- SBO does not enable efficient completion of tasks;
- SBO does not improve user productivity;
- SBO does not automate routine tasks; and
- Finding information and functionality is difficult.

6.3.3.2 Task Model

Task models may be defined as the hierarchical representation of the tasks and sub-tasks (activities) performed by users of a particular software system. Each task and activity stored in the task model should contain a trigger for the task, the steps required to successfully complete the task and the pre- and post-conditions of the task (van Tonder and Wesson 2008; Tran *et al.* 2009; Vidani and Chittaro 2009). The proposed task model (Figure 6.11) is based on the above description of a generic task model and contains:

- The name of the task;
- The goal of the task;
- The form ID linked to the task;
- The pre-condition of the task;
- The post-condition of the task; and
- The related sub-tasks (activities).

```
<taskmodel>
    <Task>
        <Name></Name>
        <Goal></Goal>
        <FormID></FormID>
        <Pre-Condition></Pre-Condition>
        <Post-Condtion></Post-Condtion>
        <Activities>
          <Activity>
              <Name></Name>
              <SequenceID></SequenceID>
              <Goal></Goal>
              <Pre-Condition></Pre-Condition>
              <Post-Condition></Post-Condition>
          </Activity>
        </Activities>
    </Task>
</taskmodel>
```

Figure 6.11: Proposed Task Model Schema

The task model can assist with adaptation by means of supporting navigation adaptation by resolving the following usability issues (Table 6.1):

- Guiding users through the correct sequence of steps (activities); and
- Providing evidence of the next sequence of steps.

Information stored in the task model would not change, as the tasks that the user will need to complete would remain the same. The manipulation of tasks is done by means of a dialog model and a presentation model. These will be discussed in the following sections.

6.3.3.3 Dialog Model

A dialog model complements the task model by describing and organising the task activities that need to be performed on the user interface. Dialog models are also hierarchically organised in a similar way to task models. The structure of a dialog model typically contains the specification of when users can invoke functions, interaction media, select or specify inputs and when the software system can query and present information using the various controls (Puerta 1997; Menkhaus and Fischmeister 2003; Ahmed and Ashraf 2007).

Presentation models, as with dialog models, are derived from the task model and assists in describing the visual appearance of the user interface. Developers often combine the dialog and presentation models because of their similarity.

The proposed dialog model (Figure 6.12) is based on the description of a generic dialog model and a presentation model. The proposed model would support the process of presentation adaptation by implementing the following adaptation techniques (Figure 6.3):

- Removing Fragments;
- Disabling Fragments;
- Sorting Fragments – the sequence ID could change in the dialog model, based on the user's interaction with the user interface; and
- Sorting controls.

The values of the various controls per activity could change, depending on the user's interaction with the user interface.

The proposed dialog model could assist with the adaptation process, by contributing to presentation adaptation (Figure 6.3) in order to resolve the following usability issues (Table 6.1):

- Too many steps required to complete a task (Disabling fragments);
- Too many unused controls (Removing fragments);
- The layout of the user interface does not contribute to the efficient completion of tasks (Sorting fragments); and
- Steps need to be manually recorded the first time, in order to be remembered for future use (Disabling fragments).

```
<dialogmodel>
    <taskname>
      <Activities>
        <Activity>
          <Name></Name>
          <SequenceID></SequenceID>
          <Enabled></Enabled>
          <ParentControlID></ParentControlID>
          <Controls>
            <Control>
              <ID></ID>
              <Name></Name>
              <Left></Left>
              <Width></Width>
              <Height></Height>
              <Top></Top>
              <GroupWith></GroupWith>
              <Enabled></Enabled>
              <Focus></Focus>
            </Control>
          </Controls>
        </Activity>
      </Activities>
    </taskname>
</dialogmodel>
```

Figure 6.12: Proposed Dialog Model Schema

6.3.4 Discussion

This section has proposed the design of an AUI for ERP systems. The design comprises an adaptation taxonomy (Figure 6.3), a system architecture (Figure 6.8) and a set of adaptive interface components (Figures 6.10 – 6.12). The adaptation taxonomy was proposed to suggest different ways in which content, presentation and navigation adaptation techniques could be applied to ERP systems – by making use of HCI patterns. Whilst the proposed system architecture (Figure 6.8) is specific to the domain of ERP systems, the declarative models are generic, and could be applied to any domain.

The next section presents a discussion of the implementation of an AUI for an existing ERP system (SBO), which makes use of the proposed adaptation taxonomy, the proposed system architecture and the proposed declarative models.

6.4 Adaptation Implementation

Selected adaptation techniques from the proposed adaptation taxonomy and selected components from the proposed system architecture were implemented (using the SDK of SBO). This was done in order to develop an adaptive version of SBO.

6.4.1 Implementation of the Proposed Adaptation Taxonomy

Only those adaptation techniques (from the proposed adaptation taxonomy) which presented a theoretical efficiency benefit were implemented (Figure 6.13). Most of the decisions made for the implementation of the adaptation techniques were determined by an analysis of the identified task (processing a purchase order), using the keystroke-level model (Card *et al.* 1983), and by the adaptation requirements as specified in Section 6.2. The following adaptive techniques were implemented to address the identified usability issues of SBO, namely Inserting and Removing Fragments (content adaptation), Disabling Fragments (presentation adaptation) and Guidance (navigation adaptation). The implementation of each of these adaptation techniques is discussed in the following sections under the headings of their respective adaptation types.

6.4.1.1 Content Adaptation

Content adaptation was implemented by using the Inserting and Removing Fragments technique and the Good Defaults HCI pattern from the proposed adaptation taxonomy (Figure 6.13). In order to achieve the list-based adaptation for content adaptation, the proposed log file (Figure 6.9) and the proposed user model (Figure 6.10) were implemented and used to provide the MRU and MFU items. Calculation of the MRU and MFU items was done by using the Base Adaptive Algorithm, as proposed by Findlater and McGrenere (2008) (Figure 6.14).

The software development kit (SDK) of SBO did not allow for the re-ordering of items in the list (for dropdown boxes) at run-time. To overcome this challenge, new dropdown boxes containing “smart lists” were created. Figure 6.15 illustrates the difference between the original list and the smart list. The list at the top of Figure 6.15 is the original list of Sales Employees in SBO in alphabetical order. The list at the bottom of Figure 6.15 re-orders the items on the list, based on their recency (MRU) and frequency (MFU). The remainder of the list has remained the same (in alphabetical order), as specified in the Base Adaptive Algorithm (Figure 6.14).

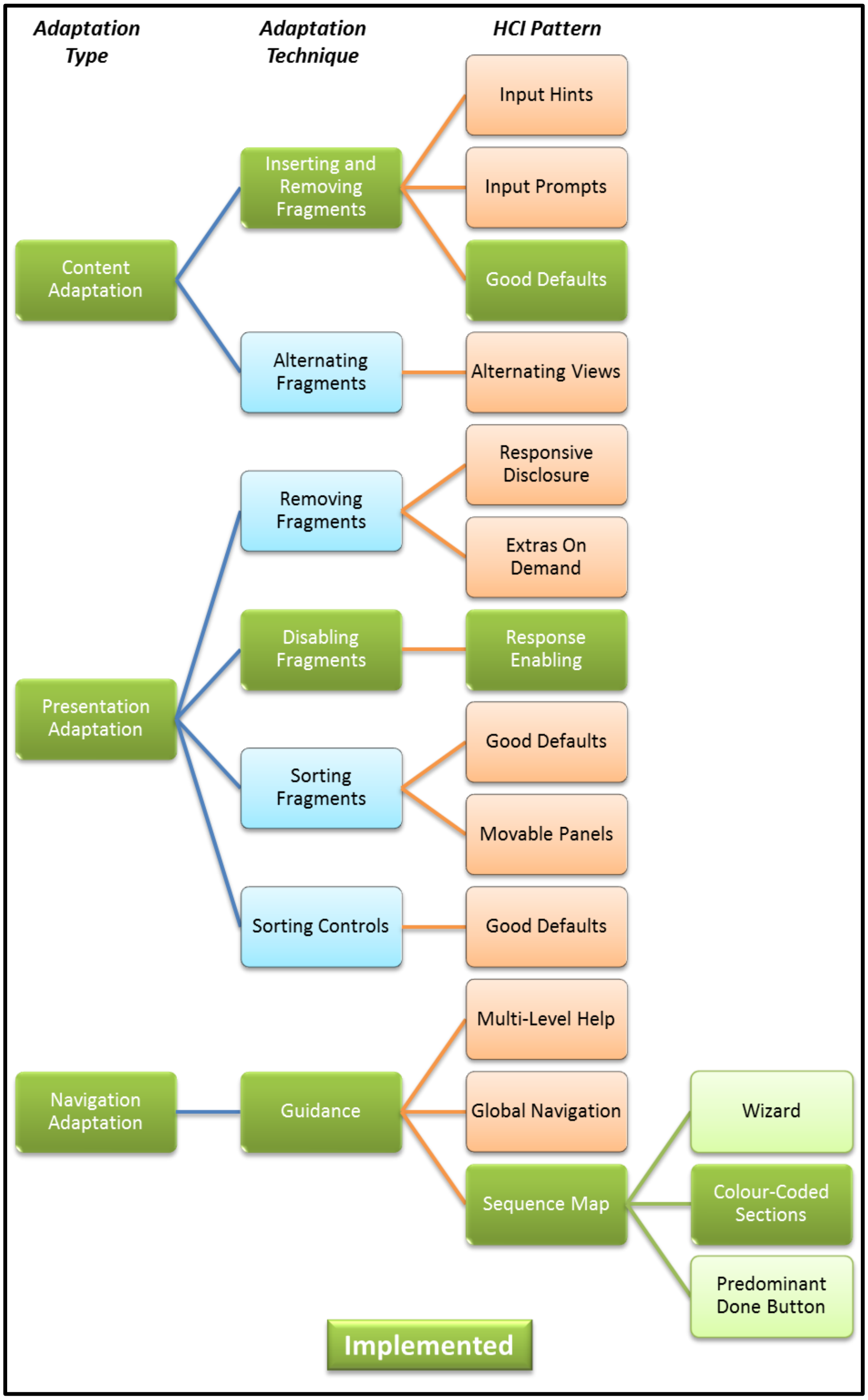


Figure 6.13: Implemented Adaptation Techniques

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 6.14: Base Adaptive Algorithm (Findlater and McGrenere 2008)

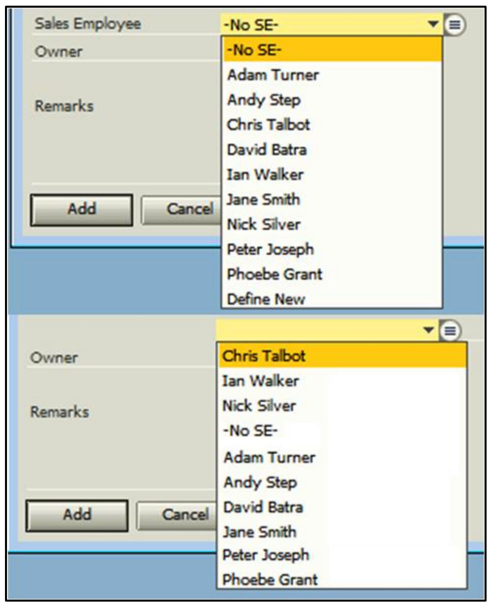


Figure 6.15: Original List (Top) vs. Smart List (Bottom)

6.4.1.2 Presentation Adaptation

Presentation adaptation was implemented to improve the user interface of SBO. In order to achieve the adaptation requirements for presentation adaptation, specified in Section 6.2.2, the Disabling Fragments adaptation technique and the Response Enabling HCI pattern were implemented from the proposed adaptation taxonomy (Figure 6.13). The remaining adaptation techniques and HCI patterns for presentation adaptation were not implemented, as the keystroke-level analysis revealed that visually re-ordering controls would not make any significant impact on efficiency.

The fragments (a collection of controls within a grouped section of a form in SBO) were enabled and disabled, based on the sequence order in the task model and the dialog model. Once all the controls for a particular fragment are complete, the controls in the next fragment are enabled.

The task model was used for the activity sequence and the dialog model provided the necessary list of controls for each activity that needed to be completed. Figure 6.16 illustrates the use of Enabling and Disabling Fragments (tabs) in SBO.

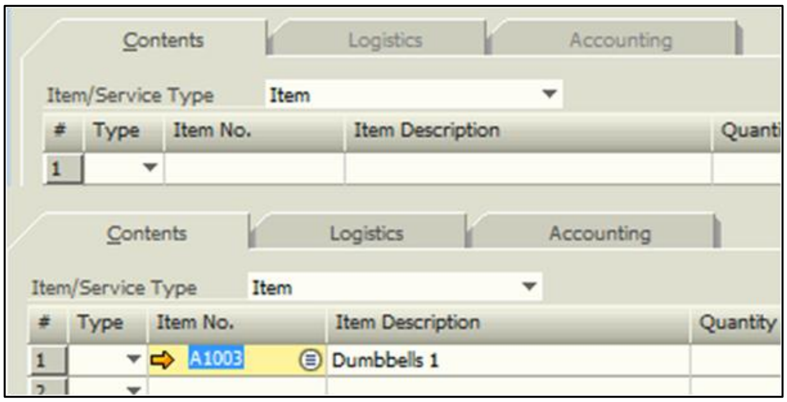


Figure 6.16: Original Enabled Fragments (top) vs Disabled Fragments (bottom)

Originally, all of the tabs (in SBO) for entering an item for a purchase order are enabled (top). After the implementation of the task model and the dialog model, the tabs are disabled (bottom) and only re-enabled once an item has been entered. This was done in order to improve the learnability of SBO by providing task support and guidance.

6.4.1.3 Navigation Adaptation

Navigation adaptation was implemented to support the need for improved guidance with regard to task completion in SBO (Section 6.2.3). In order to support the adaptation requirements for navigation adaptation, the Guidance technique and the Sequence Map (Colour-Coded Sections) HCI pattern were implemented from the proposed adaptation taxonomy (Figure 6.13). The implementation of the Guidance technique made use of the proposed task model (Figure 6.11) and the proposed dialog model (Figure 6.12). These models assisted the Inference Engine (Figure 6.8) in deciding which fragments of the form to enable and which to disable at a particular point in time.

The Colour-Coded Sequence was applied to the controls of SBO in the Purchase Order form. Currently, SBO only uses three colours to indicate the status of the form controls (Figure 6.17), namely white for enabled, grey for disabled and orange for currently selected.

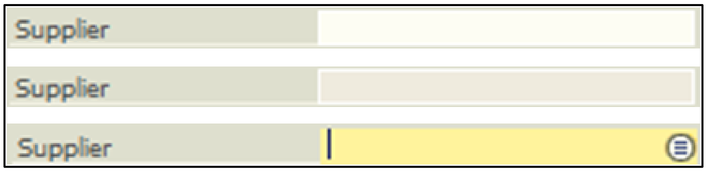


Figure 6.17: Original Colour-Coded Status of SBO

A fourth colour status (green) was added to support guidance and task completion by indicating that the required data was entered (Figure 6.18).

Figure 6.18: New Colour-Coded Status of SBO

The colour of the relevant control will change to green once the user has entered and moved to the next control in the activity.

6.4.2 Implementation of the Proposed System Architecture

Several components from the proposed system architecture were implemented to make the user interface adaptive (Figure 6.19). The decision to implement specific components was based on the functionality provided by the SDK of SBO. The implemented components of the proposed system architecture are discussed in more detail in the sections that follow.

6.4.2.1 Database Layer

Only the Adaptation Rules and Adaptive Interface Components were implemented as part of the Database Layer (Figure 6.19). The ERP system tables and the Application Data did not have to be implemented, as these were created during the installation of SBO.

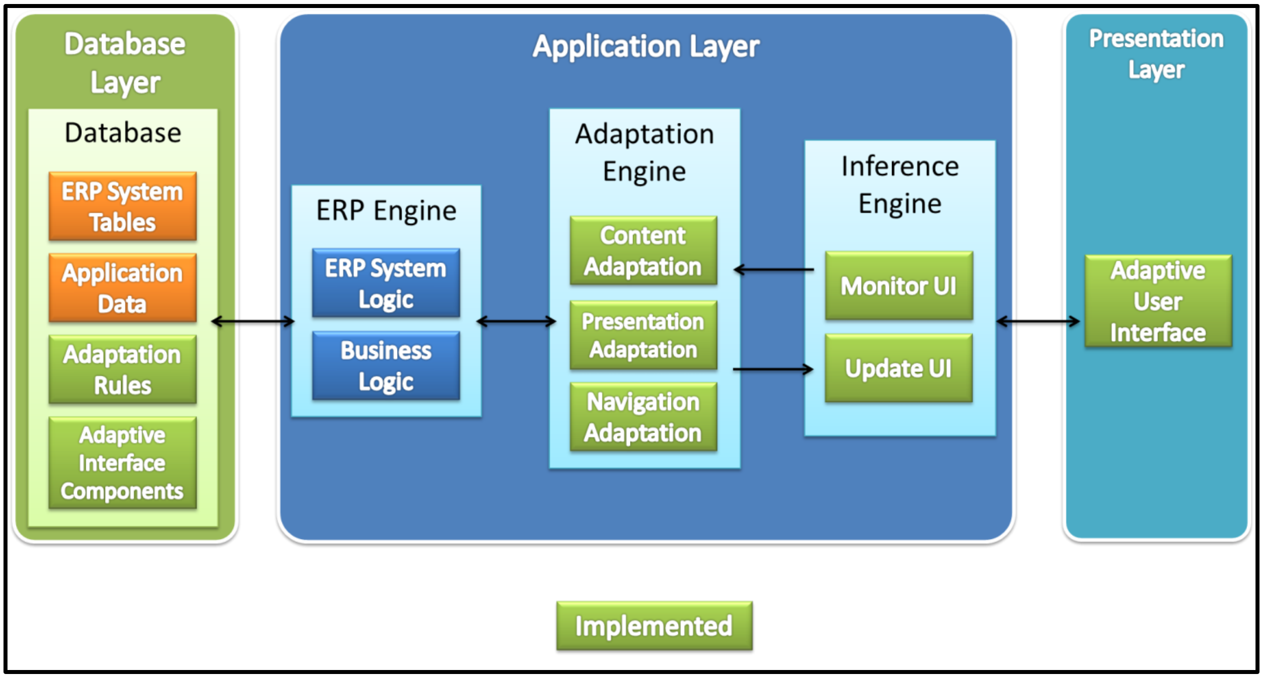


Figure 6.19: Implemented System Architecture Components

The implemented Adaptation Rules supported the decision-making on which types of adaptation should be delivered and how the adaptation should be delivered. This component of the Database Layer relied on the Adaptive Interface Components (user model, task model and dialog model) stored in the database. All of the Adaptive Interface Components were implemented, according to their proposed design in Section 6.3.3. These components were called by the Adaptation Engine in SBO to execute the required adaptation type.

6.4.2.2 The Application Layer

Only the Adaptation Engine and Inference Engine were implemented from the Application Layer of the proposed system architecture (Figure 6.19). The ERP Engine did not have to be implemented, as SBO through its SDK was capable of separating the application logic from the user interface logic. The application logic of SBO is represented by the ERP Engine in the proposed system architecture. Only the interactions of the user with the user interface had to be captured, as the SDK of SBO handled all of the other events. These included updating the Application Data and ERP System Tables in the database.

The Inference Engine is a key contribution to the proposed system architecture, as it is responsible for monitoring the user's interactions with the AUI and updating the user interface with the appropriate adaptation type (Section 6.3.2.2). The Monitor UI component of the Inference Engine was implemented and made use of the proposed log file (Figure 6.9). All interactions with the Purchase Order form by a particular user were logged in the log file. The Update UI component was implemented and made use of the implemented user model to deliver content adaptation (Section 6.4.1.1). This component also made use of the task and dialog model to deliver presentation adaptation (Section 6.4.1.2) and navigation adaptation (Section 6.4.1.3).

The implementation of the Adaptation Engine was achieved by implementing each adaptation type as a separate component. The content adaptation component either updated the user model based on the information in the log file once a purchase order was processed, or it provided the MRU and MFU items to the Update UI component of the Inference Engine. The presentation adaptation component made use of the task model and dialog model in order to determine which controls belonging to a particular activity should be enabled or disabled. This information was passed to the Inference Engine, so that the user interface could be updated accordingly. The Navigation adaptation component also made use of the task and dialog models to determine the state of specific controls once a particular section was enabled (through presentation adaptation). This information was also sent to the Inference Engine in order to update the user interface accordingly.

6.5 Discussion

Section 6.4 has shown that the proposed AUI design can be used to design an AUI for an existing ERP system (that is widely used by small enterprises).

In order to achieve the required level of adaptation, several changes were made to both the user interface and the ERP system. The changes that were made to the user interface were made to the controls and the fragments in which these controls were placed. These changes were:

- Re-ordering of items in a list, based on usage (Figure 6.14);
- Enabling and disabling fragments, based on the progress of a task (Figure 6.15); and

- Changing the colour of controls, based on their state (Figure 6.16).

These changes were based on the adaptation techniques proposed in the proposed adaptation taxonomy (Figure 6.3). Only those techniques that could offer a theoretical improvement in efficiency (according to the keystroke-level model) were selected. These techniques are illustrated in Figure 6.13.

Achieving the adaptation techniques and HCI patterns was made possible through the implementation of the proposed system architecture (Figure 6.8) and the proposed adaptive components (Figures 6.9 – 6.12). Selected components of the proposed system architecture were implemented. Only the Inference Engine and the Adaptation Engine had to be implemented in the Application Layer. The ERP Engine did not have to be implemented as SBO through its SDK supported the separation of business logic and user interface logic. The database for SBO was updated to include the various adaptation rules and adaptive interface components. All of the adaptive components in the proposed design were implemented (with no changes). The adaptive components were included to support and inform the decisions made by the Adaptation Engine. The implementation of the Application Layer and updates to the database in the Database Layer supported the implementation of the AUI in the Presentation Layer.

6.6 Conclusion

This chapter proposed a set of adaptation requirements to address the usability issues of SBO. The requirements were proposed in terms of content adaptation, presentation adaptation and navigation adaptation. The proposed adaptation requirements have shown that the application of AUIs to address the usability issues of SBO could potentially improve the efficiency and learnability of SBO.

This chapter has presented the design of an AUI, comprising an adaptation taxonomy, a system architecture and a set of adaptive interface components. The proposed adaptation taxonomy presented a unique combination of existing adaptation techniques and HCI design patterns to support the implementation of the different types of adaptation.

The proposed system architecture is a specialised version of the AUI architecture selected in Chapter 5. The architecture is specialised for the domain of ERP systems through the inclusion of an ERP Engine in the Application Layer and ERP System Tables and Application data in the Database Layer. The proposed adaptive interface components are based on the description in Section 5.4, but were updated to interface with the user interface of SBO.

A prototype AUI was successfully implemented for SBO, using the proposed AUI design. The prototype AUI demonstrated that the proposed AUI design could be used to develop an AUI for an existing ERP system, specifically SBO. Selected techniques from the proposed adaptation taxonomy and selected components from the proposed system architecture were implemented. All of the proposed adaptive components were also implemented.

The next chapter presents a discussion on the results of a comparative user study that was conducted to determine whether the usability of the prototype AUI for SBO is better than the non-adaptive version of SBO.

Chapter 7: Evaluation

7.1 Introduction

The aim of this chapter is to present and discuss the findings of a usability evaluation on SAP Business One (SBO). Two versions of SBO were compared in the evaluation, an adaptive version and a non-adaptive version. The evaluation aimed to identify whether any benefits were achieved by incorporating an adaptive user interface (AUI) into SBO.

The AUI developed for SBO was based on the AUI design for ERP systems proposed in Chapter 6. The prototype AUI supported three types of adaptation, namely content adaptation, presentation adaptation and navigation adaptation. These specific types of adaptation were implemented to determine whether AUIs could improve the usability of the SBO, in terms of efficiency, learnability and satisfaction. These attributes were identified as the areas which negatively affect the usability of enterprise resource planning (ERP) systems (Chapter 3).

The remainder of this chapter is structured as follows: firstly, a discussion on evaluation methods for AUIs is presented. Based on this, an outline of the design of the evaluation and a description of the evaluation is discussed. This is followed by the results of the evaluation. Finally, a discussion based on the results is presented and is followed by conclusions.

7.2 Evaluating AUIs

The evaluation of AUIs involves assessing the effectiveness or the usability of the adaptive system. Although no standard methodology exists for evaluating AUIs, research has shown that either an empirical or a layered approach can be adopted (Lavie *et al.* 2005; 2006; Letsu-Dake and Ntuen 2010). Empirical evaluations are conducted in the format of a controlled experiment and evaluate the usability of the adaptive system. Layered evaluations of AUIs present an alternative approach whereby both the interaction component and the decision-making component of the adaptive system are evaluated (Paramythis *et al.* 2010).

Over the years, empirical evaluations have emerged as the most predominant approach for evaluating AUIs (Peng and Silver 2007; van Tonder and Wesson 2008; Findlater and McGrenere 2010; Lavie and Meyer 2010; Letsu-Dake and Ntuen 2010; Park and Han 2011). This is in response to earlier research emphasising the value of empirical evaluations and the need for more empirical evaluations to be conducted (Lavie *et al.* 2005; Álvarez-Cortés *et al.* 2007; Hou *et al.* 2011). Traditional HCI methodologies (expert reviews, interviews, questionnaires, and think-aloud protocols) are often used in conjunction with empirical evaluations in order to evaluate the usability of AUIs (Gena 2005; Lavie *et al.* 2005; Álvarez-Cortés *et al.* 2007; van Velsen *et al.* 2008).

As this chapter focuses on evaluating the usability of SBO with an AUI, the empirical evaluation approach was selected. The layered approach was not selected, as it is not the objective of this chapter to evaluate the effectiveness of the decisions made by the adaptive engine.

7.2.1 Empirical Evaluations

An empirical evaluation is a means of determining the user's performance and attitude to a particular system. This is achieved by conducting a summative usability evaluation, in which feedback on the effectiveness, efficiency and satisfaction of a particular system is gathered from the users, through the utilisation of various usability evaluation methodologies (Gena 2005; van Velsen *et al.* 2008).

Research has shown that empirical evaluations typically consist of eleven steps, which can be grouped into four phases (Gena 2005; Findlater and McGrenere 2010; Letsu-Dake and Ntuen 2010; Park and Han 2011). These phases are illustrated in Figure 7.1.

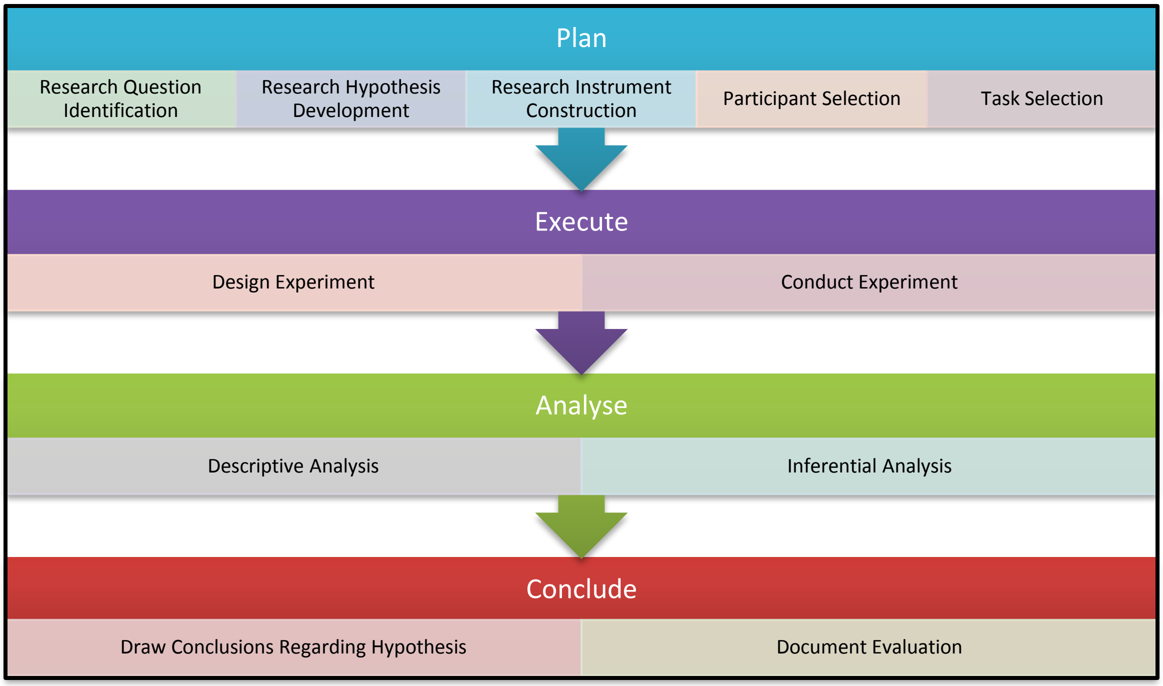


Figure 7.1: Evaluation Protocol for Empirical Evaluations

Phase 1: Plan

The first step in phase 1 involves identifying the main research question (the purpose of the evaluation). This research question is then formalised into one or more research hypotheses, which can either be rejected or fail to be rejected – based on the results of the evaluation. Once the null and alternative hypotheses have been established, the independent and dependent variables are then identified.

The dependent variables in this case are often the adaptive and non-adaptive system, while the independent variables are the attributes against which the dependent variables are measured. This is followed by constructing the various research instruments (questionnaires and task lists) that will be used in the evaluation. Next, participants need to be selected according to a set of criteria. These criteria are determined based on the nature of the study and should be aimed at identifying those participants who are representative of the actual users (Gena 2005).

Empirical evaluations of AUIs are typically task-based. Therefore, tasks which the users need to perform on the system need to be representative of the routine tasks performed by the actual users, as this has been shown to be more useful (Lavie *et al.* 2005).

Phase 2: Execute

The second phase of an empirical evaluation involves designing the controlled experiment and conducting the evaluation. Typically, experimental designs involving AUIs are either between-subject designs, within-subject designs, or mixed between-within subject designs. Comparative evaluations (between the system with the AUI and the non-AUI system) are often used to support the experimental design, in order to determine whether a statistically significant difference exists between the two systems (Álvarez-Cortés *et al.* 2007). The comparative evaluation also assists in determining whether the AUI provides any improvements in terms of efficiency. This is achieved by means of collecting additional task-related data (Letsu-Dake and Ntuen 2010).

Phase 3: Analyse

Phase three involves statistically analysing the data collected from the evaluation. Descriptive and inferential analyses are typically performed on the data collected from usability evaluations. The descriptive statistics are used to assist in summarising a set of data, whilst the inferential statistics are utilised for the purposes of making inferences on the population (Witte and Witte 2009).

