Title
Predicting Information Needs: Adaptive Display in Dynamic Environments

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 30(30)

ISSN
1069-7977

Authors
Love, Bradley C.
JOnes, Matt
Tomlinson, Marc T.
et al.

Publication Date
2008

Peer reviewed

Predicting Information Needs: Adaptive Display in Dynamic Environments

Bradley C. Love (brad_love@mail.utexas.edu)
Department of Psychology
Austin, TX 78712 USA

Matt Jones (mcj@colorado.edu)
Department of Psychology
Boulder, CO 80309 USA

Marc T. Tomlinson (mtomlinson@love.psy.utexas.edu)
Department of Psychology
Austin, TX 78712 USA

Michael Howe (michael.howe@mail.utexas.edu)
Department of Psychology
Austin, TX 78712 USA

Abstract

Although the information available to human operators can increase without obvious bound, human information processing capacities remain fixed. Finding and selecting the relevant information to display in this deluge of options imposes a burden on the user. We describe a domain-general system, Responsive Adaptive Display Anticipates Requests (RADAR), that learns to highlight the information a user would select if the user searched through all possible options. By offloading this selection process to RADAR, the user can concentrate on the primary task. Tests with human subjects in a tank video game environment that required monitoring several information channels while maintaining situation awareness revealed that players performed better with RADAR selecting which channel to display. RADAR can customize its predictions to a user to take into account individual differences and changes within a user over time. RADAR's emphasis on learning by observing minimizes the need for explicit guidance from subject matter experts.

Introduction

We increasingly find ourselves in information-rich environments. Often, many information sources are potentially useful for completing a task. For example, in coordinating disaster relief, sources of potentially useful information include video feeds, weather forecasts, inventories of relief supplies, GPS tracking of support vehicles, etc. Likewise, the many sensors, gauges, and navigation systems in a modern automobile are potentially useful to the driver.

One key challenge people face is identifying which source of information is desired at the current moment. Although the information available to a human operator can increase without obvious bound, our basic information processing capacities remain fixed. Each additional information source incurs a cost to the human operator by increasing the complexity of the selection process. As informational channels are added, at some point, the marginal costs (in terms of cognitive load) eclipse the marginal benefits.

In this report, we propose and evaluate a system that eases this selection process by highlighting the information channel desired by the user. The system, Responsive Adaptive Display Anticipates Requests (RADAR), learns to approximate the selection process of the human operator by observing the user's selection behavior. In cases where RADAR successfully approximates the human's selection process, the cognitive cost of information selection can be offloaded to RADAR.

RADAR is named after the character Radar O'Reilly from the television series M*A*S*H. Radar O'Reilly had an uncanny ability to deliver information to his commander moments before the commander formulated his request, much like how RADAR learns to anticipate the information needs of the user to reduce cognitive load. Before presenting RADAR and empirically evaluating it in a well-controlled experiment, we briefly review related work.

Related Efforts

The topic of plan recognition in AI is concerned with correctly attributing intentions, beliefs, and goals to the user. Plan recognition models tend to subscribe to the Belief-Desires-Intention framework (McTear, 1993). This line of work relies on knowledge-based approaches for user modeling and encoding insights from domain-specific experts (Goodman & Litman, 1992). These approaches can involve identifying a user's subgoals through task-analysis (Yi & Ballard, 2006). Once a user's beliefs, intentions, and goals are understood, display can be adapted appropriately (Goodman & Litman, 1992).

Instead of focusing on identifying the internal state of the user, other approaches rely on input from domain experts to adapt display to emphasize the information to which the user *should* attend. For example human experts can label episodes and these episodes can serve as training instances for machine learning models that prioritize display elements (St. John, Smallman, & Manes, 2005). Alternatively, input from human experts can be used to build expert systems or Bayesian models to prioritize display (Horvitz & Barry, 1995).

