AdWiL: Adaptive Windows Layout Manager

Mona Haraty
School of Interactive Arts and
Technology, SFU, Surrey, BC, CA
mha48@sfu.ca

Syavash Nobarany
School of Interactive Arts and
Technology, SFU, Surrey, BC, CA
sna35@sfu.ca

Steve DiPaola
School of Interactive Arts and
Technology, SFU, Surrey, BC, CA
sdipaola@sfu.ca

Brian Fisher
School of Interactive Arts and
Technology, SFU, Surrey, BC, CA
bfisher@sfu.ca

CHI 2009, April 4–9, 2009, Boston, MA, USA.
ACM 978-1-60558-247-4/09/04.

Abstract
This paper addresses a challenge for the design of visual analytics software, managing placement of multiple windows while accomplishing a cognitively challenging analysis task. We are designing an adaptive windows layout manager that will support the user's creativity by facilitating concentration on the task at hand.

Keywords
Visual analytics, layout, flow, creativity, windows management

ACM Classification Keywords
H.5.2. Information interfaces and presentation: User Interfaces—Graphical user interfaces (GUI), Interaction styles, Screen design, Windowing systems

Introduction
Investigative analysts often need a number of open windows while accomplishing an analysis task. When the number of required windows for a task increases, controlling their size and position adds an additional task to the cognitively demanding analytic process. A goal of visual analytics systems is to support the analyst in being “in the flow” [2] defined by Csikszentmihalyi as a state of engagement in an

activity with the experience of satisfaction and timelessness [3]. Bederson has reviewed characteristics of flow in the interface design context. Among them is maintaining control of the activity. In a window management task, this would suggest giving the control of placing and resizing of the open windows to the user. Although this degree of control is inevitable in adaptive interfaces, in a situation where the primary task is cognitively demanding (such as analysis), a windows layout manager would support immersion in the concentrated activity, which is important for flow [4]. Accordingly we propose automating layout of workspace windows in the context of the user's cognitive process.

In this paper, we proposed a genetic algorithm-based windows layout manager that generates appropriate layouts according to the user's current task and considering the user's reactions to the offered layouts. Our data include click patterns on windows and the relationships between the windows. Relative positions of windows may impact the primary task by making relationships that may have been overlooked (e.g. by functional fixedness) apparent. For example, two diagrams that do not have any known relationship but do show similar patterns may support the discovery of novel relationships. We hypothesize that an optimal layout will support the user's creativity in complex problem solving tasks, which may lead to new discoveries in less time.

Based on the currently open windows, a proposed layout map is offered to the user in a small part of the screen. She can select the offered layout to be applied to the workspace. Once a specific layout is selected, each window has a smooth animated motion to its new position in the layout. The smooth transition between different layouts helps the user to understand the changes in windows positions. To avoid becoming lost in different layouts, especially when the new layout is significantly different, a history of layouts is provided, which keeps the track of user's layout selection.

Related Work

There has been extensive research in the field of display space management for windows layout [9,10,12,14,15]. The main concern in most was optimizing the use of display screen space, maintaining the most task-related windows while snipping unimportant parts of a window [7] and/or making them transparent [8].

Despite the variety of proposed windows management systems, there is a need for a windows layout manager specifically designed for visual analytics tools. Some visual analytics tools like Starlight [13] have attempted a solution to this problem using traditional windows layout management techniques such as tiling and cascading. These techniques however are inadequate for larger numbers of windows seen in newer applications such as Jigsaw. Jigsaw's approach is to utilize multiple-monitors to avoid the windows management problem [5]. However, multi-monitoring does not eliminate the need for windows management for each screen. In fact it increases the complexity of the management problem which must take into account the presence of information on other monitors in fixed locations. In practice, users tend to focus on one monitor for their principal activity and use other monitors for peripheral awareness of information rather than switching frequently between monitors [6].

Goals

In the adaptive windows layout manager (AdWiL) we target analysts as users and focus on organizing windows related to a single task. Our goals are to enhance the visual scope and to better match the surface layout to the cognitive task.

Increasing the Visual Scope

One of the important issues in information visualization is representing information in such a way that the visual scope is increased. According to [11] visual scope is the degree to which the user is able to integrate information across a display of multiple windows and to grasp the whole of whatever is being displayed. The first extreme in increasing visual scope is putting all the open windows on the display screen. This extreme is not helpful for two reasons: first, it reduces the visibility of information in small windows when the number of windows increases, second, showing a large amount of information without sufficient cues and clear patterns to guide focal attention will distract user. The second extreme is displaying just the relevant windows to the user's task.

The AdWiL approach considers both extremes by showing the related windows and also some of the other windows, taking into account the likelihood that they might be enlightening or inspiring for the user in her analysis.

