THE DESIGN OF ADAPTIVE INTERFACES FOR ENTERPRISE RESOURCE PLANNING SYSTEMS

Akash Singh and Janet Wesson
Department of Computing Sciences, Nelson Mandela Metropolitan University
P.O. Box 77000, Port Elizabeth, South Africa

Keywords: AUIs, ERP systems, Small enterprises, System design, Usability.

Abstract: Research has shown that AUIs can improve the usability of software systems. Enterprise Resource Planning (ERP) systems are an example of complex software systems which suffer from several usability issues. Limited research has been published on the applicability of AUIs to the domain of ERP systems. This paper aims to address this limitation by discussing the design of AUIs for ERP systems. A secondary contribution of this paper is the proposal of an adaptive taxonomy for ERP systems. Application of the proposed taxonomy and the proposed design is demonstrated through the implementation of an AUI for an existing ERP system, namely SAP Business One (SBO). The AUI was designed to address several of the usability issues identified in SBO, specifically efficiency and learnability.

1 INTRODUCTION

Adaptive interfaces (AUIs) have the potential to reduce the complexity of user interfaces (UIs) and improve the usability of complex systems. This is achieved by means of adapting the UI automatically through the use of various types of adaptation and adaptive components (Letsu-Dake and Ntuen, 2009, Leung et al., 2006).

One such complex system which suffers from several usability issues is Enterprise Resource Planning (ERP) systems (Yeh, 2006). AUIs could potentially address the usability issues of ERP systems. Limited evidence exists, however, to demonstrate how AUIs should be designed for ERP systems.

This paper aims to address the above limitation by proposing a design of AUIs for ERP systems. The remainder of this paper comprises the following sections: Section 2 contains a discussion on related work. An adaptation taxonomy for ERP systems is proposed in Section 3. Section 4 discusses the design of AUIs for ERP systems. Section 5 presents a discussion on how the proposed design was implemented using an existing ERP system. Section 6 contains the conclusions and future work.

2 RELATED WORK

Usability studies of ERP systems indicate that ERP systems suffer from several usability issues (Topi et al., 2005). However, most of these studies were conducted on ERP systems for large enterprises. Limited research exists on the usability issues of ERP systems for small enterprises.

A usability study of an ERP system, designed specifically for small enterprises, was conducted using SAP Business One (SBO). The usability evaluation of SBO enabled the identification of 37 usability issues. These usability issues were classified based on the criteria proposed by Singh and Wesson (2009).

These usability issues were further analysed to determine which issues were a result of the UI being complex. Only 11 of the 37 identified usability issues related to the complexity of the UI. These 11 usability issues are presented in Table 1.

AUIs could potentially resolve the identified usability issues through the application of various types of adaptation. Adaptation can occur in four basic forms, namely content adaptation, presentation adaptation, navigation adaptation and device adaptation (Ramachandran, 2009, Peng and Silver, 2007).

Singh A. and Wesson J..
THE DESIGN OF ADAPTIVE INTERFACES FOR ENTERPRISE RESOURCE PLANNING SYSTEMS.
DOI: 10.5220/0003486302810286
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (ICEIS-2011), pages 281-286
ISBN: 978-989-8425-56-0
Copyright © 2011 SCITEPRESS (Science and Technology Publications, Lda.)

The research conducted for this paper did not consider device adaptation as this study focuses on ERP systems for desktop computers.

Content and presentation adaptation could be used to address usability issues which relate to task support and learnability, whilst navigation adaptation can be applied to usability issues which relate to navigation. The mapping of the various adaptation types to the identified usability issues is presented in Table 1.

3 PROPOSED ADAPTATION TAXONOMY

An adaptation taxonomy was proposed by Knutov *et al.* (2009) to illustrate the various methods, techniques and approaches that could be used to implement the various types of adaptations for hypermedia systems. 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 can be used to guide and support the design and implementation of AUIs for ERP systems.

The proposed taxonomy (Figure 1) presents several modifications over the original taxonomy proposed by Knutov *et al.* (2009). Components of the original taxonomy are illustrated in blue and the new proposed components are illustrated in orange. Only those methods that could be used to address the usability issues identified in Table 1 were selected from the original taxonomy.

