Minimalist User Modelling in a Complex Commercial Software System

Linda Strachan[1,2], John Anderson[1], Murray Sneesby[2], and Mark Evans[2]

[1] Department of Computer Science, University of Manitoba, Canada
[2] Emerging Information Systems Inc., Winnipeg, Manitoba, Canada

Abstract

While user modelling has produced many research-based systems, comparatively little progress has been made in the development of user modelling components for commercial software systems. The development of minimalist user modelling components which are simplified to provide "just enough" assistance to a user through a pragmatic adaptive user interface is seen by many as an important step toward this goal. This paper describes the development, implementation, and evaluation of a minimalist user modelling component for the Tax and Investment Management Strategizer (TIMS), a complex commercial software system for financial management. This user modelling component manages several levels of adaptations designed to assist novice users in dealing with the complexity of this software package. Important issues and considerations for the development of user modelling components for commercial software systems and the evaluation of such systems in commercial settings are also discussed.

Keywords: Pragmatic User Modelling, Knowledge-Based Systems, Empirical Evaluation, Commercial Software Systems, Financial Planning.

1. Introduction: Pragmatic User Modelling

Modern software systems are complex, often supporting a broad variety of tasks and diverse groups of users with widely varying problems and needs. This complexity has led to an increased focus on the user in commercial software development and an increasing research focus on user modelling and user-adapted interaction.

A user model is an explicit representation of the properties of a particular user (Jameson et al., 1997). The goal of user modelling as a research area is to create systems that are adaptive to an individual's needs, abilities, and preferences. User modelling techniques have been developed and evaluated by researchers in a number of fields, including: artificial intelligence, education, psychology, linguistics, human-computer interaction, and information science.

In recent years, user modelling has become more mature as a research field. Current research systems show great promise in their ability to deliver a wide range of adaptability through user modelling techniques such as stereotyping and the separation of user knowledge into a user model database. However, it is also being recognized that in spite of the demonstrated capabilities of many research-based systems, comparatively little progress has been made in taking advantage of individualized interactions in commercial software systems. This lack of progress, like the lack of emphasis on empirical studies, is considered to be one of the most important concerns in modern user modelling research (McTear 1993; Langley 1999).

Still, some limited successes in the commercial arena are becoming known. The first prominent example is Microsoft Office Assistant, which was shipped as a component of the Office 97 and Office 2000[1] suite of products. It is meaningful to the research area that a major software producer has bundled this technology with one of its most visible products (Kay and Fisher, 1997). Intelligent agents (or personal assistants) are also a highly touted recent development in software, particularly because of the easy access to people and the vast amounts of information through the World Wide Web. Software agents that actively assist the user include the Firefly passport, which provides protected user profile information to participating web sites (Firefly 1998). Personalized web search services and website appearance tailoring are also becoming more common. A good example of a customized search engine on the web is My Yahoo! (My Yahoo! 1998). By setting up a user profile, individuals can personalize the site, tailor information such as news and sport team scores, and modify the site's appearance to suit their personal preferences. More practically from a commercial standpoint, advertising displays are also customized based on user profiles. Also from a commercial standpoint, Fink and Kobsa [2000] illustrate the profound potential dollar value in personalized e-commerce applications in the near future. Student modelling is also represented on the web, a good example being a web-based LISP course that adapts to the student (Weber and Specht 1997).

The academic community is certainly aware of the current trends. The two major conferences that focus on user adaptive interactions, the *International Conference on User Modeling* (UM) and the *International Conference on Intelligent User Interfaces* (IUI) have in recent years included debates on how to translate the research developments into commercial practice. This issue was also addressed directly in two workshops that were organized at the recent international conferences of user modelling; *The Commercial Potential of User Modelling* (1996) and *User Models in the Real World* (1997). The overall theme of the Intelligent User Interfaces conference in 1999 was the transition of new scientific and technological advances to real world applications (IUI 1999)

While current research momentum appears to be moving in the right direction, particularly so in the case of web-based applications, there is still a lack of commercial deployment in user modelling systems. Part of the reason for this is the fact that there are important considerations in commercial software systems that are largely ignored in research systems. The most obvious consideration is the performance overhead that must be incurred when a user modelling system is employed. Also important are the time and expense that are involved in including a user modelling component, in comparison to the often unproven advantages that the added expense may bring. Less obvious but equally

[1] Microsoft Office is a copyright of Microsoft Corporation.

important factors include the design changes that may be necessary for the inclusion of a user modelling component, as well as the costs of training and support in an environment where a system's interaction is not necessarily consistent between one user and the next.

Despite the many details of design, implementation, maintenance, and support overhead, there is a much broader reason for the lack of commercial user modelling systems that has been speculated upon in the literature. Many of the approaches embodied in research systems are too complex or impractical for use in commercial software systems (Kobsa 1993; McTear 1993, 1996; Oppermann 1994; McCalla et al. 1996). The identification of this shortcoming has prompted some researchers to develop *pragmatic* user modelling architectures and techniques that have been simplified from theoretically motivated approaches. From a commercial standpoint, this "good enough" user modelling involves the inclusion of a minimalist user modelling component that has some of the major advantages demonstrated by large research systems, with minimal cost and commercial disruption.

It is anticipated that minimalist approaches to user modelling will become more popular as practical applications of user modelling receive greater consideration. McTear (1993) cites several systems designed for information filtering and adaptive hypertext documents, which employ simplified user modelling techniques to provide "good enough" user modelling in their respective domains. There is also the potential that minimalist user modelling components will provide a practical degree of interface adaptation while avoiding the common problem of invoking unrealistic expectations from users (Erickson 1997). In a 1996 workshop report (Malinowski 1996), the cost of building adaptive components for real-world systems was identified as a universal concern, and several of the contributed papers involve simplified techniques for use in commercial environments. The amount by which a user modelling component slows overall system performance is also an issue even in existing deployed systems (e.g. [Linton, 2000]), leading to the hope that minimalist approaches will balance the power of user modelling techniques with less impact on application performance. In addition to these factors, Shneiderman (1997) points out that users require a predictable and consistent interface in order to feel in control of a system. It is hoped that practical minimalist approaches will also allow a balance between interface predictability and software usability in complex commercial systems.

Recently, McCalla et al. (1996) employed a "slightly intelligent" design they called *analogical user modelling* for document retrieval that was relevant to an individual. Here, the user model was not complete, but it captured just enough information to allow the relevant analogies to be found. The user was then required to guide and catch the system when it failed. A proof of concept system was empirically tested and was shown to be effective. McCalla et al. argued that the user and system could work together symbiotically and that, given the current state of the art in the field of artificial intelligence (AI), minimalist approaches may be a pre-requisite for the successful application of AI techniques both inside and outside the domain of user modelling.

While the application of minimalist user modelling techniques to commercial systems shows promise, many questions remain. Many involve the efficacy and cost-benefit ratio of a minimalist user modelling component. Can a relatively small number of user-specific adaptations help to support the user in navigating a complex computer application? Can these adaptations be employed practically in a commercial system? What balance is needed between sophistication and cost of a user modelling system and the practical considerations of commercial software development? There are also many practical issues, such as user involvement and management support, which must be addressed in supporting user modelling in ongoing commercial software development.

This paper describes the design, implementation, and evaluation of a pragmatic user modelling component for the Tax and Investment Management Strategizer, or TIMS[2], a complex commercial financial management software system. The addition of this component is intended to address the needs of novice users while minimizing the impact on expert users. This is achieved by reducing a novice's perception of the complexity of the system through the use of constraints and by exposing some of the system's underlying processing. A careful balance is required in order to adequately support novices while trying to avoid irritating expert users with extraneous interactions.

This paper presents an analysis of the domain, a description of the functionality of the user modelling component, and the description and results of a study that was designed to identify the impact of this component on user satisfaction levels. This work is concerned with the entire development cycle of an adaptive system, and touches upon areas from task analysis and design to implementation and evaluation. The user modelling component that is described here is designed to address many of the concerns that system developers have with applying user modelling in commercial systems. As such, the final section of this paper describes some of the important issues involved in applying user modelling to commercial systems from the standpoint of this research.

2. Tax and Investment Management Strategizer (TIMS)

The Tax and Investment Management Strategizer (TIMS) is a sophisticated commercial software system for financial planning developed by Emerging Information Systems Inc. (EISI) of Winnipeg, Canada. The current version of this system is used in the Canadian and US marketplaces by banks, insurance companies, trust companies and brokerage firms. TIMS allows a financial planner to analyze the financial status of a client and to demonstrate and evaluate various financial planning strategies in order to improve the client's financial situation. The planning process combines data such as: (1) assets, liabilities, incomes, and expenses; (2) a number of applicable planning strategies; (3) interactions among planning strategies; and (4) objectives set by the client. The system's objectives are to maximize the effectiveness of expert planners and to increase the effectiveness of non-experts in the application of financial planning strategies. The latter

[2]TIMS is a copyright of Emerging Information Systems, Inc.

will enable individuals with limited training in financial planning to employ both simple and more complex financial planning strategies.

TIMS uses a desktop metaphor and operates with datasets known as *plans,* which represent the financial situation of a client at a particular time. A TIMS user creates a plan through a series of data entry dialogs, and the system creates a graphical display of the client's financial situation in a dialog window known as the *Plan Window*. The financial planner can then implement and evaluate numerous financial planning strategies to produce alternative plans.

Much of the user's interaction with TIMS is performed through system components known as *strategies*. Strategies implement financial planning techniques that human planners suggest to their clients or those that clients already have in place, which properly reflect their overall financial picture. For example, a Regular Savings strategy exists that represents the commitment of assets with a particular regularity over a particular period of time or until some event (e.g. saving $100/month until retirement). Nineteen different strategies are implemented in the current version of TIMS, including a variety of savings strategies, debt reduction strategies, and asset redemption strategies. Strategies are intended to provide an intuitive method of implementing financial planning tasks and to free users from excessively detailed input. Strategies can be invoked at a high level in order to automatically contribute their expertise by defining various low-level transactions (e.g. numerous $100 saving events in the above example) that will then be automatically carried out over time. The financial planner can also insert the appropriate transactions at a more detailed level of input (e.g. define repetitive low-level $100 savings events in this situation). Strategies serve to insulate the novice: by invoking the strategy at a high level, novice users are not exposed to an excessive level of detail. Indeed, the standard level of menus in TIMS does not provide access to dialogs allowing the insertion of detailed individual transactions. Novices are thus presented with the basic features of the system in a structured environment, moving to detailed levels only when what is required goes beyond the strategies implemented.

TIMS also provides three unique *Assistants*, which are knowledge-based components that provide intelligent support for system interaction. Assistants provide an even higher level of interaction by allowing the automatic analysis of situations and the generation of recommendations for system tasks. These assistants are the *Planning Assistant, Cash Flow Assistant*, and *Strategy Assistant*. Each assistant performs a unique supporting role. For example, users may analyze and implement financial planning strategies in a variety of ways. As described above, expert users may directly invoke built-in strategies as a novice would, or may actually perform the detailed purchases and redemptions directly in the system. Novice TIMS users are encouraged to use the *Planning Assistant* (Figure 1) for the analysis of tentative strategies and for final plan creation.