Phase 4: Conclude

The last phase of the empirical evaluation involves drawing conclusions (based on the results of the inferential statistics) on the research hypotheses defined in the planning phase of the evaluation. Lastly, the results of the experiment along with a description of the four phases of the evaluation (outlined in this section) are documented.

The phases for conducting an empirical evaluation, described in this section, are used in the next section which discusses the design of the evaluation.

7.3 Evaluation Design

7.3.1 Goals and Objectives

The primary objective of this evaluation is to determine whether AUIs can offer any significant improvements in the usability of SBO. This will be done by comparing an adaptive version of SBO with a non-adaptive version of SBO.

Participants used for the evaluation needed to satisfy three criteria, namely they needed to have background knowledge of ERP systems, they needed to have an equivalent level of ERP expertise and experience, as the real-world users of SBO and they needed to have no prior knowledge of SBO. The evaluation required that the participants be evaluated twice. This was done to measure for learnability and to determine the usefulness of the adaptation effects.

7.3.2 Hypotheses

SBO suffers from several usability issues that could potentially be addressed by AUIs (Chapter 5). To determine the impact that AUIs have on the usability of SBO, the following hypotheses were proposed:

H_0: An AUI for SBO cannot improve the usability (*efficiency, learnability, and satisfaction*) of SBO.

$H_{0,1}$: An AUI for SBO cannot improve the *efficiency* of SBO.

$H_{0,2}$: An AUI for SBO cannot improve the *learnability* of SBO.

$H_{0,3}$: An AUI for SBO cannot improve the *satisfaction* of SBO.

7.3.3 Metrics

The following metrics were evaluated in order to assess the null hypotheses established in the previous section:

- *Efficiency*: Performance metrics were used to measure efficiency (Tullis and Albert 2008). The performance metrics that were used were: task success, time-on-task, number of mouse clicks and number of errors. Efficiency data were captured by each version of SBO and then stored in a log file for each user.
- *Learnability*: Learnability was calculated as the difference in the efficiency results over the two-day evaluation period. This was calculated as the difference in task success rates, the difference in the amount of time spent on tasks, the difference in the number of mouse clicks and the difference in the number of errors committed.
- *Satisfaction*: Subjective metrics were used to capture user satisfaction. The subjective metrics were measured using a 5-point Likert scale, and were grouped into four categories, namely overall satisfaction, task support and efficiency, learnability, and adaptivity. The subjective metrics assisted in obtaining the data for perceived efficiency, perceived learnability and perceived satisfaction. The findings from the satisfaction metrics were used to support the findings from the performance metrics.

Biographical metrics were also captured as part of the study. The data gathered from these metrics were used to establish the participants' profile.

7.3.4 Instruments

The following instruments were provided to the participants:

- *Informed Consent form* – Candidate participants were provided with a consent form that was signed before the experiment commenced.
- *Biographical questionnaire* – This was provided to the participants in order to obtain some background information and to establish a participant profile (Appendix H).

- *Task list* – The task list comprised eight tasks, all relating to processing a purchase order. Processing a purchase order was identified, from the interview study of SBO users in the Manufacturing sector, as the most common and frequently performed task (Chapter 6). The same task list was provided to the participants, who used the adaptive and the non-adaptive versions of SBO (Appendix I).
- *Post-test satisfaction questionnaire* – A post-test satisfaction questionnaire was issued to the participants in order to obtain subjective satisfaction data. The questionnaire was a modified version of the Questionnaire for User Interaction Satisfaction (QUIS) (Chin *et al.* 1988). Two variants of the questionnaire were created – one for the participants who used the adaptive version of SBO and one for the participants who used the non-adaptive version of SBO. The difference in the two questionnaires was the inclusion of an extra section on adaptation, for those participants who used the adaptive version of SBO (Appendix J).

7.3.5 Participant Selection

Twenty-two participants were used in this experiment. These participants were selected from a third-year ERP systems module offered by the Department of Computing Sciences, at the Nelson Mandela Metropolitan University (NMMU) in Port Elizabeth, South Africa. These particular students were selected, as they fitted the profile of a typical user of an ERP system and because they met the requirements outlined in Section 7.3.1. Currently, the students are exposed to an SAP R/3 Internet Demonstration and Evaluation System (IDES), on a weekly basis, in order to complete their practical assignments.

The sample of students comprised eleven third-year students and eleven Honours students, of varying academic strengths. To divide the sample into two equal groups, a stratified sampling approach was used (Barnett 2003; Watkins *et al.* 2010). Two non-overlapping groups (one for the adaptive version of SBO and one for the non-adaptive version of SBO) were created based on the class marks of the students. Permission to make use of the class marks was obtained from the Deputy Vice Chancellor: Academic of the NMMU, the Head of Department (Department of Computing Sciences), the lecturer of the module and from the students themselves.

Ethical clearance for this study was obtained from the NMMU REC-H committee (Ref no. H11-Sci-CS-003).

7.3.6 Environment and Equipment

The evaluation was conducted in the usability lab at the Department of Computing Sciences at the NMMU. A controlled environment was used to conduct the experiment, so that the users' performance and behaviour could be observed (without influencing the outcomes of the evaluation) from an observer room (fitted with a one-way glass).

Equipment used for the evaluation was a laptop computer running SBO, with Microsoft Visual Studio 2010. The adaptive version of SBO was modified, to provide support for content, presentation, and navigation adaptation. Content adaptation was supported through the implementation of the Inserting and Removing Fragments adaptation technique. This adaptation technique was applied to the list of buyers (Figure 6.4). The enabling and disabling of user interface elements (controls) was implemented to support presentation adaptation by means of providing guidance. This adaptation technique was implemented for all of the controls and interacted with the task model in order to determine which controls should be enabled, and which should be disabled for a particular activity of the task (Section 6.5). The adaptation technique of Colour-Coded controls was implemented to facilitate navigation adaptation. Implementation of this adaptation technique assisted in improving guidance and learnability.

7.3.7 Experimental Design

A between-subjects design was used for the evaluation. This assisted in identifying any differences between the two treatment groups (adaptive and non-adaptive versions of SBO) in terms of the metrics that were specified in Section 7.3.3 (Watkins *et al.* 2010). Two groups, each containing 11 participants were created: one group was exposed to the adaptive version of SBO and the other group to the non-adaptive version of SBO.

7.3.8 Statistical Analysis

Descriptive and inferential statistics were used to analyse the performance and satisfaction data that were collected.

7.3.8.1 Descriptive Statistics

Descriptive statistics were used to summarise and describe the data gathered over the two-day evaluation period. The use of descriptive statistics assisted in providing a comparison between the various samples means: the adaptive and the non-adaptive version of SBO, and between day one and day two.

7.3.8.2 Inferential Statistics

Non-parametric inferential statistics were used because of the small sample size (n=22). The inferential statistics assisted in determining the statistical and practical significance of any inferences made with regard to the two versions of SBO. All of the inferential statistics used in the analysis were calculated at the five-percent level of significance. The non-parametric statistics used in the evaluation are discussed in this section.

a) Chi-Square Test of Independence

The chi-square test of independence was used to determine whether any association existed between the adaptive and the non-adaptive version of SBO with regard to a specific independent variable (Watkins *et al.* 2010).

b) Wilcoxon-Mann-Whitney U-Test

The Wilcoxon-Mann-Whitney U-Test was used to determine whether a statistically significant difference existed between the adaptive and the non-adaptive version of SBO with regard to a specific independent variable (Sprent and Smeeton 2007; Witte and Witte 2009).

c) Wilcoxon Matched Pairs T-Test

The Wilcoxon Matched-Pairs T-Test was used to determine whether a statistically significant difference existed for a particular version of SBO (adaptive or non-adaptive) over the two-day evaluation period (Witte and Witte 2009).

d) Cohen's d-test

The Cohen's d-test was used as a measure of practical significance for any significant difference that was identified from the *Wilcoxon-Mann-Whitney U-Test* or the *Wilcoxon Matched-Pairs T-Test*. To determine the effect size of the difference, the following scales for d, were used (Witte and Witte 2009):

- Effect size is *small*, if d is in the vicinity of 0.2;
- Effect size is *medium*, if d is in the vicinity of 0.5; and
- Effect size is *large*, if d is in the vicinity of 0.8.

e) Cronbach's Alpha (∝)

Cronbach's ∝ was used to assess the reliability of the scales used in the post-test satisfaction questionnaires (Kottner and Streiner 2010). The threshold value for ∝ is 0.7. Values larger than 0.7 indicate that the scale used is reliable. Values that are less than 0.7 indicate that there is not enough evidence to conclude that the scale is reliable. These cases are often the result of too few items being assessed, or that the participants have provided high response values for these items. Such cases require that these items be assessed individually.

7.3.9 Tasks

Over the two-day evaluation period, the participants were required to complete a series of tasks. These tasks related to processing a purchase order. Two task lists were provided – one on day one, and the other on day two of the evaluation. Copies of the task list are provided in Appendix I. Participants from both samples were exposed to the same set of task lists, due to the nature of the experiment (between-subjects design). This was necessary in order to evaluate for any improvements in efficiency, learnability, and satisfaction.

The following is an example of the tasks that needed to be completed:

Jane Smith has requested that a purchase order be made out for the purchase of 10 bicycles from Totalsports.com.

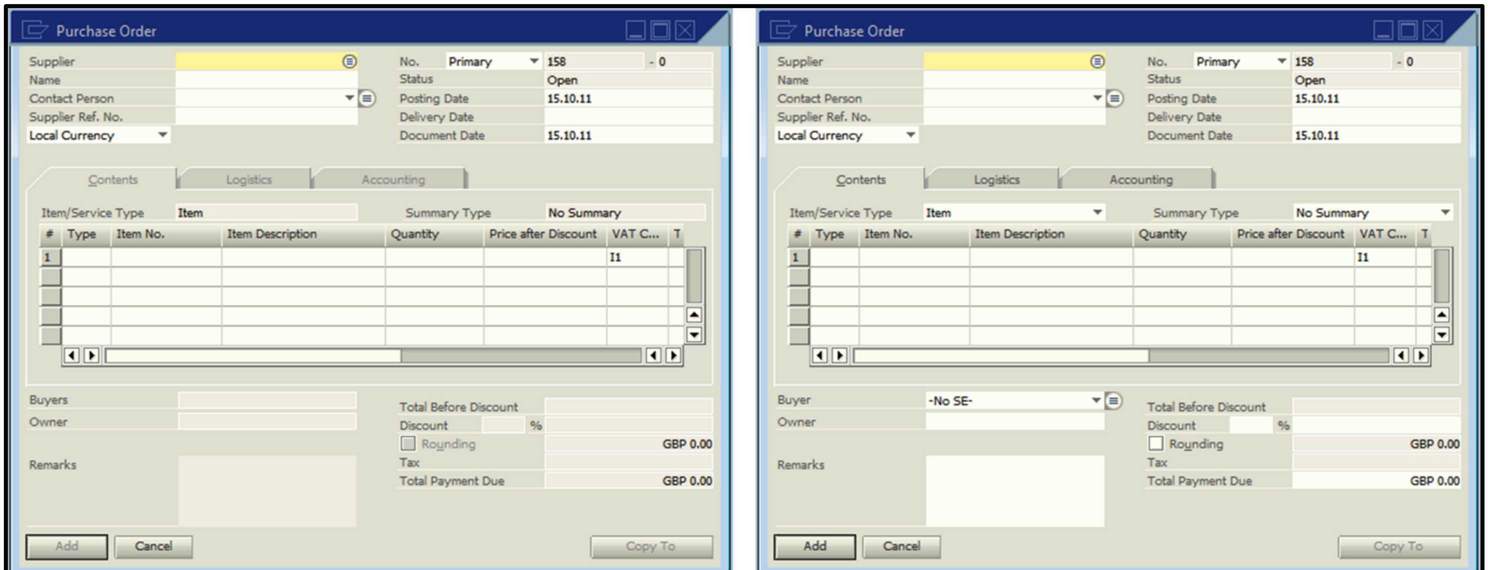
Successfully completing the above task requires that the following activities be completed:

1) Selecting a supplier (Figure 7.3);

2) Selecting an item and entering an item quantity (Figure 7.4); and

3) Selecting a buyer (Figure 7.5).

Figure 7.2 illustrates a key difference between the adaptive version of SBO, as opposed to the non-adaptive version of SBO. In the adaptive version of SBO, certain sections of the user interface are disabled (when the form is loaded) to guide the user through the correct sequence of activities when completing a task. The non-adaptive version of SBO does not provide this level of guidance, as all of the controls on the form are already enabled.

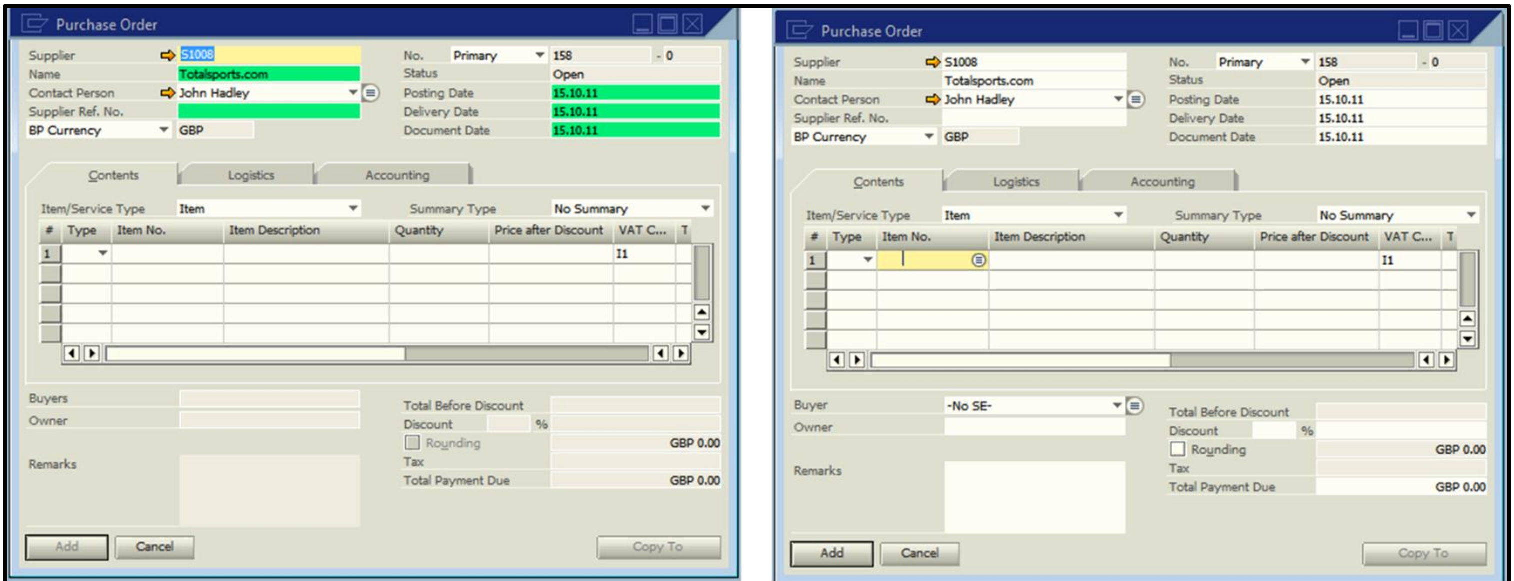


(a) Adaptive SBO (b) Non-adaptive SBO

Figure 7.2: User Interface Differences

Figure 7.3 illustrates the differences in the user interface of the two systems with regard to selecting a supplier. Once a supplier has been selected, the adaptive version of SBO highlights the associated controls in green, to show that these controls have been populated with the correct data. The non-adaptive version of SBO, however, does not support this, hence not providing the user with any immediate indication of whether the appropriate controls have been populated with the correct data.

Once the correct data have been entered, the adaptive version of SBO then enables the set of controls relating to the next activity, which in this case comprises the controls for selecting an item.

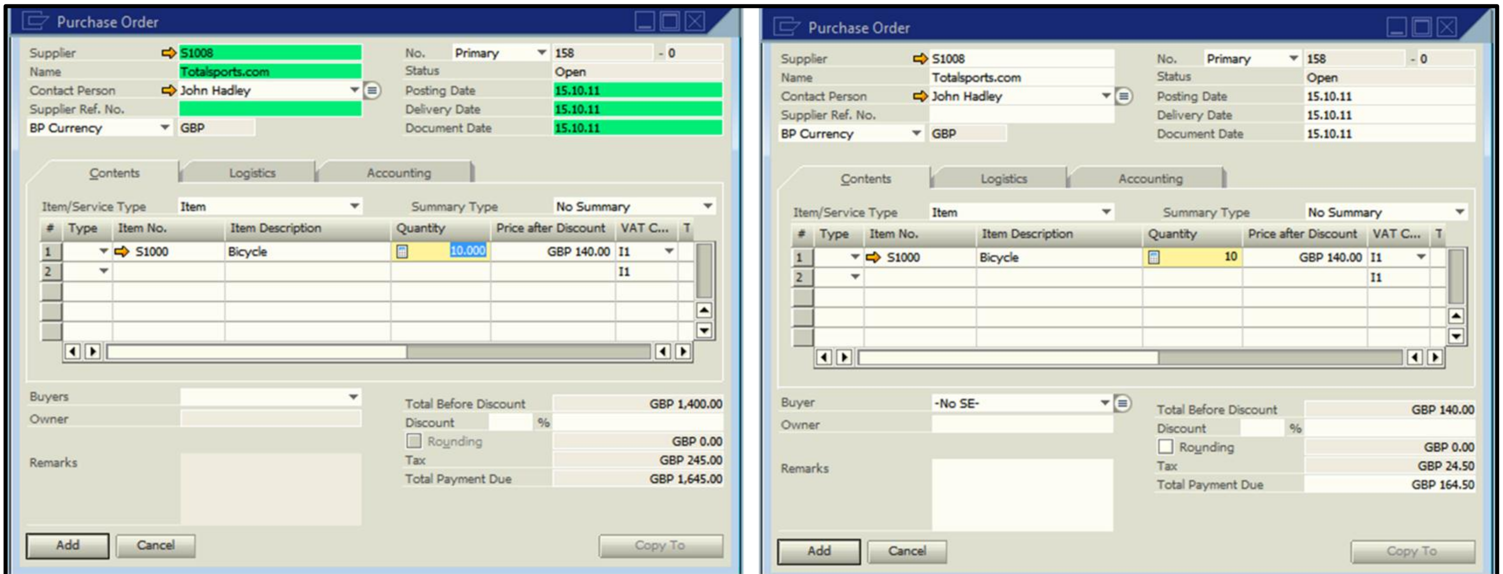


(a) Adaptive SBO (b) Non-adaptive SBO

Figure 7.3: Selecting a Supplier Activity

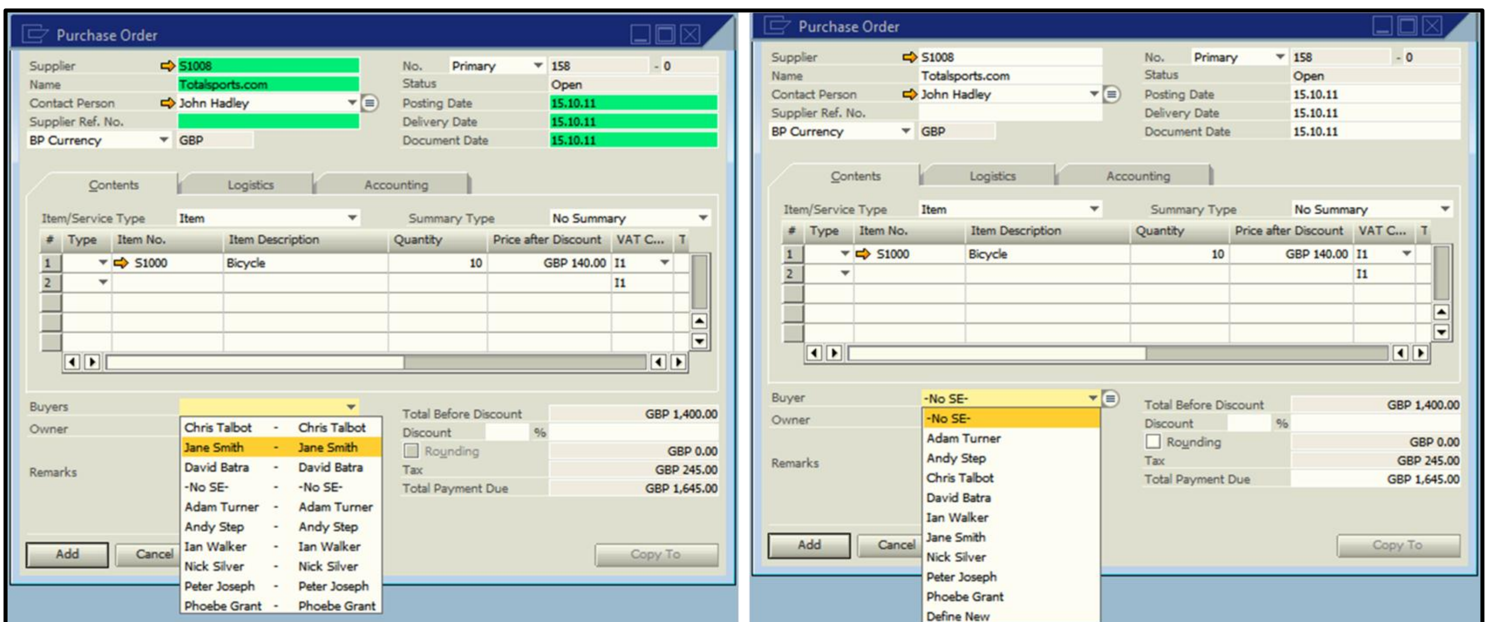
Selecting an item is the same in both versions of SBO (Figure 7.4). The only difference is that once an item has been selected, and the appropriate quantity entered, the adaptive version of SBO then enables the next set of controls – which in this case comprises those controls relating to the buyer.

The process of selecting a buyer differs between the two versions of SBO (Figure 7.5). In the adaptive version of SBO, the most recently used (MRU) buyer is placed at the top of the buyer list and is then followed by the two most frequently used (MFU) buyers. This is different from the non-adaptive version of SBO, where the list of potential buyers is simply sorted alphabetically - with no intelligence behind the sorting.



(a) Adaptive SBO (b) Non-adaptive SBO

Figure 7.4: Selecting an Item



(a) Adaptive SBO (b) Non-adaptive SBO

Figure 7.5: Selecting a Buyer

7.3.10 Procedure

The empirical evaluation was conducted over a period of two days. This was necessary in order to test for learnability. The procedure that followed on each day is outlined and discussed in this section.

Day 1

The participants were briefed on the purpose of the experiment and their role in the experiment. On agreement to participate, the participants were provided with an informed consent form to read and sign. This was followed by the participant completing the pre-test biographical questionnaire. Next, the participants were provided with the task list for day one, and were asked to complete all eight tasks in a sequential manner. The participants were requested to indicate when all eight tasks were completed. The participants were also asked to use the think-aloud protocol during the experiment, so that additional notes could be recorded, to assist in determining the participant's mental model when completing the tasks (Tullis and Albert 2008). On completion of the eight tasks, the participants were provided with a post-test satisfaction questionnaire (specific to the version of SBO used) to complete. Lastly, a follow-up session for day two of the evaluation was scheduled.

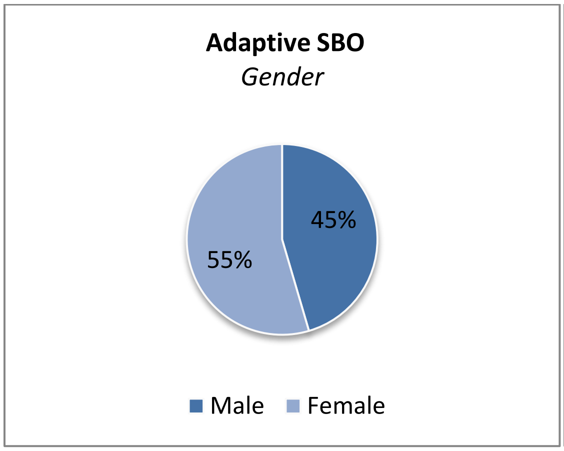
Day 2

The participants were reminded of the purpose of the experiment and their role. Once this was completed, the participants were issued the task list for day two and were asked to complete all eight tasks in a sequential manner. The participants were then requested to indicate when all eight tasks had been completed. As with day one, the participants were asked to use the think-aloud protocol during the experiment (Tullis and Albert 2008). On completion of the eight tasks the participants were provided with a post-test satisfaction questionnaire (specific to the version of SBO used) to complete. Once the session for day two was completed the participants were thanked for being a part of the experiment.

7.4 Results

7.4.1 Biographical Data

A total of twenty-two participants were used in this evaluation. Fifty percent of the participants were males and fifty percent were females. Figure 7.6 illustrates the composition of males and females, with regard to the version of SBO used.



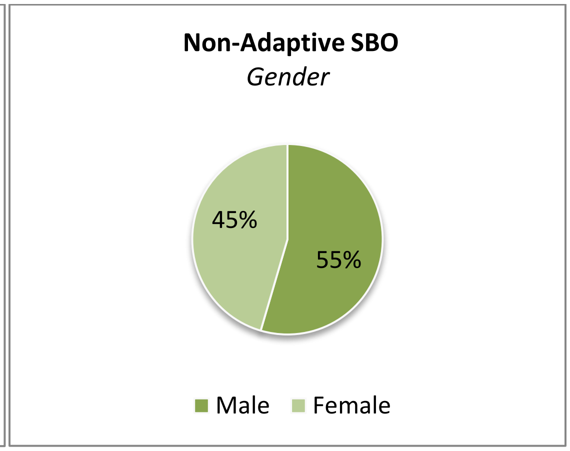
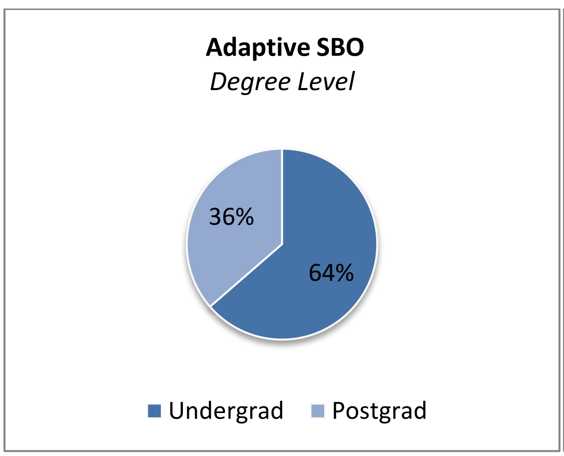


Figure 7.6: Gender Split per System (n=22)

Fifty-nine percent of the participants were enrolled for an undergraduate degree and forty-one percent were enrolled for a post-graduate degree. Figure 7.7 illustrates the composition of participants with regard to their degree and the version of SBO used.



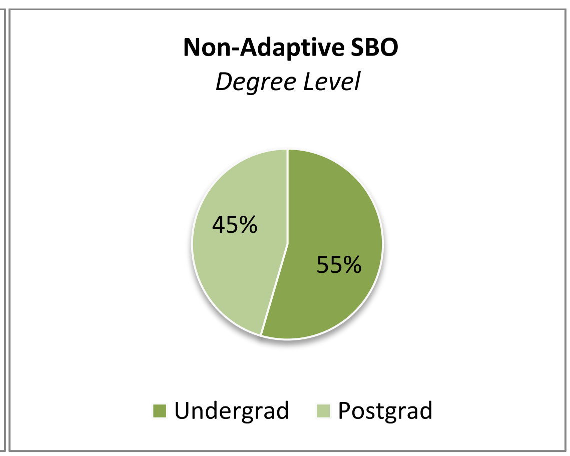


Figure 7.7: Degree Split per System (n=22)

Based on the responses from the participants, 50% felt that their previous exposure to an ERP system classified them as novice ERP users, whilst the remaining 50% felt that they fitted into the intermediate category. Figure 7.8 illustrates the composition of ERP skills with regard to the version of SBO used.

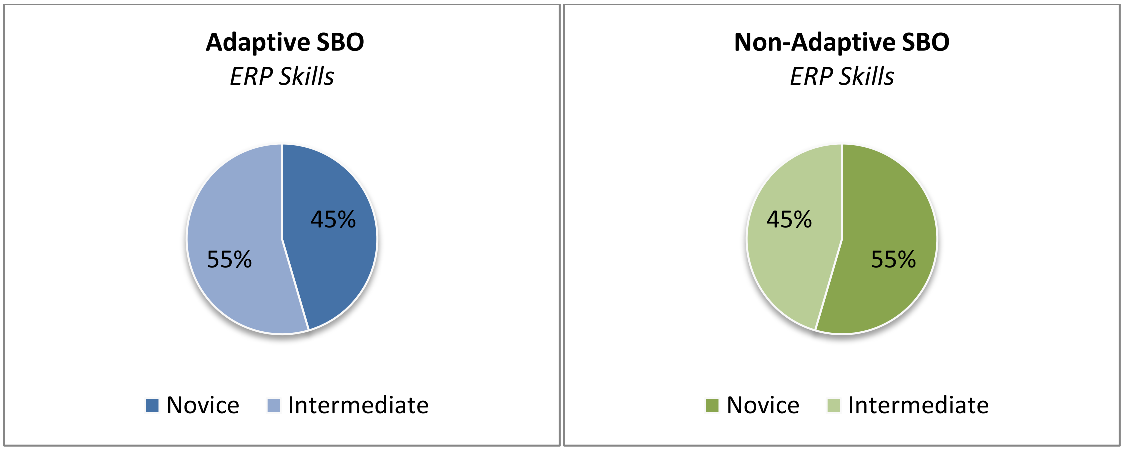


Figure 7.8: ERP Skills Split per System (n=22)

In terms of ERP usage, 91% of the participants stated that they used an ERP system occasionally, whilst 9% were of the opinion that they used an ERP system frequently. Figure 7.9 illustrates the composition of ERP usage with regard to the version of SBO used.

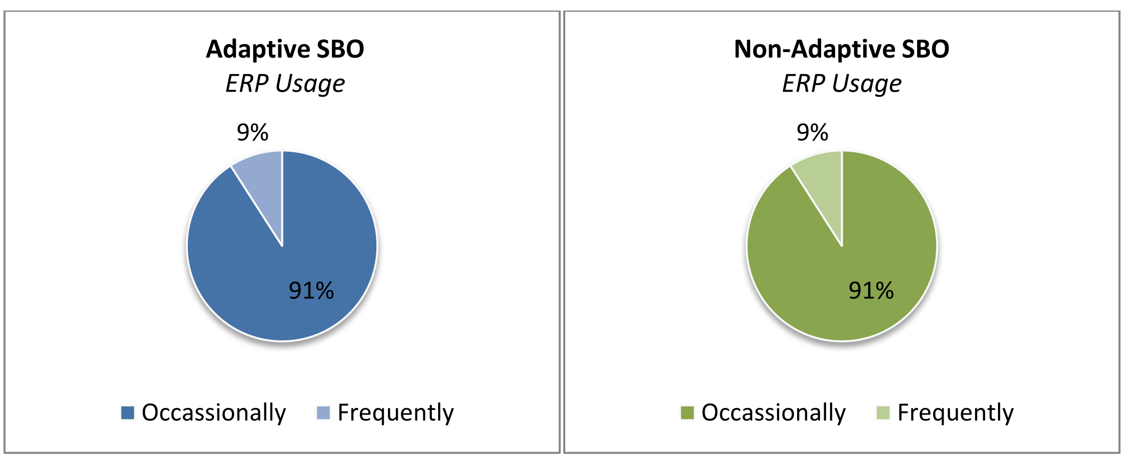


Figure 7.9: ERP Usage Split per System (n=22)

The results in Figure 7.5 illustrate an even distribution amongst the participants (between the two groups) with regard to their ERP usage. Based on these results, it may be assumed that the two groups have used ERP systems for the same, or for a similar length of time.

