CMRadar: A Personal Assistant Agent for Calendar Management

Pragnesh Jay Modi, Manuela Veloso, Stephen F. Smith, Jean Oh
Department of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213
{pmodi,mmv,sfs,jeanoh}@cs.cmu.edu

Introduction

One of the more compelling visions for agents research is the development of "personal assistant agents" that are tasked with making people and organizations more efficient by autonomously handling routine tasks on behalf of their users (Maes 1994) (Mitchell *et al.* 1994) (Chalupsky *et al.* 2001). Most recently, several researchers including ourselves have embarked on a large research project, called The Radar Project (Radar 2004), whose overall goal is to develop a personalized agent that is able to assist its user in a wide range of everyday tasks. Within this larger project, we are concerned with the more focused task of managing a user's calendar. While isolated aspects of calendar management have been investigated before (Sen & Durfee 1998), in this paper, we present CMRadar, a *complete* agent with capabilities ranging across the full spectrum of calendar management, from natural language processing of incoming scheduling-related emails, to making autonomous scheduling decisions, to negotiating with other users, to user interfacing and visualization. Although many research issues remain, we believe CMRadar is the first end-to-end agent for automated calendar management.

A key contribution of the design of CMRadar is the specification of a basic representation, called a Template, for communicating calendar scheduling related information. The Template data structure is used as the language for the communication between the components in CMRadar and as the "glue" that binds them together. In addition, Templates are also used to normalize unformatted natural language emails into a machine readable format. We offer the Template data structure as a flexible approach to the general design of a meeting scheduling agent.

The CMRadar architecture contributes a modular design in which the core scheduling functions of the agent are separated from the multiagent aspects of calendar management. Rather than an approach that tightly couples schedule optimization and negotiation, CMRadar has a separate Manager component which handles the sending and receiving of messages from other agents and more generally, manages the negotiation with others. The Manager then communicates via Templates with a separate Scheduler component that handles the core optimization problems. We found that this modular architecture facilitates the integration of existing scheduling systems and indeed, a core component of CMRadar is the Ozone scheduler (Smith, Lassila, & Becker 1996) originally designed for and used in several real-world logistics planning domains.

The primary underlying emphasis of the Radar project is to learn to improve performance, adapt to unexpected situations and to customize to different users. The emphasis on learning is reflected in our design of the CMRadar architecture in which all components read and write data to a central knowledge base that can be used by a separate learning process to provide feedback to the decision making components (see Figure 1). Indeed, it is the need to collect real-world data to support learning that drives our development of a complete end-to-end agent.

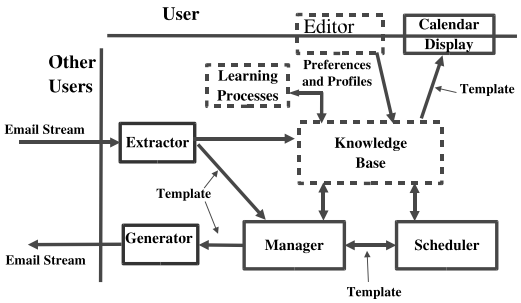

Figure 1: Architecture of a single user's CMRadar

The CMRadar Agent

Our CMRadar is developed as a personalized agent that interacts with other users or agents. Figure 1 shows the overall architecture of CMRadar with its functional modules. Dotted lines represent components not yet implemented. We present two main modules in detail in this paper in the coming sections, namely the Manager and the Scheduler. We now overview the complete ar-

chitecture, briefly describing each of the modules.

Extractor: We assume that multiagent interaction in calendar meeting scheduling occurs through email message exchange. The Extractor is responsible for parsing email messages into a *template* normalized format representing the meeting request or reply to a request. The email messages can be sent directly by other Radar agents or by users in natural language. We have followed research on applying state-of-the-art natural language parsing techniques,[1] as well as successfully defining and applying special purpose parsing rules for language specific to meeting scheduling.[2]

Manager: Calendar management is in its essence a multiagent problem as meetings involve more than one person. The Manager module in CMRadar explicitly handles the multiagent aspects of calendar management, including flexible negotiation with other agents and control of email threads. Meeting scheduling is a complex process dependent on many factors, and different users schedule meetings with other users according to many different strategies. We view this variety of possible multiagent interactions similar to a *playbook* approach that we have previously developed in robot soccer (Bowling, Browning, & Veloso 2004). The Manager can represent and reason about several different multiagent (team) strategies and learn to select the ones that are more effective when interacting with other specific agents.

