Proceedings of the Sixth International Conference on User Modelling, Sardinia, Italy, 1997, pp. 189-200

Pragmatic User Modelling in a Commercial Software System

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

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

Abstract: While user modelling has become a mature field with demonstrable research systems of great power, comparatively little progress has been made in the development of user modelling components for commercial software systems. The development of minimalist user modelling components, 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 empirical evaluation of a minimalist user modelling component for TIMS, a complex commercial software system for financial management. The experimental results demonstrate that a minimalist user modelling component does improve the subjective measure of user satisfaction. Important issues and considerations for the development of user modelling components for commercial software systems are also discussed.

1 Introduction: Pragmatic User Modelling

Modern software systems are complex, and they often support a wide variety of tasks and diverse groups of users with differing problems and needs. This 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.

User modelling as a field has become more mature in recent years, to the point where research systems show great promise and demonstrate the feasibility of user modelling techniques such as stereotyping and the separation of user knowledge into a user model database (see Kobsa, 1993, and McTear, 1993 for overviews). However, it is also being recognized that in spite of the demonstrated capabilities of many research systems, comparatively little progress has been made in taking advantage of individualized interactions in commercial software systems. This lack of progress, along with a lack of emphasis on empirical studies, are considered to be two of the most important concerns in modern user modelling research (McTear, 1993).

Part of the reason for lack of commercial deployment of user modelling systems 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 included. Also important are the time and expense that are involved in including a user modelling component, in comparison to the often unproven advantages which the added expense may bring. Other less obvious factors include the design

[*] Special thanks to the EISI team who developed the original TIMS system and provided support for this research.

changes that may be necessary for the inclusion of a user modelling component and the very real 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, Oppermann, 1994, McCalla et al., 1996). The identification of this shortcoming has prompted researchers to create pragmatic user modelling architectures and techniques, 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.

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 that must be addressed in supporting user modelling in ongoing commercial software development.

This paper describes the design, implementation, and empirical evaluation of a pragmatic user modelling component for the Tax and Investment Management Strategizer, or TIMS, a complex commercial financial planning software system. The addition of this component is intended to address the needs of novice users while minimizing the impact on expert users. The paper begins with an analysis of the domain, a description of the functionality of the user modelling component, followed by the description and results of an empirical study designed to identify the impact of this component on user satisfaction levels.

2 Tax and Investment Management Strategizer (TIMS)

The Tax and Investment Management Strategizer (TIMS) is a sophisticated commercial software system for financial planning that was developed by Emerging Information Systems, Inc. (EISI) of Winnipeg, Canada. 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 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 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 *situations*, that represent the financial situation of a client at a particular time. A TIMS user creates a situation through a series of data entry dialogs, and the system creates a graphical display of the client's financial situation in a window known as the Situation Window. The financial planner can then implement and evaluate numerous financial planning strategies to produce alternative situations.

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 to properly reflect their overall financial picture. 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, 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. buying, selling) that will then be automatically carried out over a specific period of time, or upon a specific date or event. The financial planner can also insert the appropriate transactions at a more detailed level of input. Strategies serve to insulate the novice. By invoking the Strategy at a high level, novice users need not be exposed to an excessive level of detail.

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, allowing the automatic analysis of situations and the generation of recommendations for system tasks. For example, novice TIMS users are encouraged to use the *Planning Assistant* for the analysis of tentative strategies and for final plan creation. The Planning Assistant analyzes aspects of each situation and displays the analysis and the customized recommendations in a window. The user can implement recommended Strategies in the Planning Assistant, and also has access to a menu of Data Entry dialogs to directly make changes to the client's data, Strategies used, and other Assistants provided by TIMS. 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 and allows the financial planner to make changes to the client's current situation and view the results, while the Strategy Assistant is designed to help the user easily select and apply multiple financial planning Strategies. 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 was too intrusive.

Like other 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, however, 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 in gaps in their own knowledge. Even among expert financial planners, the extent of their knowledge of the domain will vary from area to area. For example, an expert in life insurance might have little or no knowledge of leveraged investments.

A study conducted on the first commercial version of TIMS, released in 1995, validated 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 begin developing 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 to the system the benefits of user-adapted interaction with a minimum of cost and commercial disruption.

3 Supporting User-Adapted Interaction in TIMS

In the initial stages of developing a user modelling component for TIMS, it was decided to focus on the necessary adaptations for novice users. This was primarily because TIMS was originally developed in collaboration with high-end expert financial planners and had already been structured to reflect the organization of material that was helpful to them. A focus on novices, however, does not eliminate expert users from consideration. The usability of the software must not be limited by the adaptations that are necessary to support novices.