Our approach diverges from the aforementioned work. Rather than prescribe which information source a user should prioritize, we attempt to highlight the information a user would select if the user searched through all possible options. Unlike work in plan recognition, we sidestep the problem of ascribing and ascertaining the user's internal mental state. Instead, RADAR learns to directly predict a user's desired display from contextual (i.e., situational) features. Our approach emphasizes learning as opposed to preprogrammed interfaces (Mäntyjärvi & Seppänen, 2002). Adopting a learning approach to adaptive display has a number of positive consequences, including the ability to take into account individual differences across users (Schneider-Hufschmidt, Kühme, & Malinowski, 1993). Another positive consequence is that minimal input from subject matter experts is required to build a system. Like other keyhole approaches (Albrecht, Zukerman, & Nicholson, 1998), our approach is based on observing the user's behavior without interfering with or directly querying the user.

Overview of RADAR

RADAR is designed to operate in task environments in which the user must select which display among numerous displays to monitor. For example, we evaluate RADAR in an arcade game environment in which players select which of eight possible displays to show on a Head-Up Display (HUD). Figure 1 illustrates how RADAR operates in such task environments. RADAR takes as input the current context (e.g, recent game history) encoded as a feature vector and outputs to the HUD the display it thinks the user wishes to view. The user is free to override RADAR's choice. RADAR learns from the user's acceptance or rejection of its display choices and over time converges to selecting the displays the user desires. Alternatively, RADAR can observe and learn to mimic a user's display preferences offline. After online training, RADAR can be used to select displays. In the studies reported here, offline training was used.

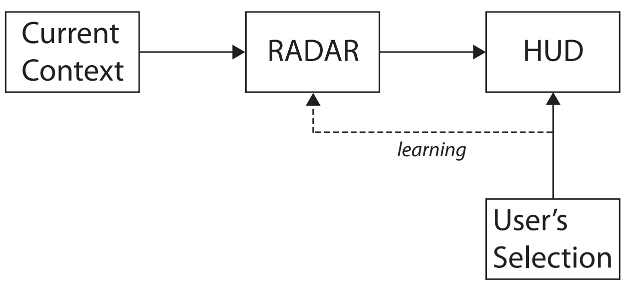

Figure 1: RADAR takes as input the current context (e.g., recent game history) and outputs its preferred display to the HUD. The user (e.g., the game player) can override RADAR's choice. Such corrections serve as learning signals to RADAR and increase the likelihood that RADAR will select the user's preferred display in similar situations in the future. Over time, RADAR approximates the information preferences of a specific user, allowing the user to offload the task of selecting the relevant information source (i.e., display) from numerous competing options.

In terms of current implementation, RADAR employs a two-stage stochastic decision process at every time step. In the first stage, RADAR estimates the probability that a user will update the HUD given the current context. When the sampled probability from the first stage results in a display update, RADAR proceeds to the second stage (otherwise the current display remains unchanged). In the second stage, RADAR estimates the probability distribution for the next display choice given the current context, and samples this probability distribution to select the next display.

The motivation for the two-stage approach is both computational and psychological. Separating display prediction into two stages improves RADAR's ability to predict display transitions. The same display currently desired is highly likely to be desired in 250 ms. This constancy would dominate learning if both stages were combined. The second stage's focus on display transitions allows for improved estimation of these relatively rare, but critical, events. Psychologically, the first stage corresponds to identifying key events in a continuous (unsegmented) environment, whereas the second stage corresponds to predicting event transitions. To make an analogy to speech perception, people segment the continuous speech stream into words (akin to RADAR's first stage) in the absence of reliable acoustical gaps between words (Saffran, 2003). Akin to RADAR's second stage, people anticipate which word (i.e., event) is likely to follow given the proceeding words (McRae, Spivey-Knowlton, & Tanenhaus, 1998).

The probability distributions associated with both stages are estimated by simple buffer networks (Cleeremans, 1993). As shown in Figure 2, buffer networks represent time spatially as a series of slots, each containing the context (e.g., game situation) at a recent time slice, encoded as a feature vector. The buffer allows both ongoing events and events from the recent past to influence display prediction. Despite their simplicity, buffer networks have been shown to account for a surprising number of findings in human sequential learning (Gureckis & Love, 2007). At each time step, weights from the buffer are increased from activated features to the display option shown in the HUD, whereas weights to the other display options are decreased. Over time, this simple error correction learning process approximates a user's information preferences.