The Planning Assistant is designed to help automate common financial planning tasks, and it offers practical planning advice that is calculated to help a user maximize a client's overall financial situation. Each plan undergoes an analysis of its incomes, expenses, assets, liabilities, savings programs, and taxes. The analysis of any individual category is displayed in the top portion of the window, while the customized recommendations are displayed in the bottom window. If appropriate, strategies that have been recommended by the Planning Assistant are displayed in quick access buttons below the recommendations. From within the Planning Assistant, the user has access to a menu of Data Entry dialogs that allow him or her to modify the client's data and the strategies that have been used, and invoke the other Assistants that TIMS provides. In general, assistants require direction from the user to implement strategies because the final decision rests with the planners. Financial planning is very subjective and the system would not be accepted if it intrusively dictated general solutions.

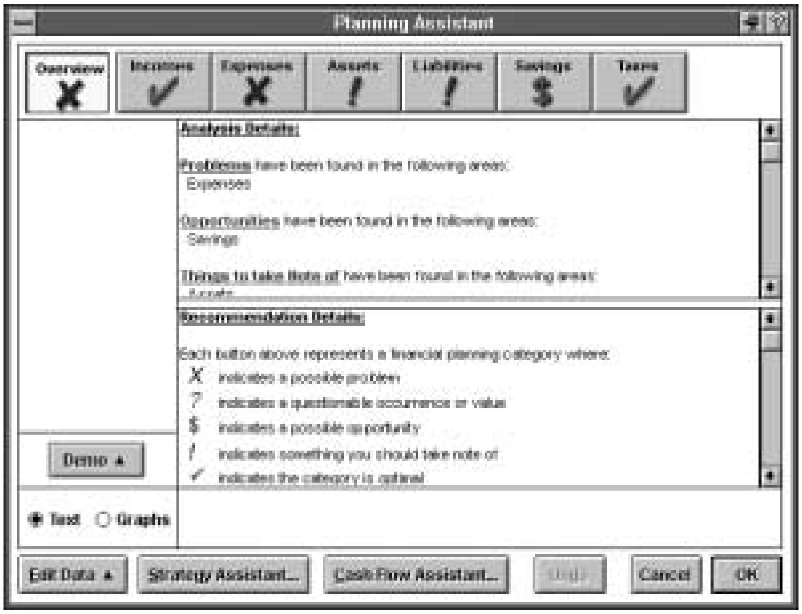

Figure 1. The Planning Assistant showing the overall analysis of a situation.

TIMS also provides intelligent *Cash Flow* and *Strategy Assistants* that perform analogous functions for their particular aspects of the system. The Cash Flow Assistant provides a complete analysis of a client's cash flow for any given year. The Cash Flow Assistant allows the financial planner to make changes to the client's current situation (by entering data directly or through the use of strategies), and view the results. It is also possible to display the ongoing transactions that meet a client's investment objectives. The Strategy Assistant is designed to help the user easily select and apply multiple financial planning strategies. It is also useful when the financial planner requires a quick overview of the number of strategies currently being applied in a particular situation.

Once the desired result has been achieved by implementing strategies and by modifying the initial client data, a completed financial plan can be created by using built-in Document Packages. These Document Packages can be used to export one or more reports or graphs produced by the system into a template to create a Rich Text Format (RTF) file. This type of file can be read by most current word processors. This complex process is made easier by allowing users to simply drag-and-drop icons representing client situations onto icons representing currently available Document Packages.

The basic construction of the TIMS program provides a sophisticated tool that can provide complete support to a computer literate, experienced financial advisor. The problem is that not all TIMS users are members of this group – and in fact, the group is a very select one. Like other commercial systems functioning in complex domains, TIMS must support a diverse range of users along two major dimensions. While many financial planners still have little computer expertise, some are sophisticated computer users with little or no need of lengthy demonstrations and extensive explanation facilities. Just as importantly, many TIMS users are expert financial planners with extensive knowledge of the domain, and use TIMS mainly for convenience. Others will have only limited financial planning skills and will rely on TIMS to fill gaps in their own knowledge. Even among expert financial planners, the extent of their knowledge of the domain varies from area to area. For example, an expert in asset allocation may have much less knowledge of life insurance or estate planning.

A study (Fossay and Sawchuk 1994) that was conducted on the first commercial version of TIMS (released in 1995) confirmed the power and sophistication of the system, but identified concerns with the complexity of the system and the impact of this on novice users. These concerns, along with the breadth of user experience that must be supported and the goals of TIMS itself, led to the decision to develop an adaptive user interface for TIMS. The use of a minimalist user modelling component to implement this adaptive user interface arose from commercial concerns and the issues described in Section 1. These were essentially a desire to bring the benefits of user-adapted interaction to the system with a minimum of cost and commercial disruption. The design and implementation of this user modelling component and the nature of its adaptations are described in the following section.

3. Supporting User-Adapted Interaction in TIMS

In the initial stages of developing a user modelling component for TIMS, the researchers decided to focus on the necessary adaptations for novice users in order to help them navigate through the system and perform required tasks. This was primarily because TIMS was originally developed in collaboration with high-end expert financial planners, and it had already been structured to reflect the organization of material that was helpful to them. This type of user recognizes that the power of the system is essential, and he or she feels that it is worthwhile to invest time in order to use it properly. A focus on novices, however, does not eliminate these expert users from consideration. The usability of the software by more experienced users must not be limited by the adaptations that are necessary for the support of novice users.

A number of areas where novice support could be increased in TIMS via a minimalist user modelling component were identified through interactions with users during the testing of the initial release of the system. Input was gathered through informal user interviews, training sessions with novice users, and telephone support calls. Two guidelines were used for the selection of adaptations for the TIMS user interface. The first guideline was that the selected adaptations had to be expected to assist the novice user. The second guideline was that these adaptations must require only a minimal user modelling component to support them. This selection process resulted in four major categories of user adaptations: the addition of animated demonstrations of tasks considered to be more difficult for new users; a Strategy Interaction Detector; a Strategy Recommendation component; and assistance in the data entry process. These adaptations were implemented in several stages, and they resulted in the development of Personalized-TIMS (P-TIMS), an adaptive version of the TIMS system.

3.1 A Minimalist User Modelling Component for TIMS

The basic architecture of the P-TIMS user modelling component (UMC) is depicted in Figure 2. It consists of a long-term user model (stored in a user model database); a task model containing descriptions of a subset of tasks in the system (such as rankings of the complexity of strategies and Assistants); and an application model containing information about the relationships between a subset of the tasks in the system (such as information about the strategies or combinations of strategies that are equivalent to each other). The inference engine contains simple rules about the relationships between these three models and the possible adaptations that can be performed on the application and the interface.

The P-TIMS user modelling component has a two-way exchange of information with the user interface and the application. The information that is received by the UMC is essentially an ongoing description of the user's interaction with the system. From the interface, the UMC receives the user's interaction history as well as explicit changes that have been made to the user model via the system's Preferences dialog. The application's current status is received from the application itself. Information that is sent from the UMC to the other components causes adaptations to be realized. These adaptations result in additional or modified dialogs, animated demonstrations, and modifications or limits to the functionality of the system.

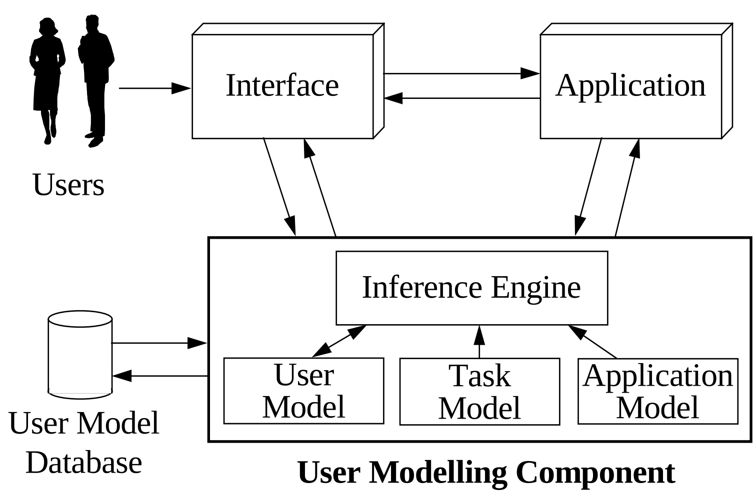

Figure 2. The user modelling component architecture of Personalized-TIMS.

Each P-TIMS user has a unique user model, which is generated the first time the user logs into the TIMS program and is reused for subsequent entries into the system. However, the term *user model* is not employed directly in the system, as many users are already familiar with the concepts of a User ID and a User Preferences option through exposure to other software packages. This was done partly to avoid introducing needless concepts to the user, but more importantly from a commercial standpoint, to keep potential differences between the adaptive and static systems as small as possible.

The user model is initially based on a set of stereotypes that are triggered by the user's job title (chosen from a menu of generic job titles) and a self-assessment of Windows experience and previous TIMS experience. The use of such stereotyped user expertise levels was first demonstrated in (Chin 1989). In this approach, each of the two experience categories is divided into Novice, Intermediate, and Expert levels. Because of the two-dimensional diversity of TIMS users described in the previous section, a decision was made to have four stereotypes within P-TIMS: Novice TIMS users (NTIMS), Novice Financial Planners (NFP), Experienced TIMS users (ETIMS), and Experienced Financial Planners (EFP). These four categories were chosen as a reasonable representation of the TIMS user population in order to keep the number of stereotypes to a minimum. Four possible generic user models can be created from these four stereotypes, as illustrated in Table 1.

Table 1. User model created as a result of the initial user questionnaire.

TIMS Expertise	Financial Planning Expertise	
	Novice	*Experienced*
Novice	NTIMS + NFP	NTIMS + EFP
Experienced	ETIMS + NFP	ETIMS + EFP

	Experienced Financial Planner	Novice Financial Planner
User Name:	User supplied	User supplied
Job Title:	User supplied	User supplied
Frequency of Use:	0	0
Date of Last Access:	Today's date	Today's date
Last Update	Today's date	Today's date
User Type:	EFP	NFP
Financial Planning Expertise:	High	Low
TIMS Experience:	User supplied	User supplied
Windows Experience:	User supplied	User supplied
User Complexity Rating:	3	1
TIMS Strategies: name: (for each individual strategy)		
counter:	0	0
date of last use:	Date	Date
demo counter:	0	0
date of last demo:	Date	Date
demo refusals:	0	0
Assistant Modules: name: (for each of the assistants)		
counter:	0	0
date of last use:	Date	Date
demo counter:	0	0
date of last demo:	Date	Date
demo refusals:	0	0
	Experienced TIMS	**Novice TIMS**
User Settings:		
Show Task Demonstrations	False	True
Show Strategy Interactions	True	True
Suggest Strategies	True	True
Data Entry Assistance	False	True

Figure 3. The attributes of the four P-TIMS stereotypes.