Eighty-two percent of the participants stated that they had used an ERP system less than five hours a week, whilst 18% of the participants stated that they had used an ERP system between five and ten hours a week. Figure 7.10 illustrates the composition of weekly ERP usage with regard to the version of SBO used.

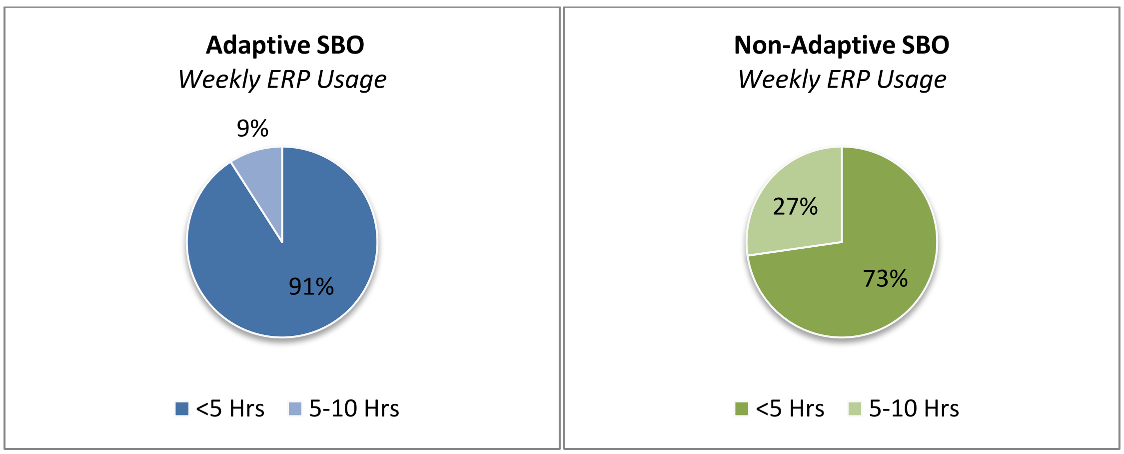


Figure 7.10: Weekly ERP Split per System (n=22)

A Chi-square test of independence was conducted to examine the relation between the various categories in the biographical questionnaire and the version of SBO used. The results from this test (Appendix K) revealed that no association exists between the two versions of SBO and each category from the biographical questionnaire, as all of the p-values were greater than the five percent level of significance.

Furthermore, the results also revealed that no significant differences exist between the two groups, and that the participants were placed in independent groups with no overlap. This indicates that the process of stratified sampling was conducted accurately.

7.4.2 Performance Results

The use of performance metrics assisted in comparing the efficiency and determining the learnability of the adaptive and the non-adaptive versions of SBO. The performance data comprised the following metrics: *Task Success, Time-on-Task, Number of Mouse Clicks and Errors Committed*. These data were automatically logged and stored in a log file for each user.

7.4.2.1 Task Success

Task success was calculated to determine whether a participant successfully completed a task. This was determined by the sequence of activities that the participant had followed. Task success was coded as a 1, if the participant had successfully completed a task and as a 0 if the participant had not successfully completed a task.

Figure 7.11 compares the means for task success for all eight tasks performed on day one and on day two of the evaluation. This is illustrated with regard to the adaptive and the non-adaptive versions of SBO.

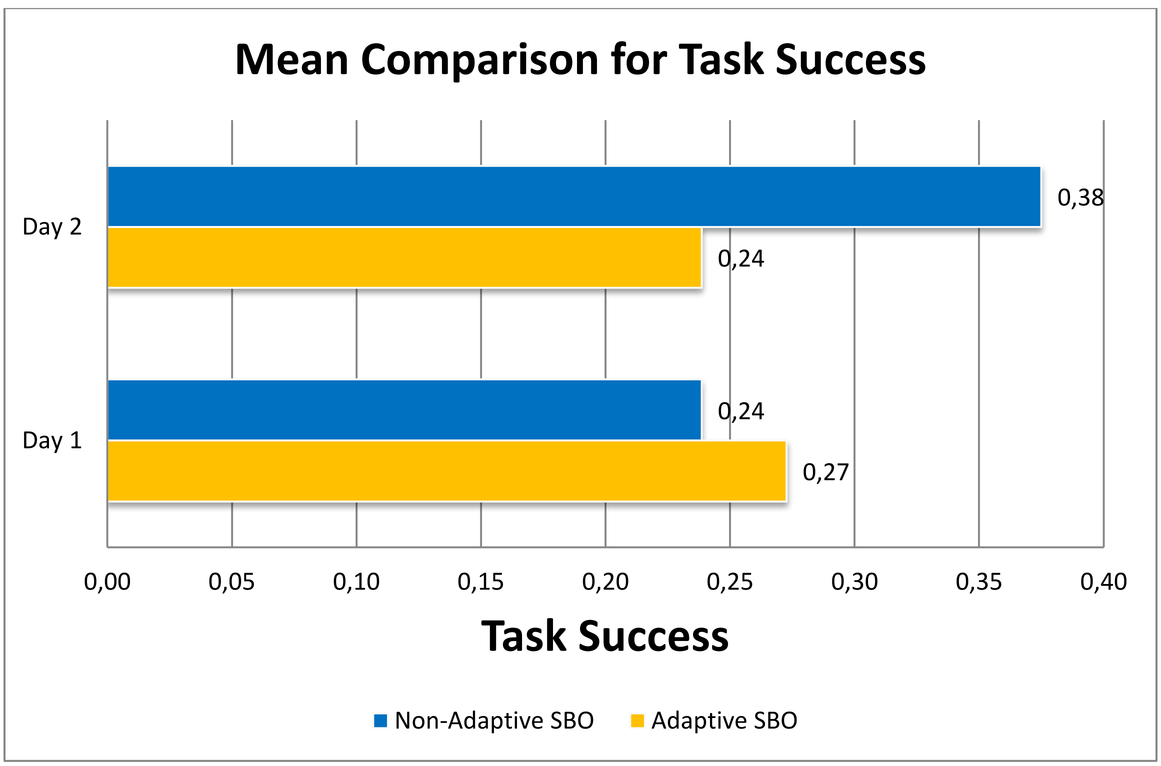


Figure 7.11: Mean Comparison for Task Success (n =22)

Based on the mean values for day one (Figure 7.11), the adaptive version of SBO had a marginally better (0.03) task success rate. This, however, differed on day two where the non-adaptive version of SBO had a greater (0.14) task success rate. The results from the Wilcoxon

Matched-Pairs T-Test (Appendix K) showed that the difference in each system, over the two-day evaluation period, was not significant (both p-values were greater than the 5% level of significance). This implies that although the task success rate declined for the adaptive version of SBO by 0.03 (over the two-day evaluation period), the decline could not be regarded as statistically significant. The same applies for the improvement in task success for the non-adaptive version of SBO.

The results from the Chi-square test of independence for task success (Appendix K) indicate that no association existed between the task success rate and the version of SBO system used (p-values for both days were greater than the 5% level of significance). This result was confirmed by the Wilcoxon-Mann-Whitney U-Test (Appendix K), where all of the p-values for task success were greater than the 5% level of significance.

7.4.2.2 Time-on-Task

Time-on-task was calculated to determine the amount of time taken for participants to complete a task. This was calculated, based on the starting time and the ending time for each task. Starting time was recorded when the participant opened the Purchase Order form. Ending time was recorded when the participant closed the Purchase Order form. Time-on-task values were stored in the log file for each task performed by the participant over the two-day evaluation period. A program was developed which parsed each log file to determine the total time (seconds) spent on each task.

Figure 7.12 compares the means for time-on-task for all eight tasks, performed on both days of the evaluation. Based on the results in Figure 7.12, the tasks completed on the adaptive version of SBO, on the first day, were completed in a shorter amount of time (5.52 seconds). The time in which tasks were completed for the second day was marginally better on the non-adaptive version of SBO by 0.09 seconds.

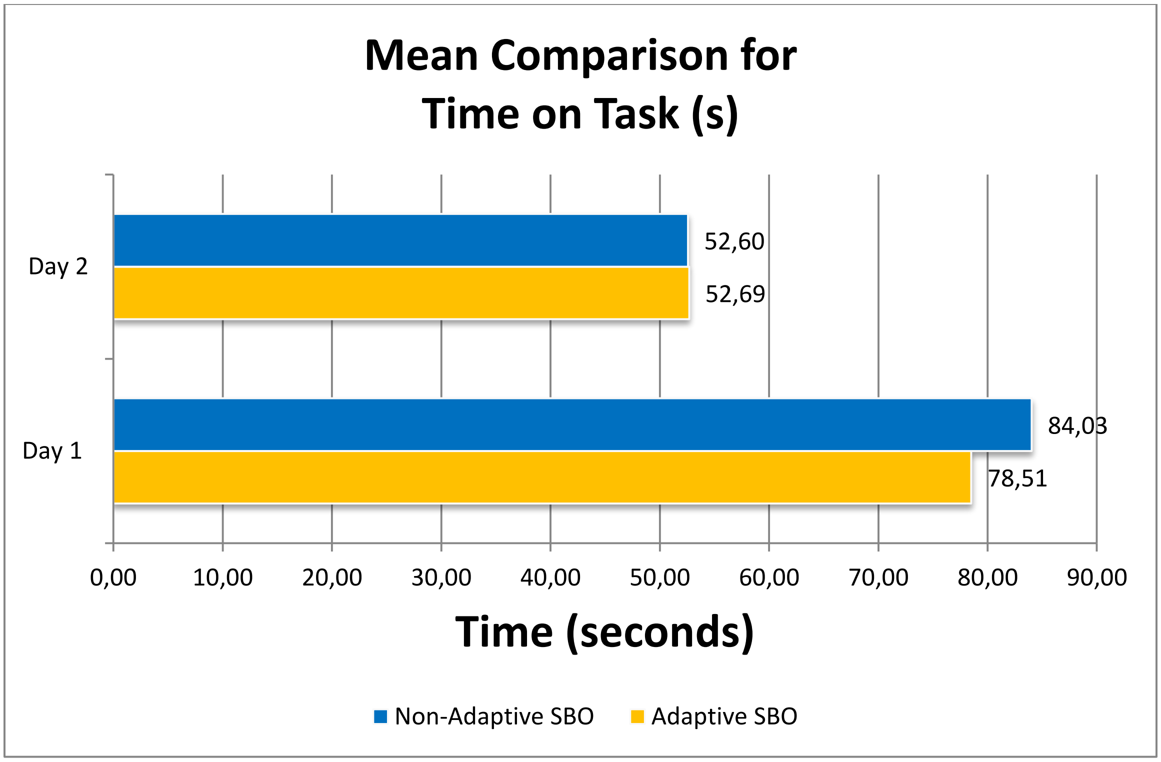


Figure 7.12: Mean Comparison for Time-on-Task (n=22)

Results from the Wilcoxon Matched-Pairs T-Test (Appendix K) indicated that there was a significant improvement in the task completion times for both versions of SBO over the two-day evaluation period (p-values were less than 5% level of significance). The Cohen's d value for the adaptive version of SBO was 1.60, and 1.67 for the non-adaptive version of SBO. This indicates that there was a large improvement in the task completion times for both versions of SBO, with the non-AUI ERP version having a marginally bigger improvement by 0.07.

The results from the Chi-square test of independence for time-on-task (Appendix K) revealed that no association existed between the time it took to complete a task and the version of SBO used (p-values for both days were greater than the 5% level of significance). This was confirmed by the results of the Wilcoxon-Mann-Whitney U-Test (Appendix K), where the p-values were also greater than the 5% level of significance.

7.4.2.3 Mouse Clicks

The number of mouse clicks was calculated to contribute to the measure of efficiency. Mouse clicks were captured and stored in the log file (for each participant), whenever a participant clicked on a control.

Figure 7.13 compares the means for the number of mouse clicks for all eight tasks on each user interface. The results for the first day showed that the participants who used the adaptive version of SBO used fewer mouse clicks in completing a task. This, however, differed on the second day, where the participants who used the non-adaptive version of SBO used fewer mouse clicks to complete a task.

The results from the Wilcoxon Matched-Pairs T-Test for mouse clicks (Appendix K) indicate that the increase in the number of mouse clicks for the adaptive version of SBO was not significant (p-value was greater than the 5% level of significance). These results were different for the non-adaptive version of SBO, where the decrease in the number of mouse clicks could be regarded as significant (p-value was less than the 5% level of significance). The Cohen's d value (0.91) indicated that there was a significantly larger improvement in the number of mouse clicks over the two-day evaluation period for the non-adaptive version of SBO.

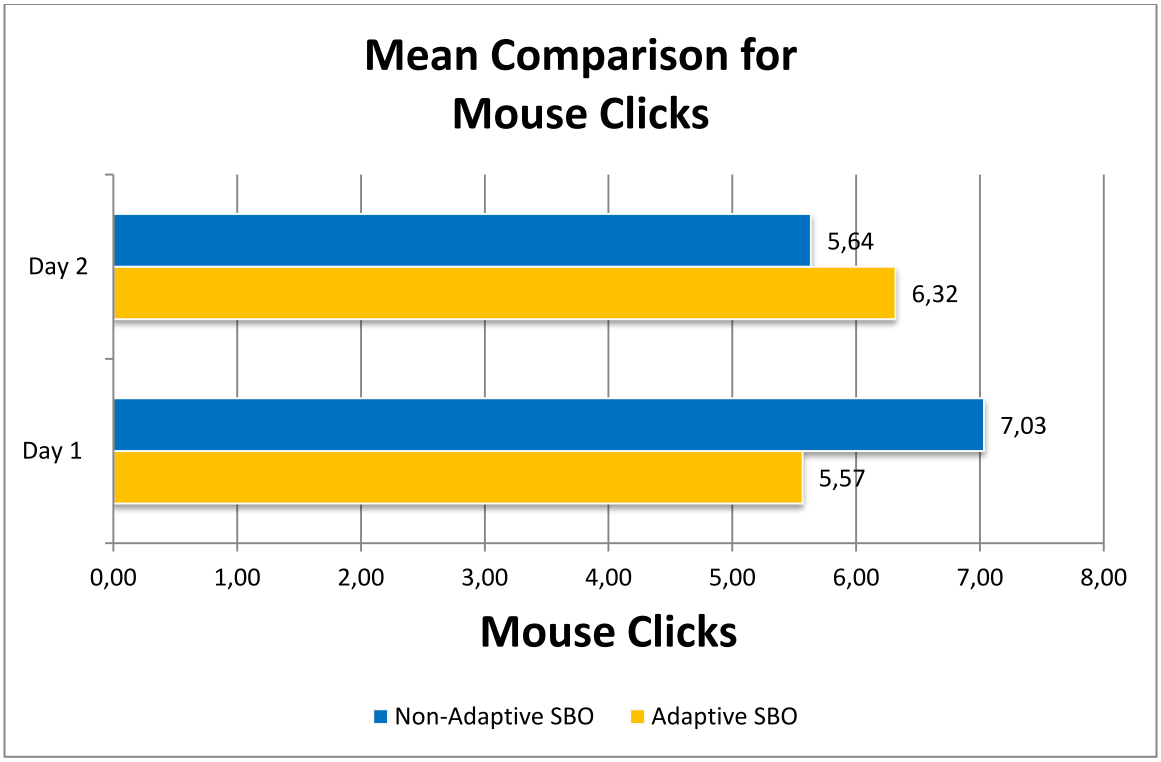


Figure 7.13: Mean Comparison for Mouse Clicks - Day 1 and Day 2 (n=22)

The results from the Chi-square test of independence (Appendix K) showed that there was an association between the number of mouse clicks and the version of SBO used over the two-day evaluation period (p-values were less than the 5% level of significance). These results were confirmed by the Mann-Whitney U-Test (Appendix K) (p-value was less than the 5% level of significance). The Cohen's d statistic (1.13) revealed that a large association existed between the number of mouse clicks and the version of SBO used. It may thus be inferred that the significant improvement in the number of mouse clicks on the non-adaptive version of SBO could be attributed to the association.

7.4.2.4 Errors Committed

The number of errors committed was calculated, as a contribution to the measure of efficiency. Determining whether or not a participant committed an error was based on the number of incorrect selections made on the Purchase Order form. The sequence in which a participant selected a control was also taken into account, when determining whether an error had been committed or not.

Figure 7.14 compares the means for the number of errors committed for all eight tasks on each version of SBO for each day. The mean values in Figure 7.14 indicate that more errors were made on the adaptive version of SBO on both days.

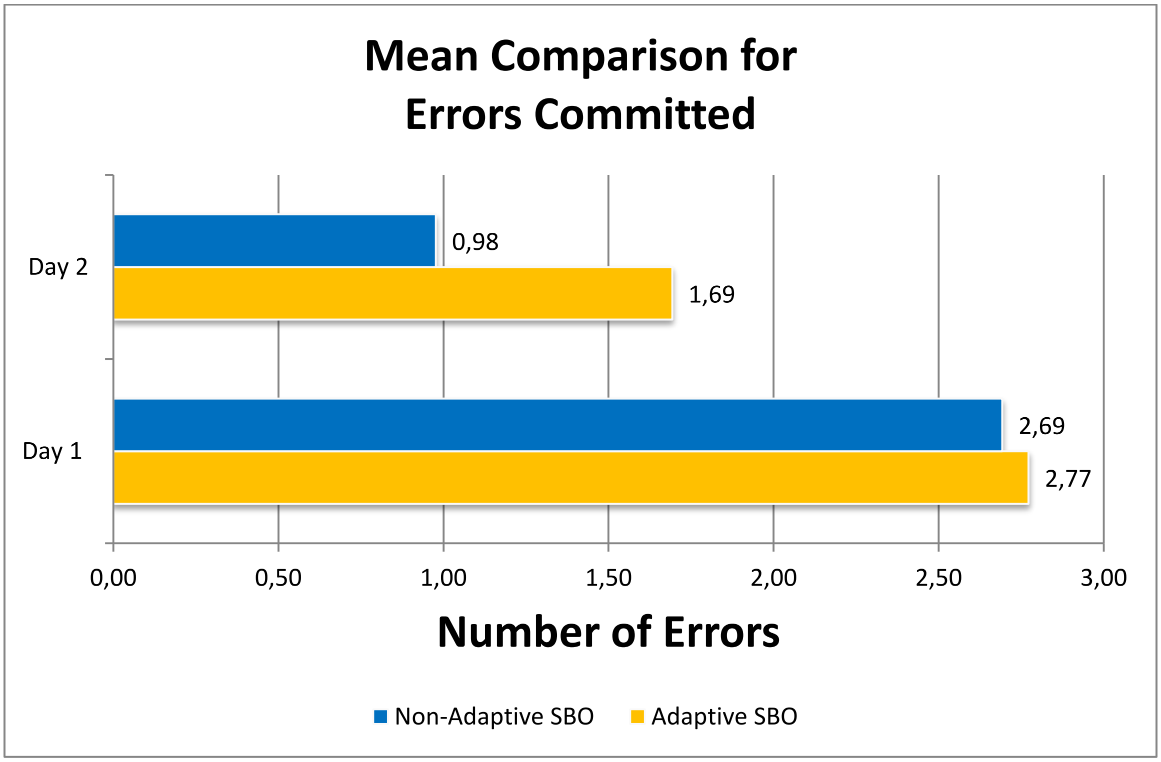


Figure 7.14: Mean Comparison for Errors Committed (n=22)

The results from the Wilcoxon Matched-Pairs T-Test (Appendix K) indicate that the reduction in the number of errors made on the adaptive version of SBO was not significant (p-value was greater than the 5% level of significance). The test did, however, reveal that the decrease in the number of errors made, from day one to day two, on the non-adaptive version of SBO was statistically significant (p=0.008, and is less than the 5% level of significance). The Cohen's d value (1.19) indicated that this was a significantly large improvement. These results were confirmed by the results from the Mann-Whitney U-Test (Appendix K), which also revealed a significant association between the number of errors committed and the type of system used on day two of the evaluation (p=0.034 was less than the 5% level of significance).

The Cohen's d value (1.15) indicated that the association on the second day between the number of errors and the version of SBO used was significantly large. Based on the results of the Wilcoxon Matched-Pairs T-Test, it becomes evident that this association was related to the significant decrease in the number of errors made on the non-adaptive version of SBO on day two of the evaluation.

7.4.3 Satisfaction Metrics

The aim of the satisfaction metrics was to obtain subjective responses from the participants to support the performance data. Two post-test questionnaires were used to obtain the subjective data – one for those participants who used the adaptive version of SBO and the other for those participants who used the non-adaptive version of SBO.

A Cronbach's $\propto$ test was conducted on the data obtained to assess the reliability of the scales used in the questionnaires. The results of the test (Appendix K) indicate that the scales used for the following sections: overall satisfaction, task support and efficiency and learnability, were reliable. The low $\propto$ – values obtained for adaptivity indicate that there is not enough evidence to consider the scale used as being reliable, and that the items in this category should be discussed individually.

Satisfaction results, based on the subjective data, will be discussed in the sections that follow.

7.4.3.1 Overall Satisfaction

The aim of the *Overall* section in the post-test questionnaire was to obtain a generalised overview from the participants of the version of SBO which they used. The overall section evaluated items relating to the overall reaction to the system, screen design and layout, task support and efficiency, learnability and navigation.

Figure 7.15 compares the means response of all the participants on day one of the evaluation. The mean responses are illustrated per item and per version of SBO. The results show that the participants who used the adaptive version of SBO rated it more satisfying than the participants who used the non-adaptive version of SBO. The greatest difference in satisfaction was with regard to overall task support. The participants who used the adaptive version of SBO felt that they had received a greater level of task support, as opposed to those participants who had used the non-adaptive version of SBO.

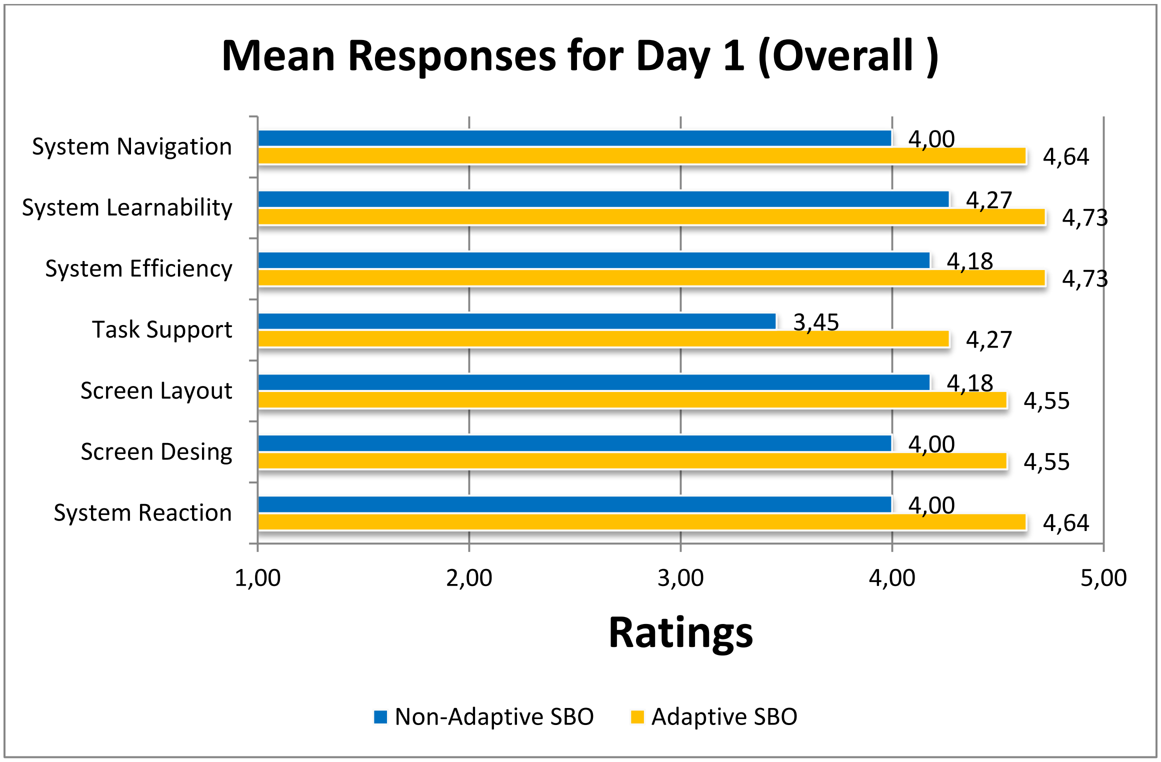


Figure 7.15: Comparison of Overall Satisfaction for Day 1 (Overall) (n=22)

Figure 7.16 compares the overall means response for day two. The mean responses on day two show an improvement over day one. The greatest difference was in terms of the task support provided by the two versions of SBO. Participants who had used the adaptive version of SBO were more satisfied than the participants who had used the non-adaptive version of SBO.

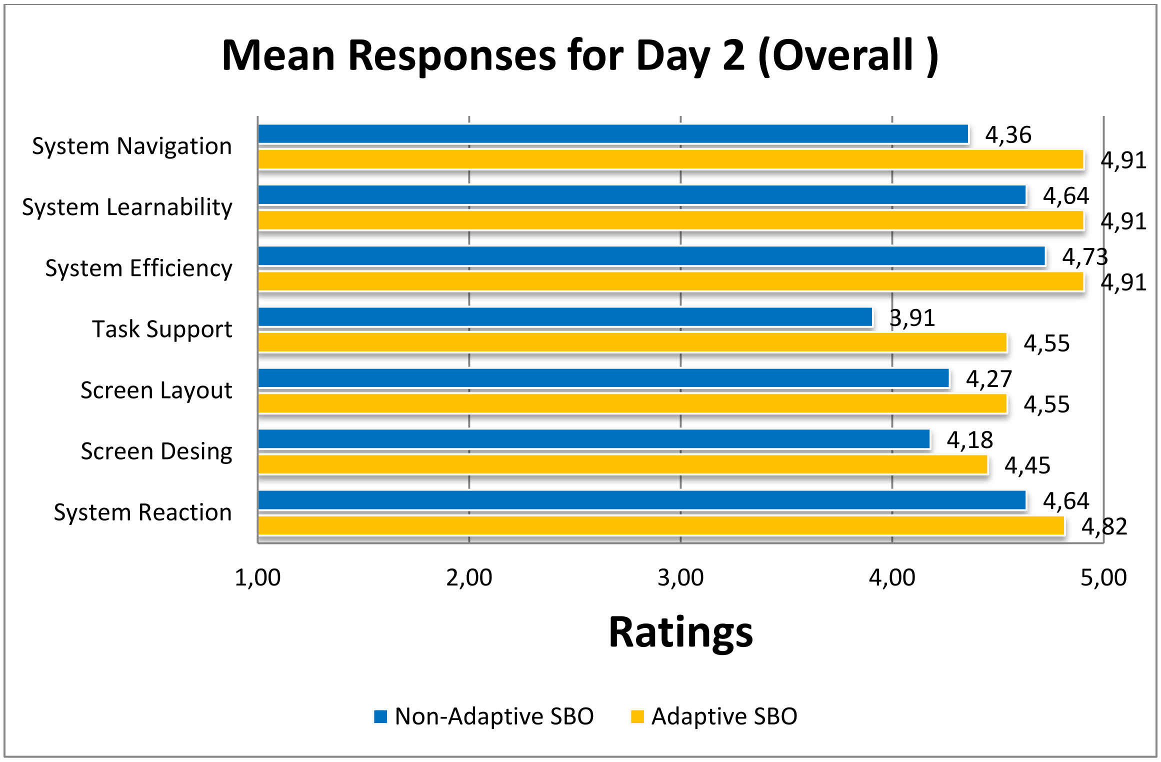


Figure 7.16: Comparison of Overall Satisfaction for Day 2 (n=22)

The results from the Wilcoxon-Matched Pairs T-Test for overall satisfaction (Appendix K) indicated that there were no significant improvements in overall satisfaction for both versions of SBO over the two-day evaluation period (p-values were greater than the 5% level of significance).

The results from the Mann-Whitney U-Test (Appendix K) revealed that no association existed between the overall satisfaction and the version of SBO used (p-values were greater than the 5% level of significance).

7.4.3.2 Task Support and Efficiency

The aim of the Task Support and Efficiency section of the post-test questionnaire was to determine how satisfied the participants were with their task completion and system response time.

Figure 7.17 illustrates the mean responses for day one for Task Support and Efficiency. These responses show that the participants who used the adaptive version of SBO achieved greater levels of satisfaction for task support and efficiency, as opposed to the participants who used the non-adaptive version of SBO. The greatest difference in terms of satisfaction was with regard to finding and selecting a supplier and a buyer. Those participants who used the adaptive version of SBO felt that it was easier to perform this operation. The second area where there was a noticeable difference was in terms of the number of steps required to complete a task and the overall process of completing a task. Participants who had used the adaptive version of SBO found that there were fewer steps required to complete a task, and that it was easier to complete the overall task, as opposed to those participants who used the non-adaptive version of SBO.

The mean responses for day two (Figure 7.18) for task support and efficiency show an improvement over the mean responses for day one. The greatest difference in satisfaction for day two occurred with regard to the number of steps required to complete a task and the overall response time. Participants who used the adaptive version of SBO felt that there were fewer steps required in completing a task, and that the system responded in a short amount of time, as compared with the participants who used the non-adaptive version of SBO.

The results from the Wilcoxon Matched-Pairs T-Test for task support and efficiency (Appendix K) showed that there was no significant difference in the satisfaction levels over the two-day evaluation period for both versions of SBO (p-values were greater than the 5% level of significance).