Matching the surface layout to the cognitive layout

Surface layout is referred to the physical positions of objects on the screen while the cognitive layout is the mental representation of the visual components' arrangement [11]. The way in which a user interacts with the windows should reflect their cognitive thread in accomplishing a specific task [1]. Therefore, AdWiL attempts to match the offered surface layout with the cognitive one in the user's mind. It tries to achieve this based on relevant data such as the user's preferences and interactions, which are used to change the fitness function's coefficients in the genetic algorithm. This leads to more productivity as the user sees what she expects. Beside the available information on user preferences and interactions, some rules of thumb such as having less overlapping sections and more visible windows are considered in choosing suitable layouts.

AdWiL Design

In this section, we discuss the factors that are taken into account in designing the genetic algorithm-based windows layout manager, how AdWiL designs a layout based on how different users treat the display screen, and how AdWiL can learn from the user's interactions.

Problem formulation: genetic algorithm (GA) approach

Each chromosome is representative of a layout of the open windows, which is a one-dimensional array of genes. Each gene is a pair of (x, y) indicating the position of a window on the screen. A Layout is determined by the positions of the windows and its appropriateness is determined by the GA's fitness function. We designed a multi-component fitness function that includes several criteria in a weighted linear function. The distances between windows, and the overlapping area of windows are the two components of the fitness function, which are weighted by windows' importance and windows' relationships. These two factors are determined by the user's interaction with the windows such as windows' switching sequence.

Window's importance
In the design of AdWiL, widows' importance refers to the degree of the window's "relatedness" to the user's task. Importance of a window is estimated by the user's interaction with that window considering the frequency and the temporal pattern by which it receives input from the user. Since window's importance is a changing attribute during user's task, the temporal distance between current usage of the window and the previous usage is used to weight the importance of that window. This window's feature is used in the GA's fitness function to weight the overlapping areas. Important windows should not be covered with unimportant ones; it has a negative score in the fitness function.

Windows relationships
A variety of relationships exist between windows in visual analytic (VA) software. When the analyst is working on a specific data set, she may open a table related to that data set, including multiple views on that data set, as well as reports, and notes that are all related to that specific data set. This type of interaction is characteristic of visual analytics tasks, which places an emphasis on integration of multiple sources of information. The second kind of similarity is type-similarity between windows, for example, having multiple graph views open on different data sets that might be similar. The closeness of windows according to the type-similarity may provide insight for the user in interpreting unexpected patterns when a similar pattern is seen on a different data set. The last type is in the form of a parent-child relationship, which refers to the relationship between the notes or annotations on a specific data included in a window and that window.

In this prototype, related windows are shown by the same background color, see figure 1. Once the type of the relationships are extracted or specified by users, it is considered in the GA's fitness function as the coefficient of the fitness function's distance component.

Overlapping Windows
Dealing with overlapping windows is an inevitable part of a windows layout manager. Although some of the windows layout managers avoid overlapping windows by tiling [9], this limits visibility of the data due to their smaller size. AdWiL allows overlapping between windows and also considers the overlapping area in each layout as a component in its GA's fitness function.

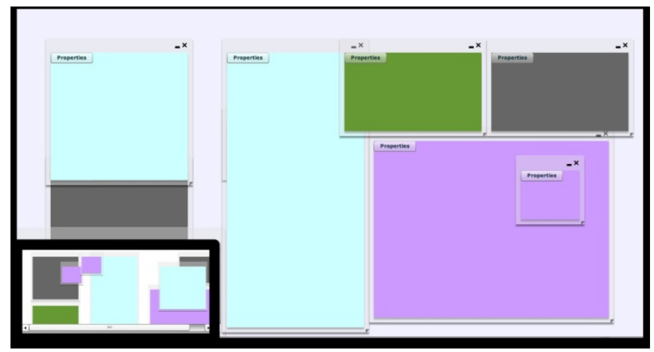

Figure 1: Sample layout of the windows. The small window in the bottom left corner of the screen shows the offered layout for the open windows.

Focus area in the Display screen
Users usually attend to different parts of the display screen in different ways; hence, they have different focus areas. AdWiL accepts user-defined focus area and it considers putting the important windows in the focus area in its fitness function.

Learning module of AdWiL

As the analyst goes through the process of analysis, AdWiL learns what kind of layouts can best be fit into the analysts' task at hand by noting the ongoing user interactions with the current windows and the suggested layouts. Selecting a specific layout and continuing to work using that layout for a relatively long time without changing the position and size of the windows supports the decision that the suggested layout is appropriate for the current task. If the analyst works with a layout in which there are a number of obscured windows for some time, the system learns that those windows can be removed from the future offered layouts. This feature of the system would be helpful as the analysts used to open a lot of windows and forget to close the ones that are no longer required.

