A Demonstration of the RADAR Personal Assistant

Andrew Faulring, Brad Myers, Ken Mohnkern, and Michael Freed[†]

School of Computer Science, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
{faulring, bam, kem}@cs.cmu.edu

[†]SRI International
333 Ravenswood Avenue, Menlo Park, CA 94025
freed@ai.sri.com

Abstract

Email clients were not designed to serve as a task management tools, but a high volume of task-relevant information in email leads many people to use email clients for this purpose. Such usage aggravates a user's experience of email overload and reduces productivity. Prior research systems have sought to address this problem by experimentally adding task management capabilities to email client software. RADAR (Reflective Agents with Distributed Adaptive Reasoning) takes a different approach in which a software agent acts like a trusted human assistant. Many RADAR components employ machine learning to improve their performance. Human participant studies showed a clear impact of learning on user performance metrics.

Introduction

Email clients are not typically designed to help people track and manage tasks, even though they serve as critical conduits for task-relevant information. This has been shown to be an important contributor to *email overload* with negative impact on work performance (Dabbish and Kraut 2006). Research systems, such as TaskMaster (Bellotti *et al.* 2003; Bellotti *et al.* 2005), have experimented with incorporating task management capabilities into email clients, producing hybrid applications with both message-centric and task-centric views. The RADAR system takes an alternative approach in which email and task management functions are kept separate, but are coordinated by a personalized software assistant with capabilities and roles modeled on those of a trusted human assistant.

RADAR components identify tasks contained within email messages and provide task management assistance, such as prioritization advice and neglected task warnings. Once a user is ready to carry out a task, RADAR also provides substantial task execution assistance. Most system components learn to improve their provided assistance by watching the user, a capability that is especially useful for exploiting learning from expert users to help novices.

System Overview

The RADAR system consists of three sets of components. The first set of components, MEssage-TAsk Linking (METAL), analyzes messages to identify task-related content and extracts task parameters. The Email Classifier treats task identification as a text classification problem and uses a regularized logistic regression algorithm to determine whether an email contains zero, one, or more of eight known task types (Yang *et al.* 2005). Additionally, an NLP pipeline of annotators attempt to extract task-relevant parameters from the email text, many using the trainable MinorThird system (Cohen 2004). The Task Manager provides services for defining tasks, specifying intra-task dependencies, instantiating tasks, managing task execution status, and recording user actions and task results. The Action[1] List displays the existing tasks in a task-centric to-do style interface. For email-derived tasks, the Action List displays both the task details and the source email, which allows the user to check and, if necessary, correct any errors made by METAL components. The user can also easily create tasks that the Email Classifier missed.

The second set of components, Multi-task Coordination Assistance (MCA), provides learning-enabled task management support to the user. The Task Prioritizer suggests an ordering for performing tasks based upon observations of the order in which experts perform similar tasks. As time progresses the Task Prioritizer updates the suggested ordering to reflect time-dependent factors. The Attention Manager takes a more assertive role by alerting the user when high-priority tasks have not been completed, based upon an analysis of user tolerance for interruptive reminders (Smailagic *et al.* 2007). The Task Shedding Advisor, currently under development, computes what tasks to drop when the expected workload is projected to exceed the available time. The results of both the Task Prioritizer and Task Shedding Advisor will be combined to form a high-quality schedule that minimizes costs of non-completed tasks due to time constraints.

The third set of components are Powertools, each an application program for carrying out a particular kind of task as might appear on the Action List. Powertools incorporate learning-enabled assistive capabilities that make use of METAL-supplied information to determine situationally correct assistance. The Virtual Information Officer (VIO) Powertool assists with information retrieval and update tasks that occur when human assistants are asked to lookup or update data contained within an information system

[1] Usability testing revealed that the term *action* is more understandable to users than *task*, so the former term is used throughout the user interface

(Zimmerman *et al.* 2007). Processing a request typically involves reading the email and then locating, completing, and submitting a form corresponding to the expressed intent. VIO uses machine learning algorithms to suggest the appropriate form and fill in its fields. The SpaceTime Planner (STP) and the CMRadar Powertools are used to schedule complex events (such as a conference). STP incorporates a new optimization algorithm designed for mixed-initiative allocation with uncertainty (Fink *et al.* 2006). CMRadar learns regularities in the environment, such as the likelihood of success in seeking new resources, that suggest ways to relax STP planning problems and thereby produce better solutions (Oh and Smith 2004). The Briefing Assistant Powertool assists users in authoring status reports that summarize what has and has not yet been accomplished (Kumar *et al.* 2007).