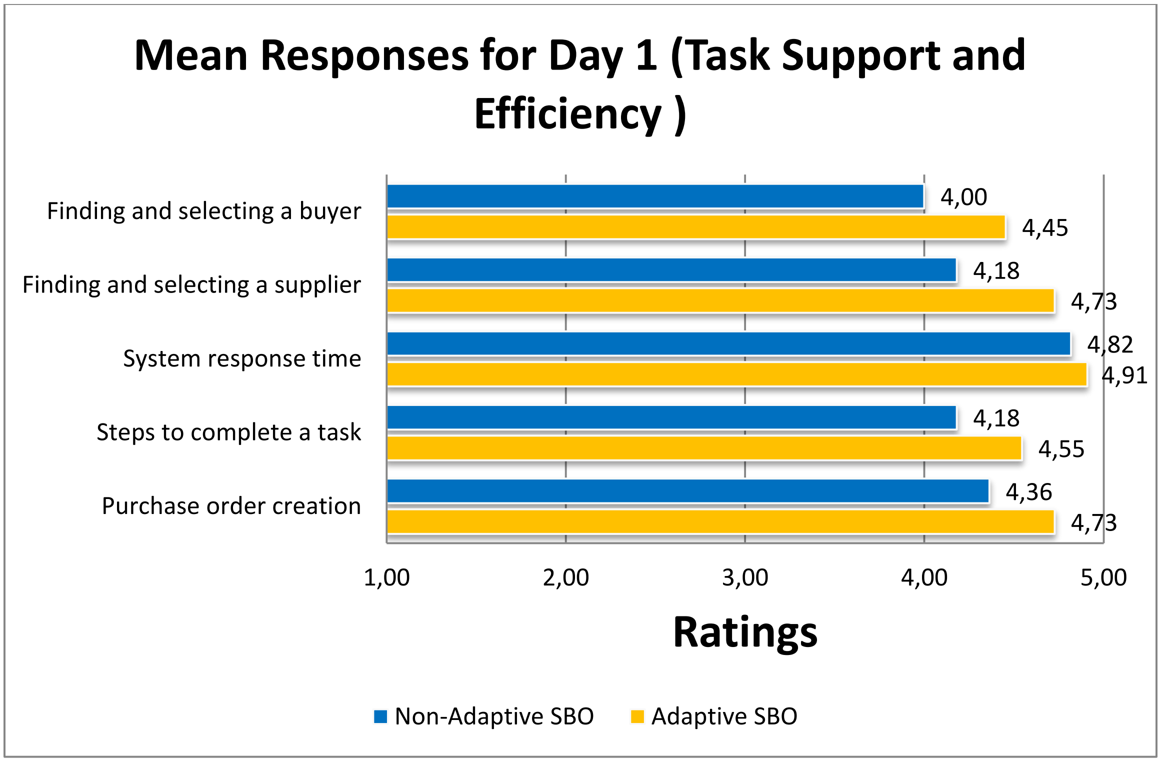


Figure 7.17: Comparison of Task Support and Efficiency for Day 1 (n=22)

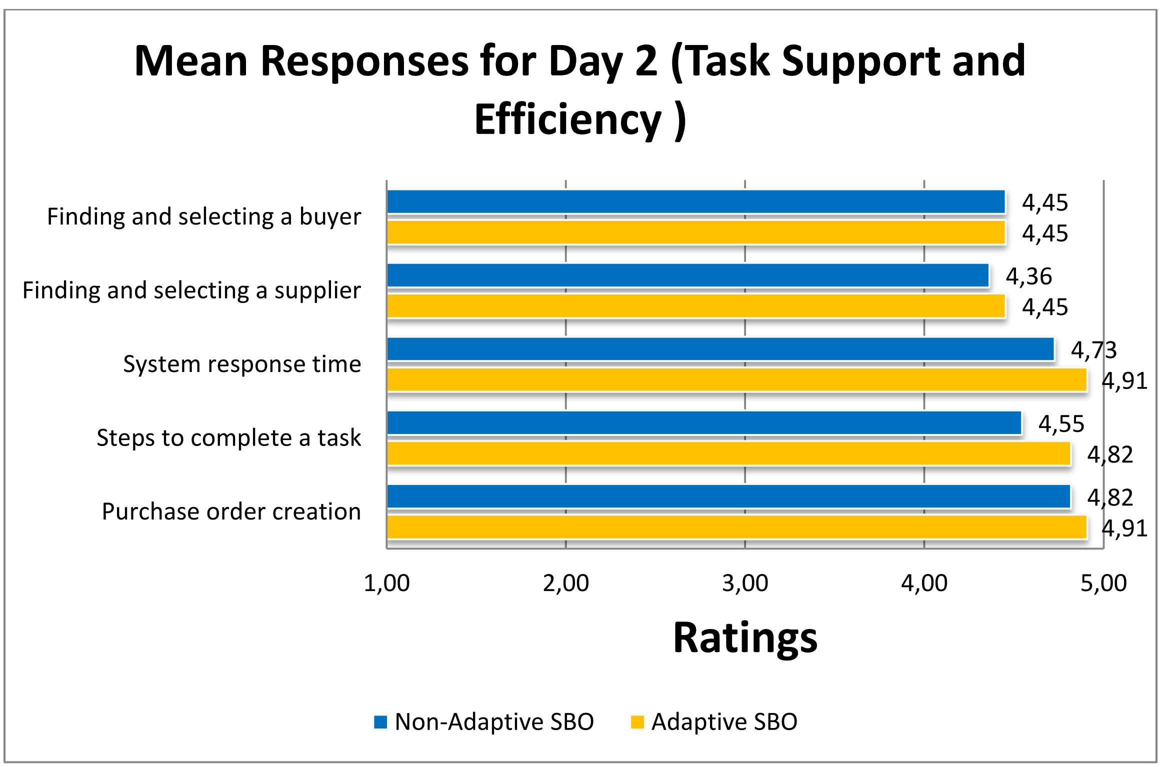


Figure 7.18: Comparison of Task Support and Efficiency for Day 2 (n=22)

The results of the Mann-Whitney U-Test for task support and efficiency (Appendix K) revealed that there was no significant association between the satisfaction levels for task support and efficiency and the version of SBO used (p-values are greater than the 5% level of significance).

7.4.3.3 Learnability

The learnability section of the post-test questionnaire was intended to determine how easy it was to learn and become skilled with SBO. This section also intended to determine how much support was provided by the system to improve learnability and task completion.

On day one (Figure 7.19), the participants (from both groups) felt that they had achieved the same level of satisfaction with regard to the overall learnability. The greatest difference, however, was in terms of the guidance provided by the system. This was followed by the amount of time required to become skilled with the system. Those participants who used the adaptive version of SBO felt that there was a noticeable amount of guidance provided by SBO, which assisted them in their task completion. This shortened the time period that it took to become skilled with SBO, as opposed to the participants who used the non-adaptive version of SBO.

The mean responses for learnability on day two of the evaluation (Figure 7.20) indicate that the participants who used the adaptive version of SBO still experienced a noticeable amount of guidance in terms of task completion. Those participants who used the adaptive version of SBO also felt that SBO was simpler to use on the second day than it had been on the first day of the evaluation.

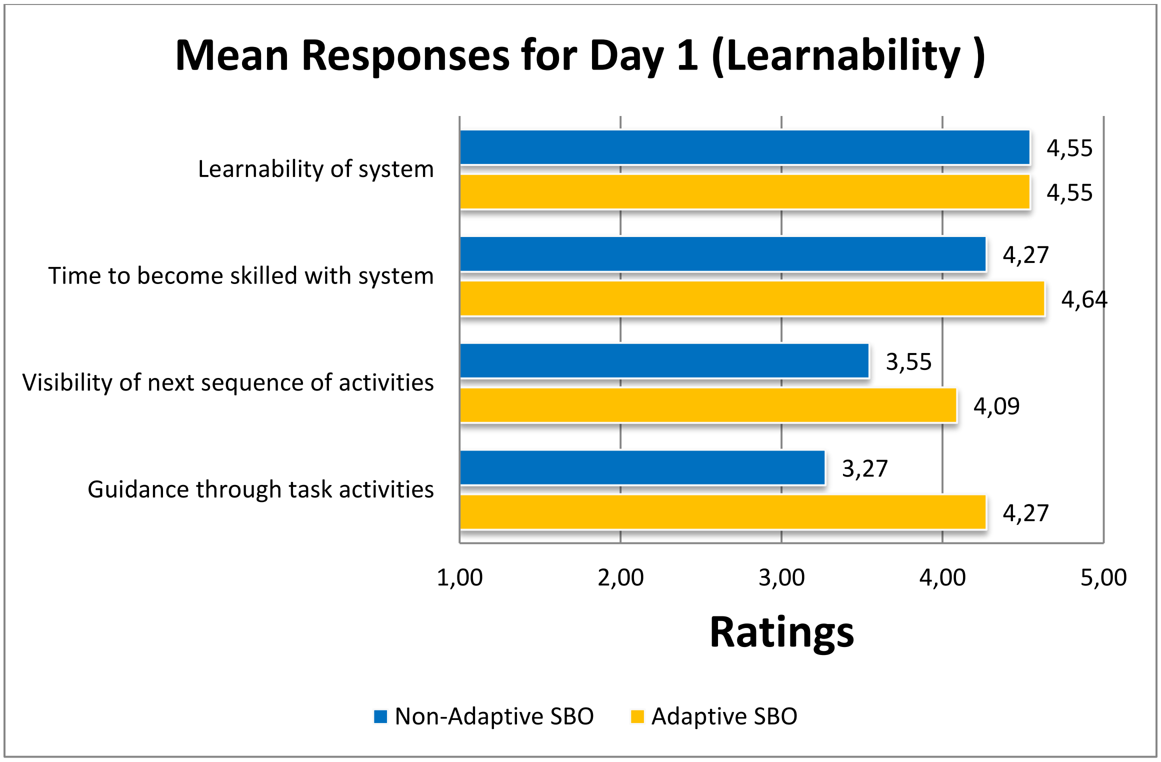


Figure 7.19: Comparison of Learnability for Day 1 (n=22)

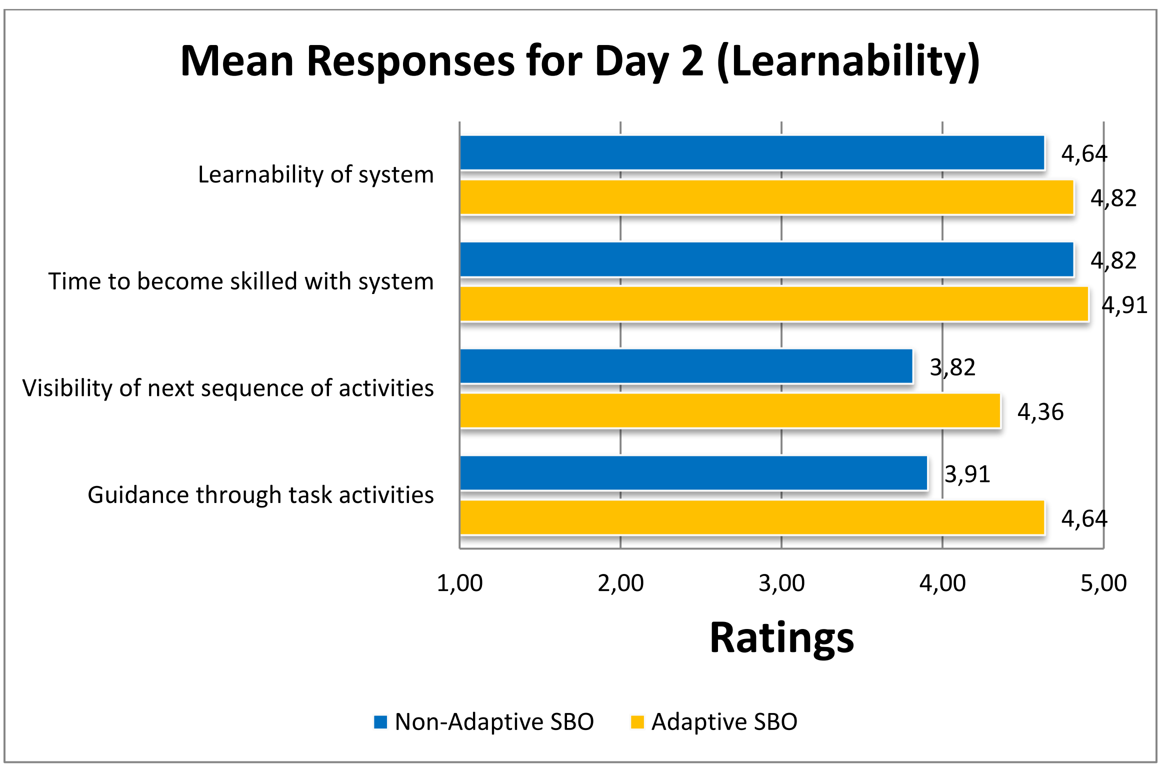


Figure 7.20: Comparison of Learnability for Day 2 (n=22)

The results from the Wilcoxon Matched-Pairs T-Test for learnability (Appendix K) revealed significant improvements in the satisfaction levels for learnability for both versions of SBO over the two-day evaluation period (p-values were less than the 5% level of significance). The Cohen's d value for the adaptive version of SBO was 0.59, indicating that there was a medium amount of improvement over the two days. The Cohen's d value for the non-adaptive version of SBO was 1.49, indicating a significantly large improvement. From these results, it can be inferred that the learning curve for the participants who used the adaptive version of SBO was relatively flat. These participants performed better on the first day of the evaluation and therefore, there was less improvement on the second day.

The results from the Wilcoxon-Mann-Whitney U-Test for learnability (Appendix K) revealed that an association existed on day two between the satisfaction levels, in terms of learnability and the version of SBO used (p-value = 0.047, which is less than the 5% level of significance). The Cohen's d value (0.93) indicated a large association. Based on these results, it may be inferred that the adaptive version of SBO was easier to use.

7.4.3.4 Adaptivity

The Cronbach's $\propto$ statistic (Appendix K) for the adaptivity section of the post-test questionnaire was too low (less than 0.6) to discuss collectively. Therefore, the individual items assessed will be discussed with regard to the adaptation technique used, and the impact on the adaptive version of SBO.

Figure 7.21 illustrates the mean responses for the adaptivity section in the post-test questionnaire for both days. These responses only applied to those participants who had used the adaptive version of SBO. Each item in the adaptivity section was evaluated, to determine whether or not the participants had noticed the adaptation technique.

Figure 7.21 shows that over the two-day evaluation period, there were minor improvements in the visibility of the adaptation. The results from the Wilcoxon Matched-Pairs T-Test for adaptivity (Appendix K) indicated that these improvements were not statistically significant (p-values were greater than the 5% level of significance).

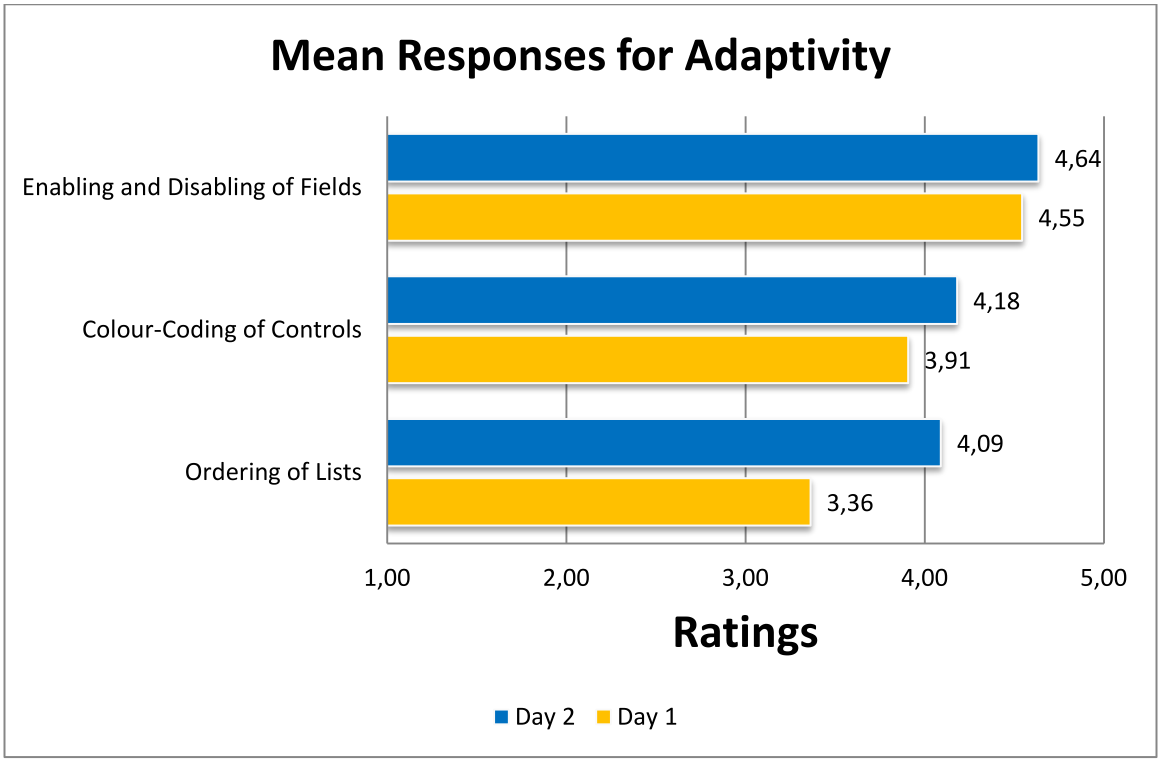


Figure 7.21: Comparison of Adaptivity for Day 1 and Day 2 (n=11)

Navigation adaptation (enabling and disabling fragments) proved to be useful, as the participants who used the adaptive version of SBO found that there were fewer steps required to complete a task, and that it was relatively easy to complete a task over the two-day evaluation period (Section 7.4.3.2). In terms of learnability, this particular adaptation technique also proved to be beneficial, as the participants found that the adaptive version of SBO provided a noticeable amount of guidance to aid task completion and learnability (Section 7.4.3.3). This also applied to the technique of colour-coding controls, as it was implemented to facilitate presentation adaptation and to assist in improving guidance and the learnability of SBO.

Content adaptation (ordering of lists technique) proved to be beneficial whereby the participants who used the adaptive version of SBO found it easier to find and select a buyer from a list, as opposed to those participants who used the non-adaptive version of SBO (Section 7.4.3.2).

7.4.4 Qualitative Feedback

Qualitative data were obtained from the participants to support the data gathered from the performance and the subjective metrics. The same questions were asked of the participants from both groups over the two-day evaluation period. The intention of these questions was to obtain any positive and negative feedback, and to obtain general comments or suggestions for the improvement of SBO. A thematic analysis was performed on the feedback obtained. The results from the thematic analysis revealed that the feedback could be further classified into user interface feedback, task support and efficiency feedback and learnability feedback.

7.4.4.1 Positive Feedback

User interface

User interface feedback mostly related to the layout of the controls. Users felt that the controls were logically placed and were easy to locate. This view was shared by 55% of the participants who used the adaptive version of SBO, and by 45% of the participants who used the non-adaptive version of SBO on the first day of the evaluation. Only 18% of the participants who used the adaptive version of SBO, and 27% of the participants who used the non-adaptive version of SBO provided positive feedback on these aspects on the second day of the evaluation.

Task support and efficiency

Task support and efficiency feedback were mostly provided by those participants who had used the adaptive version of SBO. These participants appreciated the fact that SBO suggested buyers and made it easy to complete tasks. Some of the positive comments that confirmed these statements were:

- “Having the system suggests the buyers – it makes life very easy...”
- “Task flow was easy to follow...”
- “Tasks could be conducted quickly and easily...”

- "There are not many steps to complete a purchase order..."

Learnability

Forty-five percent of the participants who had used the adaptive version of SBO felt that it was easy to learn and use on day one of the evaluation. This increased to 91% on day two, as opposed to the participants who used the non-adaptive version of SBO, where only 36% found it easy to learn and use on day one, and 73% on day two. A positive comment that was made regarding the adaptive version of SBO to support this conclusion was:

"*The system is very, very simple to learn and user friendly, in the sense that it guides you on what the next step is to be done by activating the required fields..."*

7.4.4.2 Negative Feedback

User interface

The size of the font was commented on by nine percent of the participants who had used the adaptive version of SBO, and by eighteen percent of the participants who had used the non-adaptive version of SBO. Those participants felt that the font size should be increased in order for the labels of the controls to become more legible.

Nine percent of the participants who used the adaptive version of SBO, and eighteen percent of the participants who used the non-adaptive version of SBO felt that there were too many controls and that these controls were not placed in a logical order. A comment that was made to support this conclusion was:

"The flow of the controls should be better, e.g. placing a supplier and buyer below each other for better flow..."

Lastly, nine percent of the participants who had used the non-adaptive version of SBO felt that the buyers list could become difficult to search through if the number of buyers increased significantly.

Task support and efficiency

Most of the task support issues related to system notifications. Nine percent of the participants who used the non-adaptive version of SBO felt that SBO should notify the user when a specific control was not selected or updated. A comment that was made to support this conclusion was:

"*I found it odd that the system did not notify me when I did not select a buyer...*"

Nine percent of the participants who used the non-adaptive version of SBO commented that the system did not indicate whether a task had been completed successfully, and that the error messages were too cryptic. Comments that confirmed this were:

- "The system does not notify that details were saved correctly..."
- "System error messages were not that clear..."

7.4.4.3 Suggestions for Improvement

User interface

Eighteen percent of the participants who had used the adaptive version of the SBO, and nine percent of the participants who had used the non-adaptive version of SBO felt that the fonts used in SBO could be bolder (to increase visibility and legibility). Nine percent of the participants who used the adaptive version of SBO, and nine percent of the participants who had used the non-adaptive version of SBO felt that the placement of the controls could be improved. Eighteen percent of the participants who had used the adaptive version of SBO felt that the colour scheme of SBO needed to be improved. A suggestion which confirmed this was:

"*The system is too dull , there should be more colour...*"

Task support and efficiency

All of the suggestions for task support and efficiency were provided by users of the non-adaptive version of SBO. Nine percent of the users felt that the notifications and error messages provided by SBO could be improved. Furthermore, nine percent of the users felt that sorting of the buyers and suppliers' list would improve the efficiency of the system.

7.5 Hypothesis Testing

Several hypotheses were established in Section 7.3.2 to assist in determining whether AUIs could improve the usability of SBO with regard to efficiency, learnability and satisfaction. This section aims to assess those hypotheses by reflecting on the evaluation results. Statistically, significant results are represented by a "✓". Results that are not statistically significant are represented by a "✗".

7.5.1 AUIs and Efficiency

Table 7.1 presents the Mann-Whitney U-Test results for efficiency, according to the performance metrics used in the evaluation. This result shows that no statistically significant data were obtained to prove that AUIs offered an improvement in the efficiency of SBO.

	Task Success	Time-on-Task	Mouse Clicks	Errors Committed
Day 1	✗	✗	✗	✗
Day 2	✗	✗	✗	✗

Table 7.1: Hypothesis Testing Results for Efficiency (Mann-Whitney U-Test)

Based on the results in Table 7.1, we fail to reject the null hypothesis for efficiency ($H_{0,1}$), as there is not sufficient evidence to support the claim that the AUI made any significant improvement to the efficiency of SBO.

7.5.2 AUIs and Learnability

Table 7.2 presents the Mann-Whitney U-Test results for learnability, according to the difference in the performance metrics over the two-day evaluation period, when comparing both versions of SBO.

	Task Success	Time-on-Task	Mouse Clicks	Errors Committed
Difference	✗	✗	✗	✗

Table 7.2: Hypothesis Testing Results for Learnability (Mann-Whitney U-Test)

The results in Table 7.2 show that there was no significant difference with regard to the number of mouse clicks over the two-day evaluation period.

Table 7.3 presents the Wilcoxon Matched-Pairs T-Test results for learnability (improvements in each version of SBO). The results in Table 7.3 show that only time-on-task could be regarded as significant for the adaptive version of SBO (Section 7.4.2.1).

	Task Success	Time-on-Task	Mouse Clicks	Errors Committed
Difference	✗	✓	✗	✗

Table 7.3: Hypothesis Testing Results for Learnability (Wilcoxon Matched-Pairs T-Test)

Based on the results in Tables 7.2 and 7.3, we reject the null hypothesis for learnability ($H_{0,2}$), as there is sufficient evidence to support the claim that the AUI made a significant improvement (with regard to time-on-task) on the learnability of SBO.

7.5.3 AUIs and Satisfaction

Table 7.4 presents the Mann-Whitney U-Test results for satisfaction, according to the subjective metrics over the two-day evaluation period. The learnability on the second day of the evaluation was the only metric to present a significant result. This result was statistically significant for the adaptive version of SBO (Section 7.4.3.3).

	Overall	Task Support and Efficiency	Learnability
Day 1	✗	✗	✗
Day 2	✗	✗	✓
Difference	✗	✗	✗

Table 7.4: Hypothesis Testing Results for Satisfaction (Mann-Whitney U-Test)

Table 7.5 presents the satisfaction results for the Wilcoxon Matched-Pairs T-Test. Significant results were obtained for learnability with regard to the adaptive version of SBO (Section 7.4.3.3).

	Overall	Task Support and Efficiency	Learnability
Difference	✗	✗	✓

Table 7.5: Hypothesis Testing Results for Satisfaction (Wilcoxon Matched-Pairs T-Test)

Based on the results in Tables 7.4 and 7.5, we reject the null hypothesis for satisfaction ($H_{0,3}$), as there is sufficient evidence to support the claim that the AUI made a significant improvement to the satisfaction of SBO. This result also further supports the previous decision to reject the null hypothesis for learnability (Section 7.5.2).

7.5.4 AUIs and ERP Systems

Table 7.6 presents the overall results of the evaluation in terms of the null hypothesis. The results show that sufficient evidence exists to:

- **Fail to reject** $H_{0,1}$ for *efficiency;*
- **Reject** $H_{0,2}$ for *learnability;*
- **Reject** $H_{0,3}$ for *satisfaction.*

Efficiency	Learnability	Satisfaction
✗	✓	✓

Table 7.6: Overall Hypothesis Testing Results

Based on the results in Table 7.6, we reject the null hypothesis (H_0), as there is sufficient evidence to support the claim that the AUI made a significant improvement to the usability of SBO in terms of learnability and satisfaction.

7.6 Discussion

Literature has shown that AUIs have the potential to reduce the complexity of a system and enhance its usability by means of task simplification and improved user satisfaction (Chapter 5). This evaluation has demonstrated that this also applies to an AUI for SBO. Participants who interacted with the adaptive version of SBO, over the two-day evaluation period, appreciated the level of task support provided by the adaptive version of SBO, which contributed to a greater sense of overall user satisfaction, as opposed to those participants who interacted with the non-adaptive version of SBO.

AUIs can only be considered useful if it can be demonstrated that the inclusion of adaptivity in the user interface makes the system easier and more efficient to use (Chapter 5). The satisfaction results from the evaluation revealed that the participants who had used the adaptive version of SBO found the system easier to learn on day one than those participants who had used the non-adaptive version. This outcome was supported by the learnability results from the performance metrics.

Satisfaction responses for the adaptive version of SBO were rated higher than those for the non-adaptive version of SBO. Despite this, most of the differences between the ratings of the two versions of SBO could not be regarded as statistically significant. One of the reasons for this could be attributed to the current exposure of the participants to SAP R/3 (which is more complex than SAP Business One). For this reason, SAP Business One was perceived as being more satisfying to use by both groups.

The use of an AUI proved to be beneficial with regard to learnability. Learnability results from the evaluation showed that the participants were able to complete tasks in a shorter amount of time than the users of the non-adaptive version of SBO on day one of the evaluation; and they were able to maintain these levels on day two of the evaluation.

AUI theory states that the predictability and accuracy of the AUI has an impact on the satisfaction levels of the user (Section 5.5.1). Based on the high satisfaction results obtained from the evaluation, it may be implied that the predictability and accuracy of the AUI was satisfactory. Literature has stated that AUIs have the potential to violate some direct manipulation usability issues, namely uncertainty, transparency and predictability, privacy and trust, and controllability (Section 5.5.2). The results from the evaluation have shown that the adaptive version of SBO, which maintains its direct manipulation style of user interfaces, does not violate any of these usability principles.

One of the surprising outcomes of the evaluation was the fact that the adaptive version of SBO did not have any significant improvement in terms of the number of mouse clicks and errors committed over the two-day evaluation period. These metrics should have improved, as the Disabling Fragments adaptation technique (Figure 6.13) that was implemented should have prevented the participants from clicking on any unnecessary controls. Furthermore, it should also have supported the participants in constructing the correct mental model with regard to completing a purchase order.

The evaluation results revealed that the AUI did not significantly improve the efficiency of SBO. The efficiency of the adaptive version of SBO was not statistically different from the efficiency of the non-adaptive version of SBO. Based on the qualitative feedback from the participants, factors that could have contributed to the lack of improved efficiency include: the number of controls and the flow of these controls. These issues could impact on the efficiency in terms of the participants having to spend additional time locating controls. Additional adaptation techniques for presentation adaptation (Removing Fragments and Sorting Controls) could be implemented to address these issues and possibly to improve efficiency.

Content adaptation, presentation adaptation and navigation adaptation were proposed to resolve the identified usability issues of SBO (Table 5.1). Several HCI patterns from the proposed adaptation taxonomy were implemented in order to deliver the required adaptation effects. The Good defaults HCI pattern (in the form of list-based adaptation) was implemented to the list of buyers to support content adaptation. Presentation adaptation was supported through the implementation of the Enabling and Disabling Fragments HCI pattern, which was applied to various sections of the Purchase Order form in order to separate the different activities.

Lastly, the Colour-Coded Sections HCI pattern was implemented to support navigation adaptation. Based on the results of the evaluation, Table 7.7 shows which of the identified usability issues of SBO were resolved with the proposed adaptation type.

Table 7.7 indicates that 73% of the identified usability issues were resolved through the inclusion of adaptivity in the user interface. Content adaptation through the implementation of the Good Defaults HCI pattern assisted in:

- Providing a perceived efficiency gain by allowing the participant to select the most recently used (MRU) buyer and the most frequently used (MFU) buyers from the top of the buyers' list. This, however, was not enough to make a statistically significant impact on the efficiency of the adaptive version of SBO, as opposed to the non-adaptive version of SBO. A possible reason for this was that there were too few items in the list to make a significant improvement in the overall efficiency;
- Providing a level of personalisation with regard to the user interface. The buyers' list in the Purchase Order form was sorted differently for each participant who used the adaptive version of SBO. This was based on how each participant interacted with the adaptive version of SBO; and
- The routine task of searching through a list of potential buyers was automated to provide the participants with quick access to the most recently used and the most frequently used buyers at the top of the buyers list. This assisted by preventing the participants who used the adaptive version of SBO from having to scroll up and down the list of buyers to select a buyer.