RADAR's Formal Description

Player Model Our model of the player's choice behavior assumes that the player's preferred channel at any time, t, is determined by the state of the game at that time, S^t, together with the recent history of the game, $(S^t - l)_{1 \leq l < L}$. The recent history is included, in addition to the current state, to allow for fixed delays in information need (e.g., the player wants to see channel Y, l timesteps after event X occurs). The parameter L determines the maximum delay, that is, the longest time that past information can remain relevant to the player's choice. This parameter is currently set to 10 (i.e., 2.5 s).

For compactness, we write the sequence of current and recent game states as

$$\mathbf{S} = (\mathbf{S^{t-l}})_{\mathbf{0 \leq l < L}} \tag{1}$$

Because changing channels incurs a cost in terms of attention and motor resources, we do not assume that the player changes the HUD to his or her preferred channel whenever that preference changes. Instead, we assume a two-step stochastic process, in which at every timestep there is a probability that the player will change channels and, if the channel is changed, a probability distribution over the channel to be selected. The probability of switching channels is given by

$$P^t_{change}(C^t, \mathbf{S}) = \mathbf{P}[\mathbf{change}(\mathbf{t+1}) | \mathbf{C^t}, \mathbf{S}] \tag{2}$$

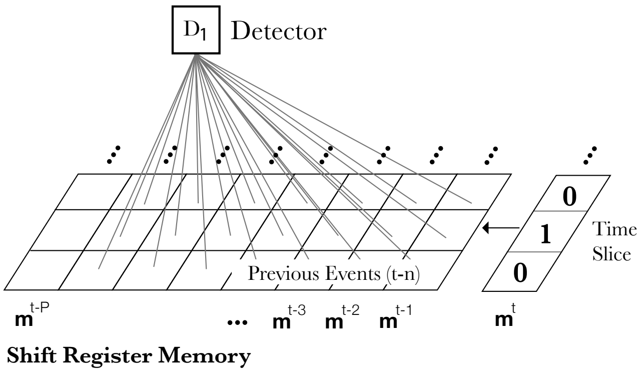

Figure 2: RADAR utilizes a buffer network to represent and learn from recent context (e.g., game history). Context is represented as a series of time slices. The tank game results are based on a context consisting of ten time slices of 250 ms each. The buffer functions as a shift register — the slice from the immediate time step enters one side of the buffer, all other time slices shift over one slot to accommodate the new entry, and the least recent time slice is removed from the buffer. Each time slice consists of a feature vector describing the current situation. Table 1 lists the features used for the tank game. Each possible display in the HUD has a detector that collects evidence to determine whether it is the situationally appropriate display. Association weights between features at various positions along the buffer and each detector are learned through error correction learning. For example, if a user prefers to have the fuel scope displayed when fuel is low, the weight from the fuel level feature's low value at various positions along the buffer to the fuel scope display detector will develop large, positive weights.

where C^t is the current channel. If the player does change channels, the probability of selecting channel j is equal to

$$P^t_{choice}(j,\mathbf{S}) = \mathbf{P}[\mathbf{C^{t+1}} = \mathbf{j} | \mathbf{change(t+1)}, \mathbf{C^t}, \mathbf{S}] \qquad (3)$$

Context Representation The state of the game at any time, t, is represented by a vector of F feature values:

$\mathbf{S^t} = (\mathbf{S^t_f})_{\mathbf{1 \leq f \leq F}}$

These features used in the studies reported here are listed in Table 1. Continuous features are discretized, and all features are coded to take on values $0 \leq S_f < V_f$ (where V_f is the number of possible values of feature f).

Prediction The display system operates by predicting two sets of probabilities, corresponding to the two steps in the model of the player's choice behavior: p_{change}, the probability that the player will change channels; and p_{choice}, the distribution over the new channel if the channel is changed. Both types of probabilities are predicted from the information in the game history, **S**. The system learns a separate set of weights for the two types of predictions, each indexed by the current channel (C^t), feature (f), value for that feature (v), and lag (l); the weights for p_{choice} are also additionally indexed by the value of the candidate new channel (j). The system's predictions are derived as a linear combination of these weights, weighted by the corresponding unit activation values:

$$P^t_{change}(C^t,\mathbf{S}) = \sum_{\mathbf{f,l,v}} \mathbf{w^{change}_{C^t,f,l,v}} \cdot \mathbf{a^t_{f,l,v}} \qquad (4)$$

$$P^t_{choice}(C^t,j,\mathbf{S}) = \sum_{\mathbf{f,l,v}} \mathbf{w^{choice}_{C^t,j,f,l,v}} \cdot \mathbf{a^t_{f,l,v}} \qquad (5)$$

Operation At each timestep the system changes the channel with probability pchange(C^t,**S**). When it does change the channel, it selects the channel j that maximizes $p_{choice}(C^t, j, \mathbf{S})$ subject to $j \neq C^t$.