Scheduler: The core task of meeting scheduling involves determining which times for the meetings. The Scheduler module in CMRadar handles all the time analysis. It receives (or initiates) a specific request for a meeting and returns the user's time availability by considering the user's preferences and its calendar with different kinds of commitments. Calendar management is handled by the Scheduler under a rich set of soft and hard constraints, and agents can reason about truthfully and rationally about their preferences towards optimizing the general social welfare. [3]

Calendar Data and Display: A human user is used to maintaining a calendar using the existing available COTS calendars. Our Calendar Data and Display module aims at having CMRadar use the same calendar programs. The current system is integrated with MS Outlook as shown in Figure 2. [4]

Learning Although not yet part of the system, learning is necessary because obtaining ill-structured user preferences and customizing behavior to different users by hand is infeasible. The agent must acquire much of its required knowledge about its specific user over time through experience. The current CMRadar, as we present in this paper, is the base step towards a complete CMRadar agent which will be truly a learning agent.

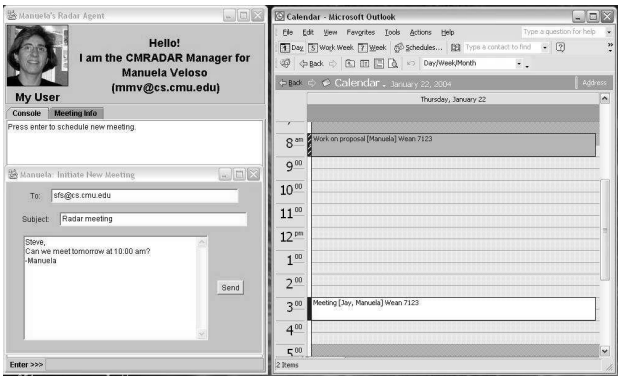

Figure 2: A single user's CMRadar integrated with MS Outlook

Acknowledgments

This work is supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHC030029. We thank the following people for their work on CMRadar: Andrew Faulring, Brad Meyers, Kerry Hannan, Lawrence Lee, Akiva Leffert, Elisabeth Crawford, Donna Gates, Benjamin Han, and Lori Levin.

References

Bowling, M.; Browning, B.; and Veloso, M. 2004. Plays as effective multiagent plans enabling opponent-adaptive play selection. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS'04)*.

Chalupsky, H.; Gil, Y.; Knoblock, C.; Lerman, K.; Oh, J.; Pynadath, D.; Russ, T.; and Tambe, M. 2001. Electric elves: Applying agent technology to support human organizations. In *Proceedings of Innovative Applications of Artificial Intelligence Conference*.

Maes, P. 1994. Agents that reduce work and information overload. *Communications of the ACM* 37(7).

Mitchell, T. M.; Caruana, R.; Freitag, D.; McDermott, J.; and Zabowski, D. 1994. Experience with a learning personal assistant. *Communications of the ACM* 37(7):80–91.

Radar. 2004. "www.radar.cs.cmu.edu".

Sen, S., and Durfee, E. H. 1998. A formal study of distributed meeting scheduling. In *Group Decision and Negotiation*, volume 7, 265–289.

Smith, S.; Lassila, O.; and Becker, M. 1996. Configurable, mixed-initiative systems for planning and scheduling. In Tate, A., ed., *Advanced Planning Technology*. Menlo Park: AAAI Press.

[1] We thank Donna Gates, Lori Levin, and Benjamin Han for their NLP work.

[2] We thank Kerry Hannan for her NLP work.

[3] We thank Elisabeth Crawford for her initial work on addressing this problem under a game theoretical approach.

[4] We thank Andrew Faulring and Brad Meyers for the integration of CMRadar with MS Outlook.