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. These adaptations were implemented in several stages and resulted in the development of Personalized-TIMS (P-TIMS), an adaptive version of the TIMS system. The foundation of P-TIMS is the user modelling component illustrated in Figure 1.

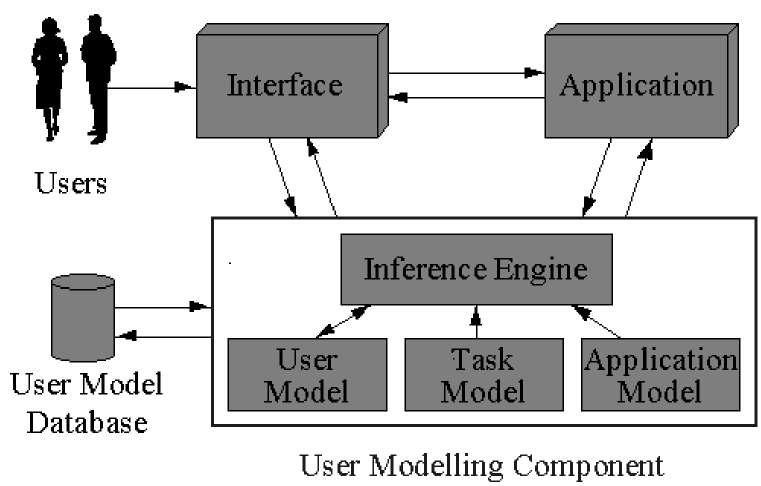

Figure 1. The user modelling component architecture of Personalized-TIMS.

The architecture consists of a long-term user model (also 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 on Strategies or combinations of Strategies that are equivalent to one another). 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.

Each P-TIMS user has their own unique user model, which is initially based on a set of stereotypes triggered by their job title (chosen from a menu of generic job titles) and a self-assessment of Windows experience and previous TIMS experience. Each of the two experience levels are divided into Novice, Intermediate, and Expert categories. Because of the two-dimensional diversity of TIMS users described in Section 2, 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. The combination of the appropriate Financial Planning and TIMS experience stereotypes result in a basic user model containing settings specific to the various adaptations that are supported by the user modelling component. The user model also 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 Strategies and Assistant demonstrations that have been previously viewed).

In an effort to make the user model visible and modifiable by the user, the user model attributes are accessible to the user through the Preferences dialog shown in Figure 2. 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. Other modifications occur when the tasks represented in the task model are used. Reclassification of a user in the initialization phase, however, 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 3).