No.	Usability Issue	Adaptation Type	Resolved
1.	Too many steps required to complete a task	Presentation	✓
2.	SBO does not enable efficient completion of tasks	Content	✓
3.	SBO does not improve user productivity	Content and Presentation	✓
4.	SBO does not automate routine tasks	Content	✓
5.	Finding information and functionality is difficult	Navigation	✓
6.	SBO cannot guide the user through the correct sequence of tasks	Navigation	✓
7.	Evidence of the next sequence of steps to complete a task is not provided	Navigation	✓
8.	There are too many unused fields on the screen	Presentation	✗
9.	Layout of the user interface does not contribute to the efficient completion of tasks	Presentation	✗
10.	Steps need to be manually recorded the first time in order to be remembered for future use	Presentation	✓
11.	Personalisation of user interfaces is not possible	Content and Presentation	✓

Table 7.7: Resolved Usability Issues of SBO

Presentation adaptation which was supported through the implementation of the Enabling and Disabling Fragments HCI pattern assisted in:

- Reducing the number of steps required to complete a task, by presenting a visual break-down of the task into several activities;
- Improving user productivity, by allowing the user to focus on one activity at a time; and
- Improving learnability, so that the steps to complete a task do not have to be manually recorded for the first time.

Areas where presentation adaptation did not improve the usability were:

- Too many unused fields; and
- Layout of the user interface does not contribute to the efficient completion of tasks.

These issues also contributed to the lack of sufficient improvement in the efficiency of completing tasks on the adaptive version of SBO. The theoretical analysis done, using the keystroke-level model showed that by changing the layout of the user interface, the efficiency would not be improved, as the same number of controls would still be present on the user interface. Feedback from the participants revealed that this feature would have been useful to have. The participants did comment on reducing the number of controls on the user interface, as this would have assisted in reducing the time it takes to locate a particular control and complete the task.

Navigation adaptation was supported by the implementation of the Colour-Coded Sections HCI pattern, and through the use of the Enabling and Disabling Fragments HCI pattern. This assisted in:

- Improving the ability to find information and functionality, by showing users which fields had been completed, and which fields or fragments were currently enabled and disabled;
- Guiding users through the correct sequence of activities for a particular task through the indication of colour and the enabling and disabling of fields; and
- Providing the next sequence of steps through the indication of colour and the enabling and disabling of fields.

7.7 Conclusion

This chapter has demonstrated that an empirical evaluation can be used to successfully evaluate the usability of an AUI for SBO. The empirical evaluation (comprising a user study) supported the collection of both quantitative and qualitative data. The quantitative data were analysed by using descriptive and inferential statistics. The qualitative data were analysed using a thematic analysis of the participants' feedback.

This chapter has shown that a between-subjects design can be used to effectively compare the usability results for two equivalent groups of participants. The between-subjects design, combined with a stratified sampling approach, supported the process of placing the participants into two equivalent groups (restricting the amount of bias that could be injected into the evaluation that would skew the results).

The benefits of an AUI can only be realised over a prolonged period of use (Chapter 5). This chapter has shown that through the use of a between-subjects design, and having the participant use the adaptive ERP system more than once, a similar effect could be achieved. The between-subjects design complemented the objectives of the evaluation, as it provided the opportunity to measure the learnability.

The results obtained from the empirical evaluation have shown that AUIs can provide some usability benefits for SBO in the form of learnability and satisfaction. No significant benefits were obtained with regard to efficiency. However, several recommendations were made involving the implementation of other adaptation techniques and HCI patterns. A statistically significant improvement in the time required for a user to complete a tasks over the two-day evaluation period was identified for the adaptive version of SBO. This improvement was confirmed by the results that were obtained relating to perceived learnability and satisfaction.

The following chapter concludes this thesis. Achievements and contributions made by this research will be discussed. Several recommendations that could be used to inform future research will be identified and discussed.

Chapter 8: Conclusion

8.1 Introduction

The aim of this chapter is to consolidate the findings made in this research and to determine the extent to which these findings have contributed to the achievement of the research objectives, as defined in Chapter 1. Some limitations and problems that were encountered in the course of achieving the research objectives are discussed. Several contributions have been made in this research and are highlighted and discussed in this chapter. Lastly, several recommendations for theory, practice and future research are proposed, based on the identified contributions.

8.2 Research Objectives Re-Visited

This research proposed the following thesis statement:

> *AUIs can be designed to improve the usability (efficiency, learnability and satisfaction) of ERP systems for small enterprises.*

In order to successfully evaluate the thesis statement, several research objectives were defined (Table 8.1).

No	*Research Objectives*
1.	To select a specific industry sector and ERP system on which to focus this research.
2.	To identify the existing usability issues of ERP systems and to determine how to evaluate the usability of an ERP system.
3.	To identify the usability issues of the selected ERP system for small enterprises.
4.	To determine how AUIs could be applied to address the identified usability issues of ERP systems.
5.	To design an AUI and develop a prototype to address the usability issues of ERP systems.
6.	To evaluate the benefits of incorporating an AUI into the selected ERP system.
7.	To discuss the theoretical and practical contributions of incorporating an AUI into the selected ERP system.

Table 8.1: Research Objectives

The research objectives for this research were converted into research questions (Table 8.2).

No	Research Question
1.	Which industry sector and ERP system for small enterprises can this research use for experimental purposes?
2.	What are the existing usability issues of ERP systems, and how can the usability of an ERP system be evaluated?
3.	What are the usability issues of the selected ERP system for small enterprises?
4.	How can AUIs be applied to address the identified usability issues of ERP systems?
5.	How can an AUI be designed and developed to address the usability issues of ERP systems?
6.	What are the benefits of incorporating an AUI into the selected ERP system?
7.	What theoretical and practical contributions have been made by incorporating an AUI into the selected ERP system for small enterprises?

Table 8.2: Research Questions

All of the objectives for this research were achieved and are discussed in the next section.

8.3 Research Achievements

The main objective of this research was to determine how adaptive user interfaces (AUIs) could be designed to improve the usability of enterprise resource planning (ERP) systems for small enterprises. This objective was realised theoretically and practically. Theoretically, this objective was met by proposing an adaptation taxonomy for ERP systems, a system architecture for ERP systems (incorporating an AUI) and a set of adaptive interface components (Chapter 6). Practically, an AUI prototype was implemented using an existing ERP system for small enterprises, namely SAP Business One (SBO) (Chapter 6). The results from an empirical evaluation showed that AUIs can present several usability benefits for SBO (Chapter 7).

8.3.1 Theoretical Achievements

Small enterprises in the Manufacturing sector in South Africa, using SBO, were selected for experimentation purposes (Chapter 2). A preliminary study conducted with three ERP vendors revealed that ERP sales amongst small enterprises were mostly in the Manufacturing sector. The preliminary study also indicated that SBO would be the most suitable ERP system (for experimental purposes), as it was the most customisable. SBO has a software development kit (SDK) that supports extensibility at the functional and user interface levels.

A literature review revealed that ERP systems typically suffer from usability issues in five main areas, namely Navigation, Presentation, Task Support, Learnability and Customisation (Chapter 3). This research has indicated that usability evaluations of ERP systems are typically conducted using an heuristic approach. Because of the inconsistency of the heuristics used, comparing usability evaluations of ERP systems is a complex task. To address this limitation, a set of heuristics (specific to ERP systems) was proposed. These heuristics were established based on the similarities of the attributes used in previous heuristic evaluations of ERP systems.

An interview study of small enterprises in the Manufacturing sector in South Africa, using SBO, was conducted. This study, in conjunction with an heuristic evaluation of SBO (using the proposed ERP heuristics), confirmed that the areas in which ERP systems for small enterprises experience usability issues are: Navigation, Presentation, Task Support, Learnability and Customisation (Chapter 4). The heuristic evaluation used a set of general heuristics (Nielsen's ten heuristics) and the proposed ERP heuristics. Use of the proposed ERP heuristics assisted in validating the ability of these heuristics to identify the usability issues of ERP systems. The results from the heuristic evaluation revealed that the proposed ERP heuristics supported some of the usability issues from the general heuristics, but were also capable of identifying different usability issues.

Three types of adaptation were proposed to address the identified usability issues, namely content adaptation, presentation adaptation and navigation adaptation (Chapter 5). These adaptation types were mapped onto the usability issues to show how AUIs could potentially address the usability issues of SBO (Table 5.1).

An adaptation taxonomy (Figure 8.1, repeated here for ease of reference) and a system architecture (Figure 8.2, repeated here for ease of reference) were proposed to support the design and implementation of an AUI for SBO (Chapter 6).

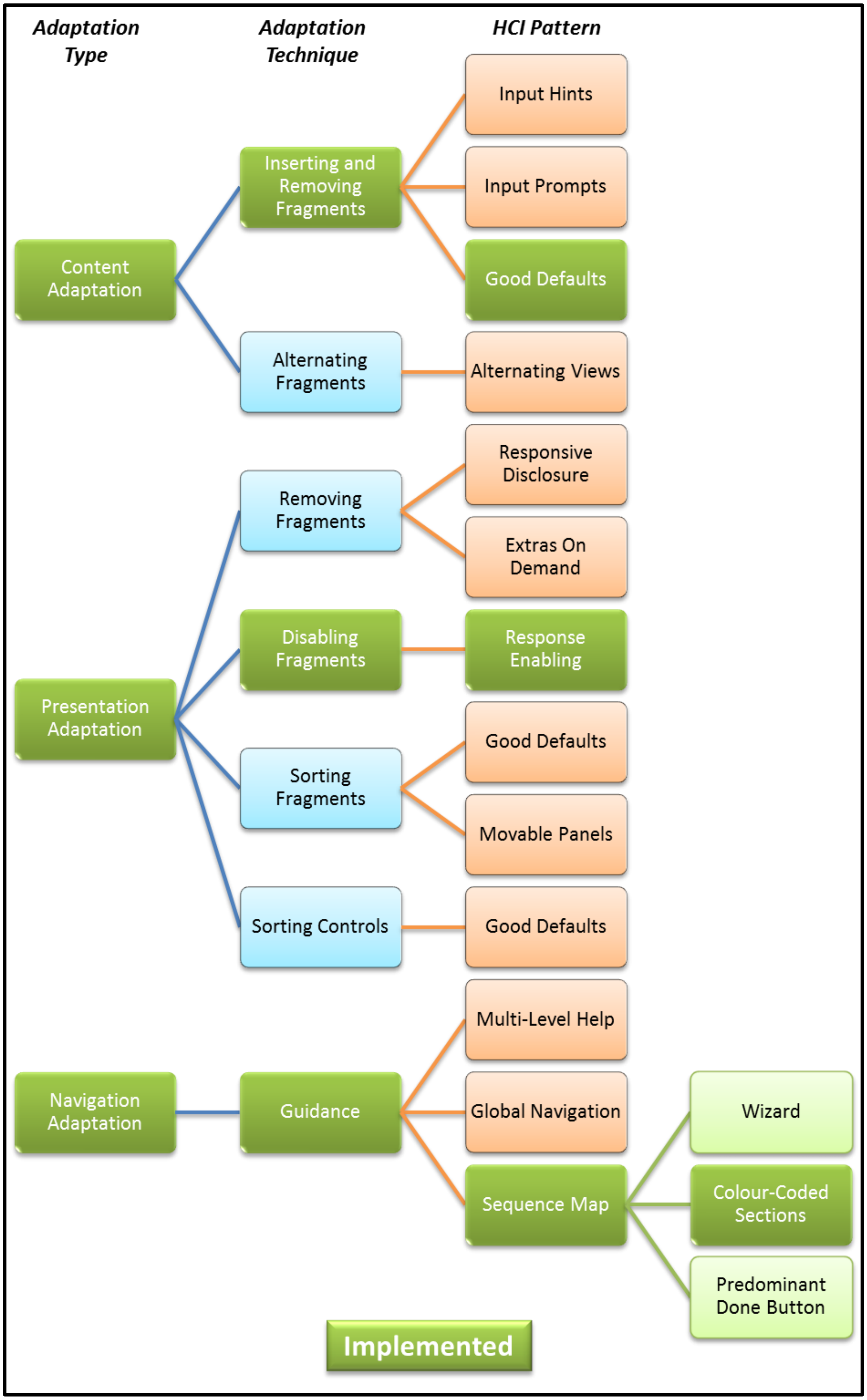


Figure 8.1: Proposed Adaptation Taxonomy with Implemented Adaptation Techniques and HCI Patterns

The proposed adaptation taxonomy (Figure 8.1) for ERP systems was based on the adaptation taxonomy for hypermedia systems, as proposed by Knutov *et al.* (2009). The proposed adaptation taxonomy for ERP systems assisted in answering the question: "How can adaptations be designed for ERP systems?"

Only those adaptation techniques that could be used to address the usability issues of SBO, as identified in Chapter 4, were selected from the original taxonomy. These adaptation techniques were (highlighted in green in Figure 8.1):

- Inserting/Removing Fragments (content adaptation);
- Sorting Fragments (presentation adaptation); and
- Guidance (navigation adaptation).

Other adaptation techniques and methods that could have been used to address the identified usability issues specific to hypermedia systems were replaced by existing human-computer interaction (HCI) design patterns for desktop applications (Table 8.3).

Adaptation Type	*Adaptation Technique*	*HCI Pattern*
Content Adaptation	Altering Fragments	Alternating Fragments
Presentation Adaptation	Dimming Fragments	Removing Fragments
		Disabling Fragments
	Layout	Sorting Controls
	Link Sorting / Ordering	

Table 8.3: Adaptive Hypermedia Techniques and HCI Design Patterns

The proposed system architecture (Figure 8.2) was based on the Simplified Adaptive System Architecture (Ramachandran 2009) selected in Chapter 5. Modifications were made to the selected architecture in order for it to be specialised to the domain of ERP systems. These modifications included:

- Provision for an Inference Engine in the Application Layer to monitor and update the AUI;
- Provision for content adaptation;

- Provision for an ERP Engine to contain the business and ERP system logic; and
- Provision for the ERP system tables in the Database Layer.

The reasons for these modifications were provided in Section 6.3.2.

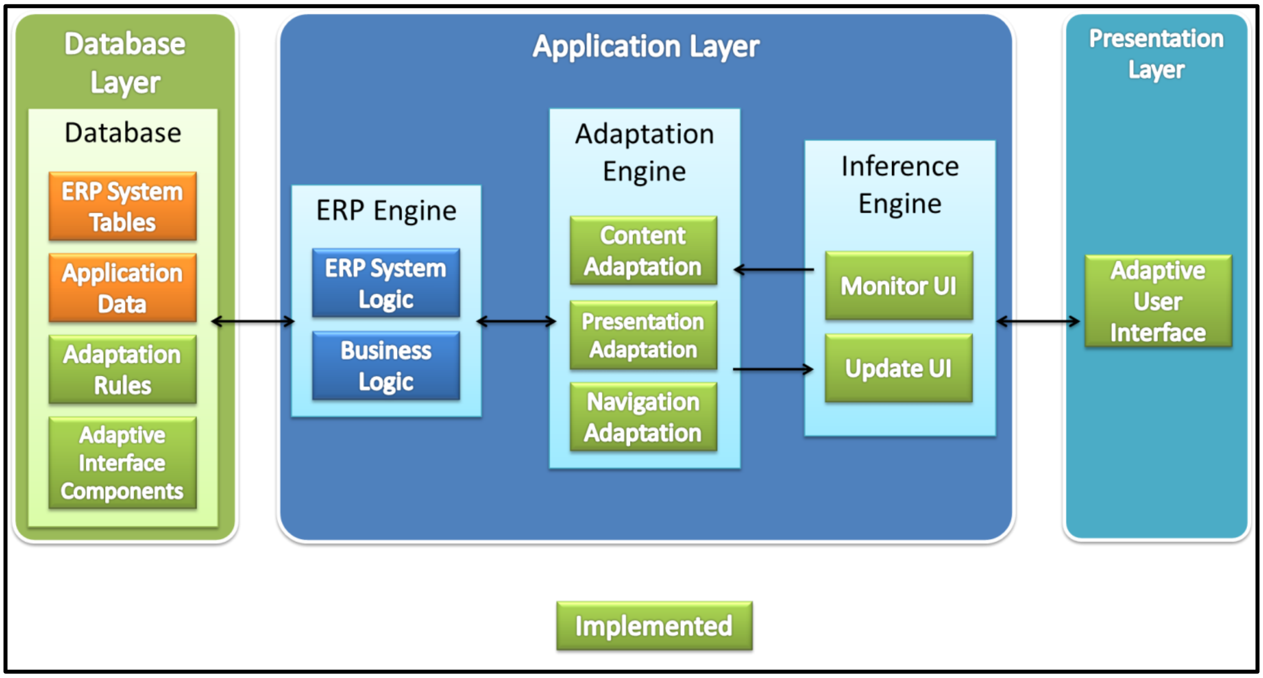


Figure 8.2: Proposed System Architecture with Implemented Components

8.3.2 Practical Achievements

An AUI prototype was implemented using an existing ERP system, namely SBO. The prototype was based on selected components from the proposed system architecture (Table 8.4) and selected adaptation techniques from the proposed adaptation taxonomy (Table 8.5). The implemented system architecture components and adaptation techniques are illustrated in green in Figures 8.1 and 8.2. The implementation demonstrated that the proposed ERP adaptation taxonomy and the proposed system architecture can be incorporated into SBO.

Layer	Component
Presentation Layer	Adaptive User Interface
Application Layer	Inference Engine
	Adaptation Engine
Database Layer	Database (Adaptation Rules and Adaptive Interface Components)

Table 8.4: Implemented System Architecture Components

Adaptation Type	Adaptation Technique	HCI Pattern
Content Adaptation	Inserting and Removing Fragments	Good Defaults
Presentation Adaptation	Disabling Fragments	Response Enabling
Navigation Adaptation	Guidance	Sequence Map (Colour-Coded Sections)

Table 8.5: Implemented Adaptation Techniques

In order to address the usability issues identified in Chapter 4 (Table 8.6, repeated here for ease of reference) several changes were made to both the user interface and SBO.

No	Usability Issue
1.	Too many steps required to complete a task
2.	SBO does not enable efficient completion of tasks
3.	SBO does not improve user productivity
4.	SBO does not automate routine tasks
5.	Finding information and functionality is difficult
6.	SBO cannot guide the user through the correct sequence of tasks
7.	Evidence of the next sequence of steps to complete a task is not provided
8.	There are too many unused fields on the screen
9.	Layout of the user interface does not contribute to the efficient completion of tasks
10.	Steps need to be manually recorded the first time in order to be remembered for future use
11.	Personalisation of user interfaces is not possible

Table 8.6: Identified Usability Issues of SBO

The changes that were made to the user interface were made to support the implemented adaptation techniques. In order to support content adaptation (Section 6.4.1.1), smart lists were created to replace the existing list of buyers. Presentation adaptation (Section 6.4.1.2) required that fragments of the user interface (supplier details, item details, and buyer details) be enabled and disabled, as the user progresses through the activities of the task of completing a purchase order. Navigation adaptation (Section 6.4.1.3) was supported by changing the colours of the controls on the Purchase Order form. The colours of the controls alternated (grey, white, green or orange), as the state (disabled, enabled, active or completed) of the control changed.

This research has shown that AUIs can improve the usability of SBO (Section 7.7). This was determined by means of an empirical evaluation which compared an adaptive version of SBO (the prototype) with a non-adaptive version of SBO. The results from the evaluation indicated that AUIs have the ability to improve the usability of ERP systems for small enterprises in terms of learnability and satisfaction (Section 7.5.4).

8.4 Limitations and Problems Encountered

Several limitations were encountered during this study. Some of these limitations are discussed below.

Access to an extensible ERP system was the first challenge that was encountered (Section 2.4.2). In order for an ERP system to be used in this study, it needed to be flexible enough to be modified at the user interface layer without affecting the business logic of the ERP system. Most ERP vendors do not provide access to modify the ERP system at the source code level. Only SBO supported this level of extensibility through an SDK. It was for this reason that SBO was selected for experimental purposes (Section 2.4.3).

Access to users was also regarded as a potential issue (Section 4.2.1.1). Many of the enterprises that use SBO in South Africa were not willing to participate in this study, as they felt that it could compromise their operations and data security. Only those users of SBO who were willing to participate in this research were consulted during the interview study.

Access to established metrics to evaluate the usability of ERP systems was a limitation (Section 3.4.1). Currently, inconsistent sets of heuristics are used to evaluate the usability of ERP systems. This made it difficult to compare the usability issues of different ERP systems from different evaluations. To overcome this challenge, a set of heuristics specific to ERP systems was proposed (Section 3.4.2).

The SDK provided with SBO restricted the level of flexibility with regard to development. This limited the ability to modify the ERP system at both functional and user interface levels (Section 6.4.1.1). An example of this limitation was the inability to re-order list items in real time. To overcome this limitation, new user interface controls had to be created to replace the old controls.

The environment in which the AUI prototype was evaluated was considered as a possible limitation (Section 7.3.6). Conducting the evaluation in a live environment with actual users would have raised several issues, such as: infrastructure costs, data privacy and integrity, bias with regard to previous exposure to SBO, as well as disruption of the daily operations of the enterprise. To overcome these issues, the experiment was conducted in a controlled environment with novice ERP users who had no previous exposure to SBO (Section 7.3.6).

8.5 Research Contributions

This research has made two key contributions. These contributions relate to the heuristic evaluation of ERP systems and the design and development of AUIs for ERP systems (Figure 8.3).

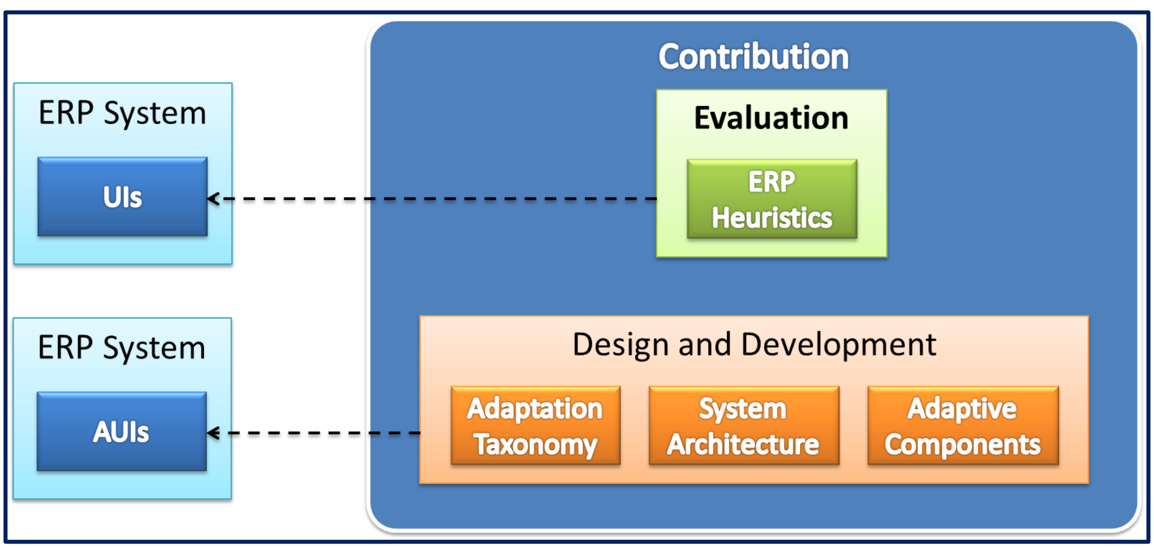


Figure 8.3: Key Contributions of Research

These key contributions have assisted in achieving the main research objective. The use of the proposed ERP heuristics (Section 3.4.2) assisted in identifying the usability issues of an existing ERP system (SBO). The proposed adaptation taxonomy (Section 6.3.1), the proposed system architecture (Section 6.3.2), and the adaptive components (Section 6.3.3) contributed to achieving the main research objective. This was done by providing a means to design and develop an AUI to address the identified usability issues of SBO.

The contributions made by this research can be classified as either theoretical or practical contributions. These contributions are discussed in this section within the context of the thesis statement (defined in Chapter 1).

8.5.1 Theoretical Contribution

Theoretical contributions were made with regard to the thesis statement in three main areas. These areas are illustrated in Figure 8.4.

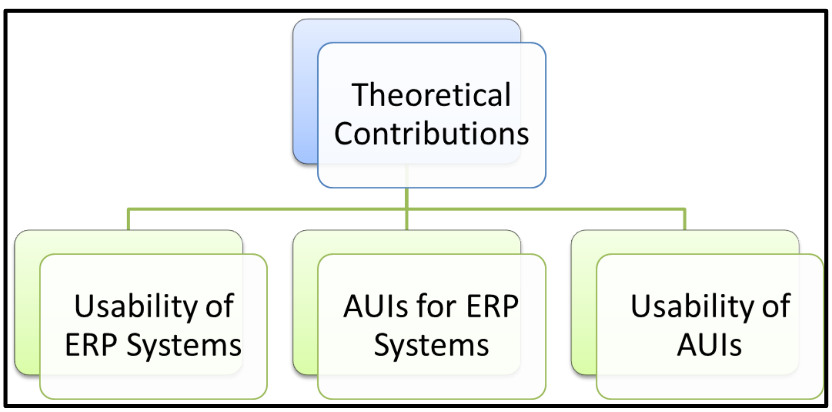


Figure 8.4: Theoretical Contributions of Research

8.5.1.1 Contribution to Usability of ERP Systems

Limited research exists to support the identification and contextualisation of usability issues for ERP systems (Section 3.4.1). To address this limitation, a set of heuristics specific to ERP systems was proposed (Section 3.4.2). The proposed ERP heuristics focused on the areas of Navigation, Presentation, Task Support, Learnability and Customisation. These areas were identified as the most predominant areas in which usability issues for ERP systems typically occur.

8.5.1.2 Contribution to AUIs for ERP Systems

Limited research exists describing how AUIs could be designed for ERP systems. This research has contributed to addressing the above limitation by proposing an adaptation taxonomy for ERP systems (Figure 8.1) (Section 6.3.1). The proposed adaptation taxonomy comprises the three main adaptation types (content adaptation, presentation adaptation and navigation adaptation), adaptation techniques (associated to a particular adaptation type) and HCI patterns. The proposed adaptation taxonomy can support the design of AUIs for ERP systems by providing a wide range of adaptation techniques and HCI patterns for an adaptation type that could provide better and more accurate adaptation results.

Several architectures and frameworks exist to support the development of AUIs (Section 5.6.2). However, none of these have been applied to the domain of ERP systems. This research has contributed to addressing the above limitation by proposing a three-tiered system architecture for an ERP system that incorporates an AUI (Figure 8.2). The architecture consists of a Presentation Layer, an Application Layer, and a Database Layer. The architecture aims to separate the business logic of the ERP system from the functional logic of the system needed to adapt the user interface (Section 6.4.2.2). This level of separation is achieved by allowing the ERP Engine to process the business logic independently of the interface events. Interface events are processed by the Inference Engine (Figure 8.2). The Adaptation Engine serves as the mediator in deciding which type of adaptation to apply to the data provided by the ERP Engine and the Inference Engine.

Several adaptive components (user model, task model, and dialog model) were proposed to support the design of AUIs for ERP systems. These components reside in the adaptive interface component of the proposed system architecture and assist in delivering the adaptations to the user interface (Section 6.4.2.1).

8.5.1.3 Contribution to Usability of AUIs

Whilst AUIs have become the preferred solution for developing personalised user interfaces for complex systems (Chapter 5), earlier work on AUIs has shown that AUIs can negatively affect good HCI design. This research has contributed to overcoming this limitation by proposing an adaptation taxonomy (specific to ERP systems) (Figure 8.1), which incorporates the use of existing HCI design principles for desktop systems (Section 2.4.3).

HCI design principles for desktop systems were selected, as this research focuses on desktop ERP systems. Further evidence to support this can be obtained from the results of the empirical evaluation (Section 7.6), which has shown that the adaptive version of SBO (which made use of selected HCI patterns) did not violate any of the usability principles established for direct manipulation user interfaces, namely uncertainty, transparency and predictability, privacy and trust and controllability.

8.5.2 Practical Contributions

This research has made practical contributions with regard to the thesis statement in three main areas. These areas are illustrated in Figure 8.5.

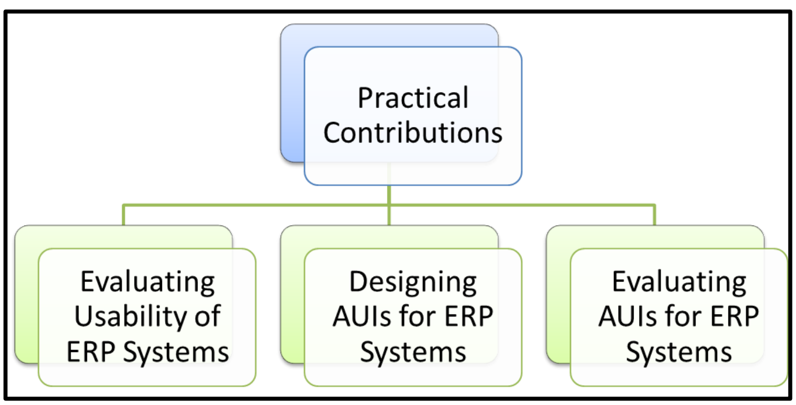


Figure 8.5: Practical Contributions of Research

8.5.2.1 Contribution to Evaluating the Usability of ERP systems

This research has made a practical contribution by demonstrating that the proposed ERP heuristics are capable of (Section 4.6):

- Supporting usability issues identified by general usability heuristics; and
- Identifying significantly different usability issues.

The proposed heuristics were used by usability experts (in combination with general usability heuristics) and assisted in determining the usability issues of SBO.

8.5.2.2 Contribution to Designing AUIs for ERP systems

Literature has shown that no ERP systems exist with an AUI. This research has contributed to addressing this limitation by implementing an AUI prototype that is based on an existing ERP system, namely SBO.

An adaptation taxonomy and a system architecture specific to ERP systems were proposed in Chapter 6. Selected adaptation techniques (Table 8.5) and selected components of the system architecture (Table 8.4) were implemented. The implementation of the proposed adaptation techniques and adaptive components demonstrated that these adaptation techniques can be applied to an existing ERP system to provide adaptation at the Presentation Layer. These adaptation techniques were used to address the usability issues of SBO identified in Chapter 4 (Table 8.7).

8.5.2.3 Contribution to Evaluating AUIs for ERP systems

This research has made a contribution by demonstrating that an empirical evaluation (comprising a user study) can be used to evaluate an AUI for ERP systems. The empirical evaluation conducted in Chapter 7 supported the evaluation of the users' performance and attitude to the adaptive and non-adaptive versions of SBO without having the need to evaluate the effectiveness of the decisions made by the adaptation engine (Section 7.2). The results from the empirical evaluation showed that AUIs can improve the usability of ERP systems in terms of learnability and satisfaction. These results show that usability benefits can be achieved if the proposed adaptation taxonomy and the proposed system architecture are successfully implemented.