The most important initial attribute that is requested of the user is his or her job title, as it is used as a basic estimate of the user's financial planning expertise and also as the trigger of the appropriate financial planning stereotype (EFP or NFP). The advantage of using a job title as such an indicator is that it is a very simple parameter, and it is easily incorporated into the first login registration procedure. Other simple variables that may be appropriate in other systems include: professional designation, location, or rank. The second component of the user model is appended to the first stereotype when the TIMS experience level is selected during this same procedure. Novice and Intermediate TIMS users trigger the NTIMS stereotype, while Experienced TIMS users trigger the ETIMS stereotype.

Figure 3 displays the individual components of the Financial Planning and TIMS experience stereotypes. The combination of the appropriate Financial Planning and TIMS experience stereotypes results in a basic user model, an example of which appears in Figure 4. The user model contains information about the user's interaction with each of the TIMS strategies and Assistants (allowing a simple interaction history to be available to the user and to indicate the strategy and Assistant demonstrations that have been previously viewed) and specific settings for the various adaptations (described in detail below) that are supported by the user modelling component.

User Name:	Peter Planner
Job Title:	Company Representative
Frequency of Use:	11 Days
Date of Last Access:	November 3, 1999
Last Update:	September 30, 1999
User Type:	NFP
Financial Planning Expertise:	Low
TIMS Experience:	2
Windows Experience:	1
User Complexity Rating:	2
TIMS Strategies: name: *	RRSP Maximizer
counter:	6
date of last use:	September 30, 1999
demo counter:	2
date of last demo:	August 20, 1999
demo refusals:	0
Assistant Modules: name: **	Planning Assistant
counter:	8
date of last use:	November 3, 1999
demo counter:	1
date of last demo:	August 20, 1999
demo refusals:	0
User Settings:	
Show Task Demonstrations	False
Show Strategy Interactions	True
Suggest Strategies	True
Data Entry Assistance	True
*For each individual strategy	
**For each of the Assistants	

Figure 4. User model of a novice financial planner.

The task model (Figure 5) in the UMC contains simple information about various system functions. Although the primary focus is on the modelling of strategies and Assistants, other complex functions such as client report creation are also represented. Each strategy and Assistant is represented in the task model with a complexity rating, a Lotus ScreenCam[3] demonstration filename, two levels of explanation, and an appropriate index to the hypertext help system. Any function that is not represented in the task model is assumed to be at the base level of complexity (represented by the numeric value 1, and indicating that most users will find these components reasonably straightforward with minimal training and support). Complexity ratings were chosen as a simple method to determine if a task was appropriate given the user's level of experience. In a more complex system, a task hierarchy might be more appropriate; only expert users would have access to the entire hierarchy (Vassileva 1994). The ratings themselves are heuristic in nature based on our experience in developing and working with TIMS. These measurements are also minimalist compared to other approaches (such as that of Linton et al. [2000], where the model of how an expert would view a task is actually based on a continually updated ranking of individuals) and certainly suffice for the scope of the adaptations described in this paper.

[3]Lotus ScreenCam is a trademark of Lotus Corporation.

Task Model*	
Name:	RRSP Maximizer
Complexity Rating:	2
Demo Available:	"RRSPMAX.SCM /SCH"
Explanations:	
Low-level	explain[rrspmax].low
High-level	explain[rrspmax].high
Availability to user type:	All
Help Screen:	2:4610 3025
* One entry for each strategy and Assistant	

Figure 5. RRSP Maximizer Strategy representation in the task model.

At the current time, all user levels are able to access all system functions, but the UMC has the ability to disable tasks based on the task model. Certain user groups may not be authorized by their corporations to perform certain types of financial analysis. For example, front-line bank employees might be permitted to perform simple analysis tasks for a client, but they would be required to refer more complex analysis to another employee who has the appropriate expertise and authority.

The user modelling component's application model (see Figure 6) contains specific information that provides support for financial planning strategy recommendations and for the analysis of interactions between financial planning strategies. In the current version of the system, sixteen equivalent strategies (the upper portion of Figure 6, where *Location* indicates the more complex strategy and *Recommendation*, the simpler alternative) are represented, and seventeen possible combinations of strategy interactions can be detected. The detection of interactions is an important task because certain combinations of strategies are a potential problem if they are active for the same asset or assets and are applicable during an overlapping period of time. The use of this knowledge will be explained in Section 3.

Application Model (a small portion)	
Equivalent Strategies:	
Location:	Detailed Assets
Transaction:	Future Returns
Selection:	N/A
Asset Type:	All
Recommendation:	Investment Returns Strategy
Location:	Detailed Liabilities
Transaction:	Principal Modifier
Selection:	Periodic
Asset Type:	All
Recommendation:	Regular Debt Reduction Strategy
Strategy Interactions:	
Strategy_combo:	Surplus Cash Savings & RRSP Maximizer & Surplus Debt Reduction
Asset Applicable Period Overlap:	No
Explanation:	
"This combination of strategies will be performed in the following order: 1) RRSP Maximizer, 2) Surplus Debt Reduction and, 3) Surplus Cash Savings. There may be less surplus cash for debt reduction because of the RRSP Maximizer Strategy. Savings will take place only if there is a remaining surplus following debt reduction."	
Strategy_combo:	(Surplus Cash Savings or Regular Savings or Lump Sum Savings) & Complete Regular Redemption
Asset Applicable Period Overlap:	Yes
Explanation:	
"The asset(s), *asset_names*...., have both a savings strategy and the Complete Regular Redemption strategy associated with them. This may prevent these assets from being fully redeemed by the end of the period."	

Figure 6. Representative entries in the application model.

The inference engine sub-component of the UMC contains the processing components that support the events that occur during system interaction with the user model. These include: user model creation; user model retrieval from, and storage to, the database; initialization and re-classification of the user; updates and modifications to the user model made during system usage; triggering of appropriate demonstrations and dialogs; and the interaction of the users with these dialogs. In order to simplify the processing as much as possible, both the task and application models are static, and they are not modified by the UMC. They are intended to provide information about a specific application's structure and functionality.

This simple architecture embodies a very important underlying assumption: *the user model is the property of the user, not the user modelling component*. One design objective of the adaptive system was to make the user model visible and modifiable by the user. This was done in order to take advantage of the benefits outlined by others (Cook and Kay 1994) including: the user's right to access information about themselves, the programmer's accountability, the user's ability to correct and verify information, and the importance of making complex systems more comprehensible. In P-TIMS, the user model attributes have been made accessible to the user through a Preferences dialog (Figure 7). Thus, while the most significant updating of the user model occurs during the initialization phase of the system, the user can also update his or her own user model at any time.

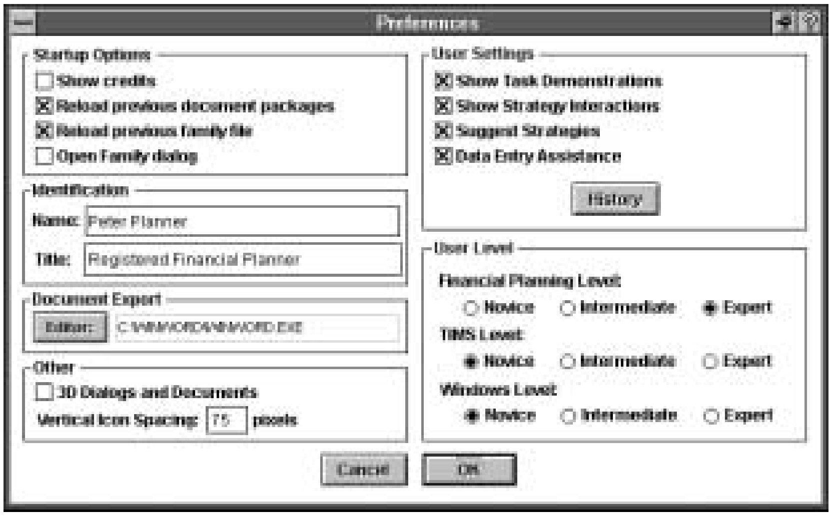

Figure 7. Preferences dialog showing all adaptations activated.

Modifications to the user model also occur as the user performs tasks represented in the task model. Reclassification of a user in the initialization phase is done only once per day in order to minimize potential problems with users repeatedly entering and exiting the program during initial experimentation (see Figure 8). This design decision also incorporates an element of common sense, in that registering ten system uses over a period of ten days would typically imply a different degree of system familiarity than the same number of uses in one day.

```
If the date has changed do the following:
    If the user has been away too long (currently 30 days)
        Downgrade TIMS experience level by 1 category
    If the user has the used the system long enough since the last update (currently every 10 sessions)
        Upgrade TIMS experience level by 1 category
    Examine all strategies/functions used recently (within the last 5 sessions) by the user
        If more than 2 are more complex than the user complexity level then
                Increase user complexity level by 1
        If all are less complex then
                Decrease user complexity level by 1
        Update financial planning expertise
    Update counter fields
```

Figure 8. The basic algorithm for update of the user model in the initialization phase.

The user modelling component activates the user model maintenance procedures during the initialization of the system on the first session of the day. The user model is updated with the new access date, and the frequency of access attribute in the user model is incremented. Frequency of use has been established as a major determinant of a user's capability, and a significant decrease in user capability occurs after a break of some weeks (Maskery, 1984). Based on this principle, the TIMS experience attribute is upgraded if the system has been used more than a threshold frequency, and is downgraded if the system has not been used for a lengthy period of time. The UMC then checks the history of the user's interaction with the system and updates the user model. For example, the user complexity rating for the user model will be increased if the user has implemented more than two strategies that were more complex than the last updated user complexity rating. The evolution of user models in this scheme will be described in further sections within the context of the specific adaptations available in P-TIMS.

Over time, the adaptive system gradually diminishes into the original, non-adaptive system. That is, the adaptive system has a planned obsolescence. Because the user modelling component is focusing on the support of novice users through the system, support is terminated with the user's permission once a user has achieved a minimum level of proficiency and has used the system for a predetermined period of time. This approach is similar to the Microsoft Office 97 adaptive Office Assistant, which stops giving advice when the user has mastered the task at hand (Microsoft 1998). When the P-TIMS system determines that the user should be re-classified, a proposal is presented to the user that explains the implications of a reclassification and asks the user to confirm or reject the proposal. Others have also argued that individualized systems are most required for supporting novices, and that the significance of such features fades as users become more adept (Vassileva 1996). Different types of adaptations in P-TIMS have different termination points. The intermediate or expert user starts with fewer supports, which gradually erode once the system has been used for a period of time or once the user has decided to cancel the adaptations. In all cases, the user has ultimate control over the termination of supports. Any reclassification that is proposed is presented to the user, with an explanation of the implications of reclassification, for the user's confirmation or rejection.