Learning The weights w_{change} and w_{choice} are computed from the player's manual choice behavior, by miminizing the following error terms

$$E^{change} = \begin{cases} (p_{change})^2 & C^{t+1} = C^t \\ (1-p_{change})^2 & C^{t+1} = C^t \end{cases} \qquad (6)$$

$$E^{choice} = \left[1 - p_{choice}(C^{t+1})\right]^2 + \sum_{j \neq C^t, C^{t+1}} p_{choice}(j)^2 \qquad (7)$$

The former is summed over all timesteps, and the latter is summed over all timesteps on which the player changed channels. In practice, the weights in RADAR's buffer networks are estimated directly and efficiently using optimized linear algebra routines (Anderson et al., 1999) rather than trial-by-trial error correction procedures. Both methods converge to the same solution, but trial-by-trial learning takes longer to do so.

Prescience The model is trained so as to predict players' display-selection behavior in advance of when that behavior would actually occur. This is accomplished by shifting the channel values relative to the feature values in the training set. The sequence of channel values (i.e. on all timesteps during play) is moved earlier by τ steps, which effectively teaches the model to predict players' behavior τ steps into the future. Thus, when allowed to control the display, the model is able to immediately select the player's (predicted) preference τ steps into the future. The shift, τ, is currently set to 2 timesteps, i.e. 500 ms.

Evaluating RADAR

RADAR was evaluated in a video game task environment in which human players battled robot tanks. The task environment was adapted from the open source BZFlag 3D tank battle game (see www.bzflag.org). Modifications to BZFlag included expanding the state of a player's tank to include limited ammunition, fuel, and health. Players could pick up corresponding flags in the game to replenish these assets. Additionally, the display was modified to include a pop-up menu that allowed players to select one of eight possible displays to view on the HUD.

The eight possible displays for the HUD correspond to the first eight features listed in Table 1. Three of the displays provided the levels of the aforementioned assets. Three other displays were player-centered scopes that indicated the location of flags to replenish the corresponding asset. The remaining two displays consisted of a terrain map and a line-of-sight unit radar that provided the positions of enemy tanks and fire when not obscured by building structures. Figure 3 illustrates the menu for selecting which display to send to the HUD display as well as an example HUD.

Table 1: The features used to describe the current game context are listed. These features serve as inputs to RADAR. From these inputs, RADAR predicts which display the user wishes to view.

Feature Type	Feature Name	
Display Shown (1-8)	Terrain Map	Unit Radar
	Ammo Status	Ammo Scope
	Health Status	Health Scope
	Fuel Status	Fuel Scope
Tank Condition (9-12)	Ammo Level	Health Level
	Fuel Level	Out of Fuel
Flag in View (13-16)	Any Flag	Ammo Flag
	Health Flag	Fuel Flag
Flag Picked Up (17-20)	Any Flag	Ammo Flag
	Health Flag	Fuel Flag
Dynamic/Battle (21-23)	Tank is moving	Tank hit
	Number of enemy tanks in view	

RADAR's task was to anticipate the displays a player wished to have shown on the HUD, thus allowing the player to offload display selection to RADAR and devote full attention to game play. Successful game play requires maintaining situation awareness of the state of one's tank, the locations of flags to replenish assets, and the position of enemy tanks. Our prediction is that players using RADAR should outperform those in control conditions.

Below, we discuss results from three studies comparing player performance under RADAR to various controls. In each study, subjects were evaluated in game situations involving two enemy (robot) tanks. A game ended when the subject's tank was destroyed. When an enemy tank was destroyed, it was replaced by a new enemy tank at a random location.