No	*Usability Issue*	*Adaptation Type*	*Adaptation Technique (HCI Pattern)*
1.	Too many steps required to complete a task	Presentation	Disabling Fragments (Response Enabling)
2.	SBO does not enable efficient completion of tasks	Content	Inserting and Removing Fragments (Good Defaults)
3.	SBO does not improve user productivity	Content and Presentation	Inserting and Removing Fragments (Good Defaults) Disabling Fragments (Response Enabling)
4.	SBO does not automate routine tasks	Content	Inserting and Removing Fragments (Good Defaults)
5.	Finding information and functionality is difficult	Navigation	Sequence Map (Colour-Coded Sections)
6.	SBO cannot guide the user through the correct sequence of tasks	Navigation	Sequence Map (Colour-Coded Sections)
7.	Evidence of the next sequence of steps to complete a task is not provided	Presentation	Disabling Fragments (Response Enabling)
8.	There are too many unused fields on the screen	Presentation	Disabling Fragments (Response Enabling)
9.	Layout of the user interface does not contribute to the efficient completion of tasks	Presentation	Disabling Fragments (Response Enabling)
10.	Steps need to be manually recorded the first time in order to be remembered for future use	Presentation	Disabling Fragments (Response Enabling)
11.	Personalisation of user interfaces is not possible	Content and Presentation	Inserting and Removing Fragments (Good Defaults)

Table 8.7: Application of Adaptation Techniques to Identified Usability Issues

8.6 Recommendations

The aim of this section is to highlight several recommendations, based on the contributions made (Section 8.5) to AUIs for ERP systems. These recommendations are made in terms of their applicability to theory, practice and future research.

8.6.1 Recommendations for Theory

Theoretical recommendations are made with regard to the thesis statement in three main areas. These areas are illustrated in Figure 8.6.

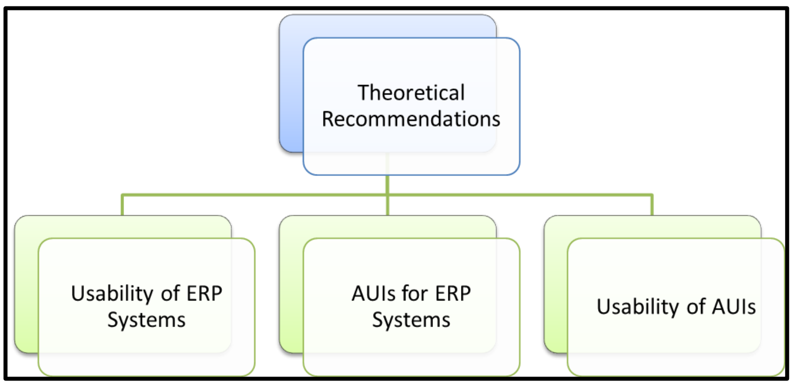


Figure 8.6: Theoretical Recommendations of Research

8.6.1.1 Recommendations for Usability of ERP Systems

A set of ERP usability heuristics (Section 3.4.2) was proposed to assist in the identification of usability issues of ERP systems. It is recommended that the proposed heuristics be used in future heuristic evaluations of ERP systems. This would assist in supporting a comparison of the severity of usability issues amongst ERP systems. A second suggestion is that the heuristics be used for other types of ERP systems, such as those designed for medium and large enterprises. This would assist in determining the applicability of the proposed heuristics to a variety of ERP systems.

The proposed heuristics should be used together with existing general heuristics (i.e. Nielsen's ten heuristics) to form a comprehensive set of heuristics that could evaluate both the general and ERP-specific aspects of ERP systems. The use of the proposed heuristics could also provide a common basis for comparing the results of heuristic evaluations for ERP systems. This would also address the issue of inconsistent heuristics when evaluating ERP systems.

8.6.1.2 Recommendation for AUIs for ERP Systems

This research has proposed a system architecture incorporating an AUI for ERP systems (Figure 8.2). It is recommended that the designers of ERP systems wanting to incorporate an AUI refer to the architecture as it has been successfully implemented. The proposed system architecture is very generic and could apply to any ERP system wanting to incorporate an AUI. The proposed system architecture could be specialised to a specific ERP system, in terms of the following (Sommerville 2007):

- Environment specifications:
    - Specialising the architecture, so that it reflects the operating environments and devices on which the AUI is deployed.
- Functional specifications:
    - Specialising the architecture in the application layer and database layer, so that it is representative of the different functional requirements.
- Process specifications:
    - Specialising the architecture in the application layer (ERP Engine), so that those business processes, which are different or omitted, can be updated and included.

8.6.1.3 Recommendations for Usability of AUIs

An adaptation taxonomy specific to desktop ERP systems was proposed to support the implementation of content, presentation and navigation adaptation (Section 6.3.1). Only three adaptation techniques were implemented, namely Inserting and Removing Fragments (content adaptation), Disabling Fragments (presentation adaptation), and Guidance (navigation adaptation) (Table 8.5). These particular adaptation techniques were implemented as they were identified as being the most suitable adaptation techniques to address the identified usability issues of SBO (Table 8.6) in terms of efficiency and learnability (Section 6.4). It is recommended that more adaptation techniques and HCI patterns be implemented (from the proposed adaptation taxonomy), in order to determine how these adaptation techniques and HCI patterns could impact on the usability of ERP systems.

This research has suggested that additional adaptation techniques for presentation adaptation (Removing Fragments and Sorting Controls) be implemented in order improve the efficiency of SBO. This was based on the user feedback from the empirical evaluation (Section 7.6).

8.6.2 Recommendations for Practice

Practical recommendations are made with regard to the thesis statement in three main areas. These areas are illustrated in Figure 8.7.

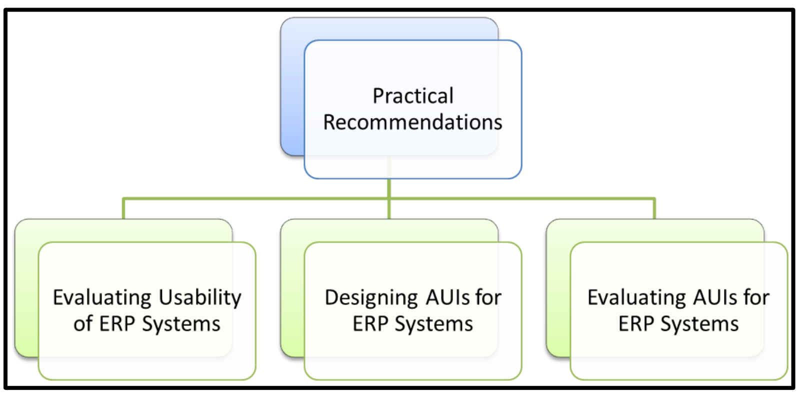


Figure 8.7: Practical Recommendations of Research

8.6.2.1 Recommendations for Evaluating the Usability of ERP systems

A set of ERP heuristics was proposed in Chapter 3. It is suggested that these heuristics be used by usability professionals to assist ERP vendors in the evaluation of their products. This could potentially result in the design and development of more usable ERP systems. The proposed heuristics could also be used by usability professionals to perform comparative heuristic evaluations of existing ERP systems. This would also support a comparison of usability issues amongst similar ERP systems.

8.6.2.2 Recommendation for Designing AUIs for ERP Systems

This research proposed and implemented a system architecture (Figure 8.2) for ERP systems incorporating an AUI (Chapter 6). It is recommended that the proposed system architecture be implemented in this manner to separate the business logic from the user interface logic. This approach would allow for quick and dynamic adaptation to be made to the user interface without affecting the integrity or the structure of the business logic.

This research has successfully shown that SBO can be used to develop an AUI for ERP systems. It is therefore recommended that SBO be used as a suitable experimentation platform for developing AUIs for ERP systems for small enterprises.

8.6.2.3 Recommendation for Evaluating Usability of AUIs

An adaptation taxonomy specific to ERP systems (Figure 8.1) was proposed in Chapter 6. It is recommended that the adaptation techniques and HCI patterns be implemented on several different ERP systems. This could assist in determining which adaptation techniques and HCI patterns would be most applicable to ERP systems for small, medium and large enterprises, or if all of the adaptation techniques and HCI patterns could be applied to different types of ERP systems. The implementation of the adaptation techniques and HCI patterns could also assist in determining which adaptation techniques and HCI patterns are technically and practically feasible to implement on new and existing ERP systems.

8.6.3 Recommendations for Future Research

Several opportunities exist for future research. Some of these possibilities are discussed below.

Future research could include a comparative study that makes use of the proposed ERP heuristics to evaluate the usability of two of more ERP systems (of a similar nature). The results from the study could assist in determining the extent to which the heuristics support the comparison of usability issues and whether any modifications need to be made to the proposed heuristics.

Only three of the seven proposed adaptation techniques were implemented from the proposed adaptation taxonomy, namely Inserting and Removing Fragments, Disabling Fragments and

Guidance (Section 6.4). Future research could be conducted to determine the extent to which the proposed adaptation techniques and HCI patterns might address the usability issues of other ERP systems. Further research could also be conducted to determine whether more adaptation techniques and HCI patterns should be included to enhance the proposed adaptation taxonomy.

The implemented adaptation techniques (Table 8.5) did not contribute to significantly improving the efficiency of SBO. Future research could involve the implementation of additional adaptation techniques and HCI patterns from the proposed taxonomy. It was suggested that the Removing Fragments and Sorting Controls adaptation techniques be implemented to determine whether an improvement in efficiency could be obtained for SBO (Section 7.6). These particular adaptation techniques were suggested as they could address the feedback provided by the users from the empirical evaluation (Section 7.4.1).

The results from the empirical evaluation showed that an empirical evaluation (comprising a user study) could be used to successfully evaluate the usability of an AUI for SBO. It is therefore recommended that future research, which involves the usability evaluation of AUIs for ERP systems adopt this approach.

Only 22 participants were used for the empirical evaluation. Future research could conduct an empirical evaluation using a larger sample size. This could potentially assist in acquiring more precise statistics.

Another recommendation would be to conduct the empirical evaluation with actual users of SBO, provided that the environmental constraints (discussed in Section 8.4) can be overcome.

8.7 Concluding Remarks

This research has successfully met its objectives. The outcomes of this research have resulted in a theoretical contribution to the field of AUIs for ERP systems, as well as a practical contribution to the domain of ERP systems for small enterprises. Future work could investigate some of the outlined possibilities, in order to provide improvements to the proposed adaptation taxonomy, system architecture and the proposed ERP heuristics.

Considering the literature review, the empirical study and the discussion presented in this thesis, it may be concluded that AUIs can be designed to improve the usability of ERP systems for small enterprises.

References

Abran, A., Khelifi, A., Suryn, W. and Seffah, A. (2003): Usability Meanings and Interpretations in ISO Standards. *Software Quality Journal*, **11** (4):325-338. November 2003.

Ahmed, S. and Ashraf, G. (2007): Model-based user interface engineering with design patterns. *The Journal of Systems & Software*, **80** (8):1408-1422. August 2007.

Alexandre, S., Renault, A. and Habra, N. (2006): OWPL: A Gradual Approach for Software Process Improvement In SMEs. In *Proceedings of 32nd EUROMICRO Conference on Software Engineering and Advanced Applications*, Cavtat, Dubrovnik:328 - 335. IEEE Computer Society.

Álvarez-Cortés, V., Zayas-Pérez, B.E., Zárate-Silva, V.H. and Uresti, J.A.R. (2007): Current Trends in Adaptive User Interfaces: Challenges and Applications. In *Proceedings of Electronics, Robotics and Automotive Mechanics Conference*, Morelos, Mexico:312 - 317 IEEE Computer Society.

Babaian, T., Lucas, W. and Topi, H. (2004): Collaborating to Improve ERP Usability. In *Proceedings of International Conference on Enterprise Information Systems*, Porto, Portugal:164-168.

Barnett, V. (2003): *Sample Survey Principles and Methods*. Third Edn, John Wiley & Sons, Inc. 978-0470685907

Barnum, C.M. (2002): *Usability Testing and Research*. Longman. 0-205-31519-4.

Benyon, D. R. (1996): Domain models for user interface design. In *Critical issues in user interface systems engineering*:3-19. D.R. Benyon and P. Palanque (Eds.) London, Springer-Verlag.

Benyon, D.R., Innocent, P. and Murray, D.M. (1987): *System adaptivity and the modelling of stereotypes*. Great Britian, National Physical Laboratory, Division of Information Technology and Computing.

Botta-Genoulaz, V. and Millet, P.A. (2005): A classification for better use of ERP systems. *Computers in industry*, **56** (6):573-587.

Botta-Genoulaz, V. and Millet, P.A. (2006): An investigation into the use of ERP systems in the service sector. *International Journal of Production Economics*, **99** (1-2):202-221. January-February 2006.

Boudreau, M.C. (2003): Learning to Use ERP Technology: a Causal Model. In *Proceedings of 36th Hawaii International Conference on System Sciences*, Big Island, Hawaii:1-19. IEEE Computer Society.

Brun, M.H. and Lanng, C. (2006): Reducing barriers for e-business in SME's through an open service oriented infrastructure. In *Proceedings of 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the Internet* Fredericton, New Brunswick, Canada:403-410. ACM.

Bruque, S. and Moyano, J. (2007): Organisational determinants of information technology adoption and implementation in SMEs: The case of family and cooperative firms. *Technovation*, **27** (5):241-253. May 2007.

Brusilovsky, P. (1996): Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction* **6**(2-3):87-129. July 1996.

Brusilovsky, P. and Cooper, D.W. (2002): Domain, task, and user models for an adaptive hypermedia performance support system. In *Proceedings of 7th international conference on Intelligent user interfaces* San Francisco, California, USA:23-30. ACM.

Bueno, S. and Salmeron, J.L. (2008): Fuzzy modeling Enterprise Resource Planning tool selection. *Computer Standards and Interfaces,* **30** (3):137-147. March 2008.

Calisir, F. and Calisir, F. (2004): The relation of interface usability characteristics, perceived usefulness, and perceived ease of use to end-user satisfaction with enterprise resource planning (ERP) systems. *Computers in Human Behavior*, **20** (4):505-515. July 2004.

Card, S.K., Moran, T.P. and Newell, A. (1983): *The Psychology of Human-Computer Interaction*. Hillsdale, NJ, Lawrence Erlbaum Associates. 978-0898598599

Cheng, D., Deng, F. and Li, H. (2006): Critical Factors for Successful Implementation of ERP in China. In *Proceedings of IEEE International Conference on e-Business Engineering*, Shanghai, China:358-365. IEEE Computer Society.

Chew, J., Orlov, L.M. and Herbert, L. (2003): *App User Interfaces Still Need Work*. Forrester Research. Available at http://www.forrester.com/ER/Research/Brief/Excerpt/0,1317,16184,00.html.

Chin, J.P., Diehl, V.A. and Norman, K.L. (1988): Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of Conference on Human factors in computing systems*, Washington, D.C.:213-218. ACM.

Chou, S. and Chang, Y. (2008): The implementation factors that influence the ERP (enterprise resource planning) benefits. *Decision Support Systems*, **46** (1):149-157. December 2008.

Clerckx, T., Luyten, K. and Coninx, K. (2005): Dynamo-AID: a Design Process and a Runtime Architecture for Dynamic Model-Based User Interface Development. In *Engineering Human Computer Interaction and Interactive Systems* 77-95. R. Bastide, P. Palanque and J. Roth (Eds.) Berlin, Springer-Verlag.

Constantine, L.L. and Lockwood, L.A.D. (1999): *Software for Use: A Practical Guide to the Models and Methods of Usage-Centred Design*. New York, NY, Addison-Wesley Publishing Co. 978-0201924787.

Costabile, M.F. and Matera, M. (2001): Guidelines for hypermedia usability inspection. *IEEE MultiMedia*, **8** (1):66-69.

da Silva, B.S., Netto, O.A.M. and Barbosa, S.D.J. (2005): Promoting a separation of concerns via closely-related interaction and presentation models. In *Proceedings of 2005 Latin American conference on Human-computer interaction*, Cuernavaca, Mexico:170 - 181. ACM.

Daniels, J., Fels, S., Kushniruk, A., Lim, J. and Ansermino, J.M. (2007): A framework for evaluating usability of clinical monitoring technology. *Journal of Clinical Monitoring and Computing*, **21** (5):323-330. July 2007.

das Neves, D., Fenn, D. and Sulcas, P. (2004): Selection of enterprise resource planning (ERP) systems. *South African Journal of Business Management*, **35** (1):45-52.

Dieterich, H., Malinowski, U., Kühme, T. and Schneider-Hufschmidt, M. (1993): State of the Art in Adaptive User Interfaces. In *Adaptive User Interfaces: Principle and Practice*:13-48. M. Schneider-Hufschmidt, T. Kühme and U. Malinowski (Eds.) Amsterdam: North-Holland.

Dillard, J.F., Ruchala, L. and Yuthas, K. (2005): Enterprise resource planning systems: A physical manifestation of administrative evil. *International Journal of Accounting Information Systems*, **6** (2):107-127. June 2005.

Dix, A., Finlay, J.E., Abowd, G.D. and Beale, R. (2004): *Human-Computer Interaction*. Third Edn, Prentice Hall. 978-0130461094

Edwards, P.J., Moloney, K.P., Jacko, J.A. and Sainfort, F. (2008): Evaluating usability of a commercial electronic health record: A case study. *International Journal of Human-Computer Studies*, **66** (10):718-728. October 2008.

ERPwire.com (2008): *What are the common problems for ERP in SME's?* [online]. Available at http://www.erpwire.com/erp-articles/common-problems-of-erp-sme.htm. [Accessed on January 8 2008].

Feng, S., Liu, M. and Wan, J. (2006): An agilely adaptive User Interface based on Design Pattern. In *Proceedings of 6th International Conference on Intelligent Systems Design and Applications*, Jinan, China:1117 - 1122. IEEE Computer Society.

Findlater, L. and McGrenere, J. (2004): A Comparison of Static, Adaptive, and Adaptable Menus. In *Proceedings of Conference on Human Factors in Computing Systems* Vienna, Austria:89 - 96. ACM.

Findlater, L. and McGrenere, J. (2008): Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces. In *Proceedings of Conference on Human Factors in Computing Systems*, Florence, Italy:1247-1256. ACM.

Findlater, L. and McGrenere, J. (2010): Beyond performance: Feature awareness in personalized interfaces. *International Journal of Human-Computer Studies*, **68** (3):121-137. March 2010.

Folmer, E. and Bosch, J. (2004): Architecting for usability: a survey. *Journal of Systems and Software*, **70** (1-2):61-78. February 2004.

Folmer, E., van Gurp, J.I. and Bosh, J. (2003): A Framework for Capturing the Relationship between Usability and Software Architecture. *Software Process: Improvement and Practice*, **8** (2):67-87. June 2003.

Gajos, K.Z., Czerwinski, M., Tan, D.S. and Weld, D.S. (2006): Exploring the design space for adaptive graphical user interfaces. In *Proceedings of Working Conference on Advanced Visual Interfaces*, Venezia, Italy:201 - 208. ACM.

Gajos, K.Z., Everitt, K., Tan, D.S., Czerwinski, M. and Weld, D.S. (2008): Predictability and accuracy in adaptive user interfaces. In *Proceedings of 26th Conference on Human Factors in Computing Systems*, Florence, Italy:1271-1274. ACM.

Gajos, K.Z. and Weld, D.S. (2004): SUPPLE: Automatically Generating User Interfaces. In *Proceedings of 9th international conference on Intelligent user interfaces*, Funchal, Madeira, Portugal:93-100. ACM.

Gena, C. (2005): Methods and techniques for the evaluation of user-adaptive systems. *The Knowledge Engineering Review*, **20** (1):1-37. December 2005.

Gena, C. (2006): A User-Centered Approach for Adaptive Systems Evaluation. In S. Weibelzahl, A. Paramythis and J. Masthoff (Eds.)*Proceedings of 5th Workshop on User-Centred Design and Evaluation of Adaptive Systems held in conjunction with the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Dublin, Ireland:430-439.

Ghédira, C., Maret, P., Faynb, J. and Rubel, P. (2002): Adaptive user interface customization through browsing knowledge capitalization. *International Journal of Medical Informatics*, **68** (1-3):219-228. December 2002.

Goldstuck, A. (2006): *SME Survery 2006 - Unlock the power of the Entrepreneur*. World Wide Worx.

Gonzalez-Calleros, J.M., Guerrero-Garcia, J., Vanderdonckt, J. and Munoz-Arteaga, J. (2009): Towards Canonical Task Types for User Interface Design. In *Proceedings of 7th Latin American Web Congress*, Meridia, Yucatan, Mexico:63-70. IEEE Computer Society.

Gupta, M. and Kohli, A. (2006): Enterprise resource planning systems and its implications for operations function. *Technovation,* **26** (5-6):687-696. May-June 2006.

Haas, M.W. and Hettinger, L.J. (2001): Current Research in Adaptive Interfaces. *The International Journal of Aviation Psychology* **11** (2):119-121. November 2009.

Harwood, S. (2003): *ERP: The Implementation Cycle*. Butterworth-Heinemann. 978-0585458076.

Hayes, N. (2000): *Doing Psychological Research* Open University Press. 978-0335203796

Henry, P. (1998): *User-Centered Information Design for Improved Software Usability* Artech House Publishers. 978-0890069462

Herbert, L., Ragsdale, J. and Gaynor, E. (2006): *Put Business Applications To The Usability Test - Steps To Improve Packaged, Composite, and Homegrown Applications*. Forrester Research Inc.

Hix, D. and Hartson, H.R. (1993): *Developing User Interfaces: Ensuring Usability Through Product & Process*. John Wiley and Sons, Inc. 978-0471578130.

Hofstee, E. (2006): *Constructing a Good Dissertation: A Practical Guide to Finishing a Masters, MBA or PhD on Schedule*. Johannesburg, South Africa, Exactica. 978-0958500715.

Hollingsed, T. and Novick, D.G. (2007): Usability inspection methods after 15 years of research and practice. In *Proceedings of 25th Annual ACM International Conference on Design of Communication*, El Paso, Texas:249-255. ACM.

Höök, K (2000): Steps to take before intelligent user interfaces become real. *Interacting with Computers,* **12** (4):409-426. February 2000.

Höök, K. (1998): Evaluating the utility and usability of an adaptive hypermedia system. *Knowledge-Based Systems,* **10** (5):311-319. March 1998.

Hornbæk, K. (2006): Current practice in measuring usability: Challenges to usability studies and research. *International Journal of Human-Computer Studies*, **64** (2):79-102. February 2006.

Hornbæk, K. and Law, E. L. (2007): Meta-analysis of correlations among usability measures. In *Proceedings of Conference on Human Factors in Computing Systems* San Jose, California:617-626. ACM.

Hou, M., Zhu, H., Zhou, M. and Arrabito, G.R. (2011): Optimizing Operator-Agent Interaction in Intelligent Adaptive Interface Design: A Conceptual Framework. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, **41** (2):161-178. March 2011.

Howarth, J., Smith-Jackson, T. and Hartson, R. (2009): Supporting novice usability practitioners with usability engineering tools. *International Journal of Human-Computer Studies*, **67** (6):533-549. June 2009.

Huin, S.F. (2004): Managing deployment of ERP systems in SMEs using multi-agents. *International Journal of Project Management*, **22** (6):511-517. August 2004.

Hyde, K.F. (2000): Recognising deductive processes in qualitative research. *Qualitative Market Research: An International Journal*, **3** (2):82-89.

Iansiti, M. (2007): ERP End-User Productivity: A Field Study of SAP and Microsoft, Keystone Strategy, LLC.

ISO (1998): Ergonomic Requirements for Office Work with Visual Display Terminals. Part 11: Guidance on Usability. *9241-11*, International Standards Organisation (ISO). **9241**(11).

ISO (2001): Software Engineering - Product Quality - Part 1: Quality Model. *9126-1*, ISO. **9126**(1).

ISO/IEC (1999): Information technology - Software product evaluation - Part 1: General Overview. *14598-1*, International Standards Organisation (ISO). **14598**(1).

Jameson, A. (2003): Adaptive Interfaces and Agents. In *Human-Computer Interaction Handbook*:305-330. A.J. Jacko and A. Sears (Eds.) Mahwah, USA, Lawrence Erlbaum Associates.

Jaspers, M.W.M. (2009): A comparison of usability methods for testing interactive health technologies: Methodological aspects and empirical evidence. *International Journal of Medical Informatics*, **78** (5):340-353. May 2009.

Jiang, F., Li, J. and Zhu, Z. (2008): A User-centric Task Computing Architecture for Pervasive Computing. In *Proceedings of 3rd International Conference on Pervasive Computing and Applications (ICPCA 2008)*, Alexandria, Egypt:491-496. IEEE Computer Society.

Johnson, R.R., Salvo, M.J. and Zoetewey, M.W. (2007): User-Centered Technology in Participatory Culture: Two Decades "Beyond a Narrow Conception of Usability Testing". *IEEE Transactions on Professional Communication*, **50** (4):320-332. December 2007.

Juristo, N., Moreno, A.M. and Sanchez-Segura, M. (2007a): Analysing the impact of usability on software design. *Journal of Systems and Software*, **80** (9):1506-1516. September 2007.

Juristo, N., Moreno, A.M. and Sanchez-Segura, M. (2007b): Guidelines for Eliciting Usability Functionalities. *IEEE Transactions on Software Engineering*, **33** (11):744-758. October 2007.

Keeble, R.J., Macredie, R.D. and Williams, D.S. (2000): User Environments and Individuals: Experience with Adaptive Interface Agents. *Cognition, Technology & Work*, **2** (1):16-26.

Kjeldskov, J., Skov, M.B. and Stage, J. (2010): A longitudinal study of usability in health care: Does time heal? *International Journal of Medical Informatics*, **79** (6):e135-e143. June 2010.

Knutov, E., De Bra, P. and Pechenizkiy, M. (2009): AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. *New Review of Hypermedia and Multimedia*, **15** (1):5-38. May 2009.

Kottner, J. and Streiner, D.L. (2010): Internal consistency and Cronbach's α: A comment on Beeckman et.al (2010). *International Journal of Nursing Studies*, **47** (7):926-926. July 2010.

Kristiansen, R. (2005): Task-based tailoring of ERP-systems user interface. In *Proceedings of 12th Doctoral Consortium on Advanced Information Systems Engineering*, Porto, Portugal.

Kühme, T. (1993): User-centered approach to adaptive interfaces. *Knowledge-Based Systems*, **6** (4):239-248. December 1993.

Kühme, T. and Schneider-Hufschmidt, M. (1993): Introduction. In *Adaptive User Interfaces: Principles and Practice*. Human Factors in Information Technology. Elsevier Science Publishers.

Kwahk, K. (2006): ERP Acceptance: Organizational Change Perspective In *Proceedings of 39th Hawaii International Conference on System Sciences*, Kauai, Hawaii:172b-172b. IEEE Computer Society.

Langley, P. (1999): User Modeling in Adpative Interfaces. In J. Kay (Ed.)*Proceedings of 7th International Conference on User Modeling*, Banff, Canada:357-370. Springer-Verlag.

Lauesen, S. (2005): *User Interface Design: A Software Engineering Perspective*. Addison-Wesley Publishing Co. 978-0321181435.

Lavie, T. and Meyer, J. (2010): Benefits and costs of adaptive user interfaces. *International Journal of Human-Computer Studies*, **68** (8):508-524. August 2010.

Lavie, T., Meyer, J., Bengler, K. and Coughlin, J.F. (2005): The Evaluation of In-Vehicle Adaptive Systems. In S. Weibelzahl, A. Paramythis and M. J. (Eds.)*Proceedings of Fourth Workshop on Empirical Evaluation of Adaptive Systems, held at the 10th International Conference on User Modeling*, Edinburgh:9-18.

Law, C.C.H. and Ngai, E.W.T. (2007): ERP systems adoption: An exploratory study of the organizational factors and impacts of ERP success. *Information & Management*, **44** (4):418-432. June 2007.

Lea, B., Gupta, M.C. and Yu, W. (2005): A prototype multi-agent ERP system: an integrated architecture and a conceptual framework. *Technovation*, **25** (4):433-441. April 2005.

Lefebvre, E., Lefebvre, L.A. and Roy, M. (1995): Technology Penetration and Cumulative Benefits in SMEs. In *Proceedings of 28th Hawaii Conference on System Sciences*, Maui, Hawaii, **3**:533-541. IEEE Computer Society.

Letsu-Dake, E. and Ntuen, C.A. (2009): A conceptual model for designing adaptive human–computer interfaces using the living systems theory. *Systems Research and Behavioral Science*, **26** (1):15-27. September 2008.

Letsu-Dake, E. and Ntuen, C.A. (2010): A case study of experimental evaluation of adaptive interfaces. *International Journal of Industrial Ergonomics*, **40** (1):34-40. Janauary 2010.

Leung, A., Morisson, S., Wringe, M. and Zou, Y. (2006): Developing an adaptive user interface in eclipse. In *Proceedings of The Eclipse Technology Exchange Workshop at European Conference on Object Oriented Programming*, Nantes, France.

Liang, H. and Xue, Y. (2004): Coping with ERP-related contextual issues in SMEs: a vendor's perspective. *The Journal of Strategic Information Systems*, **13** (4):399-415. December 2004.

Light, B. (2005): Going beyond 'misfit' as a reason for ERP package customisation. *Computers in Industry*, **56** (6):606-619. August 2005.

Liu, J., Wong, C.K. and Hui, K.K. (2003): An Adaptive User Interface Based on Personalised Learning *IEEE Intelligent Systems*, **18** (2):52-57.

López-Jaquero, V., Montero, F., Molina, J. P., Fernández-Caballero, A. and González, P. (2003): Model-Based Design of Adaptive User Interfaces through Connectors. In *Interactive Systems. Design, Specification, and Verification* **2844/2003**:427-437. Springer Berlin / Heidelberg.

López-Jaquero, V., Montero, F., Molina, J.P., González, P. and Fernández-Caballero, A. (2005): A Seamless Development Process of Adaptive User Interfaces Explicitly Based on Usability Properties In *Engineering Human Computer Interaction and Interactive Systems* **3425/2005**:112-117. Springer-Verlag.

Matthews, D. (2008): Usability as an ERP Selection Criteria, IFS. Available at http://ifs.datahost.com/shop/images/wp-usability.pdf.

McGrenere, J., Baecker, R.M. and Booth, K.S. (2002): An evaluation of a multiple interface design solution for bloated software. In *Proceedings of Conference on Human factors in computing systems: Changing our world, changing ourselves*, Minneapolis, Minnesota, USA:164-170. April 20-22. ACM.

McNamara, N. and Kirakowski, J. (2005): Defining usability: quality of use or quality of experience? In *Proceedings of IEEE International Professional Communication Conference*, Limerick, Ireland:200-204. July 10-13, 2005.

Menkhaus, G. and Fischmeister, S. (2003): Dialog Model Clustering for User Interface Adaptation. In *Proceedings of International Conference on Web Engineering (ICWE '03),* Oviedo, Spain, Lecture Notes in Computer Science, **2722**:194-203. Springer-Verlag.

Menkhaus, G. and Pree, W. (2002): A Hybrid Approach to Adaptive User Interface Generation. *Journal of Computing and Information Technology*, **10** (3):171-179.

Microsoft (2000): *Usability in Software Design* [online]. Available at http://msdn.microsoft.com/en-us/library/ms997577(printer).aspx. [Accessed on].