The benefit of the planned obsolescence of the adaptations in the P-TIMS system lies in its ability to minimize the problems that users have with systems that are inconsistent between users and over periods of time. These problems are minimized due to the fact that experienced users will tend to have what is essentially a non-adaptive system. Problems that are associated with inconsistency are significant, yet underemphasized, in commercial settings, and they include the loss of shared experiences with other users when their system has personalized information. For example, a common frame of reference would not exist with personalized newspapers that are filtered by an intelligent agent. One could not simply ask, "did you see the article in the paper today?". Over time, readers would tend to become more and more knowledgeable about certain topics, and they may not be as aware of new issues and topics that appear periodically. Perhaps a greater problem from a commercial prospective, user-specific adaptations can add a level of complexity to product support or help desk support to individualized systems. Even the informal networks of peer support, user communities, or "power users" lending support to others may be jeopardized. If an adaptive system is designed to support collaborative work, then the personalized views can't be too different or it will effectively terminate the collaboration (Vassileva 1994).

The architecture and basic functionality of the user modelling component have been described only in general terms. A greater understanding is provided by examining the user modelling component within the context of the currently supported adaptations in P-TIMS. The adaptations were implemented in three major phases, and are described in the following sections.

3.2 Phase 1: Animated Demonstrations

During training sessions and marketing presentations, it was noted that novice users, and particularly first-time users of TIMS, experienced a particularly high level of uncertainty when trying to perform new tasks. This uncertainty led to errors and wasted time and energy. Expert financial planners required assistance with tasks such as: mapping their domain knowledge into the system, understanding the types of functionality provided by the system, and knowing how to perform specific tasks with it. Less experienced planners also needed some assistance in the form of suggestions on tasks to perform and extra confirmation for the successful completion of these tasks. These types of TIMS novices can be better supported by demonstrations of tasks before such tasks are attempted for the first time. For that reason, the first major phase of adaptations involved adding support for animated Lotus ScreenCam demonstrations.

This type of support is familiar to Microsoft Office users who have experienced the "Wizards" and animated Demos that are available in the help subsystems of software such as Microsoft Word[4] and Microsoft Excel[5]. It was anticipated that the result of this addition to the system would be a decreased number of errors, less uncertainty in performing a task for the first time, and increased user satisfaction with the system. An important consideration of this work was the validation of these hypotheses through empirical methods. Details of the experimental process and the findings that resulted from it are outlined in Section 4.

Lotus ScreenCam was chosen to create animated demonstrations of significant TIMS tasks because of its ability to support recorded speech and text captions along with the visual presentation. Text captions were ultimately chosen over recorded speech in order to enhance and clarify the display. There were two main reasons for this. Not all TIMS users have the necessary hardware support for audio presentations, and the use of recorded speech results in substantially larger files. The issue of file size was an important one, due to great variability in current and potential TIMS users' hardware. (In general, large variations in hardware platforms are another significant problem in commercial user modelling.) The complete adaptive system, which includes a relatively small number of movies, contains just over six megabytes of ScreenCam files. Narration would have made these files approximately 7.5 times larger.

Each animation provides for one of three possible supports: (1) familiarization of the user with the overall structure of the TIMS system (including the Preferences dialog, to make

---

[4] Microsoft Word is a copyright of Microsoft Corporation.
[5] Microsoft Excel is a copyright of Microsoft Corporation.

users aware of their direct control over the adaptations they experience); (2) familiarization of the user with the structure of the supporting TIMS components (including all 19 strategies and three assistants); and (3) demonstrations of how to perform particular TIMS tasks. Each demonstration is generally less than two minutes in length and shows the user how to perform any necessary functions. Limiting the demonstrations to two minutes was enforced because pilot testing of early versions of the system revealed that this was the approximate maximum user attention span for such a demonstration. Most demonstrations, such as those explaining strategies, are available for replay through a small movie camera icon in the appropriate dialog's title bar (see the top right corner of Figure 7).

Like the other adaptations to be described, the demonstration facility is directly controlled by the User Modelling Component (UMC) described in Section 3.1. When the user attempts to implement a TIMS strategy or to invoke an Assistant, the UMC is called to determine if a demonstration is to be shown to the user. The UMC determines if the Show Task Demonstrations attribute (visible in the Preferences dialog depicted in Figure 7) has been deactivated. This could occur through a variety of methods: by the user directly, indirectly by the user refusing further demonstrations, or by the user achieving an expert TIMS experience level. If the option is determined to be active, the user is asked if he or she would like to view a demonstration of that particular component. If the user refuses a particular demonstration, a second dialog is presented which allows the user to discontinue all prompting regarding the demonstrations. This deactivates the Show Task Demonstrations checkbox in the Preferences dialog, but a user is free to reactivate it again.

P-TIMS' demonstration facility supports the most inexperienced users of TIMS. The impact of this facility on users is fairly minimal once they have passed the initial start-up phase of the system. Unfortunately, the first few sessions require the greatest amount of input from the user, but these same sessions are also where the greatest level of assistance is required. Major changes for the users of the adaptive system include the login procedure, occasional viewing of animated demonstrations, and prompting to terminate demonstrations. In the worst possible case, users who were averse to any interruptions in their use of the system would be forced to respond to the P-TIMS demonstration query and the termination of demonstrations query immediately after entering the system. If the demonstration facility is deactivated immediately, the user would only need to respond to the first dialog.

This amount of interruption was deemed manageable. Earlier versions of P-TIMS triggered a demonstration the first time that any individual strategy or Assistant was used. However, during pilot testing of the original adaptive system, it became obvious that this level of intrusion irritated some users. This led to a re-design of the animated demonstration triggering conditions. The next iteration of P-TIMS triggered only the first two distinct demonstrations. From a design standpoint, it seemed that two short demonstrations would be sufficient to help the new user become familiar with the system and to establish how to invoke further demonstrations.

A review of user comments, following the subsequent re-design and testing cycle clearly showed that the users did not like any automatic triggering of demonstrations and that they preferred to choose whether or not they viewed each demonstration. It was deemed particularly annoying because the ScreenCam mechanism to exit from the demonstration was hidden to avoid confusing the user. Automatic triggering was especially confusing to very new users who were not familiar with any other software packages: to this group of users it seemed almost magical. In one memorable incident during pilot testing, several users were observed looking away from their monitors while the first automatically triggered demonstration was playing. When they were asked why they were doing this they replied, “We don’t know what we did so we’re waiting until it’s over!”. As a result of this misunderstanding and the previously mentioned feedback from the second large pilot test of the P-TIMS system, the designer’s final decision was to prompt the user before showing any demonstration.

3.3 Phase 2: Strategy Recommendations and Strategy Interaction Detection

While demonstrations of strategies are certainly helpful to novices, some strategies are more difficult to employ than others, and may also interact in ways that are not always obvious. A good example of a strategy interaction in the TIMS system is the common occurrence of situations that include both savings and debt reduction strategies. Each strategy affects the client’s cash flow and may cause deficits to occur in the plan. Despite these potential difficulties, strategies are the preferred method of implementing ongoing transactions, particularly for the inexperienced user. Strategies provide the user with the meta-task presented in his or her own terms so that the user does need not be concerned about the low-level system details behind the strategy.

The second set of implemented adaptations was designed to recommend simpler alternatives (based on strategies) to complex actions in the system, and to point out interactions between strategies. These adaptations rely heavily on the task and application models that are maintained in the UMC (Figures 5 and 6), which is described in Section 2. All of the strategies and the more difficult system functions are represented in the task model. This representation includes a task complexity rating (a range from 1 to 4, with 1 denoting “simple” and 4 denoting “very complex”). A user complexity rating is also maintained as part of the user model (Section 3.1), and when a complex procedure is invoked, this rating is compared with the complexity rating of the task in the task model. The application model contains a mapping between complex tasks and low-level tasks (or sequences of low-level tasks). If a user’s rating is lower than the complexity rating of the task that he or she has selected, then the application model is checked for a simpler solution. If one exists, a recommendation is presented. The user may choose to disregard any suggestions and may also choose to deactivate subsequent suggestions at any time.

Most of the available recommendations involve suggesting replacements for sequences of low-level detailed transactions that are equivalent to various strategies. These strategies provide well-defined guidelines that insulate the user from the details of the transaction and provide the same functionality. In addition, the ScreenCam demonstrations that are described in Section 3.2 are also available to the user to illustrate the use of the strategy.

The user complexity rating that forms the basis for the strategy recommendation adaptations is initially set by the triggered financial planning stereotype that is described in Section 3.1. (The NFP stereotype assigns an initial value of 1; the EFP stereotype a value of 3.) However, like other components of the user model, it evolves over time as a result of the user's interaction history with the system. The user complexity level is increased by 1 if the user has invoked at least two functions or strategies that have a complexity rating that is higher than the current user complexity rating (Figure 8). This simple method is analogous in its minimalism to the learned or unlearned states associated with domain rules used by Corbett et al. [2000], and is reasonable in this situation. If a user's abilities have been underestimated, the correction is made once two slightly more complex strategies or functions are used. This would likely take place in a relatively short period of time. In a case where a severe underestimation is made, it will take consistent complex feature invocations and at least ten sessions over the course of ten days (because the user model is only updated daily) to position the complexity rating at the proper level. Normally, however, this complexity level increase would occur gradually. A user would undergo training, gain experience, and increase his or her TIMS knowledge, resulting in the use of the more complex features of the system. The UMC would then increase the user's complexity level appropriately. In a case of severe underestimation of ability, the user can (and likely would) turn off undesirable adaptations directly.

If the user's ability has been overestimated, the user complexity rating in the user model will slowly decrease over time (unlike Corbett et al.'s [2000] production rules, which do not incorporate a model of forgetting over time because of a lesser emphasis on this aspect in their domain). Like the increase of the user's complexity level, this processing is performed only once a day. The user complexity rating is decreased by 1 if the user has not recently used any functions or strategies that are at least as complex as the user complexity rating. "Recent" is defined in the system as within the last five uniquely dated sessions, typically a one-week period. The reasoning behind this processing is that a user who has not performed complex tasks in this period of time has probably been misclassified as a more advanced user. A Last Update field in the user model allows the UMC to periodically check the user complexity. Again, individuals may increase or decrease their rating at some later date depending on their function use over time.

While these adaptations allow the system to recommend particular strategies, interactions between a new strategy and previously applied strategies are still a major problem. Strategy interactions are a natural side effect of any financial planning performed using TIMS, and they reflect the conflicts that are inherent in the domain itself. The major advantage of employing strategies, the insulation of the user from low-level implementation details in the TIMS system, makes it difficult to foresee interactions. In practice, even experienced financial planners find it difficult to pin down elusive combinations.

The UMC helps to deal with this problem by checking for possible interactions that are recorded in the Application model (Figure 6) whenever a strategy is implemented. A Strategy Interaction dialog is presented only if the Application model contains interactions for the strategy in the context of the current plan. This dialog explains the represented interactions, and is simply informational – the user is not required to make any modifications to the situation. Like other adaptations, interaction detection can be deactivated and later reactivated if necessary.