The approaches and techniques originally proposed for hypermedia systems were replaced by existing HCI design patterns for desktop applications. Existing HCI patterns were selected to support the design of usable ERP systems. These patterns can assist in solving common UI design problems within a given design domain (Tidwell, 2006).

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 UI and interaction design.

The most comprehensive, recent and widely referenced catalogues include Tidwell (2006) and van Welie (2008). The HCI patterns proposed by Tidwell (2006) were selected due to 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.

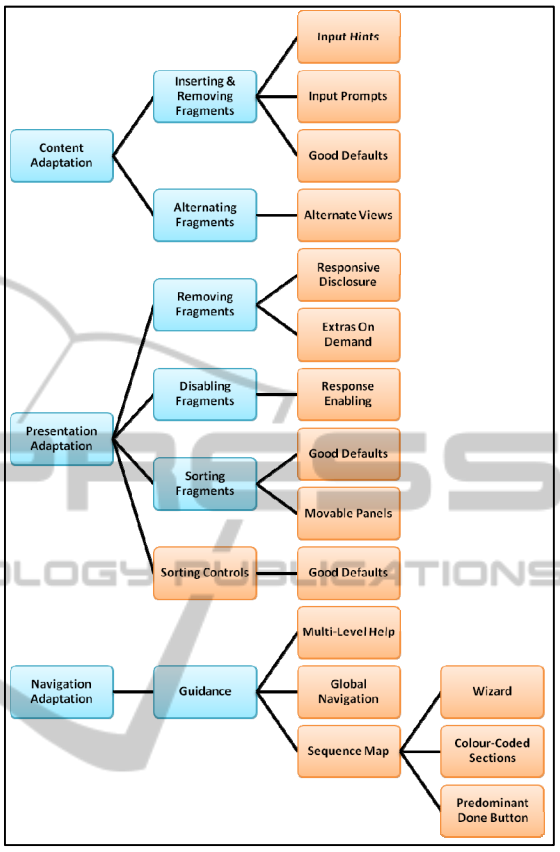

Figure 1: Proposed ERP Adaptation Taxonomy.

3.1 Content Adaptation

Based on the proposed ERP adaptation taxonomy, content adaptation can be implemented by inserting, removing, or altering fragments of information. The aim of inserting, removing or altering fragments of information is to ensure efficiency and to improve user productivity.

3.2 Presentation Adaptation

Several presentation adaptation techniques are proposed (Figure 1) to address the usability issues that were identified with regards to task support, learnability and customisation.

Removing fragments uses the techniques of responsive disclosure and extras on demand. These two techniques aim to reduce the number of controls on the UI by only displaying what is necessary at a particular point in time whilst providing an option to display more or everything (extras-on-demand).

Table 1: Usability Issues of SBO and Relevant Adaptation Types.

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

The method of disabling fragments uses the technique of responsive enabling. This technique aims to address the issue of guiding the user through a task by enabling task activities in an ordered manner.

Sorting fragments uses two techniques, namely good defaults and movable panels. This method addresses the usability issue of UI personalisation. This method will re-order the fragments of the UI based on the user's previous interactions (good defaults) and could also allow the user to re-order the fragments based on his/her own preference (movable panels).

3.3 Navigation Adaptation

Most of the usability issues relating to navigation identified in Table 1 relate to guidance. For this reason the guidance adaptation method was selected from the original taxonomy.

The following techniques are proposed which aim to ensure that users are guided through the correct sequence of tasks (sequence map); evidence of the next sequence of steps are provided (sequence map and colour-coded sections); and finding information and functionality is facilitated (multiple-level help).

4 DESIGN

This section proposes a design for AUIs for ERP systems. The proposed design is made up of two key components: system architecture and model design.

4.1 System Architecture

The proposed architecture in Figure 2 is an extension of an existing AUI system architecture proposed by Ramachandran (2009). Components of the original architecture are illustrated in blue in and the new components are illustrated in orange (Figure 2). The architecture proposed by Ramachandran (2009) was selected as it was general enough to be specialised to the domain of ERP systems, it supported presentation and navigation adaptation; it included a user model; and it separated the business logic of the system from the actual UI.

The proposed system architecture (Figure 2) differs from the original architecture through the addition of the following components:

- An inference engine (monitoring the user interactions & updating the AUI);
- Support for content adaptation (missing from the original architecture);
- An ERP engine (contains both the business logic and the ERP system logic); and
- Support for the ERP system database tables and application data tables within the database.