Evaluation

As a multi-year project under DARPA's Personalized Assistant that Learns (PAL) program, RADAR is evaluated annually to measure machine learning research progress with particular focus on *learning in the wild*: learning based on passive observations or natural interaction with a user. A brief overview of the evaluation follows; a full description of the test design and evaluation procedures is described elsewhere (Steinfeld *et al.* 2007). The participant's overall task is to take over organizing a fictional academic conference, substituting for an incapacitated organizer. Participants are instructed to go through the organizer's email inbox (119 messages) to learn what else needs doing and come as close as possible to completing the remaining tasks. Participants are given approximately two hours of instruction, including some hands-on training using the RADAR system, followed by two hours of testing. In one condition, participants use a version of RADAR that had undergone a period of simulated use, allowing learning mechanisms to train the system with the goal of improving performance. We refer to this condition as RADAR + Learning ("Radar +L"). In a second condition, participants use an untrained version of RADAR ("RADAR −L") to provide a baseline measure of performance. This design shows how well RADAR's machine learning assists a participant independent of any engineered functionality or knowledge. An overall scoring function calculates performance on the conference planning task, yielding a score between 0.000 and 1.000. Results (see table below) show that, for the most recent test (Year 3), learning algorithms improved user performance by 32%, with the contribution of learning increasing by 41% from Year 2 to Year 3.

Condition	**Year 2**		**Year 3**	
	N	**Score**	**N**	**Score**
RADAR −L	18	0.509	42	0.534
RADAR +L	18	0.630	32	0.705

Conclusion

The RADAR system represents a new approach to handling email overload: a trusted human assistant that aids a user with both task management and task execution.

Acknowledgments

The authors thank the numerous faculty, staff, and students at Carnegie Mellon University, SRI International, and elsewhere, who have worked to design and build RADAR. This material is based upon work supported by the Defense Advanced Projects Research Agency (DARPA) under Contract No. FA8750-07-D-0185.

References

Bellotti, V. *et al.* 2003. Taking Email to Task: The Design and Evaluation of a Task Management Centered Email Tool. *Proc. of CHI*, 345–352. ACM Press.

Bellotti, V. *et al.* 2005. Quality Versus Quantity: E-Mail-Centric Task Management and Its Relation With Overload. *Human-Computer Interaction* 20(1/2): 89–138.

Cohen, W.W. 2004. Minorthird: Methods for Identifying Names and Ontological Relations in Text using Heuristics for Inducing Regularities from Data. http://minorthird.sourceforge.net/.

Dabbish, L.A.; and Kraut, R.E. 2006. Email Overload at Work: An Analysis of Factors Associated with Email Strain. *Proc. of CSCW*, 431–440. ACM Press.

Fink, E. *et al.* 2006. Scheduling with Uncertain Resources: Search for a Near-Optimal Solution. *Proc. of IEEE SMC*, 137–144. IEEE Press.

Kumar, M. *et al.* 2007. Summarizing Non-textual Events with a 'Briefing' Focus. *Proc. of RIAO*, Centre De Hautes Etudes Internationales D'Informatique Documentaire.

Oh, J.; and Smith, S.F. 2004. Learning User Preferences in Distributed Calendar Scheduling. *Proc. of PATAT*, 35–50.

Smailagic, A. *et al.* 2007. Task Interruptibility Analysis. Technical Report, Institute for Complex Engineered Systems, Carnegie Mellon University, Pittsburgh, PA.

Steinfeld, A. *et al.* 2007. Evaluation of an Integrated Multi-Task Machine Learning System with Humans in the Loop. *Proc. of PerMIS*, NIST.

Yang, Y. *et al.* 2005. Robustness of Adaptive Filtering Methods in a Cross-Benchmark Evaluation. *Proc. of SIGIR*, 98–105. ACM Press.

Zimmerman, J. *et al.* 2007. VIO: A Mixed-initiative Approach to Learning and Automating Procedural Update Tasks. *Proc. of CHI*, 1445–1454. ACM Press.