3.4 Phase 3: Data Entry Assistance

The third phase of implemented adaptations is intended to address concerns with the initial data entry process. One common method of simplifying interactive programs is to have two sets of menus, one standard and one advanced. The use of two menu sets is employed in TIMS and does indeed make it easier for a novice to avoid the more complex dialogs that are not normally required for a simple financial plan. Once users find it necessary to use some of the more advanced features, they are normally more accustomed to the system and are not so overwhelmed by the additional functionality.

The data entry path in the original TIMS system also attempts to make the entry of client data as straightforward as possible. A logical flow from the client's demographic information to their incomes, assets, and liabilities is made explicit by providing Next and Previous Dialog buttons at the bottom of dialogs in this loop, in addition to the typical OK and Cancel buttons that terminate data entry (see Figure 9). By stepping from one dialog to the next, the user is assured that all of the pertinent data has been entered and that the client's financial situation is ready to be analyzed once the data entry loop is complete.

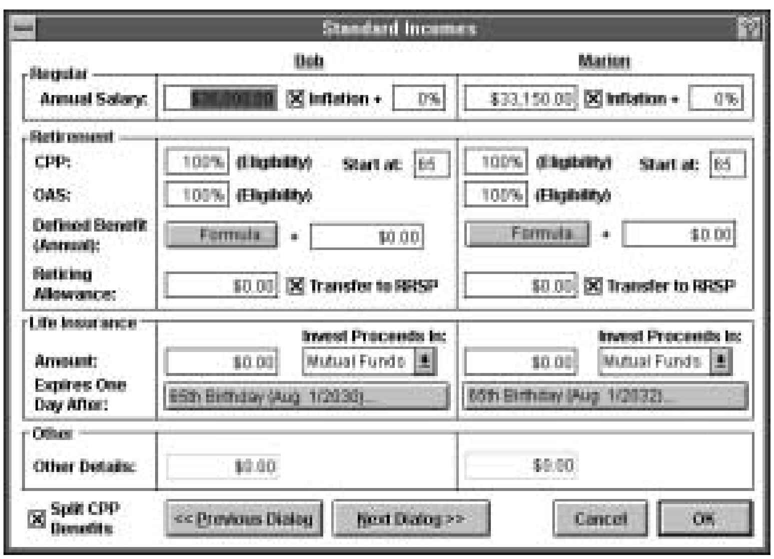

Figure 9. A typical dialog in the data entry loop.

Although the data entry support mechanisms that were in place for newer users were adequate and were at least on par with comparable commercial systems, the system's designers were open to suggestions for improvement. These suggestions occurred naturally as part of the system refinement process. Feedback from users progressed from bug identification and procedural concerns to an increased focus on usability issues. As a result of this feedback, two major modifications for new users have been introduced. The first identified concern was the early termination of the data entry loop by accidentally clicking on the OK button as opposed to the Next Dialog button. These concerns were addressed through the user modelling component by prompting the novice user and allowing him or her to safely return to the data entry loop if it was accidentally terminated. This warning was displayed only if the dialog was accessed as part of the initial data entry process, not if the dialog was accessed individually. The use of this warning dialog to reduce accidental exits is directly linked to the user model, so that it would not be displayed for experienced TIMS users.

The second difficulty that was consistently reported by novice users was uncertainty during navigation through the complex process of data entry, financial situation analysis, and the production of a final printed financial plan. TIMS uses a desktop metaphor which requires users to: enter data regarding clients; perform financial analysis by applying a variety of strategies (modifying client data and strategies appropriately); and produce a printed financial plan using document packages. The system was constructed to be as powerful and flexible as possible. Once data is entered, there is no single optimal solution. Compounding this lack of direction is the fact that financial planning is not an exact science: it relies on a large base of knowledge, heuristics, compromises, and individual preferences on the part of the advisor.

Supporting the novice user through this process is not an easy task. For example, once initial client data is entered, the system recommends the Strategy Assistant as the mechanism to enter the client's existing financial planning strategies. When the Strategy Assistant is closed, the data entry phase is complete, and a new user may be uncertain about the next logical step to perform in the system. This difficulty is dealt with by providing the novice user with ongoing information as to where he or she is in the overall process. Once the Strategy Assistant is closed in P-TIMS, the novice user is queried as to whether creation and analysis of a new plan is desired.

Generally, the user's final goal is the creation of a printed financial plan for a client. As mentioned in Section 2, TIMS uses Document Packages that accept data from the system and export it in the form of an RTF file. Each document package requires specific information in order to function. To support the user through the various steps that follow data entry, the UMC maintains the requirements of the various Document Packages explicitly. At each step, the UMC examines the currently active document package and the requirements that remain incomplete. A dialog is displayed, which asks the user if he or she wishes to follow what the UMC determines to be the next logical step. The user may then view a ScreenCam demonstration of the recommended action(s).

These additional demonstrations assist novice users by providing a direct path through the TIMS program from initial data entry through to the printing of the completed financial plan. They are especially helpful the first few times that the user enters data into the program. These adaptations are targeted directly at novice users. Experienced users will not receive any of the modifications unless they explicitly turn on the option themselves. In terms of user comments, this is by far the most useful adaptation.

3.5 Minimalism in the P-TIMS User Modelling Component

There are very clear advantages for the use of a simple user modelling architecture in complex systems like TIMS. Issues that are critical in large, complex user modelling systems can be minimized or ignored in a simple system. For example, there is no need for a conflict resolution scheme, which would be absolutely necessary in a more complex approach. Here, users are assigned a single stereotype that has been chosen from a very small set of potential candidates. This stereotype is not revoked if further information about the user is found that changes the system's perception of the user's skills. Individual parameters of the user model are simply updated as the user's interaction unfolds over time. The stereotype is used primarily as a starting point by providing an approximation of the user's expertise in different aspects of the system.

Similarly, neither sophisticated learning mechanisms nor uncertainty management schemes are necessary in an approach such as this. User models contain only a minimal set of user attributes, and each attribute is associated with an inherent risk that it may be inaccurate due to its generality. The system learns very basic information about users by monitoring their actions and gathering user information from the users themselves. This information is primarily composed of the tasks performed, the number of demonstrations viewed, and the user's direct modifications of the user model. As described earlier, the user model is refined on the basis of this interaction history and the length of time over which the system has been used.

The most serious drawback of a simple user modelling mechanism over a more complex one is the potential for misjudging users due to the generality of the attributes recorded. These types of errors have a minimal effect on an individual user in this system. Consider the two most extreme examples: incorrectly categorizing a complete novice (a new TIMS user with limited financial planning skills) as an expert user, and vice versa. In the former category, the user would experience system interactions that are similar to using a system with no user modelling component. Nothing would be gained through having an adaptive system, but nothing would be lost. As the user interacted with the system over time, the user complexity attribute in the user model would decrease and the necessary support would be available.

The second situation is somewhat more problematic. The user might take a week of system use to reach his or her true level, and during that time be irritated by extra prompts and reminders of known concepts. If it is clear to expert users that they have the ability to easily control the adaptations, it is reasonable to expect experts to access and change them for themselves. In the P-TIMS system, the user models are accessible, and they can be modified by users so that the impact of an initially inaccurate user model is low.

The accessibility of the user model to the user is an important component of this approach and of minimalist user modelling in general. Previous research on adaptive systems suggests that user control over the user modelling component is important for user acceptance (Wahlster and Kobsa 1989; Krogsæter and Thomas 1994; Krogsæter et al. 1994). Because users are often unfamiliar with adaptive behaviour in a system, care must be taken so that users do not feel monitored. Users must have the ability to inspect what the system knows about them (Cook and Kay 1994; Kay 1995). Because a user's confidence in a system that is taking actions for them changes significantly with the user's comfort level over time (Norman 1997), a significant level of user control must be provided. Oppermann (1994) proposes several methods of achieving user control, from direct control of the activation and deactivation of individual adaptations and the definition of specific parameters for adaptation, to offering adaptations in the form of proposals and informing the user of the effects of adaptation modifications.

P-TIMS attempts to maximize the feeling of user control using Oppermann's methods. Check boxes in the Preferences dialog (previously shown in Figure 7) provide a means of examining the adaptations that are currently available, and they can activate and deactivate individual adaptations. Adaptations that are offered in the form of a proposal have been implemented through dialog boxes that prompt the user before displaying demonstrations. In several places in the system, if the user is determined to have little or no experience with a concept or function, he or she is asked if a context-sensitive presentation about this feature is desired. Once a suggestion is rejected, the user is given the option to terminate further prompts. In addition, the Strategy Interactions dialog and the Strategy Recommendations dialog contain check boxes that allow the user to immediately terminate these two adaptations. Through these features, P-TIMS has provided users with direct control over whether they will accept adaptations at a particular time, and whether they will wish to continue to experience adaptations in the future.

P-TIMS also attempts to give the user information about the effects of the adaptation modification through the use of informational dialogs. For example, if the UMC has determined that a user is to be re-classified to a different user level, an appropriate dialog is posted with the suggested changes based on the user's new level. The user can refuse to be re-classified at this point, and he or she may choose to deactivate any further re-classifications. Users may, of course, re-classify themselves through the Preferences dialog at any time.

In a relatively simple manner, the P-TIMS system also allows users to have sole control over their user models. Each individual user model is stored as a file under the system directory. There is no attempt to share these files or extract information from them for any other purpose. The user could simply delete them if desired. The user would not be expected to feel threatened by the existence of a P-TIMS user model since there is little sensitive material in the models. There are no error rates, task completion times, or other types of information that could potentially be used by employers to monitor their employees.

4. Evaluation

The minimalist user modelling component and the adaptations that are described in Section 3 have proven to be very useful in practice. However, there is much more to the evaluation of a human-computer interface than such informal, subjective statements.

Indeed, the very question of how one can evaluate one interface and be able to say that it is “better” or “more easily understood” is still the subject of much debate. While there are many possible ways in which the quality of a user interface can be measured, Shneiderman (1992) lists five (non-independent) significant factors: (1) speed, the time taken to complete tasks; (2) accuracy, the number of errors made; (3) time to learn; (4) retention of acquired knowledge; and (5) user acceptance or satisfaction.

Of these, user satisfaction is one of the most neglected (Dietrich et al. 1993). The reason lies in the difficulty of measuring user satisfaction empirically, as opposed to the relative ease of measurement of the other factors. It is certainly not due to a lack of this factor’s recognized importance. Chin, Norman, and Shneiderman (1987) feel that subjective satisfaction often determines the ultimate acceptance of any system. Indeed, systems can be more than adequate on all of Shneiderman’s other measures, but are still doomed to failure if user satisfaction is not addressed (Shneiderman 1992). Shneiderman (1992) commented that, “We have come to expect that automobiles will have miles-per-gallon reports pasted to the window, appliances will have energy efficiency ratings, and textbooks will be given grade level designations; soon, we will expect software packages to show learning time estimates and user satisfaction indices from appropriate evaluation sources” (p. 495).