These additions are implemented across three different sections within the proposed system architecture: the database layer, the application layer and the presentation layer.

4.1.1 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).

4.1.2 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 UI. 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 UI. The inference engine was not part of the original system architecture.

A log file is stored for each user, in order for the inference engine to successfully monitor and capture the user's interaction with the UI.

The adaptation engine takes the input from the inference engine and determines what type of adaptation to perform on the UI 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 adaption.

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) and 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.

4.1.3 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 through the inference engine. Determining the validity of the data will be done by the ERP engine and if accepted, this data will be saved to the database.

4.2 Adaptive Components

The various types of adaptation discussed in Section 2 make use of several adaptive components in order to execute the adaptation. These adaptive components are represented in the form of declarative models.

4.2.1 User Model

User models are a critical component of any AUI as they provide the information necessary to adapt the user model between the different users (Tran et al., 2009).

The proposed user model will contain a summary of the key elements that the ERP system will require in order to adapt the UI. These elements will be grouped according to the various tasks performed by the user.

The data stored within the user model will aid the decisions made by the adaptation engine in terms of supporting presentation adaptation and content adaptation. Data stored within the user model includes the sequence in which controls were selected for a particular task. This could assist with presentation adaptation in terms of re-ordering the controls on the UI. All of the data stored within the user model will be continuously updated based on the users' interaction with the ERP system.

4.2.2 Task Model

Task models can be defined as the hierarchical representation of the tasks and sub-tasks (activities) performed by users of a particular software system. The proposed task model is based on a generic description of task models and contains:

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

The task model can assist with adaptation by means of supporting navigation adaptation in terms of:

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

Manipulation of tasks is done by means of a dialog model and a presentation model, which are discussed in the following sections.

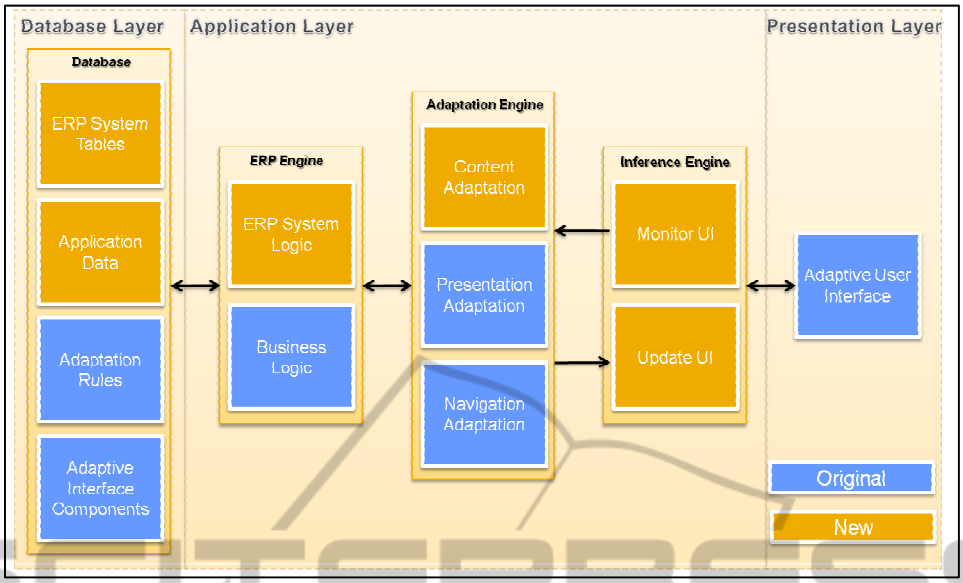

Figure 2: Proposed System Architecture.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

4.2.3 Dialog Model

A Dialog model complements the task model by means of describing and organising various task activities that need to be performed on the UI. The proposed dialog model is based on the description of the generic dialog model and will support the process of presentation adaptation (Figure 1) by means of visually:

- Sorting fragments and the controls within these fragments; and
- Enabling and disabling fragments.

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

5 PROTOTYPE

The proposed design and adaptation taxonomy was implemented using SBO. SBO was selected as it is widely used by small enterprises, it was the most extensible ERP system, and it allowed for the separation of business logic and AUI logic.