Study 1: Group Model Evaluation

Experimental Methods Five undergraduate student volunteers in the laboratory served as the research subjects. These students each had over ten hours experience playing the tank game without RADAR operational (i.e., all displays were manually selected from the menu). Because this is the first evaluation of RADAR, the testing procedure was simplified to the greatest extent possible. A single set of weights that predict display preferences was calculated, as opposed to deriving a separate set of predictive weights for each subject. Thus, at test, each subject interacted and was evaluated with

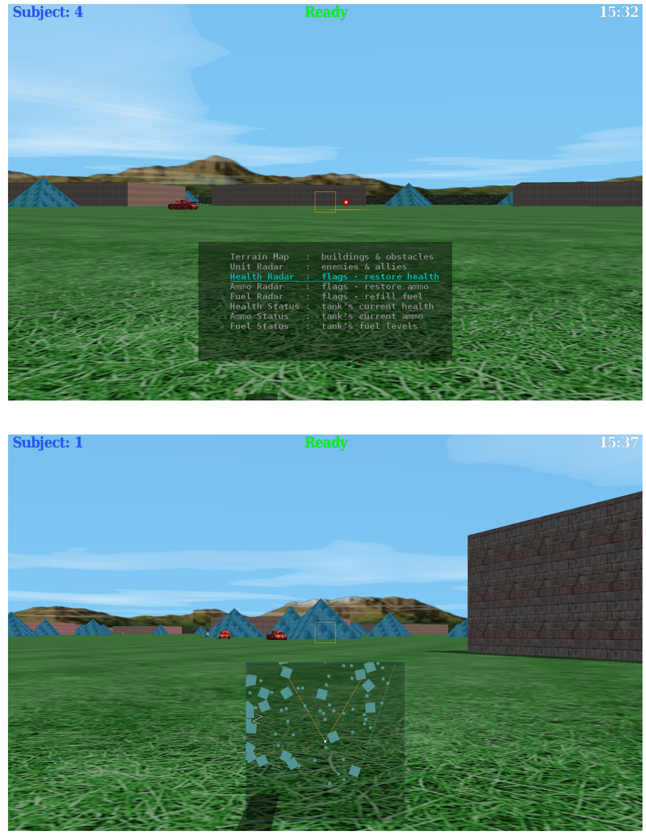

Figure 3: Screenshots from our modified version of the BZFlag tank game are shown. The top panel shows the selection menu listing the eight possible displays from which players can choose. These eight possible displays correspond to the first eight features listed in Table 1. Once a display is selected, the menu is replaced with the chosen display in the HUD, as shown in the bottom panel. Players can offload the task of selecting relevant displays to RADAR.

the same version of RADAR rather than a user-customized version. To further simplify evaluation, eight hours (across all five subjects) of game data without a functioning adaptive display (i.e., all display choices were determined by the subject) were used to derive RADAR's weights, as opposed to incrementally training RADAR online. These evaluation choices make interpretation of the results clearer, but potentially reduced RADAR's benefits as individual differences in information preferences and drift within an individual's preferences over time are not captured by this procedure. The features that describe the game history for each time slice are listed in Table 1.

To provide a stringent test of the adaptive display system, subjects' ability to manually select displays (i.e., override RADAR) was disabled. Removing this ability forces subjects to completely rely on RADAR for information updates and simulates conditions in which operators do not have the option of scrolling through menus while on task. Performance with RADAR functioning was compared to a closely matched control condition. In the control condition, displays were shown for the same durations as the experimental condition (i.e., the base rates and mean durations of the eight displays were matched), but transitions between displays were determined at random rather than selected by RADAR. Thus, any benefit of RADAR over the control condition is attributable to RADAR's selecting the situationally appropriate displays for the HUD, as opposed to RADAR's merely learning which displays are most valuable in general. For each game, the probabilities of a subject being assigned to the experimental or control display conditions were 80% and 20%, respectively. Each player completed fifty test games.

Experiment Results The primary dependent measure was the mean number of enemy tanks destroyed per game. As predicted, subjects killed significantly more (4.54 vs. 3.29) enemy tanks in the experimental than in the control condition, $t(4) = 10.60, p < .001$. All five subjects showed an advantage with RADAR. These results indicate RADAR's effectiveness.

