

An Adaptive Window Management System

Stefan Stille, Shailey Minocha and Rolf Ernst

Institut für Datenverarbeitungsanlagen
 Technische Universität, Braunschweig
 Hans-Sommer-Str. 66, D-38106 Braunschweig, Germany
 stille@ida.ing.tu-bs.de

ABSTRACT Modern complex and information intensive computer applications involve multi-window operations. To simultaneously visualize all the necessary information for a task requires time-consuming window management operations by the user. Since these activities are not directly related to the user's task domain, time spent on window management results in loss of user's mental context and an increase in the actual task completion time. We propose an Adaptive Window Manager (AWM) which automates the layout of the windows on the display screen according to the current user and his current task domain and learns the user's layout requirements. In this paper, we shortly address what user model we used for building AWM.

KEYWORDS Adaptive Window Management, Adaptive User Interfaces, User Model

1. Introduction

With an ever-increasing rate of advances in computers, a user is burdened with a cluttered multi-window desktop and has to deal with rearranging the display layout (that is, positioning and sizing of windows) quite often to suit his current task or information requirement. In order to relieve the user from these activities, we are currently developing an Adaptive Window Manager (AWM), a realization of the idea of Adaptive Automatic Display Layout (A²DL)¹.

In (Stille et al., 1996), we have treated the motivation, software ergonomic requirements, the architecture and working of the proposed AWM. AWM is an auto-adaptive layout system, which would be able to identify the user's current working context, on the basis of each working context's distinguishing software-tools and their states, and would be able to adapt the layout characteristics to suit the current user and his current working context. It is *auto-adaptive* in the sense that during

the user-computer interaction, it *learns* about the user's layout preferences for his various working contexts from his layout actions, *updates* this 'learnt' information with the change in user's preferences, and uses this 'learnt' knowledge to *adapt* the layouts to suit the working contexts of the user in his future interactions. AWM maintains a profile that is specific to each user and contains the learnt contexts and layout features for each context as per the user's preferred layout characteristics for that context. During the user's work, AWM not only learns new contexts and the associated layout features but also updates the information for the previously learnt contexts through feedback (layout corrections or modifications made by the user) and thereby, presents layouts for the learnt contexts which are appropriate for the user and his tasks.

In order to be able to provide adaptivity, AWM needs to have a user model which represents general knowledge about user's layout wishes. The appropriateness of this model is crucial for user acceptance of the adaptive system as it represents the information that is the basis for adaptation.

¹This research work is being supported by German Research Foundation under grant no. Er 168/7-2

2. The User-Model of AWM

In research prototypes described in literature, a user model is built by classifying the user's expertise on the basis of his behavior and action patterns with the computer system for a *particular task*, or by querying the user. Since AWM's support should not be restricted to a particular user or task domain, a more general user model had to be developed.

Our user model has been derived from an analysis of user behavior with a UI in a multi-window setup. The model is non-specific as it does not pertain to each *individual* user or does not model a group of users with common characteristics. Rather it is a *single* user model for all users, describing how users perform their activities in a multi-window environment. We, first, performed a theoretical analysis of user behaviour and verified the results by an empirical study based on protocols of user's actual interaction with a computer system.

An observational study of user-computer interaction indicates that a user views his work as composed of tasks to achieve certain goals and a goal is the desired output from a performed task. Typical tasks of users could vary from writing a document to debugging a program. A user divides his job into several tasks and each task may constitute of several sub-tasks. For each sub-task, the user may choose a software tool from a tool group (a group of software tools of the same functionality) that supports him in the best possible manner. On a UI level, each tool constitutes of several graphical objects, like windows, menus, list boxes, combo boxes, etc. The task or sub-task-status can be determined by observing the states of the active software tool and the graphical objects of this tool.

It is obvious that a user manipulates the layout of windows on the screen to suit his current context and the current activity within the context. The layout characteristics may vary with the current activity within a task. We conclude that the layout wishes of a user are closely related to the set of graphical objects which he currently works with and also to their current states.

Our source of information for our empirical study has been protocols of user interaction while performing their *everyday work* under UNIX/X11 over the window manager *fvwm*. We took log-files of 17 users into account and could clearly verify the results of (Henderson and Card, 1986) in the sense that we could detect working sets of usually no more than 5 windows

that users tend to use for accomplishing a task. An additional analysis of layout-activities (moving, resizing or iconifying of windows) showed that these are normally bound to a change of the working set. We thereby proved our theoretical analysis described above.

However, we also found out that the information we can get on the window manager level is not sufficient to explain the user's layout activities completely. This is due to the lack of information about the current state of tools. Hence, we decided to define a protocol for communication between applications and AWM so that they can supply information about their states. The decision which states are to be supplied can be made during the task analytic design of the application. This will be discussed in other communications.

3. Future Work

We are currently designing a UI for AWM which would enable the user to initiate or inhibit the layout actions performed by the system. It would also allow the user to specify a distinct task which he wants to perform. AWM will then start the appropriate applications in the desired state and arrange the appropriate layout on the screen. This will extend AWM to a session manager similar to *xsm* in X11R6.

A working prototype of AWM will be soon ready to evaluate the system by performing usability studies with actual users through questionnaires and interviews to discover whether the proposed system can really support users while working with a multi-window system aided by AWM. We will especially examine what kind of strategy for AWM's layout proposals users prefer and whether our approach to context recognition is appropriate.

References

- Henderson, D. A. and Card, S. K. (1986). Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention in a Window-Based Graphical User Interface. *ACM Transactions on Graphics*, 5(3):211-243.
- Stille, S., Minocha, S., and Ernst, R. (1996). A²DL—An Adaptive Automatic Display Layout System. In *Proceedings of the IEEE 3rd Annual Symposium on Human Interaction with Complex Systems (HICS 96)*, Dayton, OHIO, USA. IEEE Computer Society Press.