In fact, Shneiderman’s other factors may be dependent on user satisfaction as well. A large metastudy provided evidence that user satisfaction or preference has a strong positive association with users’ average task performance (Nielsen and Levy 1994). Nielsen and Levy (1994) also reported that when choosing between systems “...one has a reasonably large chance of success if one chooses between interfaces based solely on users’ opinions” (p. 75).

Because of the importance of user satisfaction and the desire for an empirical evaluation of the adaptations provided in P-TIMS, an experiment was initiated whose purpose was to investigate the impact of an adaptive user interface on user satisfaction levels. It was hypothesized that computer programs with user models produce increased levels of user satisfaction over programs with no such user models. More specifically, it was hypothesized that the TIMS program with a user model (P-TIMS) would produce increased levels of user satisfaction over a TIMS program with no user model. This section presents an overview of the methodologies employed and the results obtained in this study; further details are available in (Strachan 1999).

Participants. Forty-four participants were selected from a pool of 3,500 employees of a large Canadian financial services company that had adopted the TIMS program as their financial planning software. Ages ranged from 27–64, with a mean of 43.6 years. Five of the 44 participants were female and 39 were male. All of the participants had relatively high levels of education and moderate amounts of previous computer experience. Minimal previous experience with the TIMS system was reported by 32% of the participants, while the majority (68%) reported no previous experience. All participants were judged by the experimenters to be novice TIMS users.

Questionnaire. An instrument to measure user satisfaction was chosen on the basis of its established reliability and validity (Wong and Rengger 1990). The Questionnaire of User Interface Satisfaction (QUIS), developed by the Human-Computer Interaction Laboratory at the University of Maryland, was designed to evaluate a user's subjective satisfaction with the human-computer interface of an interactive computer system (Chin et al. 1988; see also Shneiderman, 1992, chap. 13). QUIS was also selected because it continues to be updated and refined as the focus of a long-term research project at the University of Maryland, and it has documented user interface evaluations in various industrial and academic environments. Participants were given a modified short version of the Questionnaire for User Interface Satisfaction (QUIS) v5. The twenty-one item questionnaire is arranged in a hierarchical format and contains: (1) a demographic questionnaire, (2) six scales that measure overall reaction ratings of the system, and (3) four measures of specific interface factors: screen factors, terminology and system feedback, learning factors, and system capabilities.

Procedure. Two programs were compared – the original TIMS system and an adaptive TIMS system (P-TIMS). The experiment took place over a period of four days. Participants were assigned to one of four groups. One TIMS group and one P-TIMS group started the first session of the experiment on Day 1, and the second TIMS and P-TIMS groups started on Day 2. Day 1 participants returned to complete the experiment on Day 3, while Day 2 participants returned on Day 4. The design produces a main effect for program, a main effect for day, and a program-day interaction.

The first session of the experiment for all of the participants in the experiment consisted of a half-day TIMS introductory training session. This training program is based on the TIMS Quick Start manual and provides a basic level knowledge of the TIMS program. At the completion of the training, users were expected to be able to do the following: enter client data, perform simple analyses of the client situation, and generate a client report.

Each participant returned 2 days later for the second half-day session in order to perform the second component of the experiment. Each individual was given a data worksheet containing sample client financial data and a set of instructions to assist them in performing certain tasks. The client data was fairly simple, and it was similar to the data that was entered during the TIMS training session. The worksheet detailed tasks that reflected the typical system usage that the participants needed to perform in order to re-familiarize themselves with the TIMS system. For the group with the adaptive systems, it gave them an opportunity to experience the customization of their system. Each participant filled out an anonymous questionnaire (QUIS) following the completion of the worksheet.

Results. A 2 × 2 between-subjects multivariate analysis of variance (MANOVA) was performed on six dependent variables: the overall impression of the system (*Impression*), the subjective satisfaction (*Satisfaction*), whether the system was dull or stimulating (*Stimulation*), ease of use (*Ease of Use*), whether the system was ineffective or powerful (*Power*), and whether the system was rigid or flexible (*Flexibility*). Independent variables were Program (TIMS and P-TIMS) and Day (Day 1 and Day 2). Table 2 shows the means and standard deviations for all four groups.

Table 2. Means (and standard deviations) for the six dependent variables broken down into four groups.

	TIMS		P-TIMS	
Dependent variable	Day 1[a]	Day 2[b]	Day 1[c]	Day 2[d]
Impression	7.3 (1.3)	6.9 (1.2)	7.0 (1.2)	6.6 (1.3)
Satisfaction	6.4 (2.1)	5.7 (1.1)	6.1 (1.8)	6.7 (1.0)
Stimulation	7.4 (1.4)	6.5 (1.2)	7.6 (1.0)	7.2 (1.2)
Ease of Use	6.2 (2.1)	4.8 (1.8)	4.9 (1.8)	5.2 (1.8)
Power	7.9 (0.8)	6.9 (1.1)	8.2 (0.9)	7.9 (1.0)
Flexibility	7.4 (1.3)	6.3 (1.3)	7.0 (1.3)	6.8 (1.1)

Note. Maximum score = 9. [a]$n = 11$. [b]$n = 11$. [c]$n = 10$. [d]$n = 12$.

SPSS* MANOVA was used for the analyses. All cases contained a complete set of data, resulting in a total *n* of 44. There were no univariate or multivariate within-cell outliers. The assumptions underlying MANOVA were tested and found to be satisfied.

Wilks' criterion for the combined DVs was significant for Program, $F(6, 35) = 2.73$, $p < .05$, but not for Day, $F(6, 35) = 1.73$, $p > .05$, or for the overall Program and Day interaction, $F(6, 35) = 0.99$, $p > .05$. Because the omnibus MANOVA shows a significant main effect for the Program variable, it is appropriate to further investigate the nature of the relationships among the IVs and DVs.

Univariate analyses were done for each variable by Program, Day, and the interaction between Program and Day. *Power* was the only significant variable ($p < .05$), both for Program and Day but not for the interaction effect.

Discussion. In this experiment, the questions to be answered were whether or not the means of the six overall reaction factors measured by QUIS, representing various aspects of subjective user satisfaction differed as a function of the program used and the day on which the experiment was performed. Are there differences in the way that the user perceives each system in aspects such as overall user satisfaction, ease of use, power or flexibility? Also, are there differences between groups that performed the experiment on different days?

An overall positive effect on user satisfaction levels was indeed observed as a result of the adaptations outlined in Section 3. In terms of the individual measures, four of the six user satisfaction factors were rated higher by the P-TIMS groups versus the TIMS groups: *Satisfaction*, *Stimulation*, *Power*, and *Flexibility*. P-TIMS users rated the factor of *Power* significantly higher than the TIMS users. This was the only factor that showed a significant difference with a confidence level of 95%. Based on the informal comments by the participants in the experiment, the *Power* factor represented the impression that the system was more powerful because it could perform tasks for them (it was more helpful) and that it made it easier for them to see the capabilities of the system. While it would be more gratifying for the system with the user modelling component to show a higher rating for each of the six factors, the amount of adaptation was minimal and therefore, a smaller effect was expected.

This small effect was expected by the researchers as a by-product of the commercial setting in which this experimentation was performed. One of the foremost goals of empirical research is the exercise of control over all aspects of the setting. However, if complete control were to be imposed, the experimental situation would no longer even remotely resemble a realistic commercial setting, nor would results from such a scenario be transferable to this type of setting. This phenomenon has been studied extensively by ecological psychologists (e.g. Barker et al., 1978; Barker 1968). In a realistic commercial setting, however, even elements such as the perfectly random selection of subjects are extremely difficult to achieve. The choice of performing evaluation in such an environment is often an unavoidable one when working with a commercial product. In work such as that presented here, integration of research with both commercial development of the software product itself and commercial use of that software is vital. In this particular case it was simply impossible to take random users away from their daily employment activities purely for the purpose of engaging in the study itself. Like many aspects of research in commercial settings, it was necessary to balance research needs with the needs of commercial partners, and perform the study through the presentation of TIMS training. This provided on one hand a very realistic setting for the subjects' experience with both systems. It also, however, provided possible interactions that must be considered when interpreting the results above.

In our case, it was clear during the study (and in the results of Table 2) that the first TIMS group was overly enthusiastic about the original non-adaptive TIMS system. In this particular group, it is likely that the results are skewed positively toward the original TIMS system (and thus illustrate an artificially small impact on the part of P-TIMS) due to the comments on day 1 of an individual participant who was particularly enthusiastic about the ease of use of the original TIMS system. Many of the participants in the group were this individual's subordinates, and the results in this particular group indicate greater initial enthusiasm than the other groups for the non-adaptive system (overall average of 7.1 versus 6.2). On the other hand, the two P-TIMS groups are remarkably similar (overall averages of 6.8 and 6.7). In particular, the *Ease of Use* factor is much higher in the group in question than the other three groups. In this case, if the first group's data is excluded then P-TIMS performs even better by comparison. Five of the factors are rated higher (*Satisfaction*, *Stimulation*, *Ease of Use*, *Power*, and *Flexibility*) and *Stimulation* is very close to being a statistically significant difference ($p < .06$). While these observations are not empirically based, they indicate the possibility of even more significant effects of minimalist user modelling than were observed here, and lay the groundwork for a future study (see Section 5.2).

5. Conclusions

This research has resulted in a number of different findings. These findings are presented in two subsections. The first subsection presents some important practical considerations when performing applied research in the commercial arena. Like other papers in this volume [Corbett et al., 2000; Linton et al., 2000] we view these insights as some of the most important contributions of this work from the standpoint of deploying user models and performing user modeling research in the real world. The second subsection presents some general conclusions about the focus of the adaptations, the empirical testing component of the research, and the directions that future work will follow.

5.1 Issues in User Modelling in Commercial Settings

There are many practical considerations that must be made when performing applied research in the commercial arena. Research prototypes need not be as concerned with certain issues to the degree necessitated in a commercial system. Desmarais (1997) has emphasized several lessons learned in performing joint research with commercial partners, including close contacts, good prototypes, and sufficient funding. This work certainly reinforces these needs. The most critical issues that arose during the course of this work were the following:

- *User involvement.* The adaptations that are chosen for a commercial system must reflect concerns that real users have with an existing system, and be discovered by studying real users' needs and problems. It is easy to make these types of decisions in isolation and to judge proposed changes on the basis of one's intuition. It takes much more time and effort to actually observe users in realistic situations and to take note of their questions and concerns. In this research, formal one-day TIMS system training programs presented excellent opportunities to monitor hundreds of first-time users of the system as they became acquainted with it.

- *Quality of Adaptations.* There is an expectation of very high adaptation quality when experimental testing is performed on the actual users of a commercial system. Normally, there is much more latitude when it is understood that the system being tested is a prototype or a mock-up of anticipated changes. If users believe that the experimental system that they are being shown will be the actual system in place at their desks, then they will expect a higher level of refinement. For this reason, the adaptive components in P-TIMS were limited to certain aspects of the system in order to bring them up to production level quality in a reasonable amount of time. The adaptations must appear seamless, and must match the original system in look and feel so that it is not obvious that modifications have been made.