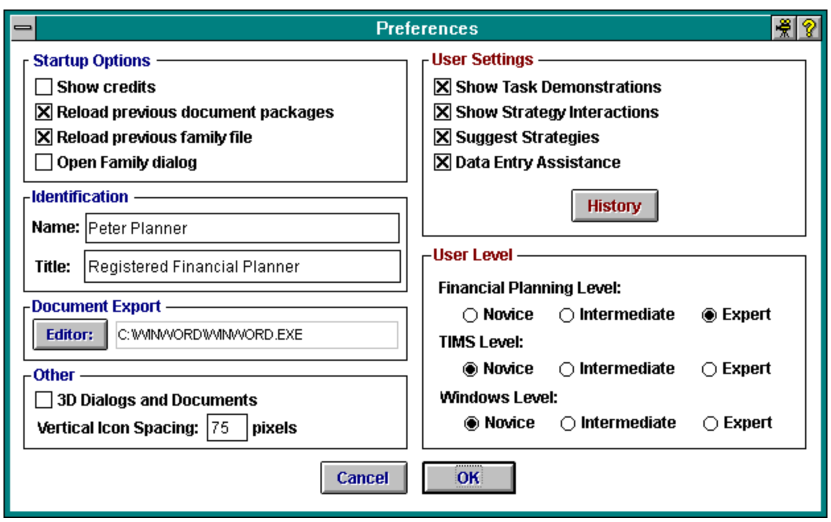

Figure 2. Preferences dialog for Personalized-TIMS.

```
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 (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 > 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 3. The basic algorithm for update of the user model in the initialization phase.

Over time, the adaptive system gradually diminishes into the original, non-adaptive system. 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. Once the 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. Different types of adaptations in the system 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.

3.1 Adaptations supported by the User Modelling Component

The first major phase of adaptations involved adding support for animated Lotus ScreenCam™ demonstrations of the general structure of the system, its components, and the various tasks the user can perform. These demonstrations (generally less than 2 minutes in length[1]) include explanations of the Preferences dialog for modifying the user model, assist in data entry, and demonstrate each of the Assistants and Strategies. After viewing any demonstration, the user is able to replay the demonstration through a movie camera icon in the dialog title bar (see the top right corner of Figure 2). The user is also instructed as to alternate methods of accessing the demonstration in future.

Like the other adaptations to be described, the demonstration facility is directly controlled by the User Modelling Component (UMC) described previously. When the user attempts to implement a TIMS Strategy or Assistant, the UMC is called to determine if a demonstration is to be shown to the user. The UMC determines if the demonstrations have been deactivated (by the user directly, indirectly by refusing further demonstrations, or by having an expert TIMS

[1] Pilot testing revealed that the maximum user attention span for the demonstrations was about 2 minutes.

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 demonstrations. This deactivates the Show Task Demonstrations checkbox in the Preferences dialog, but a user is free to reactivate it again if desired (see Figure 2). These adaptations benefit the novice user most during the first few sessions, and conversely, require the greatest amount of input from the user in the first few sessions. The ultimate ability of the user to refuse further demonstrations at any time minimizes the level of intrusion of this series of adaptations.

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. The second set of implemented adaptations were designed to recommend simpler alternatives to complex actions in the system, as well as to point out interactions.

These adaptations rely heavily on the task and application models maintained in the UMC. All of the Strategies and the more difficult system functions are represented in the task model and include a numeric complexity rating. A complexity rating is also maintained as part of the user model (initially set by the triggered financial planning stereotype), 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), and will be checked for equivalent Strategies if the users' rating is lower than the complexity rating of the task they have selected. The user may choose to disregard any suggestions, and may also choose to deactivate subsequent suggestions at any time. Similarly, Strategy interactions are recorded in the application model and when the UMC recognizes the interaction the user is informed of the potential difficulty. Like other adaptations, interaction detection can be deactivated and later reactivated if necessary.

Like other components of the user model, the user's complexity rating evolves over time as a result of the user's interaction history with the system. The basic algorithm for updating the user complexity level was seen in the initialization routine in Figure 3. The user complexity level is increased by one if the user has invoked at least two functions or Strategies that have a complexity rating higher than the current user complexity rating, and is decreased by one if the user has not recently (within the last five uniquely dated sessions) used any functions or Strategies that are at least as complex as the user complexity rating.

The third phase of implemented adaptations is intended to address concerns with the initial data entry process. Like many other systems, TIMS gathers data through a series of dialogs that flow in a logical sequence, and again like many other systems, it is possible to accidentally terminate the series of dialogs by hitting the wrong button on the dialog. This was 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 abnormally terminated. This warning was displayed only if the dialog was accessed as part of the initial data entry process, not if it 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. This was dealt with by recognizing that the ultimate goal of the user is the creation of a printed financial plan for a client. Printed financial plans are generated using Document Packages that accept data from the system and export it in the form of a Rich Text Format file that can be used by any modern word processing program. Each document package requires specific information in order to be used. For example, the default document package, Retirement Plan, requires two situations: the current financial situation and an improved situation created by applying financial planning strategies to the first situation. Because the user is performing tasks to satisfy the requirements of a document package, the user modelling component assists in advising on the next logical step by maintaining these requirements explicitly. At each step, the UMC examines the currently active document package and its outstanding requirements, and 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. 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 was by far the most useful adaptation.

On an informal basis, the adaptations and the minimalist user modelling component described in this Section have all proven to be useful. A study was conducted in order to more rigourously examine the effect of these adaptations on user satisfaction. The nature and results of this investigation are described in the next Section.

4 Empirical Evaluation

The purpose of this experiment was to investigate the impact of an adaptive user interface on user satisfaction levels. It was hypothesized that complex 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 for novice users over a TIMS program with no user model.

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. Of the 44 participants, 5 were female, and 39 were male. All 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.

The Questionnaire. An instrument to measure user satisfaction was chosen based on 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 Shneiderman 1987 for an early version of the questionnaire). 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 has documented user interface evaluations in various industrial and academic environments (QUIS, 1997). Participants were given a modified short version of the Questionnaire for User Interface