Microsoft (2009a): *Microsoft Dynamics Industry Solutions* [online]. Available at http://www.microsoft.com/dynamics/industry/default.mspx. [Accessed on 10 January 2009].

Microsoft (2009b): *Solutions by business need: Overview* [online]. Available at http://www.microsoft.com/dynamics/businessneeds/solutions_overview.mspx. [Accessed on 14 January 2009].

Molina, M. and Flores, V. (2008): A presentation model for multimedia summaries of behavior. In *Proceedings of 13th international conference on Intelligent user interfaces*, Gran Canaria, Spain:369-372. ACM.

Montabert, C., McCrickard, D.S., Winchester, W.W. and Pérez-Quiñones, M.A. (2009): An integrative approach to requirements analysis: How task models support requirements reuse in a user-centric design framework. *Interacting with Computers*, **21** (4):304-315. August 2009.

Montazemi, A.R. (2006): How they manage IT: SMEs in Canada and the U.S. *Communications of the ACM*, **49** (12):109-112. December 2006.

Morabito, V., Pace, S. and Previtali, P. (2005): ERP Marketing and Italian SMEs. *European Management Journal*, **23** (5):590-598. October 2005.

Nach, H. and Lejeune, A. (2008): Implementing ERP in SMEs: Towards an Ontology Supporting Managerial Decisions. In *Proceedings of International MCETECH Conference on e-Technologies*, Montreal, QC, Canda:223-226. IEEE Computer Society.

Naidoo, P. (2005): ERP and SMEs. *iWeek*, ITWeb. Available at http://www.iweek.co.za/ViewStory.asp?StoryID=156921. [Accessed on 1 January 2008].

Ndiwalana, A. and Tusubira, F.F. (2006): ICT in Small Businesses: How can Academic Networks be more relevant? In *Proceedings of 4th International Conference on Open Access*, Stockholm. December 12-14.

Newman, T. (2007): *Usability: the key to manufacturing ERP success* [online]. Available at http://instrumentation.co.za/article.aspx?pklArticleId=4566&pklCategoryId=71. [Accessed on 6 March 2009].

Ngai, E.W.T., Law, C.C.H. and Wat, F.K.T. (2008): Examining the critical success factors in the adoption of enterprise resource planning. *Computers in Industry*, **59** (6):548-564. August 2008.

Nielsen, J. (1993): *Usability Engineering*. San Francisco, Morgan Kaufmann. 978-0125184069

Nóbrega, L., Nunes, N.J. and Coelho, H. (2005): DialogSketch: dynamics of the canonical prototypes. In *Proceedings of 4th international workshop on Task models and diagrams*, Gdansk, Poland:19-25. ACM.

Ohigashi, Y. and Omori, T. (2006): Modeling of autonomous problem solving process by dynamic construction of task models in multiple tasks environment. *Neural Networks*, **19** (8):1169-1180. October 2006.

Olivier, M.S. (2004): *Information Technology Research - A Practical Guide for Computer Science and Informatics*. Pretoria, Second Edn, Van Schaik Publishers. 978-0627025761.

Olivier, M.S. (2009): *Information Technology Research - A practical guide for Computer Science and Informatics*. Pretoria, Third Edn, Van Schaik Publishers 978-0627027581.

Olsen, K.A. and Sætre, P. (2007): ERP for SMEs – is proprietary software an alternative? *Business Process Management Journal*, **13** (3):379 - 389.

ORACLE (2009a): *Oracle E-Business Suite* [online]. Available at http://www.oracle.com/applications/e-business-suite.html. [Accessed on 14 January 2009].

ORACLE (2009b): *Oracle Industry Solutions* [online]. Available at http://www.oracle.com/industries/index.html. [Accessed on 10 Janurary 2009].

Orlikowski, W.J. and Baroudi, J.J. (1991): Studying Information Technology in Organizations: Research Approaches and Assumptions. *Information Systems Research*, **2** (1):1-28.

Pang, C., Dharmasthira, Y., Eschinger, C. and Motoyoshi, K. (2008): *Market Share: ERP Software, Worldwide, 2007* Gartner, Inc.

Paramythis, A., Weibelzahl, S. and Masthoff, J. (2010): Layered evaluation of interactive adaptive systems: framework and formative methods. *User Modeling and User-Adapted Interaction* **20** (5):383-453.

Park, J. and Han, S.H. (2011): Complementary menus: Combining adaptable and adaptive approaches for menu interface. *International Journal of Industrial Ergonomics*, **41** (3):305-316. May 2011.

Paternò, F. (1999): ConcurTaskTrees: An Engineered Approach to Model-based Design of Interactive Systems. In *The Handbook of Analysis for Human-Computer Interaction*:1-18. London, Lawrence Erlbaum Associates.

Paymans, T.F., Lindenber, J. and Neerincx, M. (2004): Usability Trade-Offs for Adaptive User Interfaces: Ease of Use and Learnability. In *Proceedings of 9th international conference on Intelligent user interfaces*, Funchal, Madeira, Portugal:301-303. ACM.

Peng, Q., Chung, C., Yu, C. and Luan, T. (2007): A networked virtual manufacturing system for SMEs. *International Journal of Computer Integrated Manufacturing*, **20** (1):71-79. February 2007.

Peng, X. and Silver, D.L. (2007): Interface Adaptation Based on User Expectation. In *Proceedings of 21st International Conference on Advanced Information Networking and Applications Workshops*, Niagara Falls, Ontario, Canada:264 - 269. IEEE Computer Society.

Preece, J., Rogers, Y. and Sharp, H. (2007): *Interaction Design Beyond Human-Computer Interaction*. Second Edn, John Wiley & Sons, Inc. 978-0471492788.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. and Carey, T. (1994): *Human-Computer Interaction*. First Edn, Addison-Wesley Publishing Co.

Puerta, A.R. (1993): The study of models of intelligent interfaces. In W.D. Gray, W.E. Hefley and D. Murray (Eds.)*Proceedings of 1st international conference on Intelligent user interfaces*:71-78. ACM.

Puerta, A.R. (1997): A model-based interface development environment *IEEE Software*, **14** (4):40-47. July/August 1997.

Puerta, A.R., Eriksson, H., Gennari, J.H. and Musen, M.A. (1994): Beyond data models for automated user interface generation. In G. Cockton, S.W. Draper and G.R.S. Weir (Eds.)*Proceedings of British Conference on Human-Computer Interaction*, Glasgow, United Kingdom:353-366. Cambridge University Press.

Ragowsky, A. and Gefen, D. (2008): What makes the competitive contribution of ERP strategic. *ACM SIGMIS Database*, **39** (2):33-49. May 2008.

Ramachandran, K. (2009): *Adaptive user interfaces for health care applications* [online]. Available at http://www.ibm.com/developerworks/web/library/wa-uihealth/. [Accessed on].

Reichenbacher, T. (2004): *Mobile Cartography – Adaptive Visualisation of Geographic Information on Mobile Devices* The Instituite for Photogrammetry and Cartography, Munich Technical University. Munich, Germany.

SAGE (2009): *Our Products and Services* [online]. Available at http://www.sage.com/ourbusiness/aboutus/ourproductsandservices. [Accessed on 10 January 2009].

SAP (2004): *User Day Toolkit* [online]. Available at http://www.sapdesignguild.org/resources/user_day_toolkit/index.htm. [Accessed on 24 March 2009].

SAP (2009a): *Business Solutions by Industry | Enterprise Solutions and Services* [online]. Available at http://www36.sap.com/industries/index.epx. [Accessed on 10 January 2009].

SAP (2009b): *Enterprise Resource Planning (ERP) Software from SAP* [online]. Available at http://www.sap.com/solutions/business-suite/erp/index.epx. [Accessed on 14 January 2009].

Scott, J.E. (2008): Technology acceptance and ERP documentation usability. *Communications of the ACM*, **51** (11):121-124.

SEDA (2008): *Small Enterprises as Users and Suppliers of Information on the ICT Sector*. Small Enterprise Development Agency (SEDA). 978-0980273359.

Seffah, A. (2010): The evolution of design patterns in HCI: from pattern languages to pattern-oriented design. In *Proceedings of 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems* Berlin, Germany:4-9. ACM.

Seffah, A., Donyaee, M., Kline, R.B. and Padda, H.K. (2006): Usability measurement and metrics: A consolidated model *Software Quality Journal*, **14** (2):159-178.

Seffah, A., Kececi, N. and Donyaee, M. (2001): QUIM: a framework for quantifying usability metrics in software quality models. In *Proceedings of Second Asia-Pacific Conference on Quality Software*, Hong Kong, China:311-318. IEEE Computer Society.

Şen, C.G., Baraçli, H., Şen, S. and Başlıgil, H. (2009): An integrated decision support system dealing with qualitative and quantitative objectives for enterprise software selection. *Expert Systems with Applications*, **36** (3):5272-5283. April 2009.

Seymour, L.F. and Roode, J.D. (2008): Investigating affective response and job impact with ERP adoption. *South African Computer Journal*, **40**:74-82.

Shackel, B. (1991): Usability - Context, framework, definition, design and evaluation. In *Human Factors for Informatics Usability*:21-38. B. Shackel and S. Richardson (Eds.) Cambridge, Cambridge University Press.

Shneiderman, B. (1992): *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, MA., Second Edn, Addison-Wesley Publishing Co.

Shneiderman, B. (1998): *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, MA, Third Edn, Addison-Wesley Publishing Co.

Shoniregun, C.A. (2004): Classification and taxonomy of TEISMEs. In J. Janssen, H.G. Sol and R.W. Wagenaar (Eds.)*Proceedings of 6th international conference on Electronic commerce* Delft, The Netherlands:196-200. ACM.

Singh, A. and Wesson, J. (2011): The Design of Adaptive Interfaces for Enterprise Resource Planning Systems. *13th International Conference on Enterprise Information Systems*. R. Zhang, J. Cordeiro, X. Li, Z. Zhang and J. Zhang. Beijing, China, SciTePress. **4**: 281-286. [Accessed on 8 -11 June].

Singh, A. and Wesson, J.L. (2009): Evaluation criteria for assessing the usability of ERP systems. In *Proceedings of In Proceedings of the 2009 Annual Research Conference of the South African institute of Computer Scientists and information Technologists (SAICSIT '09)*, Vanderbijlpark, Emfuleni, South Africa: 87-95. October 12 - 14. ACM.

Sommerville, I (2007): *Software Engineering*. Eight Edn, Addison-Wesley Publishing Co. 978-0321313799.

Sprent, P. and Smeeton, N.C. (2007): *Applied Nonparametric Statistical Methods*. Fourth Edn, Chapman & Hall/CRC. 978-1584887010

Stephanidis, C., *et al.* (1998): Adaptable and Adaptive User Interfaces for Disabled Users in the AVANTI Project. In S. Trigila, A. Mullery, M. Campolargo, H. Vanderstraeten and M. Mampaey (Eds.)*Proceedings of 5th International Conference on Intelligence in Services and Networks*, Antwerp, Belgium:153-166. Springer-Verlag.

Strachan, L., Anderson, J., Sneesby, M. and Evans, M. (2000): Minimalist User Modelling in a Complex Commercial Software System *User Modeling and User-Adapted Interaction*, **10** (2-3):109-145.

Svensson, G. (2009): A counter-intuitive view of the deductive research process: Clockwise versus anti-clockwise approaches. *European Business Review*, **21** (2):191 - 196.

Te'eni, D., Carey, J.M. and Zhang, P. (2006): *Human-Computer Interaction: Developing Effective Organizational Information Systems*. John Wiley & Sons, inc. 978-0471677659.

te Brake, G., de Greef, T., Lindenberg, J., Rypkema, J. and Smets, N. (2006): Developing adaptive user interfaces using a game-based simulation environment. In F.F.B. Van de Walle and M. Turoff (Eds.)*Proceedings of 3rd International Information Systems for Crisis Response and Management and Newark Conference*, Newark, NJ.

Tidwell, J (2011): *Designing Interfaces*. Second Edn, O'Reilly Media, Inc. 978-1449379704

Topi, H., Lucas, W. and Babaian, T. (2005): Identifying usability issues with an ERP implementation. In *Proceedings of 7th International Conference on Enterprise Information Systems*, Miami, USA:128-133.

Tran, V., Vanderdonckt, J., Kolp, M. and Faulkner, s. (2009): Generating User Interface from Task, User and Domain Models. In *Proceedings of 2nd International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services*, Porto, Portugal:19-26.

Trimi, S. (2008): ICT for small and medium enterprises *Service Business* **2**(4):271-273.

Tsandilas, T. and Schraefel, M.C. (2004): Usable adaptive hypermedia systems. *New Review of Hypermedia and Multimedia* **10** (1):5-29.

Tsandilas, T. and Schraefel, M.C. (2005): An empirical assessment of adaptation techniques. In *Proceedings of Extended Abstratcts from the Conference on Human factors in computing systems*, Portland, USA. ACM.

Tselios, N., Avouris, N. and Komis, V. (2008): The effective combination of hybrid usability methods in evaluating educational applications of ICT: Issues and challenges. *Education and Information Technologies*, **13** (1):55-76.

Tullis, T. and Albert, W. (2008): *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Morgan Kaufmann. 978-0123735584.

Umble, E.J., Haft, R.R. and Umble, M.M. (2003): Enterprise resource planning: Implementation procedures and critical success factors. *European Journal of Operational Research*, **146** (2):241-257. April 2003.

Van den Bergh, J. and Coninx, K. (2004): Model-based design of context-sensitive interactive applications: a discussion of notations. In *Proceedings of 3rd annual conference on Task models and diagrams* Prague, Czech Republic:43 - 50. ACM.

Van Nieuwenhuyse, I., De Boeck, L., Vandaele, N. and Lambrecht, M. (2007): From ERP to Advanced Resource Planning: Improving Operational Performance by Getting the Inputs Right. In R.H. Sprague (Ed.)*Proceedings of 40th Hawaii International Conference on System Sciences*, Waikoloa, Big Island, Hawaii:1-9. IEEE Computer Society.

van Tonder, B. and Wesson, J. (2008): Using adaptive interfaces to improve mobile map-based visualisation. In *Proceedings of 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*, Wilderness Beach Hotel, Wilderness, South Africa:257-266. ACM.

van Velsen, L., van der Geest, T., Klaassen, R. and Steenhouder, M. (2008): User-centered evaluation of adaptive and adaptable systems: A literature review. *The Knowledge Engineering Review*, **23** (3):261-281. September 2008.

Vasilyeva, E., Pechenizkiy, M. and Puuronen, S. (2005): Towards the Framework of Adaptive User Interfaces for eHealth. In *Proceedings of 18th IEEE Symposium on Computer-Based Medical Systems*, Dublin, Ireland:139 - 144. IEEE Computer Society.

Verville, J. and Halingten, A. (2003): A six-stage model of the buying process for ERP software. *Industrial Marketing Management*, **32** (7):585-594. October 2003.

Vidani, A.C. and Chittaro, L. (2009): Using a Task Modeling Formalism in the Design of Serious Games for Emergency Medical Procedures. In *Proceedings of Conference in Games and Virtual Worlds for Serious Applications*, Coventry, United Kingdom:95-102. IEEE Computer Society.

Vilpola, I. and Kouri, I. (2006): Improving ERP Requirement Specification Process of SMEs with a Customer-Centered Analysis Method. In *Proceedings of Frontiers of e-Business Research*:140-151.

Vilpola, I., Kouri, I. and Väänänen-Vainio-Mattila, K. (2007): Rescuing Small and Medium-sized Enterprises from Inefficient Information Systems – a Multi-disciplinary Method for ERP System Requirements. In *Proceedings of 40th Hawaii International Conference on System Science*, Waikoloa, Big Island, Hawaii:1530-1605. IEEE Computer Society.

Wang, E.T.G., Klein, G. and Jiang, J.J. (2006): ERP Misfit: Country of Origin and Organizational Factors. *Journal of Management Information Systems* **23** (1):263-292.

Watkins, A.E., Scheaffer, R.L. and Cobb, G.W. (2010): *Statistics: From Data to Decision*. Second Edn, John Wiley & Sons, Inc. 978-0470458518

Weber, R. (2004): Editor's comments: the rhetoric of positivism versus interpretivism: a personal view. *MIS Quarterly*, **28** (1):iii-xii. March 2004.

Wei, C., Chien, C. and Wang, M.J. (2005): An AHP-based approach to ERP system selection. *International Journal of Production Economics*, **96** (1):47-62. April 2005.

Winter, J., Rönkkö, K., Ahlberg, M., Hinely, M. and Hellman, M. (2007): Developing Quality through Measuring Usability – The UTUM Test Package. In *Proceedings of International Conference on Software Engineering 5th Workshop on Software Quality*, Minneapolis, Minnesota:2-2. IEEE Computer Society.

Witte, R.S. and Witte, J.S. (2009): *Statstics*. Ninth Edn, John Wiley & Sons, Inc. 978-0470392225.

Wixon, D. and Wilson, C. (1997): The usability engineering framework for product design and evaluation. In *Handbook of Human–Computer Interaction*:653-688. M.G. Helander, T.K. Landauer and P. Prabhu (Eds.) Amsterdam, The Netherlands, Elsevier.

Wu, J., Shin, S. and Heng, M.S.H. (2007): A methodology for ERP misfit analysis. *Information & Management*, **44** (8):666-680. December 2007.

Wu, J. and Wang, Y. (2002): Development of a Tool for Measuring Key-User Satisfaction in an ERP Outsourcing Environment. In *Proceedings of 6th Pacific Asia Conference on Information Systems,* Atlanta, USA:555-568.

Wu, L., Ong, C. and Hsu, Y. (2008): Active ERP implementation management: A Real Options perspective. *Journal of Systems and Software*, **81** (6):1039-1050. June 2008.

Yang, J., Wu, C. and Tsai, C. (2007): Selection of an ERP system for a construction firm in Taiwan: A case study. *Automation in Construction*, **16** (6):787-796. September 2007.

Yeh, J. (2006): Evaluating ERP Performance from User Perspective. In *Proceedings of IEEE Asia-Pacific Conference on Services Computing*, Guangzhou, Guangdong, China:311-314. December 12-15. IEEE Computer Society.

Yen, G.G. and Acay, D. (2009): Adaptive user interfaces in complex supervisory tasks. *ISA Transaction*, **48** (2):196-205. April 2009.

Zarikas, V. (2007): Modeling decisions under uncertainty in adaptive user interfaces. *Universal Access in the Information Society*, **6** (1):87-101. April 2007.

Zou, Y., Lerner, M., Leung, A., Morisson, S. and Wringe, M. (2008): Adapting the User Interface of Integrated Development Environments (IDEs) for Novice Users. *Journal of Object Technology*, **7** (7):55-74. September - October 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* **13** (2):243-270.

Appendix A: ERP Industry Sector and Sub-Sectors

ERP Industry Sector and Sub-Sectors	
Industry Sector	*Industry Sub-Sector*
Financial and Public Services	Banking
	Defence & Security
	Healthcare
	Higher Education & Research
	Insurance
	Public Sector
Manufacturing	Aerospace & Defence
	Automotive
	Chemicals
	Consumer Goods
	Engineering, Construction & Operations
	High Technology
	Industrial Machinery & Components
	Life Sciences
	Mill Products
	Mining
	Oil & Gas
Services	Media
	Retail
	Telecommunications
	Professional Services
	Travel & Transportation
	Utilities
	Wholesale Distribution
Government	
Non-Profit Organisations	

ERP Industry Sectors and Sub-Sectors (Microsoft 2009a; ORACLE 2009b; SAGE 2009; SAP 2009a)

Appendix B: ERP Vendor SME Sales Questionnaire

Nelson Mandela Metropolitan University

for tomorrow

Department of Computing Sciences

SME ERP SALES PATTERNS QUESTIONNAIRE

October 2008

General Instructions	
No.	*Instruction*
1.	This questionnaire attempts to gather insights into the sales and sales patterns of ERP systems to small and medium enterprises (SMEs) in South Africa.
2.	The information gathered in this questionnaire is for research purposes with the intention of informing a PhD study pursued at the Nelson Mandela Metropolitan University.
3.	This questionnaire consists of 6 sections: Biographical Details Size & Sector Business Processes Modules Costs General
4.	The above mentioned sections address a total of 26 questions.
5.	Please complete each question to the best of your knowledge.
6.	Please complete all questions in ink with a blue or black ballpoint pen and not pencil.
7.	Indicate your response to a question by placing an X in the space provided (where required).
8.	Any errors made should not be Tipp-Exed but should rather be crossed out and signed.
9.	For further enquiries please feel free to contact me (Akash Singh) at: (Cell): 082 – 882 6128 (Email): Akash.Singh@nmmu.ac.za
10.	All the information provided within this questionnaire will remain completely confidential and no reference will be made to any individual or their responses.
11.	Thank you for taking the time to complete this questionnaire.

<table>
<tr><th colspan="6">Section A: Biographical Details</th></tr>
<tr><td>Category</td><td colspan="5">Details</td></tr>
<tr><td>Name:</td><td colspan="5"></td></tr>
<tr><td>Surname:</td><td colspan="5"></td></tr>
<tr><td>Age:</td><td>20-29</td><td>30-39</td><td>40-49</td><td>50-59</td><td>60+</td></tr>
<tr><td>Gender:</td><td colspan="2">Male</td><td colspan="3">Female</td></tr>
<tr><td>Company:</td><td colspan="5"></td></tr>
<tr><td>Position:</td><td colspan="5"></td></tr>
<tr><td>Date:</td><td colspan="5"></td></tr>
</table>

<table>
<tr><th colspan="2">Section B: Size and Sector</th></tr>
<tr><td colspan="2">1. In which sectors do SMEs typically purchase ERP systems? (Select more than one option)</td></tr>
<tr><td>Agriculture</td><td></td></tr>
<tr><td>Mining and Quarrying</td><td></td></tr>
<tr><td>Manufacturing</td><td></td></tr>
<tr><td>Electricity, Gas and Water</td><td></td></tr>
<tr><td>Construction</td><td></td></tr>
<tr><td>Trade (Retail and Wholesale)</td><td></td></tr>
<tr><td>Transport, Storage and Communications</td><td></td></tr>
<tr><td>Services (Finance, Business, Community, Social and Personal)</td><td></td></tr>
</table>

2. What size (employees) of SMEs typically purchase ERP systems? (select more than one option)	
0-5	
6-20	
21-50	
50-100	
100-150	
150-200	
3. What are the top 3 SME sectors that purchase ERP systems? (Rank your response starting from 1 as the most predominant and 3 as the least predominant)	
Agriculture	
Mining and Quarrying	
Manufacturing	
Electricity, Gas and Water	
Construction	
Trade (Retail and Wholesale)	
Transport, Storage and Communications	
Services (Finance, Business, Community, Social and Personal)	
4. What are the top 3 SMEs (size-employees) that purchase ERP systems? (Rank your response starting from 1 as the most predominant and 3 as the least predominant)	
0-5	
6-20	
21-50	
50-100	
100-150	
150-200	

5. What is the average age (years) of SMEs who purchase an ERP system? (Select an option)	
0-5	
6-10	
10-15	
16-20	
20+	
6. What are the main reasons why SMEs purchase an ERP system? (Select more than one system)	
Integration of core business activities	
Increased productivity	
Support informed decision making	
Support the transformation from inefficient business processes to best practice business processes	
Automation of certain business process	
Improved competitiveness	
Reduced costs	
Increased ability to deploy new information system functionality	
Supply chain integration between small enterprises and large enterprises	
Timely access to management information	

Section C: Business Processes			
7. Which business processes do SMEs usually automate through the implementation of an ERP system? (Select more than one option)			
Sales and Marketing		Operations	
Compliance		Administration	
Finance and Accounting		Procurement	
Human Resources		Logistics	
Inventory		Production & Manufacturing	
8. Which top 3 business processes do SMEs automate through the implementation of an ERP system? (Rank your response starting from 1 as the most predominant and 3 as the least predominant)			
Sales and Marketing		Operations	
Compliance		Administration	
Finance and Accounting		Procurement	
Human Resources		Logistics	
Inventory		Production & Manufacturing	
9. Which business processes do SMEs not automate? (Select more than one option)			
Sales and Marketing		Operations	
Compliance		Administration	
Finance and Accounting		Procurement	
Human Resources		Logistics	
Inventory		Production & Manufacturing	

10. Which business processes for SMEs require the most automation? (Select one option)			
Sales and Marketing		Operations	
Compliance		Administration	
Finance and Accounting		Procurement	
Human Resources		Logistics	
Inventory		Production & Manufacturing	

Section D: Modules

11. Which ERP modules do SMEs typically purchase? (Select more than one option)			
Administration		Inventory	
Human Resources		Business Intelligence	
Financials		Production	
E-Commerce		Project Management	
Sales		MRP	
Manufacturing		Purchasing	
Services		SCM	
Banking		Report	
CRM		Portal	

12. Which ERP module is the most complex to use? (Select one option)			
Administration		Inventory	
Human Resources		Business Intelligence	
Financials		Production	
E-Commerce		Project Management	
Sales		MRP	
Manufacturing		Purchasing	
Services		SCM	
Banking		Report	
CRM		Portal	
13. Which ERP module takes the longest to learn? (Select one option)			
Administration		Inventory	
Human Resources		Business Intelligence	
Financials		Production	
E-Commerce		Project Management	
Sales		MRP	
Manufacturing		Purchasing	
Services		SCM	
Banking		Report	
CRM		Portal	

14. Which ERP module do users enjoy the most? (Select one option)			
Administration		Inventory	
Human Resources		Business Intelligence	
Financials		Production	
E-Commerce		Project Management	
Sales		MRP	
Manufacturing		Purchasing	
Services		SCM	
Banking		Report	
CRM		Portal	

Section E: Costs

15. How much are SMEs willing to spend (Rands) on an ERP system in terms of purchase costs?	
<R1000	
R1000-R3999	
R4000-R6999	
R7000-R9999	
>R10000	

16. How much does your base ERP system cost in terms of initial purchase costs? (Select one option)	
<R1000	
R1000-R3999	
R4000-R6999	
R7000-R9999	
>R10000	
17. How much does your ERP system cost to support annually? (Select one option)	
<R1000	
R1000-R3999	
R4000-R6999	
R7000-R9999	
>R10000	
18. How much does an annual single user licence cost? (Select one option)	
<R1000	
R1000-R3999	
R4000-R6999	
R7000-R9999	
>R10000	
19. How much does training cost per user? (Select one option)	
<R1000	
R1000-R3999	
R4000-R6999	
R7000-R9999	
>R10000	

Section F: General
Please add any relevant comments that would assist in addressing factors affecting the sales of ERP systems to SMEs in South Africa

Appendix C: Common Usability Attributes

Common Usability Attributes					
Author	*Usability Attributes*				
Shackel (1991)	Effectiveness (Speed)	Learnability (time to learn)	Learnability (Retention)	Effectiveness (Errors)	Attitude
Shneiderman (1992)	Speed of Performance	Time to Learn	Retention over time	Rate of Errors by Users	Subjective Satisfaction
Hix and Hartson (1993)	Long-term performance	Learnability	Retainability		Long-term User Satisfaction
Nielson (1993)	Efficiency of Use	Learnability	Memorability	Errors / Safety	Satisfaction
Preece, Rogers, Sharp, Benyon, Holland and Carey. (1994)	Throughput	Learnability		Throughput	Attitude
Wixon and Wilson (1997)	Efficiency	Learnability	Memorability	Error Rates	Satisfaction
Shneiderman (1998)	Speed of Performance	Time to learn	Retention over Time	Rate of Errors by Users	Subjective Satisfaction
ISO 9241-11(1998)	Efficiency				Satisfaction
Constantine and Lockwood(1999)	Efficiency in Use	Learnability	Memorability	Reliability in Use	User Satisfaction
ISO 9126 (2001)	Operability	Learnability		Operability	Attractiveness
Dix, Finlay, Abowd and Beale (2004)		Learnability			
Seffah, Donyaee, Kline and Padda (2006)	Efficiency	Learnability		Safety	Satisfaction
Preece, Rogers and Sharp (2007)	Efficiency	Learnability	Memorability	Safety	Utility

Appendix D: ERP Heuristic Comparison

ERP Heuristic Comparison										
	Matthews (2008)	Topi, Lucas and Babaian (2005)	Herbert, Ragsdale and Gaynor (2006)	Scott (2008)	SAP Design Guild (2004)	Keystone (2007)	Wu and Wang (2002)	Newman (2007)	Yeh (Yeh 2006)	Calisir and Calisir (2004)
General Heuristics										
Visibility of system status										
Match between system and real world	✓									✓
User control and freedom		✓				✓				
Consistency and standards	✓									
Error Prevention										
Recognition rather than recall			✓			✓				
Flexibility and efficiency of use					✓		✓	✓		✓
Aesthetics and minimalist design										
Help recognising, recovering and diagnosing errors		✓				✓				✓
Good help and documentation		✓		✓	✓					✓

	Matthews (2008)	Topi, Lucas and Babaian (2005)	Herbert, Ragsdale and Gaynor (2006)	Scott (2008)	SAP Design Guild (2004)	Keystone (2007)	Wu and Wang (2002)	Newman (2007)	Yeh (Yeh 2006)	Calisir and Calisir (2004)
ERP Specific Heuristics										
Ease of Customisation (Customise effectively and efficiently with greatest re-use)	✓		✓				✓			✓
Ease of Navigation (Consistent and Understandable Navigation)	✓		✓	✓				✓		
Ease of Searching (Simple and Efficient Searching)	✓									
Identification and Access to correct information		✓	✓							
Appropriate System Output		✓					✓	✓	✓	
Appropriate use of Terminology		✓								
Appropriate presentation & Layout of screen			✓	✓						
Application Performance and Stability			✓				✓			✓
Ease of Learning		✓		✓	✓	✓	✓			

Appendix E: Small Enterprise Interview Study Questionnaire

Department of Computing Sciences

Small Enterprise Interview Study Questionnaire

Objectives
The purpose of this research is to identify possible usability problems that could exist with SAP Business One (SBO). Both views of the technology driver and the users of SBO are required in order to obtain a holistic view on the usage, of SBO, within an enterprise. Note that this research is not a reflection on the user but rather an indication of the user's perspective of SBO.

Questions
Technology Driver
Why did your enterprise decide to purchase SBO?
How many users do you currently have on SBO?
How many departments are currently and actively using SBO?
What do your users like about SBO?
What do your users dislike about SBO?
What was the biggest challenge in terms of initiating and using SBO within your enterprise?
What would you like to see in SBO that will allow your users to use it more proficiently?
Does SBO ensure an accurate alignment between itself and your business processes, SBO and your users?
Why did your enterprise decide to purchase SBO?