- *Scheduling.* Scheduling users to perform the testing and feedback on the adaptive system created many problems and is likely to do so in any commercial environment. Final testing does not only depend upon a completed system. Involving users in the construction and testing of a software system involves anticipating any cycles of the domain in question. For the financial planning domain there is a reasonably predictable annual cycle; January and February are very busy months for Canadian financial planners due to tax filing deadlines and other deadlines in Canadian financial legislation (e.g. Registered Retirement Savings Plans). Corporate training is also less likely to be done in December, July, or August because these are traditionally the months that are popular for employee holidays. As P-TIMS was to undergo empirical testing in a large Canadian financial service organization, some coordination with the TIMS program launch training programs were necessary.

Scheduling is also an important issue on a more finely-grained temporal scale. Existing corporate training schedules are difficult to adapt to because a researcher has little control over the dates, times and the number of users to be trained in any one session. This was an issue in this research because of the training component of the experiment necessary to get the participants up to a minimum level of proficiency with the system.

- *Management Support.* Upper-level management of any company that is involved in testing must be convinced that their employees should be involved in the experiment. The researcher must address concerns about the employees' productivity and ensure that the testing of the system will not have any adverse effect on employees' normal duties.

- *Corporate Benefits.* Corporations reasonably expect that they will receive at least some tangible benefits in return for their cooperation. In our own experience, one large financial service corporation's participation in the experimental testing of P-TIMS was possible because they were interested in seeing if the proposed adaptations would be helpful to their employees. The TIMS program had recently been adopted as the financial planning software for their representatives across Canada, and the company's support was achieved by emphasizing a beneficial consequence of the experimental process – introductory TIMS training. Introductory training was provided to all participants to ensure that new users were trained to a minimum criterion before being asked to compare the adaptive and non-adaptive systems. This hands-on training is very expensive for geographically-distributed corporations, and through participation in the experiment described in this paper they obtained this training at a minimal cost.

  A second benefit for both corporation and developer is feedback from employees about a software program. Every participant in the experiment described in Section 4 filled out an anonymous questionnaire and was encouraged to make broad comments on various aspects of the system. Employees who do not fear reprisal can be expected to be more candid and the result is higher quality feedback. Both the results of the statistical analysis and the anonymous compilation of anecdotal comments were released to the corporation. If continuing evolvement and corporate-level customization of a software package is expected, as it is in this work, then this information is invaluable for future development decisions.

- *Potential for Negative Transfer.* In a corporate environment, the researcher must also be extremely careful to avoid sending users back to their normal working conditions with different training or system expectations (Meyer 1994). This potential negative transfer is a unique concern in the commercial environment. It was not a trivial task to coordinate the parallel development of the system containing the user modelling component (P-TIMS) with the latest release of the TIMS program. Because TIMS development is ongoing, there have been a number of new product releases throughout the period of this research. Some of the releases were primarily changes made to fix existing problems, but several major releases added extensive new functionality to the system.

  When attempting to compare an adaptive and a non-adaptive system, the adaptive system must be synchronized with the latest release in order for any test of preference to be valid. There are two reasons for this: to ensure that the differences that are detected are based solely on adaptations, and because one cannot train new users on an older version of the software and return those users to their working environment with a newer version. In order to address these concerns, the P-TIMS system was upgraded to the most recent TIMS release for testing. The development environment and existing internal company procedures assisted in this process by tracking the sections of code that had been modified between releases. Synchronizing the adaptive system without this type of information would be overwhelming on any non-trivial system.

- *Integration of Code.* Integrating a user modelling component with existing code can also pose challenges. A problem, which is familiar to all maintenance programmers, is the difficulty of fully understanding the coding of other programmers. The TIMS system is large and complex; programmers require several months to become familiar enough with the structure of the system and the development environment to become productive. Ideally, if problems or questions about the primary system arise, the original developers should be consulted in order to avoid wasting large blocks of time. This requires taking these developers away from their own work to assist the researcher. This is a more difficult proposition in a commercial environment than it might be in a research environment, because a for-profit organization’s goal is to get a product to market as quickly as possible.

In general, any developer of an adaptive component for a commercial product will be subject to constraints that do not normally exist in a research environment.

5.2 General Insights and Future Work

As mentioned at the beginning of this section, all of the adaptations described here have been targeted at novice users. In practice, however, it was found that some of the adaptations have proven to be more useful to experienced users of the system. The best example of this is the highlighting of the interactions between strategies. While novice users initially require more assistance with understanding interactions, the extra information seems to confuse them. It is simply a case of information overload (another major motivation behind user modelling systems, and one explored elsewhere in this special issue [Billsus and Pazzani, 2000]). On the other hand, experienced users find interaction information helpful because it answers some of their questions and concerns by exposing some of the underlying processing in the system.

The empirical component of this work has also re-emphasized the importance of the user in all aspects of the system development life cycle. Clearly, the assumptions made in the development environment need to be confirmed with actual users in their work environments. Introducing formal statistical analysis to this procedure was interesting because it caused the team to be more cautious about over-interpreting positive results that may have occurred by chance. At the same time, the informal results that were gathered as anecdotal stories and comments helped us to gain a better understanding of the users and their environment.

Some unanticipated problems occurred in the empirical testing in addition to the control issues discussed in Section 4. The overall level of computer familiarization of the users was expected to be higher than was actually experienced. As a result, the reactions to P-TIMS were very much also reactions to TIMS as well. During the experiment, users barely had time to digest the unmodified TIMS system and then they were asked to judge the adaptations. It is almost certain that some confusion would have resulted from this short time exposure, and would anticipate the differences in quantitative scores between TIMS and P-TIMS to be wider given a longer exposure time. Further, many users were unfamiliar with Windows, their new hardware, and the new software including Microsoft Office and TIMS. In this context, it is not at all surprising that the ease of use reaction parameter was not high.

The information gathered as a result of this work was always intended to direct changes to the production version of TIMS. Our experiences certainly are similar to that of a large adaptive user interface project (Totterdell 1990) where investigators found that the adaptations were not simply an extra facility that could differentiate between the system and its competitors but were much more a part of the design process. The three phases of adaptations all proved to be moderately successful, and modifications that were based on them have been implemented. The modifications that were felt to be most helpful to the novice user were the animated demonstrations and the prompting of the user to the next logical system function. The current release of the ScreenCam program was determined to be inadequate for the purposes of this work and did not provide the flexibility that was required. This triggered the development of a macro-language demonstration component that is available on the desktop at all times. This component, called *Hint*, was implemented in a TIMS-Lite version of the system[6], and actually performs the function while the demonstration is taking place so that the user can choose to accept the changes when the demonstration is complete. Demonstrations are thus dependent on the current context of the system. Three options are available for all the entries: "Explain It", "Show Me", and "Do It". Demonstrations are currently available for various planning and analysis functions (e.g. How to create a new client); reporting functions (e.g. How to create a client report); and general concepts (e.g. Setting up an education plan).

Deciding to spend the effort to "reinvent the wheel" when there is an existing product that could provide the functionality is a difficult decision. Integrating TIMS with the ScreenCam program was a reasonable decision for this research project but it quickly became apparent that a better solution was needed for a marketable application. A similar situation occurred with the exporting procedure in the original system. As discussed earlier, the client RTF file was exported to the Microsoft Word program as a customized client report where it could be edited and printed. TIMS launched the MS Word program directly as part of the export process. Users were observed to have some interesting problems with the integration of the two products. MS Word is a extremely powerful, complex application in its own right; it was not clear to the users in this study where the boundary was between the two systems. This contributed to a perception that TIMS was even more complicated.

It is not safe to assume that users have the necessary expertise in the integrated program, or that they will even understand that they are using a second program. These difficulties, among others, were also reported during the testing of three different applications that extended their functionality by launching Microsoft Excel (Schroeder 1996). They commented that, "Complementing your application with another feature-rich application is like getting a drink of water from a fire hydrant". The TIMS product now includes a simple component that allows users to view the client report in the TIMS program as well as in-place editing of reports. Novice users and users who do not wish to modify the report can print the report easily. The capability to export and edit the RTF file remains but can remain hidden until this functionality is necessary or desirable.

There are two significant elements of future work arising from this project. The first concerns expanding the adaptations to include those specifically designed for different user types and groups. Examples of these include restrictions of system functionality and simplifications to the system. The system would only perform the functions that are necessary for the accomplishment of the user's tasks. Experienced and intermediate users who have had some experience using the TIMS system will also be solicited for suggestions for these types of users.

The other major area of future work concerns the evaluation of the minimalist approach in this setting. A longitudinal study which tracked users of comparable versions of TIMS and P-TIMS over a longer period of time, perhaps several months, would suggest results that would be more significant and perhaps more meaningful. This would also allow the

[6] TIMS-Lite is a reduced-functionality system that was designed specifically for ease-of-use.

user model to make substantive changes, which is difficult to do over the course of a three hour session. While the user model can be expected to evolve to more closely match the user's capabilities and preferences, this will obviously take some time. It would be interesting to examine the user models to determine which attributes had been turned off explicitly, the supports that the system had terminated, and the state of the system that the user had gradually evolved into. It would also be interesting to see if different users had chosen to retain some of the adaptations and, if so, why this was done.

Longer-term experiments are lacking in the user modelling literature in part because of the difficulty and expense in conducting such research. This work has demonstrated the difficulty in controlling the environment in a two-day session in a commercial environment, even after several pilot studies. Increasing the monitoring of the adaptive and non-adaptive systems over time would introduce many extra variables. Each user would have completely different experiences with the system, using it for different tasks and over different periods of time. Formal analysis of this data would prove nearly impossible. Nonetheless, any information gathered would still be very valuable, particularly interviews with users.

Minimalist user modelling components appear to show much promise. This experience with a pragmatic user modelling component in a commercial system has shown that the time and expense of including an adaptive user interface did improve the user satisfaction with the overall system, although the difference was not very dramatic. The user modelling component had virtually no impact on the system response time. The overall impression left by this work is that the concept was sound but better decisions about the actual modifications to the system were needed. User models appear destined to be perceived as just one more knowledge source in future knowledge-based systems. Certainly, more empirical work is necessary because of items that were not clear in the experiment.

McTear (1993) emphasized two increasingly important areas for user modelling research. Firstly, that there is a need to recognize that some of the more theoretically motivated approaches may not be practical in the real world, and, secondly, that much more empirical work needs to be done. This work fits into both of these niches.

Acknowledgments

The authors wish to thank the EISI team who developed the original TIMS system and provided support for this research: Geoff Crysdale, Darren Miller, Paul Thiessen and Gerald Adams.

Bibliography

Barker, R.: 1968, *Ecological Psychology: Concepts and Methods for Studying the Environment of Human Behaviour,* Stanford, CA: Stanford University Press. 242 pp.