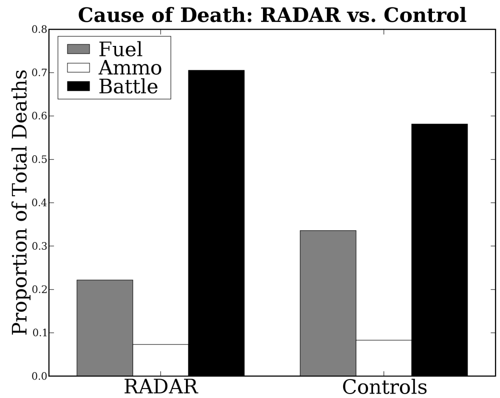

Figure 4: Study 2 demonstrates that players are more likely to lose situation awareness and die from somewhat avoidable causes, such as running out fuel, when RADAR is not operating.

Study 2: Maintaining Situation Awareness

Study 2 was patterned after Study 1. The same RADAR group model, experimental conditions, and methods were used. Nine inexperienced players participating in a 36-hour sleep deprivation study served as subjects. The question of primary interest was whether RADAR helps subjects maintain situation awareness. If so, subjects using RADAR should be aware of the state of their tank and die at lower rates from causes that are somewhat avoidable, such as running out of fuel or ammunition. Subjects who maintain awareness of the state of their vehicle are more likely to replenish fuel and ammunition when necessary. The distribution of player deaths by condition is shown in Figure 4. As predicted, a greater proportion of games ended with fuel and ammunition depleted in the control condition than when RADAR was operating, $\chi^2(2, N = 713) = 12.58$, $p < .01$. These results suggest that players were less aware of the state of their vehicle in the control condition.

Study 3: Assessing Individual Models

Experimental Methods Five undergraduate student volunteers in the laboratory served as the research subjects. These students each had over ten hours experience playing the tank game without RADAR operational prior to test data collection. Four hours of manual play data from each subject were used to train the various RADAR models evaluated at test. Unlike Study 1, subjects at test could manually override RADAR's display choices. Subjects completed test games in four conditions: Manual, Group, Individual, and Other Individual. In the Manual condition, no RADAR model was operable and subjects manually selected all displays (as in the training phase). In the remaining three conditions, a version of RADAR was operable at test. In the Group condition, a RADAR model was derived for each player using training data from all the other players combined. In the Individual condition, a RADAR model was derived for each subjects using only that player's own training data. In the Other Individual condition, subjects were assigned to another player's individual RADAR model. To evaluate RADAR's promise in contexts where minimal input from subject matter experts is available, a minimal feature set was used to predict display preferences in all RADAR models. This minimal set consisted of the "Display Shown" and "Tank Condition" features shown in Table 1. Each player completed 12 test games in each of the four conditions. Game order was randomly determined for each subject with games from the various conditions interleaved. Study 3 evaluates whether RADAR offers a potential benefit over purely manual operation.

Experiment Results Mean kills per condition are shown in Figure 5. Subjects killed significantly more tanks in the Individual and Other Individual conditions than in the Manual condition, $t(4) = 3.02, p < .05$ and $t(4) = 2.84, p < .05$, respectively. The advantage of these RADAR conditions over the Manual condition held for all five subjects. Interestingly, this advantage for RADAR did not arise because of a reduction in the rate of manual requests. In fact, subjects made significantly more requests per second (.13 vs. .12) in the Individual condition than in the Manual condition, $t(4) = 3.91, p < .05$, with the effect holding for every subject.

These results indicate that individual RADAR models are more effective than purely manual operation. The strong performance in the Other Individual condition was attributable to relatively novice subjects benefiting from using the display models of more experienced subjects. This serendipitous result suggests that RADAR may prove effective as a training system in which novice subjects train under an expert's RADAR model. The fact that more manual requests were made in the Individual condition than in the Manual condition suggests that RADAR freed cognitive resources so that subjects could seek additional information.