Implementation

An early prototype of the system is implemented using a genetic algorithm for placing windows on appropriate positions on the screen. We have selected GA as our algorithm for finding suitable layouts of windows for the following reasons: 1. GA's basic heuristic is that combination of two fit chromosomes might lead to a better chromosome, which is well suited to a position-based system like this one, 2. The large search space needs a heuristic search, 3. It offers various layout choices to the user, as the best solutions found after each generation is shown to the user, 4. As it is kind of an interactive genetic algorithm, the fitness function gets updated during the user's interaction through the learning process of the system. The proposed design guidelines are considered in the genetic algorithm's fitness function, which are prone to change due to the change in user's interaction.

Evaluation in progress

Our hypothesis is that AdWiL supports analysts in a visual analytics task and facilitates the process of decision making and knowledge discovery as it provides a situation for the user that enables her to focus on her activity by spending her cognitive resources on the task itself instead of dealing with the trivial activities like managing the size and position of windows. To test this hypothesis we will design a series of tasks and give them to two groups of subjects. The first group of subjects should perform the assigned tasks without a windows layout manager (control group) while the subjects in the second group perform the same tasks using AdWiL as their windows layout manager. After the experiment, subjects will be interviewed to see how AdWiL helps them to focus on the tasks. This data will be used to help the further extension of the system.

Conclusion and future work

We have presented AdWiL, adaptive windows layout manager, as a solution for an existing problem in the visual analytics (VA) tools, which is handling a large number of windows. AdWiL uses genetic algorithm to find real time appropriate windows layouts for suggesting to the user. All the constraints and considerations regarding to the appropriate placing of the windows on the screen are considered in its GA fitness function, including the windows' importance, overlapping areas, and windows' relationships. The overall goal of the study is to remove the user from the tedium of dealing with windows management, so as to provide the opportunity of keep the user in the flow of analysis, which supports creativity in problem solving.

By further developing the presented prototype, we are going to integrate it with real visual analytic software for a comprehensive evaluation of the system.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

References

[1] Ballard, D.H., Hayhoe, M.M., Pook, P.K., and Rao, R.P.N. Deictic codes for the embodiment of cognition. *Behavioral and Brain Sciences 20*, 04 (1997), 723-742.

[2] Bederson, B.B. Interfaces for staying in the flow. *Ubiquity 5*, 27 (2004), 1-1.

[3] Csikszentmihalyi, M. *Flow: The Psychology of Optimal Experience*. HarperCollins, 1991.

[4] Csikszentmihalyi, M. *Creativity*. Harper-Collins, New York, 1996.

[5] Gorg, C., Zhicheng Liu, Parekh, N., Singhal, K., and Stasko, J. Visual Analytics with Jigsaw. *IEEE Symposium on Visual Analytics Science and Technology, VAST 2007*, IEEE (2007), 201-202.

[6] Grudin, J. Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (2001), 458-465.

[7] Hutchings, D.R. and Stasko, J. Shrinking window operations for expanding display space. *Proceedings of the working conference on Advanced visual interfaces*, ACM (2004), 350-353.

[8] Ishak, E.W. and Feiner, S.K. Interacting with hidden content using content-aware free-space transparency. *Proceedings of the 17th annual ACM symposium on User interface software and technology*, ACM (2004), 189-192.

[9] Ishak, E.W. and Feiner, S. Content-aware layout. *CHI '07 extended abstracts on Human factors in computing systems*, ACM (2007), 2459-2464.

[10] Matthews, T., Czerwinski, M., Robertson, G., and Tan, D. Clipping lists and change borders: improving multitasking efficiency with peripheral information design. *Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM (2006), 989-998.

[11] Norman, K.L., Weldon, L.J., and Shneiderman, B. Cognitive layouts of windows and multiple screens for user interfaces. *Int. J. Man-Mach. Stud. 25*, 2 (1986), 229-248.

[12] Oliver, N., Smith, G., Thakkar, C., and Surendran, A.C. SWISH: semantic analysis of window titles and switching history. *Proceedings of the 11th international conference on Intelligent user interfaces*, ACM (2006), 194-201.

[13] Risch, J., Rex, D., Dowson, S., Walters, T., May, R., and Moon, B. The STARLIGHT information visualization system. *IEEE Conference on Information Visualization*, IEEE (1997), 42-49.

[14] Robertson, G., Horvitz, E., Czerwinski, M., et al. Scalable Fabric: flexible task management. *Proceedings of the working conference on Advanced visual interfaces*, ACM (2004), 85-89.

[15] Stumpf, S., Bao, X., Dragunov, A., et al. Predicting user tasks: I know what you're doing. *National Conference on Artificial Intelligence (AAAI'05)*, AAAI (2005).