Barker, R., and Associates: 1978, *Habitats, Environments, and Human Behaviour*, San Francisco: Jossey-Bass Publishers. 327 pp.

Billsus, D., and M. J. Pazzani: 2000, "User Modelling for Adaptive News Access", *User Modelling and User-Adapted Interaction*, special issue on Deployed User Modelling Systems.

Chin, D. N.: 1989, "KNOME: Modeling what the User Knows in UC", in Kobsa, A., and W. Wahlster (Eds.), *User Models in Dialog Systems* (Springer-Verlag: Berlin), pp. 74-197.

Chin, J. P., Diehl, V., and Norman, K.: 1988, "Development of an instrument measuring user satisfaction of the human-computer interface." *CHI88 Conference Proceedings: Human Factors in Computing Systems*, New York: Association for Computing Machinery, pp. 213-218.

Chin, J. P., Norman, K., and Shneiderman, B.: 1987, *Subjective User Evaluation of CF Pascal Programming Tools*, Technical Report CAR-TR-304, Center for Automation Research, University of Maryland, College Park.

Cook, R. and Kay, J.: 1994, "The justified user model: a viewable, explained user model." *Proceedings UM'94, Fourth Int. Conference on User Modeling*, Hyannis, MA, pp. 145-150.

Corbett, A., M. McLaughlin, and K. C. Scarpinetto: 2000, "Modelling and Student Knowledge: Cognitive Tutors in High School and College", *User Modelling and User-Adapted Interaction*, special issue on Deployed User Modelling Systems.

Desmarais, M.: 1997, "Real world Applications of User Models", *Proceedings of the Workshop on User Models in the Real World*, 6th International Conference on User Modelling (UM97), Sardinia, Italy, pp. 3-6.

Dieterich, H., Malinowski, U., Kuhme, T., and Schneider-Hufschmidt, M.: 1993, "State of the Art in Adaptive User Interfaces", in Schneider-Hufschmidt, M., Kuhme, T., and Malinowski, U. (Eds.), *Adaptive User Interfaces: Principles and Practice.* Amsterdam: Elsevier Science Publishers B.V. pp. 13-48.

Erickson, T.: 1997, "Designing Agents as if People Mattered", in Bradshaw, J. (Ed.), *Software Agents*, Cambridge: MIT Press, pp. 79-96.

Fink, J., and A. Kobsa: 2000, "Commercial User Modeling Servers", *User Modelling and User-Adapted Interaction*, special issue on Deployed User Modelling Systems.

Firefly: 1998, http://www.firefly.com [1998, April 10]

Fossay, C., and Sawchuk, J.: 1994, Marketing Strategy for TIMS©: The Tax and Investment Management System, KPMG Peat Marwick Thorne, Winnipeg, Manitoba Canada.

IUI: 1999, *1999 International Conference on Intelligent User Interfaces*, Redondo Beach, Los Angeles, CA, Available: http://sigart.acm.org/iui99 [1999, October 10]

Jameson, A., C. Paris, & C. Tasso:, 1997, Preface. In Jameson, A., C. Paris, & C. Tasso (Eds.), *User Modeling: Proceedings of the Sixth International Conference UM'97*. New York: Springer-Verlag Wien.

Kay, J.: 1995, "Vive la difference! Individualised Interaction with Users", Proceedings of IJCAI-95, Montreal, Canada, pp. 978-984.

Kay, J., and G. Fisher: 1997, "Introduction". *Proceedings of the Workshop on User Models in the Real World*, 6th International Conference on User Modelling (UM97), Sardinia, Italy, pp. 1-2.

Langley, P.: 1999, "User Modeling in Adaptive Interfaces", *Proceedings of the 7th International Conference on User Modelling* (UM99), Banff, AB., pp. 357-369.

Kobsa, A.: 1993, "User Modeling: Recent Work, Prospects and Hazards", in Schneider-Hufschmidt, M., Kuhme, T., and Malinowski, U. (Eds.), *Adaptive User Interfaces: Principles and Practice.* Amsterdam: Elsevier Science Publishers B.V. pp. 111-128.

Krogsæter, M., Oppermann, R., and Thomas, C.: 1994, "A User Interface Integrating Adaptability and Adaptivity", in Oppermann, R. (Ed.), *Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software*. Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 97-125.

Krogsæter, M., and Thomas, C.: 1994, "Adaptivity: System-Initiated Individualization", in Oppermann, R. (Ed.), *Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software*. Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 67-96.

Kuhme, T., and Schneider-Hufschmidt, M.: 1993, "Introduction", in Schneider-Hufschmidt, M., Kuhme, T., Malinowski, U. (Eds.), *Adaptive User Interfaces: Principles and Practice.* Amsterdam: Elsevier Science Publishers B.V. pp. 1-9.

Linton, F., D. Joy, H. P. Schaefer, and D. Hersey: 2000, "Recommender Systems for Learning: Building User and Expert Models through Long-Term Observation of Application Use", *User Modelling and User-Adapted Interaction*, special issue on Deployed User Modelling Systems.

Malinowski, U.: 1996, "ABIS-95: GI Workshop on Adaptivity and User Modeling in Interactive Software Systems", *User Modeling and User-Adapted Interaction* **6**(1), 69-76.

Maskery, H. S.: 1984, "Adaptive interfaces for naive users–an experimental study", in Shackel, B. (Ed.) *INTERACT "84*. Elsevier-North Holland, Amsterdam.

McCalla, G., Searwar, F., Thomson, J., Collins, J., Sun, Y., and Zhou, B.: 1996, "Analogical User Modelling: A Case Study in Individualized Information Filtering". *Proceedings of the 4th International Conference on User Modeling*, Hyannis, MA, pp. 13-20.

McTear, M.: 1993, "User Modelling for Adaptive Computer Systems: A Survey of Recent Developments", *Artificial Intelligence Review* 7, pp. 157-184.

McTear, M.: 1996, Introduction. *Proceedings of the Workshop on the Commercial Potential of User Modeling*, 5th International Conference on User Modelling (UM96), Kona, HI, p. 1.

Meyer, B.: 1994, "Adaptive Performance Support: User Acceptance of a Self-Adapting System". *Proceedings of the 4th International Conference on User Modeling*, Hyannis, MA, pp. 65-70.

Microsoft.: 1998, "It's Your Microsoft Office 97" [Online]. Available: http://www.microsoft.com/office/office97/articles/off97_1.html [1998, April 10]

My Yahoo!: 1998, Available: http://my.yahoo.com [1999, October 10]

Nielsen, J., and Levy, J.: 1994, "Measuring Usability: Preference vs. Performance", *Communications of the ACM* Vol. 37, No.4, pp. 69-75.

Norman, D.: 1997, "How People Might Interact with Agents", in Bradshaw, J. (Ed.), *Software Agents*, Cambridge: MIT Press, pp. 49-55.

Oppermann, R.: 1994, "Introduction", in Oppermann, R. (Ed.), *Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software,* Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 3-12.

Schroeder, W.: 1996, "Integrating Other Applications", *Eye for Design*, North Andover, MA: User Interface Engineering.

Shneiderman, B.: 1997; "Direct Manipulation Versus Agents", in Bradshaw, J. (Ed.), *Software Agents*, Cambridge: MIT Press, pp. 97-106.

Shneiderman, B.: 1992, "Designing the User Interface: Strategies for Effective Human-Computer Interaction". Reading, MA: Addison-Wesley, 2nd edition.

Strachan, L.: 1999, *Pragmatic User Modelling for Complex Knowledge-Based Systems*, Ph.D. Dissertation, Department of Computer Science, University of Manitoba, 226 pp.

Totterdell, P.: 1990, "Introduction", in Browne, D., Totterdell, P., and Norman, M. (Eds.), *Adaptive User Interfaces*, San Diego, CA: Academic Press Inc, pp. 1-14.

Vassileva, J.: 1994, “A Practical Architecture for User Modeling in a Hypermedia-Based Information System”, *Proceedings of the 4th International Conference on User Modelling*, Hyannis, MA., pp. 115-126.

Vassileva, J.: 1996, “User Modelling in a Hypermedia Office Documentation System”, *User Modelling and User-Adapted Interaction* 6(2-3), 185-223.

Wahlster, W., and Kobsa, A.: 1989, “User Models in Dialog Systems”, in Kobsa, A. and Wahlster, W. (Eds.), *User Models in Dialog Systems*, New York: Springer-Verlag, pp. 4-34.

Weber, G., and Specht, M.: 1997, “User modeling and adaptive navigation support in WWW-based tutoring systems”, *Proceedings of the 6th International Conference on User Modelling (UM97)*, Sardinia, Italy, pp. 289-300.

Wong, G., and Rengger, R.: 1990, *The validity of questionnaires designed to measure user-satisfaction of computer systems*. National Physical Laboratory Report DITC 169/90, Teddington, Middlesex, UK.

Authors' Vitae

(1) Dr. L. Strachan:

Emerging Information Systems Inc., 900 - 125 Garry Street Winnipeg, MB, Canada R3C 3P2

Linda Strachan is currently the Manager of Training at Emerging Information Systems Inc., continuing her work in the area of adaptive user interfaces. Prior to this, she was a Lecturer in the Department of Computer Science at the University of Manitoba. She completed her Ph.D. in 1999 at the University of Manitoba, specializing in user modelling and adaptive software. Her primary interests lie in the areas of knowledge-based financial planning systems and user modelling.

(2) Dr. J. Anderson:

Dept. of Computer Science, University of Manitoba, Winnipeg, Manitoba, Canada, R3T 2N2

John Anderson is currently an Assistant Professor in the Department of Computer Science at the University of Manitoba and runs the Autonomous Agents Laboratory in the department. He completed his Ph.D. in Computer Science at the University of Manitoba in 1995, specializing in real-time planning in complex domains. His current research interests include software agents, multiagent and multirobot systems, simulation systems for designing and evaluating intelligent agents, and adaptive software systems.

(3) Mr. M. Sneesby:

Emerging Information Systems Inc., 900 - 125 Garry Street Winnipeg, MB, Canada R3C 3P2

Murray Sneesby is a Senior Software Developer at Emerging Information Systems Inc., and holds a Bachelor's degree in Computer Science from the University of Manitoba. He conducted research in the field of user modeling while enrolled in the M. Sc. program in Computer Science at the University of Manitoba, under the supervision of Dr. Mark Evans.

(4) Dr. M. Evans:

Emerging Information Systems Inc., 900 - 125 Garry Street Winnipeg, MB, Canada R3C 3P2

Mark Evans is the President of Emerging Information Systems Inc. Prior to this, he was an Associate Professor in the Department of Computer Science at the University of Manitoba. He completed his Ph.D. in Computer Science at the University of Manitoba in 1988, specializing in distributed knowledge-based systems. His primary area of research is knowledge-based systems, and he has developed numerous applications employing this technology. He is also interested in the integration of expert systems with conventional environments, and in user interface technology for expert systems.