Satisfaction (QUIS) v5. The 21 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). Participants were assigned to one of four groups. One TIMS group and one P-TIMS group started the first component 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; 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.

All participants chosen for this experiment took part in a three hour TIMS training program as the first component of the experiment. This training program is based on the TIMS introductory training manual.

Each participant returned 2 days later for a half-day session to perform the second component of the experiment. Each individual was given a data worksheet containing sample family financial data and a set of instructions to assist them in performing certain tasks. The worksheet detailed tasks reflecting typical system usage that the participants needed to perform 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 following the completion of the worksheet.

Results. A 2 X 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 1 shows the means and standard deviations for all four groups.

Table 1. Means and standard deviations for the six dependent variables broken down into four groups.

	Treatment			
	TIMS		**P-TIMS**	
	Day 1	***Day 2***	***Day 1***	***Day 2***
Dependent Variables	Mean(S.D.)	Mean(S.D.)	Mean(S.D.)	Mean(S.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)
Participants:	11	11	10	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, $F(6,35)=1.73$, $p>.05$, nor for an 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, we were interested in whether 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 that the experiment was performed on. 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?

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 an increased level of user satisfaction over all six factors, the amount of adaptation was minimal, therefore a smaller effect was expected.

5 Conclusions

There are very clear advantages for the use of a simple user modelling architecture such as this 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 chosen from a very small set of potential candidates. This stereotype is not revoked, but is simply updated as new information is presented. Similarly, neither sophisticated learning mechanisms nor uncertainty management schemes are necessary in a simplistic scheme such as this. User models contain only a minimal set of user attributes, and each attribute always contains an inherent risk that it may be inaccurate due to its generality. Very basic learning about users is possible by monitoring user 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 based on this interaction history and the number of days 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. Such errors have only 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 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. In the opposite situation, the user might take a week of system use to reach his or her true level. However, the user models are accessible and can be modified by users, so the impact of an initially inaccurate user model is reasonably 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 (Krogsæter and Thomas, 1994, Krogsæter et al., 1994). 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. The mechanisms of user control detailed throughout this paper are directly implemented from Oppermannís methods.

In addition to the quantitative results detailed in the previous section, this work has also demonstrated that there are many practical considerations that must be made when performing applied research in the commercial arena (Strachan, 1997). Beyond the many obvious logistical issues (e.g. convincing managers the research will be useful to them financially and scheduling research to coincide with corporate training schedules), there are many logistical problems that might not be anticipated. For example, when attempting to compare an adaptive and non-adaptive commercial system during ongoing development, 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 to avoid training new users on an older version of the software and returning those users to their working environment with a newer version.

The nature of the user modelling component itself is dictated by practical concerns. The adaptations that are chosen for a commercial system, for example, must reflect concerns that real users have with an existing system. It is easy to make these types of decisions in isolation and to judge proposed changes based on one's intuition. It takes much more time and effort to actually observe users in realistic situations and to note the questions and concerns they have. In our case, the use of formal one-day training programs for the TIMS system presented excellent opportunities to monitor hundreds of first-time users of the system as they became acquainted with TIMS.

For the next step in this research, the focus of the adaptations will expand to include the implementation of adaptations designed specifically for different user types and groups. Examples of these are restrictions of system functionality and the simplification of the system to the point where it simply performs the functions that are necessary for the performance of the user's tasks. Experienced TIMS users will be solicited for suggestions.

Some unanticipated problems occurred in the empirical testing. We had expected the overall level of computer familiarization of the users to be higher than what we 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. Many users were unfamiliar with Windows, their new hardware, and the new software including Microsoft® Office and TIMS. It is not surprising that the ease-of-use reaction parameter was not that high. One user commented, "I feel just like I am learning to drive a Ferrari!". Making small changes to simplify a single program is not going to help a user who is feeling that overwhelmed.

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

6 References

Chin, J. P., Diehl, V., and Norman, K. (1988). Development of an instrument measuring user satisfaction of the human-computer interface. In *CHI'88 Conference Proceedings: Human Factors in Computing Systems*, New York: Association for Computing Machinery, 213-218. Available: http://lap.umd.edu/lapfolder/papers/cdn.html

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. 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, 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, 67-96.

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 UM'96, Fifth Int. Conference on User Modeling*, Kailua-Kona, HI, 13-20.

McTear, M. (1993). User modelling for adaptive computer systems: A survey of recent developments, *Artificial Intelligence Review* 7:157-184.

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

QUIS (1997). Available: http://www.lap.umd.edu/QUISFolder/quisHome.html

Shneiderman, B. (1987). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Reading, MA: Addison-Wesley.

Strachan, L. (1997). *Pragmatic User Modelling for Complex Knowledge-Based Systems*. Ph.D. Dissertation, Department of Computer Science, University of Manitoba, 1997 (forthcoming).

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.