Questions
Business User
What do you use SBO for?
What are your daily tasks?
How well does SBO align to the way in which your enterprise requires tasks to be executed and completed?
What do you like about SBO?
What do you dislike about SBO?
What is your biggest challenge in using SBO?
What do you think about SBO in terms of identifying and accessing the correct information and functionality of the system?
Does the presentation of the screen support the way to execute and complete your tasks and transactions?
Is the output that you receive from SBO easy to obtain, informative, easy to interpret and useful?
Was it easy to learn how to use SBO (If not, why?)?
Is the help that comes with SBO sufficient?
Does SBO support customization in terms of how you work?
What would you like to see in SBO that will allow your users to use it more proficiently?

Appendix F: Heuristic Evaluation Scenario

Nelson Mandela Metropolitan University

for tomorrow

Department of Computing Sciences

HEURISTIC EVALUATION SCENARIO

Scenario
Dearest Participant *You have been identified as an expert in the field of Human Computer Interaction and Usability. Your assistance is required to heuristically evaluate SAP Business One. The aim of this evaluation is to identify potential usability problems with the system (SAP Business One). Accompanied with this document is a heuristic evaluation checklist, to aid your evaluation. Please complete the heuristic evaluation checklist and feel free to make any comments, positive or negative, in the space provided. A scenario is provided comprising two tasks (process a purchase order and create a customer) in the pages to follow as a guide to interacting and evaluating the system. You are welcome to explore SAP Business One in order to gain a better understanding of how it works.* *Good luck!*

Test Scenario

You are the accountant at OEC Computers. Part of your daily tasks involves processing purchase orders. This task of processing a purchase order is done on the company's enterprise resource planning (ERP) system – SAP Business One. Another responsibility, of yours, within in the company is to also ensure that any new or potential customer is captured on the system (SAP Business One).

Task 1: Processing Purchase Orders

1) You are required to purchase 3 pairs of boxing gloves from Athletic Sportware & Co. The buyer for these items was Jane Smith. Jane has also mentioned to you that she receives a 5% discount from Athletic Sportware & Co. when placing orders.

2) You are required to create a purchase order for 4 sport-shirts from Teamware. The buyer responsible for this order is Andy Step.

3) You were requested by buyer David Batra that the purchase order be created for 3 Mountain Bikes from an outdoor sporting company called Highlander.

Task 2: Create a Customer

Ronnie Buis from Arial Lift Rentals recently contacted OEC Computers and mentioned that he would like to do business with OEC Computers on a regular basis. You are required to add the details of Ronnie Buis and his company on SAP Business One.

Company Number	:	C10100
Company Name	:	Arial Lift Rentals (Pty) Ltd.
Company Type	:	High Tech
Preferred Currency	:	Euros

Contact Person	:	Mr. Ronnie Buis
Contact Number (w)	:	(092) 999 9115
Contact Number (c)	:	(052) 885 5123
Email	:	Ronald.Buis@ALR.co.za
Position	:	Operations Director
Address	:	65 Two-Wheel Drive
		Motor Rad
		Biker City
		1395

Mr. Buis would prefer it if his company dealt with OEC Computers on a Cash Basis. Being a potential customer you are required to place the newly created account for Arial Lift Rentals (Pty) Ltd. on hold.

This is the end of the scenario. Please ensure that you have saved all the work that you have done and that you have filled in the heuristic checklist before leaving the testing environment.

Appendix G: Heuristic Evaluation Check List

Nelson Mandela Metropolitan University
for tomorrow

Department of Computing Sciences

ERP HEURISTIC EVALUATION CHECKLIST

General Instructions	
No.	*Instruction*
1.	This heuristic evaluation checklist assists in evaluating the usability of SAP Business One.
2.	A 5-point Likert scale is used to assess the severity of the usability problem.
3.	The 5-point scale to be used throughout this heuristic evaluation checklist is: *0. Not a usability problem* *1. Cosmetic – Will not affect the usability of the system, fix if possible* *2. Minor Usability Problem – Users can work around problem, fixing problem should be low priority* *3. Major Usability Problem – Users have difficulty and cannot workaround problem, fixing should be a high priority* 4. *Catastrophic Usability Problem – Users are unable to do their work, fixing this problem should be mandatory*
4.	This heuristic evaluation contains 15 heuristics which need to be evaluated.
5.	Please evaluate each heuristic honestly and critically. Space is provided after each heuristic for you to motivate your severity selection.
6.	Please complete all required fields in ink with a blue or black ballpoint pen and not pencil.
7.	Any errors made should not be Tipp-Exed but should rather be crossed out and signed.
9.	For further enquiries please feel free to contact me (Akash Singh) at: *(Cell): 082 – 882 6128* *(Email): Akash.Singh@nmmu.ac.za*
10.	All the information provided within this questionnaire will remain completely confidential and no reference will be made to any individual or their responses.
11.	Thank you for taking the time to complete this evaluation

1. Visibility of System Status

The system should always keep the user informed about what is going on, through appropriate feedback within reasonable time.

No.	*Review Checklist*	0	1	2	3	4	*Comments*
1.1	Every display begins with a title or header that describes the screen contents						
1.2	Menu instructions, prompts, and error messages appear in the same place(s) on each menu						
1.3	In multipage data entry screens, each page is labelled to show its relation to others						
1.4	Pop-up windows used to display error messages, show the user the field in which the error occurred						
1.5	Some form of system feedback is available for every operator action						
1.6	After the user completes an action (or group of actions), the feedback provided indicates that the next group of actions can be started						
1.7	Visual feedback in menus or dialog boxes indicates which choices are selectable						
1.8	Visual feedback in menus or dialog boxes indicates which choice the cursor is on now						
1.9	Visual feedback is provided when objects are selected or moved						
1.10	The current status of an icon is clearly indicated						
1.11	Menu-naming terminology is consistent with the user's task domain						
1.12	The system provides visibility: that is, by looking, the user can tell the state of the system and the alternatives for action						
1.13	GUI menus make it obvious which item has been selected						
1.14	GUI menus make it obvious whether de-selection is possible						
1.15	If users must navigate between multiple screens, the system uses context labels, menus, maps and place makers as navigational aids						

2. Match Between System and Real World

The system should always speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

No.	Review Checklist	0	1	2	3	4	Comments
2.1	Menu choices are ordered in the most logical way, given the user, the item names, and the task variables						
2.2	There is a natural sequence to menu choices						
2.3	Related and interdependent fields appear on the same screen						
2.4	When prompts imply a necessary action, the words in the message box are consistent with that action						
2.5	On data entry screens, tasks are described in terminology familiar to users						
2.6	Field-level prompts are provided for data entry screens						
2.7	Menu choices fit logically into categories that have readily understood meanings						
2.8	Input data codes are meaningful						
2.9	The system automatically enters leading or trailing spaces to align decimal points						

3. User Control and Freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialog. The system should support undo and redo.

No.	*Review Checklist*	0	1	2	3	4	*Comments*
3.1	In systems that use overlapping windows, it is easy for users to rearrange windows on the screen						
3.2	In systems that use overlapping windows, it is easy for users to switch between windows						
3.3	When a user's task is complete, the system waits for a signal from the user before proceeding						
3.4	Users can type-ahead in a system with many nested menus						
3.5	Users are prompted to confirm commands that have drastic, destructive consequences						
3.6	There is an "undo" function at the level of a single action, a data entry, and a complete group of actions						
3.7	Users can cancel out of operations in progress						
3.8	Users can reduce data entry time by copying and modifying existing data						
3.9	If menu lists are long (more than seven items), users can select an item either by moving the cursor or by typing a mnemonics code						
3.10	If users can go back to a previous menu, they can change their earlier menu choice						
3.11	Users can easily reverse their actions						
3.12	If the system allows users to reverse their actions, there is a retracting mechanism to allow for multiple undoes						
3.13	Users can set their own system, session, file, and screen defaults						

4. Consistency and Standards

Users should not have to wonder whether different words, situations, or actions mean the same thing.

No.	Review Checklist	0	1	2	3	4	Comments
4.1	Industry or company formatting standards been followed consistently in all screens within a system						
4.2	Abbreviations do not include punctuations						
4.3	Integers are right-justified and real numbers decimal-aligned						
4.4	There are no more than twelve to twenty icon types						
4.5	Each window has a title						
4.6	The menu structure matches the task structure						
4.7	Industry or company standards have been established for menu design, and are applied consistently on all menu screens in the system						
4.8	Menu titles are either centered or left-justified						
4.9	On-line instructions appear in a consistent location across the screens						
4.10	Field labels are consistent from one data entry screen to another						
4.11	Fields and labels are left-justified for alpha lists and right-justified for numeric lists						
4.12	Field labels appear to the left of single fields and above list fields						
4.13	There are no more than four to seven colours, and they are far apart along the visible spectrum						
4.14	The most important information is placed at the beginning of the prompt						
4.15	User actions are named consistently across all prompts in the system						
4.16	Menu choice names are consistent, both within each menu and across the system, in grammatical style and terminology						
4.17	The structure of menu choice names matches their corresponding menu titles						
4.18	Different parts of the system require completely different actions in order to complete a sequence of tasks						

5. Helps Users Recognise, Diagnose, and Recover from Errors

Error messages should be expressed in plain language, precisely indicating the problem, and constructively suggest a solution.

No.	*Review Checklist*	**0**	**1**	**2**	**3**	**4**	*Comments*
5.1	Sound is used to signal an error						
5.2	Prompts are stated constructively, without overt or implied criticism of the user						
5.3	Prompts imply that the user is in control						
5.4	Prompts are brief and unambiguous						
5.5	Error messages are worded so that the system, not the user, takes the blame						
5.6	Error messages are grammatically correct						
5.7	Error messages avoid the use of exclamation points						
5.8	Error messages avoid the use of violent or hostile words						
5.9	All error messages in the system use consistent grammatical style, form, terminology, and abbreviations						
5.10	Error messages place users in control of the system						
5.11	If an error is detected in a data entry field, the system place the cursor in that field or highlights the error						
5.12	Error messages inform the user of the error's severity						
5.13	Error messages suggest the cause of the problem						
5.14	Error messages provide appropriate semantic information						
5.15	Error messages indicate what action the user needs to take to correct the error						
5.16	If the system supports both novice and expert users, multiple level of error-messages detail available						

6. Error Prevention

An even better alternative to good error messages is careful design, which prevents a problem from occurring in the first place.

No.	Review Checklist	0	1	2	3	4	Comments
6.1	Menu choices are logical, distinctive, and mutually exclusive						
6.2	Data inputs are case-blind whenever possible						
6.3	If the system displays multiple windows, navigation between windows is simple and visible						
6.4	The system warns users if they are about to make a potentially serious error						
6.5	The system prevents users from making errors whenever possible						
6.6	The system intelligently interprets variations in the user's commands						
6.7	Data entry screens and dialog boxes indicate the number of character spaces available in a field						
6.8	Fields in data entry screens and dialog boxes contain default values when appropriate						

7. Recognition Rather than Recall

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

No.	Review Checklist	0	1	2	3	4	Comments
7.1	All the data a user needs is on display at each step in the transaction sequence						
7.2	Prompts have been formatted using white space, justification, and visual cues for easy scanning						
7.3	Text areas have "breathing space" around them						
7.4	There is an obvious visual distinction made between "choose one" menu and "choose many" menus						
7.5	The system greys out or deletes labels of currently inactive soft function keys						
7.6	Items have been grouped into logical zones, and headings have been used to distinguish between zones						
7.7	Zones are no more than twelve to fourteen characters wide and six to seven lines high						
7.8	Zones have been separated by spaces, lines, colour, letters, bold titles, rules lines, or shaded areas						
7.9	Field labels are close to fields, but separated by at least one space						
7.10	Optional data entry fields are clearly marked						
7.11	Size, boldface, underlining, colour, shading or typography are used to show relative quantity or importance of different screen items						
7.12	Borders are used to identify meaningful groups						

No.	Review Checklist	0	1	2	3	4	Comments
7.13	The same colour has been used to group related elements						
7.14	Colour coding is consistent throughout the system						
7.15	Colour is used in conjunction with some other redundant cue						
7.16	There is good colour brightness contrast between images and background colours						
7.17	The system provides mapping: that is, the relationships between controls and actions are apparent to the user						
7.18	Input data codes are distinctive						
7.19	Inactive menu items are greyed out or omitted						
7.20	If the system has many menu levels or complex menu levels, users have access to an on-line spatial menu map						
7.21	There are salient visual cues to identify the active window						
7.22	Data entry screens and dialog boxes indicate when fields are optional						
7.23	On data entry screens and dialog boxes, dependant fields are displayed only when necessary						

8. Flexibility and Efficiency of Use

Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. Provide alternative means of access and operation for users who differ from the "average" user.

No.	*Review Checklist*	0	1	2	3	4	*Comments*
8.1	If the system supports both novice and expert users, multiple levels of error message detail is available						
8.2	Users can define their own synonyms for commands						
8.3	The system allows novice users to enter the simplest, most common form of each command, and allow expert users to add parameters						
8.4	Expert users have the option for entering multiple commands in a single string						
8.5	The system provides function keys for high-frequency commands						
8.6	Data entry screens allow users to save partially complete tasks						
8.7	If the system uses a type-ahead strategy, the menu items have mnemonic codes						
8.8	If the system uses a pointing device, users have the option of either clicking directly on a field or using a keyboard shortcut						
8.9	The system offers "find next" and "find previous" shortcuts for searches						
8.10	On menus, users have the option of either clicking directly on a menu item or using a keyboard shortcut						
8.11	In dialog boxes, users have the option of either clicking directly on a dialog box option or using a keyboard shortcut						

9. Aesthetic and Minimalist Design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in dialogue competes with the relevant units of information and diminishes their relative visibility.

No.	*Review Checklist*	**0**	**1**	**2**	**3**	**4**	*Comments*
9.1	Only (and all) information essential to decision making is displayed on the screen						
9.2	If the system uses a standard GUI where menu sequences has already been specified, menus adhere to the specifications whenever possible						
9.3	Meaningful groups of items are separated by white space						
9.4	Each data entry screen has a short, simple, clear, distinctive title						
9.5	Field labels are brief, familiar, and descriptive						
9.6	Prompts are expressed in the affirmative, and use active voice						

10. Help and Documentation

It is even better if the system can be used without documentation, but it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

No.	*Review Checklist*	0	1	2	3	4	*Comments*
10.1	If users are working from hard copy, the parts of the hard copy that go on-line are marked						
10.2	On-line instructions are visually distinct						
10.3	The instructions follow the sequence of user actions						
10.4	If menu choices are ambiguous, the system provides additional explanatory information when an item is selected						
10.5	The memory aids for commands, are provided either through on-line quick reference or prompting						
10.6	The help function is visible						
10.7	The help system interface is consistent with the navigation, presentation, and conversation interfaces of the application it supports						

11. Navigation and Access to Information

The ability to identify and access appropriate information, menus, reports, options, and elements accurately and efficiently.

No.	*Review Checklist*	**0**	**1**	**2**	**3**	**4**	*Comments*
11.1	Information is easy to find						
11.2	Functionality can be found quickly and easily (e.g. Transactions)						
11.3	There is sufficient help provided for finding the correct functionality, information and screens						
11.4	The system can guide the user through the correct sequence of transactions to complete a business process						
11.5	The GUI is easy to understand to enable efficient and accurate navigation of the system						
11.6	There is a search functionality						
11.7	The search functionality supports finding information						
11.8	The information found from the search functionality matches the information required						
11.9	The navigation suits different interaction styles of the users						
11.10	There are alternative ways of navigating the system						
11.11	Guidance-type information is always available						
11.12	The next sequence steps for a transaction are clear						

12. Presentation of Screen and Output

The appropriateness of layout of menus, dialog boxes, controls, and information on the screen for data entry and output generation.

No.	*Review Checklist*	**0**	**1**	**2**	**3**	**4**	*Comments*
12.1	The visual layout is well designed						
12.2	Information is timely, accurate, complete and understandable						
12.3	The output is easy to understand and interpret						
12.4	The output is comprehensive						
12.5	The output is well structured						
12.6	The information presented supports informed decision making						
12.7	The output provided by the system provides clear visibility into the various other business units and departments						
12.8	The system has an intuitive GUI						
12.9	The system presents the user with complex and busy GUIs, resulting in information overload						

13. Appropriateness of Task Support

Accurate alignment of user tasks between system and real world to ensure effective task support and efficient task completion.

No.	*Review Checklist*	**0**	**1**	**2**	**3**	**4**	*Comments*
13.1	The terminology used is consistent with the terminology of the user						
13.2	The system provides real-time information						
13.3	The response from the system is quick and efficient						
13.4	The system supports efficient completion of user tasks						
13.5	The system improves user productivity						
13.6	The system automates routine and redundant tasks and data						
13.7	It is easy to operate and use the system						
13.8	The system supports improved information flow between the various organisational units and departments						

14. Intuitive Nature of System

The degree of ease required to learn how to use the system effectively.

No.	*Review Checklist*	**0**	**1**	**2**	**3**	**4**	*Comments*
14.1	You can learn how to use the system without a long introduction						
14.2	You can figure out the various functions of the system by trying them						
14.3	There is sufficient on-line help to support the learning process						
14.4	It is easy to become skilful at using the system within a short amount of time						
14.5	The system is intimidating and complex to learn and use						
14.6	The system contains unnecessary technical jargon and confusing acronyms						

15. Ability to Customise

How easy it is to customise the system to ensure accurate alignment between the system and business processes, the system and the user, the user and the business processes?

No.	*Review Checklist*	0	1	2	3	4	*Comments*
15.1	The system can easily be configured to an industry type						
15.2	The system supports customisation to the level of user preferences						
15.3	Customisation is supported in the way individual users complete a specific transaction or task						
15.4	The system can be customised to align the ERP transactions with the business processes of the enterprise						
15.5	The system can be configured to update existing business processes and (or) to include new business processes						
15.6	A particular module can be customised						
15.7	The system supports customisation to enable and promote business agility						
15.8	The system supports customisation of reports						
15.9	The system is easy to change and re-configure over a period of time without making the system more complicated						
15.10	The GUI can be configured without affecting the underlying business logic						

Appendix H: Biographical Questionnaire

Nelson Mandela Metropolitan University

for tomorrow

Department of Computing Sciences

BIOGRAPHICAL INFORMATION QUESTIONNAIRE

General Instructions	
No.	*Instruction*
1.	This questionnaire attempts to gather information on your (the participant) biographical background with regards to specific topics that are of relevance to this study.
2.	The information gathered in this questionnaire is for research purposes with the intention of informing a PhD study pursued at the Nelson Mandela Metropolitan University.
3.	This questionnaire consists of 1 section – Biographical Details.
4.	The above mentioned section addresses a total of 8 questions.
5.	Please complete each question to the best of your knowledge.
6.	Please complete all questions in ink with a blue or black ballpoint pen and not pencil.
7.	Indicate your response to a question by placing an X in the space provided (where required).
8.	Any errors made should not be Tipp-Exed but should rather be crossed out and signed.
9.	For further enquiries please feel free to contact me (Akash Singh) at: (Cell): 082 – 882 6128 (Email): Akash.Singh@nmmu.ac.za
10.	All the information provided within this questionnaire will remain completely confidential and no reference will be made to any individual or their responses.

11.	Thank you for taking the time to complete this questionnaire.

No.	*Biographical Details*				
1.	**GENDER**	**MALE**	**FEMALE**		
2.	**AGE (YEARS)**	**18-20**	**20-29**	**30-39**	**40-49**
3.	**DEGREE**				
4.	**DEGREE LEVEL**	**UNDERGRAD**	**POSTGRAD**		
5.	**WEEKLY COMPUTATIONAL USAGE**	**<5 Hrs**	**5-10 Hrs**	**11-20 Hrs**	**>20 Hrs**
6.	**ERP SKILLS**	**NOVICE**	**INTERMEDIATE**	**EXPERT**	
7.	**ERP USAGE (FREQUENCY)**	**NEVER**	**OCCASIONALLY**	**FREQUENTLY**	
8.	**WEEKLY ERP USAGE (FREQUENCY)**	**<5 Hrs**	**5-10 Hrs**	**11-20 Hrs**	**>20 Hrs**

Appendix I: Task Lists

Department of Computing Sciences

TASK LIST – DAY 1

General Instructions	
No.	*Instruction*
1.	This task list contains 8 tasks that you need to complete with the prototype ERP system.
2.	Before you start with the tasks, the various functionalities of the prototype ERP system will be explained to you.
3.	Please let me know if you think you have finished a task. This should be done for every task.
4.	Try to do all the tasks; however do not hesitate to ask any questions.
5.	Please "think aloud" as you attempt the various tasks – tell us what you are thinking, what you are trying to do, what you do not understand, etc.
6.	Sometimes, I might ask you to repeat a task - This will happen, when I do not have all the information I need.
7.	Sometimes, I might ask you to move on, although you have not yet finished the task. This will happen when I have all the data I need.
8.	Remember we are evaluating the prototype and not you.
9.	Your participation in this evaluation will remain completely anonymous.
10.	You are in full control of this session.
11.	This session will take no more than 30 minutes.

Scenario:

You are the purchasing clerk at a small sporting goods outlet called Multisports Limited. Multisports has recently purchased an ERP called SAP Business One, designed specifically for small and medium enterprises, to replace its existing purchasing and inventory legacy systems. Your task today is to attempt to conduct your routine activities of recording purchase orders using SAP Business One.

TASK 1

You are required to purchase 3 pairs of boxing gloves from Athletic Sportware & Co. The buyer for these items was Jane Smith. Jane has also mentioned to you that she receives a 5% discount from Athletic Sportware & Co. when placing orders.

TASK 2

Jane Smith has requested that a purchase order be created for the purchase of 1 Bicycle from Totalsports.com.

TASK 3

You were requested by buyer David Batra that the purchase order for 5 Divers Goggles and 5 snorkels be created for Kenyan Sporting Ltd.

TASK 4

The company is planning on having an internal soccer challenge and buyer Nick Silver has placed an order for 3 Type-1 footballs from a sporting goods company called Ballgames.

TASK 5

David Batra has placed an order for 6 Type-1 caps for each of the teams and their coaches participating in the internal soccer tournament from Kenyan Sporting Ltd. You are required to create the purchase order for this purchase.

TASK 6

You are required to create a purchase order for 5 Volleyballs from British Outdoors. The buyer responsible for this order is Phoebe Grant.

TASK 7

You are required to create a purchase order for 3 Type-1 dumbbells from Highlander. The buyer responsible for this order is Ian Walker.

TASK 8

You are required to create a purchase order for 4 Type-2 caps from Teamware. The buyer responsible for this order is Jane Smith.

Nelson Mandela Metropolitan University
for tomorrow

Department of Computing Sciences

TASK LIST – DAY 2

General Instructions	
No.	*Instruction*
1.	This task list contains 8 tasks that you need to complete with the prototype ERP system.
2.	Before you start with the tasks, the various functionalities of the prototype ERP system will be explained to you.
3.	Please let me know if you think you have finished a task. This should be done for every task.
4.	Try to do all of the tasks; however do not hesitate to ask any questions.
5.	Please “think aloud” as you attempt the various tasks – tell us what you are thinking, what you are trying to do, what you do not understand, etc.
6.	Sometimes, I might ask you to repeat a task – This will happen, when I do not have all the information I need.
7.	Sometimes, I might ask you to move on, although you have not yet finished the task. This will happen when I have all the data I need.
8.	Remember we are evaluating the prototype and not you.
9.	Your participation in this evaluation will remain completely anonymous.
10.	You are in full control of this session.
11.	This session will take no more than 30 minutes.

Scenario:

You are the purchasing clerk at a small sporting goods outlet called Multisports Limited. Multisports has recently purchased an ERP called SAP Business One, designed specifically for small and medium enterprises, to replace its existing purchasing and inventory legacy systems. Your task today is to attempt to conduct your routine activities of recording purchase orders using SAP Business One.

TASK 1

You are required to purchase 5 stop watches from Hardcore Sporting. The buyer for these items was Jane Smith.

TASK 2

Jane Smith has requested that a purchase order be created for the purchase of 5 pairs of Cross-Trainers from Sportsworld.

TASK 3

You were requested by buyer David Batra that the purchase order be created for 3 Mountain Bikes from an outdoor sporting company called Highlander.

TASK 4

The company is planning on having an internal soccer challenge and buyer Nick Silver has placed an order for 10 Type-1 footballs from a sporting goods company called Ballgames.

TASK 5

David Batra has placed an order for 6 Type-1 caps for each of the teams and their coaches participating in the internal soccer tournament, from Athletic Sportware & Co. You are required to create the purchase order for this purchase.

TASK 6

You are required to create a purchase order for 4 Racquet sets from Racquet Sports Limited. The buyer responsible for this order is Phoebe Grant.

TASK 7

You are required to create a purchase order for 4 sport-shirts from Teamware. The buyer responsible for this order is Andy Step.

TASK 8

You are required to create a purchase order for 4 Type-2 dumbbells from Totalsports.com. The buyer responsible for this order is Chris Talbot.

Appendix J: Adaptive Post-Test Satisfaction Questionnaire

Nelson Mandela Metropolitan University

for tomorrow

Department of Computing Sciences

POST-TEST SATISFACTION QUESTIONNAIRE

General Instructions	
No.	*Instruction*
1.	This post-test satisfaction questionnaire assists in evaluating the strengths and weaknesses of the prototype ERP system.
2.	A 5-point Likert scale is used throughout this questionnaire to assess each statement.
3.	This questionnaire contains 6 sections that need to be completed.
4.	Please evaluate each statement honestly and critically.
5.	Please complete all required fields in ink with a blue or black ballpoint pen and not pencil.
6.	Any errors made should not be Tipp-Exed but should rather be crossed out and signed.
7.	For further enquiries please feel free to contact me (Akash Singh) at: (Cell): 082 – 882 6128 (Email): Akash.Singh@nmmu.ac.za
10.	All the information provided within this questionnaire will remain completely confidential and no reference will be made to any individual or their responses.
11.	Thank you for taking the time to perform this evaluation.

Section A: General Details		
User ID:		
Day:	1	2

Section B: Overall						
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 screen	**VERY CONFUSING**				**VERY CLEAR**
		1	2	3	4	5
4.	Task support provided by the system	**VERY FRUSTRATING**				**VERY SATISFYING**
		1	2	3	4	5
5.	Efficiency of the system	**VERY FRUSTRATING**				**VERY SATISFYING**
		1	2	3	4	5
6.	Learnability of the system	**VERY DIFFICULT**				**VERY EASY**
		1	2	3	4	5
7.	Navigation of the system	**VERY DIFFICULT**				**VERY EASY**
		1	2	3	4	5

Section C: Task Support and Efficiency							
1.	Creating a purchase order	**VERY DIFFICULT**				**VERY EASY**	
		1	2	3	4	5	
2.	Number of steps to complete a task	**MANY**				**FEW**	
		1	2	3	4	5	
3.	System response time	**SLOW**				**FAST**	
		1	2	3	4	5	
4.	Finding and selecting an appropriate supplier	**VERY DIFFICULT**				**VERY EASY**	
		1	2	3	4	5	
5.	Finding and selecting an appropriate buyer	**VERY DIFFICULT**				**VERY EASY**	
		1	2	3	4	5	

Section D: Learnability							
1.	Guidance through a sequence of task activities	**NOT NOTICABLE**				**VERY NOTICIABLE**	
		1	2	3	4	5	
2.	Visibility of next sequence of activities for a particular task	**NOT NOTICABLE**				**VERY NOTICIABLE**	
		1	2	3	4	5	
3.	Time to become skilled with the system	**LONG**				**SHORT**	
		1	2	3	4	5	
4.	Difficulty to learn how to use the system	**VERY COMPLEX**				**VERY SIMPLE**	
		1	2	3	4	5	

Section E: Adaptivity						
1.	Ordering of lists	**NOT NOTICABLE**			**VERY NOTICIABLE**	
		1	2	3	4	5
2.	Colour-coding of controls	**NOT NOTICABLE**			**VERY NOTICIABLE**	
		1	2	3	4	5
3.	Enabling and disabling of fields	**NOT NOTICABLE**			**VERY NOTICIABLE**	
		1	2	3	4	5

Section F: General
Describe any positive aspects of the system that appealed to you
Describe any negative aspects of the system that you identified
Please provide any general suggestions for improvement of the system in the space below

Appendix K: Statistical Analysis

Biographical Results for Chi-Square Test of Independence

Category	Chi-Square	p-value
Gender	0.18	.670
Age	0.00	1.000
Degree Level	0.19	.665
Weekly Computer Usage	4.27	.234
ERP Skills	0.18	.670
ERP Usage	0.00	1.000
Weekly ERP Usage	1.22	.269

Wilcoxon Match-Pairs T-Test

	Adaptive		Non-Adaptive	
	T	*p-value*	*T*	*p-value*
Overall	2.50	.093	3.00	.063
Task Support and Efficiency	9.00	.753	14.00	.091
Learnability	0.00	.043	0.00	.008
Adaptivity	10.00	.139		
Task Success	16.00	.779	6.50	.107
Time-on-Task	1.00	.004	0.00	.003
Mouse Clicks	23.00	.374	2.50	.007
Errors Committed	14.50	.100	3.00	.008

Chi-Square Test of Independence Results

	Chi-Square Test	
	X^2	***p-value***
Task Success – Day 1	0.79	.375
Task Success – Day 2	1.64	.201
Task Success - Difference	3.14	.076
Time-on-Task – Day 1	0.18	.670
Time-on-Task – Day 2	0.18	.670
Time-on-Task – Difference	0.18	.670
Mouse Clicks – Day 1	0.18	.670
Mouse Clicks – Day 2	0.00	1.00
Mouse Clicks - Difference	4.55	.033
Errors Committed – Day 1	0.18	.670
Errors Committed – Day 2	2.93	.087
Errors Committed - Difference	0.00	1.00

Mann-Whitney U-Test Results for Differences

	Mann-Whitney Test		
	U	*p-value*	*Cohen's d*
Overall - Day 1	35.0	.101	
Overall - Day 2	34.0	.088	
Overall - Difference	51.0	.562	
Task Support and Efficiency – Day 1	31.0	.056	
Task Support and Efficiency – Day 2	60.0	1.000	
Task Support and Efficiency - Difference	38.5	.151	
Learnability – Day 1	37.0	.133	
Learnability – Day 2	30.5	.047	0.93
Learnability - Difference	38.5	.151	
Task Success – Day 1	59.5	.949	
Task Success – Day 2	40.5	.193	
Task Success - Difference	42.0	.243	
Time-on-Task – Day 1	51.0	.562	
Time-on-Task – Day 2	54.0	.699	
Time-on-Task – Difference	54.0	.699	
Mouse Clicks – Day 1	38.5	.151	
Mouse Clicks – Day 2	46.5	.365	
Mouse Clicks - Difference	26.0	.023	1.13
Errors Committed – Day 1	59.0	.949	
Errors Committed – Day 2	28.0	.034	1.15
Errors Committed - Difference	47.0	.401	

Cronbach's α results

	Day 1	*Day 2*
Overall	0.92	0.82
Task support & Efficiency	0.66	0.84
Learnability	0.83	0.73
Adaptivity	0.40	0.69