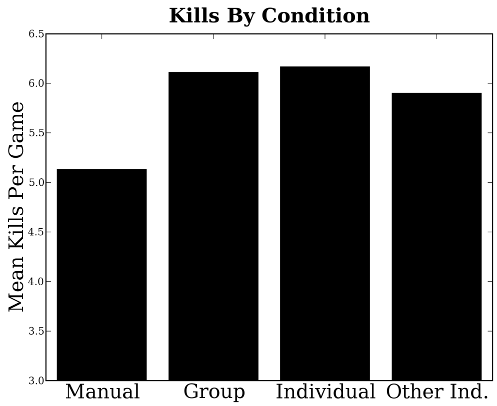

Figure 5: Mean kills per game by condition for Study 3.

General Discussion

Advances in information technology make large quantities of information available to human decision makers. In this deluge of information, finding and selecting the relevant piece of information imposes a burden on the user. This burden is particularly onerous in dynamic environments in which decisions must be made rapidly. RADAR is a domain-general system that learns to approximate the information search process of an individual user. By offloading this search process to RADAR, the user can concentrate on the primary task. Experimental results in a tank video game environment in which the player must maintain situation awareness demonstrate RADAR's promise. Players performed better with RADAR.

Systems that automate tasks for humans often result in unexpected negative consequences (Miller, Funk, Goldman, Meisner, & Wu, 2005). We believe RADAR's design makes it less likely than most systems to suffer from these problems. Users can maintain basic control by overriding RADAR's display choices (see Figure 1). Mode errors are unlikely because all automatic updates involve a change of display, which the user should notice. Trust in the system should be high as RADAR learns to approximate a user's desired display preferences, rather than prescribe what the user should view. Finally, RADAR can be incrementally deployed with increasing rates of automatization over time.

All the current studies trained variants of RADAR models from data collected under purely manual conditions. The results of Studies 1-3 demonstrate that people perform differently with RADAR operating. Thus, future work will involve training RADAR models online so that RADAR and human operators can co-evolve.

Acknowledgements

This work was supported by AFOSR grant FA9550-04-1-0226 and NSF CAREER grant #0349101 to the first author.

References

Albrecht, D. W., Zukerman, I., & Nicholson, A. E. (1998). Bayesian models for keyhole plan recognition in an adventure game. *User Modeling and User-Adapted Interaction*, *8*(1-2), 5–47.

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., et al. (1999). *LAPACK users' guide* (Third ed.). Philadelphia, PA: Society for Industrial and Applied Mathematics.

Cleeremans, A. (1993). *Mechanisms of implicit learning: Connectionist models of sequence processing*. Cambridge, MA: MIT Press.

Goodman, B. A., & Litman, D. J. (1992). On the interaction between plan recognition and intelligent interfaces. *UMUAI*, *2*, 83-115.

Gureckis, T., & Love, B. C. (2007). Behaviorism reborn? statistical learning as simple conditioning. *Proceedings of the Annual Meeting of Cognitive Science Society*.

Horvitz, E., & Barry, M. (1995). Display of information for time-critical decision making. In *Proc. of conf. on uncertainty in ai* (p. 296-305).

Mäntyjärvi, J., & Seppänen, T. (2002). Adapting applications in mobile terminals using fuzzy context information. In *Mobile hci* (p. 95-107).

McRae, K., Spivey-Knowlton, M. J., & Tanenhaus, M. K. (1998). Modeling the influence of thematic fit (and other constraints) in on-line sentence comprehension. *Journal of Memory and Language*, *38*, 283-312.

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

Miller, C., Funk, H., Goldman, R., Meisner, J., & Wu, P. (2005). Implications of adaptive vs. adaptable UIs on decision making: Why automated adaptiveness is not always the right answer. In *Proc. of the 1st inter. conf. on augmented cognition.*

Saffran, J. R. (2003). Statistical language learning: Mechanisms and constraints. *Current Directions in Psychological Science*, *12*, 110-114.

Schneider-Hufschmidt, M., Kühme, T., & Malinowski, U. (1993). *Adaptive user interfaces: Principles and practice*. North-Holland.

St. John, M., Smallman, H. S., & Manes, D. I. (2005). Assisted focus: Heuristic automation for guiding users' attention toward critical information. In *Proc. of the 1st inter. conf. on augmented cognition.*

Yi, W., & Ballard, D. (2006). Behavior recognition in human object interactions with a task model. In *AVSS* (p. 64-64).