# Modeling the Disruption to the User's Mental Model

Bowen Hui and Craig Boutilier
Department of Computer Science, University of Toronto
{bowen,cebly}@cs.utoronto.ca

The need to develop user adaptive systems have been made more evident by emerging work that applies user modeling to technologies such as automatic interface and software customization [2, 3] and health care support systems [1]. Since different people prefer different styles of interaction, intelligent systems can be designed to adapt to the user's changing needs and preferences. The sequential nature of human-computer interaction makes it possible to build user adaptive systems that learn the user's preferences overtime. These systems have the potential to increase work productivity and user satisfaction. However, the main bottleneck in building effective adaptive systems is that adaptive behaviour may cause *disruption* to the user's *mental model* of the application. For example, the user has learned that the `Exit` menu item is located near the bottom of the `File` menu. The system observes that the user frequently selects the `Exit` menu item, so it decides to move this function to the top of the `File` menu. The system also observes that the user never uses the `Options` menu item, so it decides to hide it. Although this adaptive behaviour may be intended to make function access more convenient, users may find these changes disruptive because the new application state conflicts with information in the user's mental model. In particular, the next time the user attempts to use `Exit`, he will no longer be able to find it in (or near) the location he remembers and will have to learn its new location. On the other hand, if the user has no prior information about the location of `Options`, then the disruption is not caused by contradictory knowledge in the mental model, but rather, the extra search time induced in traversing the hidden items (which may be negligible). While it is possible to introduce interaction mechanisms to ease the abrupt transition between two application states (e.g., via using notifications or animations), some users may find such mechanisms distracting or unnecessary. Therefore, a viable and scalable solution needs to model the relevant parameters in the system's adaptive behaviour and quantify the disruption of system actions. We argue that user adaptive systems need to be able to assess the amount of disruption its adaptive behaviour has on the user's mental model so that the system's decision making process can explicitly trade off the negative impact of its adaptations with the reward of assisting the user. While researchers generally agree that the user's mental model are *incomplete*, *dynamic*, and *unstable* [4], current computational approaches adopt a non-probabilistic representation and use heuristics for updates overtime [5]. In contrast, we propose a model using a dynamic Bayesian network that enables the system to capture the uncertainties and interaction dynamics in a principled manner. Our goal is to build an assistive agent that tailors to individual user needs and preferences, so that it can help the user in achieving the current task either by making the task completion process more efficient or the interaction experience easier to manage. We focus on the development of an intelligent agent in Powerpoint here, but the general principles apply more broadly.

To quantify disruption, we first build a representation of the user's mental model, $\theta$. Roughly speaking, $\theta$ reflects the user's expectations of the next application state after executing a function in the current state. Formally, we define $\theta$ as the user's belief of the application state given the current state, $S$, and system observations of user actions, $O$, $\theta : S \times O \rightarrow \Delta(S)$. We refer to the system's approximation to $\theta$ as $M$. Realistically speaking, $M$ is a belief distribution over possible mental models, i.e., $M = \Delta(\theta)$. However, since the system can observe both the application state and user actions, the system can approximate $\theta$ directly as $M : S \times O \rightarrow \Delta(S)$ as a simplification. To be more concrete, we restrict our attention to modeling the *location* of application functions. Other aspects of the application, such as the function's expected execution time or the semantic representation of widgets, can be modeled similarly. We hypothesize that ongoing usage of a function, $f$, and its neighbouring functions influence how well the user knows the location of $f$: $Pr(KnowLoc_t | KnowLoc_{t-1}, UseF_t, UseNB_t)$. Furthermore, the amount of disruption is influenced by the relative change in $f$'s location and how well the user knows the original location: $Pr(Disruption_t | KnowLoc_t, ChgLoc_t)$. To get a better sense of how realistic this model is, we conducted a pilot experiment with 5 participants using an interface designed to mimic the Powerpoint menu structure, but with labels replaced by non-computing terms (e.g. `Fruits` instead of `File`). The experiment consists of 3 tasks. The first task is a training phase that asks the user to select 10% of the available menu functions, each drawn from different frequency distributions. The second task is designed to assess the "strength" of the participant's mental mental developed from the first task by asking participants to identify the location of menu functions. Results from this task provides an indication of how well the user knows the location of a function, based on the frequency of use (from task one), the frequency of a neighbouring function being used, and major "landmarks" surrounding the function (e.g., the first item in a submenu). The third task is designed to elicit, on a Likert scale, the amount of disruption that is induced on the user by changing the location of a menu function with different usage frequencies. We found that swapping two neighbouring function locations generally go unnoticed, while hiding a function is very disruptive, especially when the user sees it unnecessary. On the other hand, changing a location to the top of the menu had varying results, which could be explained by the need to model individual preferences in trading offs convenience with disruption. Results from the last two tasks also suggest that users treat a menu location and its neighbouring locations equally. This suggests that the mental model of the function location is represented as a "region", which is supported in our model. We also use this notion of a region to guide the discretization of the change in location (ChgLoc) parameter that quantifies the disruption of an action. Although preliminary, these results support the overall structure of our model and they identify a richer set of observations to modeling $\theta$. As future work, we plan on extending this experiment to collect data for learning the model parameters.

# References


[1] J. Boger, P. Poupart, J. Hoey, C. Boutilier, G. Fernie, and A. Mihailidis. A decision-theoretic approach to task assistance for persons with dementia. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 1293–1299, Edinburgh, Scotland, 2005.

[2] K. Gajos and D. Weld. SUPPLE: Automatically generating user interfaces. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 93–100, Madeira, Portugal, 2004.

[3] B. Hui and C. Boutilier. Who's Asking for Help? A Bayesian Approach to Intelligent Assistance. In *International Conference on Intelligent User Interfaces (IUI)*, pages 186–193, Sydney, Australia, 2006.

[4] D. Norman. Some Observations on Mental Models. In D. Gentner and A. Stevens, editors, *Mental Models*. Hillsdale, NJ:Erlbaum, 1983.

[5] M. Sasse. *Eliciting and Describing User's Models of Computer Systems*. PhD thesis, School of Computer Science, The University of Birmingham, Birmingham, England, 1997.