Further analysis of the usability issues presented in Table 1 revealed that these issues could be classified as either efficiency issues (to improve task support through better task completion times) or learnability issues (how quickly and effectively can a novice SBO user learn how to use the ERP system). This section will briefly discuss how the various types of adaptation were implemented in order to improve the efficiency and learnability of SBO.

Content adaptation was implemented using list-based adaptation techniques. This was applied the dropdown boxes of SBO using the frequency and recency algorithms proposed by Gajos *et al.* (2006). Presentation adaptation techniques (Figure 1) were applied to the fragments (a collection of controls within a grouped section of a form) in SBO. Fragments were either disabled or enabled (disabling and enabling fragments pattern) based on the sequence order in the task and dialog models. Navigation adaptation was implemented using the sequence map pattern (Figure 1) in order to provide improved guidance. This was done by introducing an additional colour state (green – to symbolise appropriate data entry) to the controls, in addition to the existing colour states of the controls (white – enabled; grey – disabled; and orange – active) to support guidance and task completion.

6 CONCLUSIONS AND FUTURE WORK

ERP systems currently suffer from several usability issues which are caused by complex UIs. Research has shown that AUIs can improve the usability of software systems, but limited guidance exists to support the design of AUIs for ERP systems. This paper has proposed an adaptation taxonomy for ERP systems and the design of an AUI for ERP systems

using this taxonomy.

A prototype AUI was successfully implemented using SBO. The development of this prototype provided evidence to support the potential benefits of AUIs to improve the usability of ERP systems Future work will include conducting a comparative user study to determine if the usability of the AUI is better than the non-adaptive version.

ACKNOWLEDGEMENTS

The support of SAP Research Pretoria and SAP Meraka UTD (CSIR) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at are those of the authors and not necessarily to be attributed to the companies mentioned in this acknowledgement.

REFERENCES

Gajos, K. Z., Czerwinski, M., Tan, D. S. & Weld, D. S. (2006). Exploring the design space for adaptive graphical user interfaces. *Proceedings of the working conference on Advanced visual interfaces.* Venezia, Italy: ACM.

Knutov, E., De Bra, P. & Pechenizkiy, M. (2009). AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. *New Review of Hypermedia and Multimedia,* 15**,** 5-38.

Letsu-Dake, E. & Ntuen, C. A. (2009). A Conceptual Model for Designing Adaptive Human-Computer Interfaces Using the Living Systems Theory. *Systems Research and Behavioral Science,* 26**,** 15-27.

Leung, A., Morisson, S., Wringe, M. & Zou, Y. (2006). Developing an adaptive user interface in eclipse. *In Proc. the Eclipse Technology Exchange Workshop at European Conference on Object Oriented Programming.* Nantes, France.

Peng, X. & Silver, D. L. (2007). Interface Adaptation Based on User Expectation. *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops - Volume 02.* IEEE Computer Society.

Ramachandran, K. (2009). Adaptive User Interfaces for Health Care.

Singh, A. & Wesson, J. L. (2009). Evaluation criteria for assessing the usability of ERP systems. *In:* In Proceedings of the 2009 Annual Research Conference of the South African institute of Computer Scientists and information Technologists (SAICSIT '09), October 12 - 14 2009 Vanderbijlpark, Emfuleni, South Africa. ACM, 87-95.

Tidwell, J. (2006). *Designing Interfaces*, O'Reilly Media, Inc.

Topi, H., Lucas, W. & Babaian, T. (2005). Identifying Usability Issues with an ERP Implementation. *In:* International Conference on Enterprise Information Systems, 24-28 May 2005 Miami, USA.

Tran, V., Vanderdonckt, J., Kolp, M. & Faulkner, S. (2009). Generating User Interface from Task, User and Domain Models. *The Second International Conference on Advances in Human-Oriented and Personalised MEchanisms, Technologies and Services (CENTRIC '09)*. Porto, Portugal.

Van Welie, M. (2008). *A Pattern Library for Interaction Design*. Available: http://www.welie.com/ [Accessed July 15 2010].

Yeh, J. Y. (2006). Evaluating ERP Performance from User Perspective. *In:* IEEE Asia-Pacific Conference on Services Computing (APSCC'06), December 12-15 2006. IEEE Computer Society, 311-314.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS