Emma: An Event Management Assistant

P. M. Berry, T. Donneau-Golencer, K. Duong, M. Gervasio, B. Peintner, N. Yorke-Smith
Artificial Intelligence Center, SRI International, Menlo Park, CA 94025, USA
{berry,donneau}@ai.sri.com, khang.duong@sri.com, {gervasio,peintner,nysmith}@ai.sri.com

Abstract

Emma is a personalized calendar management assistant. It integrates commercial calendaring tools with state-of-the-art algorithms for scheduling, negotiation, and personalization. Careful design is intended to produce a user experience that reduces effort in day-to-day life. Designed to learn scheduling preferences over time, Emma is in deployment at our organization. In this extended abstract we describe the motivation, design, deployment and evaluation of the Emma system.

Introduction

Ever received that email, "When are you available next week?" Ever struggled to book a conference room with an electronic reservation system? Ever wished there was a better way of negotiating meetings than by email avalanche?

Emma is a personalized calendar management assistant. Emma is designed to help its user handle email meeting requests, reserve venues, and schedule events. It interfaces with commercial enterprise calendaring platforms, and it operates seamlessly with users who do not have Emma. Emma is designed to learn scheduling preferences, adapting to its user over time. The system is in initial deployment at our organization.

Our work to develop Emma is part of the *Personalized Time Management* (PTIME) thrust of the *CALO* project. CALO (Cognitive Agent that Learns and Organizes) is a large-scale effort to build an adaptive, interactive cognitive assistant situated in the office environment (Myers *et al.* 2007). A CALO agent is designed to support its user in various ways: project and task management, information organization, and meeting preparation and summarization.

Over several years we have explored the trade-offs between the competing forces of representation, reasoning, personalization, and negotiation (Berry *et al.* 2005; 2006; 2007). These experiences led to two objectives for Emma. First, learning from the numerous user studies in the literature (e.g., (Palen 1999; Weber and Yorke-Smith 2008)), we want Emma to be of tangible value to its user. Emma should make day-to-day time management more simple. Second, learning from the difficulty of evaluating the efficacy of machine learning techniques, primarily due to lack of data, we want the deployed Emma to collect significant data from the user population. The first objective of a deployed, usable

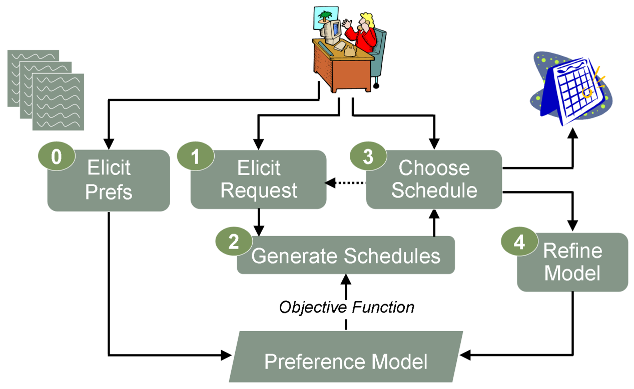
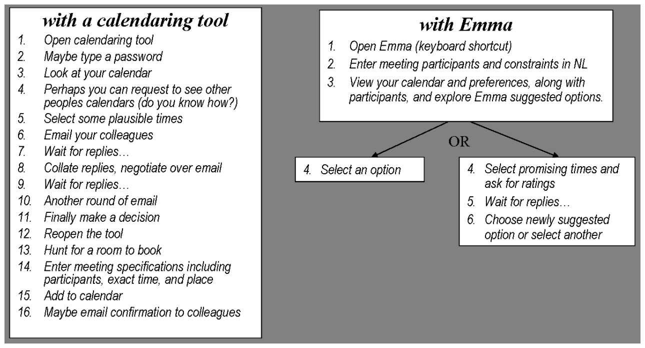


Figure 1: Scheduling with Emma.

system supports the second objective of evaluating the benefit of learning, which in turn helps us develop a more useful and usable system.

At the core of Emma is a constraint-based scheduler. Given a specification of a new event to be scheduled, including desired time window, duration, location, and participants, the scheduler generates a ranked set of candidate options. It can consider moving existing events that belong to the user: for instance, changing the location of a meeting that the user has previously organized. Preferences arise directly from the participants (e.g., "I prefer morning meetings, except on Monday"), from the event request ("As near to my office as possible"), and from Emma's learned knowledge (e.g., the user attaches more importance to the participants than the location). Relaxed solutions must be considered in the case of an over-constrained problem (e.g., the requested venue is not available within any of the requested times). The scheduler reasons over the resulting set of hard and soft constraints.

While both commercial (e.g., Microsoft Outlook and Exchange Server, Google Calendar) and open source (e.g., Yahoo! Zimbra, Mozilla Sunbird) calendaring systems abound to support centralized solutions within an institution, they leave the task of choosing a meeting time up to the user, thus avoiding the need to actively reason about constraints and user preferences. As an advanced example, *Meeting Maker* (PeopleCube, Inc. 2008) supports the user in selecting a time



with a calendaring tool

1. *Open calendaring tool*
2. *Maybe type a password*
3. *Look at your calendar*
4. *Perhaps you can request to see other peoples calendars (do you know how?)*
5. *Select some plausible times*
6. *Email your colleagues*
7. *Wait for replies…*
8. *Collate replies, negotiate over email*
9. *Wait for replies…*
10. *Another round of email*
11. *Finally make a decision*
12. *Reopen the tool*
13. *Hunt for a room to book*
14. *Enter meeting specifications including participants, exact time, and place*
15. *Add to calendar*
16. *Maybe email confirmation to colleagues*

with Emma

1. *Open Emma (keyboard shortcut)*
2. *Enter meeting participants and constraints in NL*
3. *View your calendar and preferences, along with participants, and explore Emma suggested options.*

4. *Select an option*

OR

4. *Select promising times and ask for ratings*
5. *Wait for replies…*
6. *Choose newly suggested option or select another*

Figure 2: Comparison of a typical meeting negotiation process with Emma's process.

by graphically depicting the availability of participants. The strength of this and other group calendaring systems is integration with institutional workflow.

By contrast, Emma is a calendaring assistant (Mitchell *et al.* 1994). Emma can be configured to work with any iCal-compatible calendar server. It supports information gathering, streamlines negotiation (acting as its user's proxy where authorized), and employs scheduling to propose options for new events. Emma is designed to complement, not replace, the user's preferred calendar tools.

While prior academic research (e.g., (Mitchell *et al.* 1994; Sen *et al.* 1997; Faulring and Myers 2005; Brzozowski *et al.* 2006)) has looked at one or more of the three aspects of representation (modelling and eliciting), personalization (especially, learning), and reasoning (scheduling and negotiation), the resulting systems have rarely sought to encompass all three. The closest to the approach we have taken are systems such as *CMRadar* (Modi *et al.* 2004). The larger task of automated planning of one's time and tasks is addressed by systems such as *SelfPlanner* (Refanidis and Alexiadis 2008).

Emma in Practice

Figure 1 shows schematically an interaction with Emma. Emma elicits scheduling preferences (step 0); elicits an event request (step 1); computes candidate schedules (possibly relaxations) in response to the request, and displays a subset of the candidates to the user as options (step 2); and accepts the user's choice of the desired schedule option (step 3). Based on which option the user chooses for each request among the presented options, the learning module in Emma updates the parameters of the preference model instance (step 4). The updated model becomes the basis of reasoning over candidate schedules for the next request.

Emma supports a variety of scheduling and calendaring processes that we identified through user studies. Figure 2 compares a typical Emma process with common convention. Emma supports its user in quickly and efficiently adding simple events to her calendar, manipulating events on the calendar, and deriving information from the calendar (e.g., availability) with minimal effort. Emma also serves its user in more complex processes that involve coordinating groups of people, negotiating meeting times with busy, over-constrained colleagues, rescheduling events, ensuring resource availability, and booking resources.

Figure 3 shows the most common Emma interface. The user invokes Emma by icon or by keyboard shortcut. She can view and directly manipulate her calendar, schedule a new event or reschedule an existing event, and pull up a summary of her availability. For the simple case of an event that the user wishes to add to her calendar, she can simply specify and add it; Emma suggests time and location if the user does not have them in mind. For the complex case of arranging an event where there is no ideal solution (an over-constrained meeting request), Emma generates options and helps the user weigh them.

Time management is an intensely personal undertaking, and scheduling events varies between individuals. The best solution for an over-constrained meeting request depends on the user: one person prefers to reschedule an existing meeting, another prefers times outside the specified window, while still another prefers to omit a participant. An instant messaging feature allows the user to request the opinions of others; Emma automatically falls back to email for communication to and from individuals without Emma.

Architecture and System Design

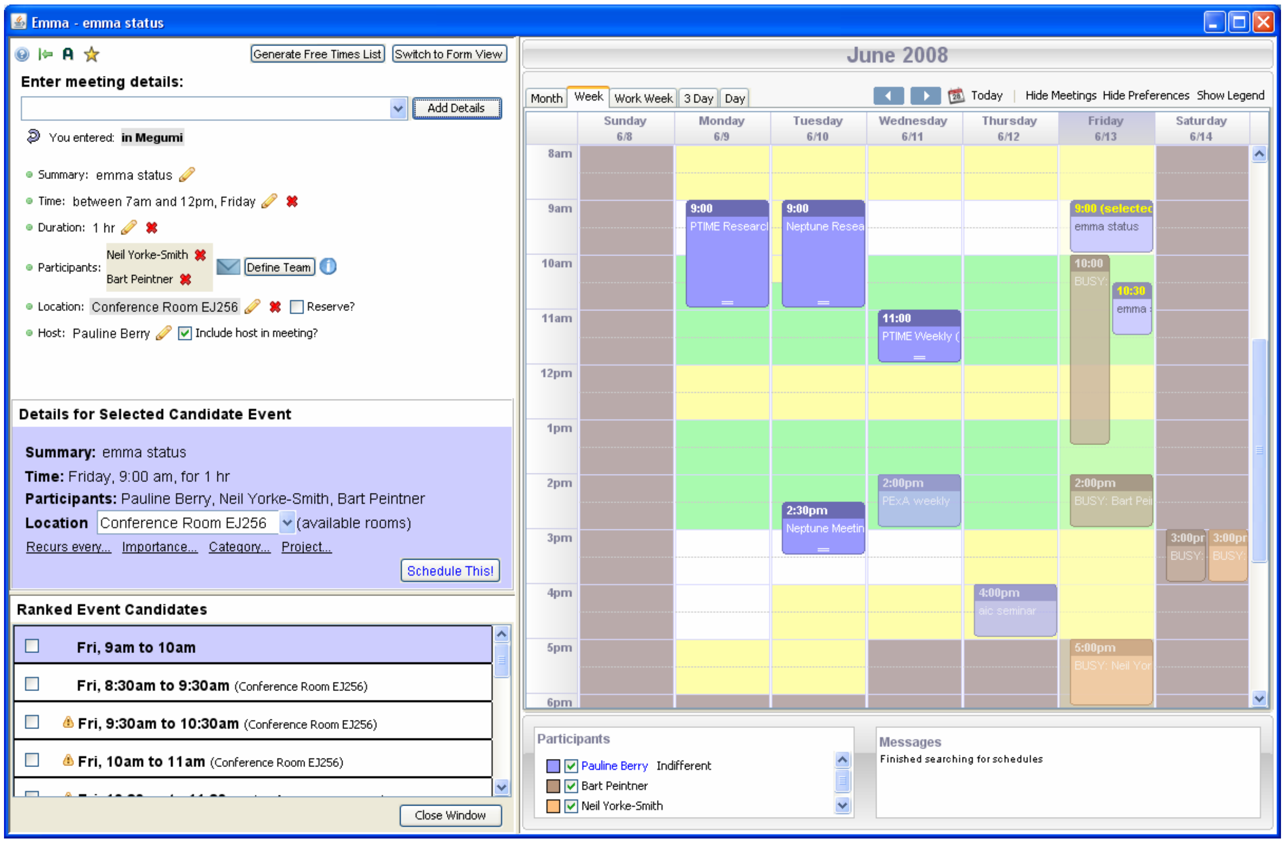
Four main components compose Emma: the user interface (UI), a calendar proxy, a constraint reasoner, and a community of learners. Shown in Figure 3, the UI allows both restricted natural language and direct manipulation.

Second, the *calendar proxy* provides a common connection between Emma and a variety of calendar servers. Currently, the proxy implements interfaces to Google Calendar, Sun Calendar, and Yahoo! Zimbra. Emma can thus simultaneously manage calendars from multiple sources, which is key for many of our users who maintain personal calendars in addition to their work calendars.

Underlying the constraint reasoner and learners is a preference model. Three forces shape the model. First, it must be expressive enough to capture the user's preferences. Second, it must be simple enough to allow efficient search for preferred scheduling options. Third, it must be limited in the number of parameters to be learned, given the importance of learning quickly from a limited amount of user feedback.

The aim of the *constraint reasoner* in Emma is to generate scheduling options in response to new details and constraints entered by the user. For example, if the user enters "next fri afternoon with Bart and Neil" in the UI, Emma's scheduler first constructs a set of soft constraints. The scope of these constraints are temporal variables that represent the start time and duration of the candidate event, and the existing events of the user and the requested participants. In addition, discrete variables are created to model the location of each event and the inclusion of each participant.

The soft constraints allow all aspects of the user's request to be relaxed by the scheduler in cases where the request cannot be satisfied. The constraints form a *Multi-Criteria Disjunctive Temporal Problem with Preferences* (Moffitt *et al.* 2006); each constraint is assigned to one or more criteria:



Enter meeting details:

You entered: **in Megumi**

- Summary: emma status
- Time: between 7am and 12pm, Friday
- Duration: 1 hr
- Participants: Neil Yorke-Smith, Bart Peintner
- Location: Conference Room EJ256 ☐ Reserve?
- Host: Pauline Berry ☑ Include host in meeting?

Details for Selected Candidate Event

Summary: emma status
Time: Friday, 9:00 am, for 1 hr
Participants: Pauline Berry, Neil Yorke-Smith, Bart Peintner
Location Conference Room EJ256 (available rooms)
Recurs every... Importance... Category... Project...

Ranked Event Candidates

- ☐ **Fri, 9am to 10am**
- ☐ **Fri, 8:30am to 9:30am** (Conference Room EJ256)
- ☐ ⚠ **Fri, 9:30am to 10:30am** (Conference Room EJ256)
- ☐ ⚠ **Fri, 10am to 11am** (Conference Room EJ256)

Participants

☑	Pauline Berry	Indifferent
☑	Bart Peintner	
☑	Neil Yorke-Smith	

Messages

Finished searching for schedules

Figure 3: Emma interface. LHS: event specification, current option, candidate options. RHS: calendar with relevant events and preferences shown, and current option.

requested time window, *existing event time*, *user preference*, and four other criteria that represent user-understandable dimensions in our preference model. The constraint reasoner scores each schedule that it generates by aggregating the level of satisfaction of the constraints for each criterion into a single value. These values are in turn aggregated by means of a Choquet integral with learned weights on each term in that integral. Details of this scheme can be found elsewhere (Berry *et al.* 2007).

Once a set of optimal — or near-optimal, in some cases — scheduling options is found, a subset is selected for asynchronous presentation to the user in the UI. At any time, the user can select an option, or create options manually by interacting with the calendar view. The set of options presented is chosen to balance three considerations: (1) preferred solutions according to the constraint reasoner's solving, (2) minimal violation of the constraints (since the current learned preference model may differ from the user's 'true' model), and (3) overall diversity to stimulate the user's understanding of feasible options.

The final component is a *community of learners*. The learners are given training instances by and queried by both the UI and the constraint reasoner. Before searching for candidate scheduling options, the constraint reasoner queries the *schedule preference learner* for the current instantiation of the preference model. After the user selects an option, all options are sent to the schedule preference learner as a training instance. Learning is therefore unobtrusive (obtaining implicit feedback from the user's normal working practice) and online (learning continually). The user has the ability to provide explicit feedback, for instance by rating an option.

A second learner predicts the importance of an event based on features such as the participants, associated project (if any), and category (if any). This learner is currently used to provide suggested values for event importance. It also serves as a feature for the event-acceptance learner, which will provide default responses when the user is presented with a meeting request from another user. A third learner predicts which room or location is appropriate for an event.

Emma is implemented in Java. All components reside in the same process but are fully independent, and can be incorporated or accessed by external components.

Design Process and Deployment

Emma's design reflects the experiences gleaned with PTIME. An iterative design process (Weber and Yorke-Smith 2008) interleaved user studies, prototyping, interaction design, and staged deployment.

The iterative deployment followed human-computer interaction principles to evolve a system by evaluating both the usability of the system and the efficacy of the learning technology. Following initial design studies and cognitive walkthroughs, we conducted a set of think-aloud user studies with an Emma prototype. After a further round of design, a second set of user subjects attended a carefully controlled training session, followed by the completion of standardized questionnaires and a period of use under normal working conditions.

The results of these two stages produced the version of the system for general deployment within our organization. In contrast to the resource-intensive CALO system as a whole, Emma is a lightweight, stand-alone Java application. Important for adoption at our institution, Emma operates under Windows and Mac OS.

Evaluation and Future Developments

The evaluation of Emma is aimed at assessing the two factors that drive the system's development: usability and user acceptance rate, and the performance of scheduling and learning algorithms and overall efficacy of the system. The deployed release includes extensive logging mechanisms for continuous, automated collection of information on the use of Emma, the performance of the constraint reasoning and learning, and the response times for negotiation instances. In addition, the user is periodically presented with a short, optional questionnaire to collect information on user acceptance and perception. Data from this evaluation will be analyzed after 12 weeks of deployment.

Future enhancements to the system include extension to the calendar proxy, dynamic reminder generation, more complex and automated email-enabled interactions, and better explanation of scheduling options to the user (Bresina and Morris 2006). Other developments include refining the adjustable autonomy capabilities to learn when an event might be accepted or rejected on behalf of the user, when a request for rating event options could be filled in automatically, and when and to whom a message should be sent when a meeting is overrunning or finishing early.

In the legacy of prior applications of AI in the time management domain, Emma retains the spirit of pioneering assistive agents such as *Calendar Apprentice* (Mitchell *et al.* 1994). Scientifically, Emma is an investigation into machine learning to inform scheduling. Pragmatically, Emma is an assistant to make users' day-to-day lives easier. We maintain that usability and UI design are prerequisites to enable the behind-the-scenes scheduler. The positive response from users to the initial deployment of Emma points to the benefits of the appropriate infusion of scheduling in this domain.

Acknowledgments We thank all the members of the CALO project, especially the PExA team, and gratefully acknowledge the alpha and beta users of Emma. Thanks to Mark Plascencia and Aaron Spaulding. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-07-D-0185/0004. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA, or the Air Force Research Laboratory.

References

P. M. Berry, M. T. Gervasio, T. E. Uribe, M. E. Pollack, and M. D. Moffitt. A personalized time management assistant: Research directions. In *Proc. of AAAI 2005 Spring Symposium on Persistent Assistants: Living and Working with AI*, pages 1–6, Stanford University, CA, March 2005.

P. M. Berry, K. Conley, M. Gervasio, B. Peintner, T. Uribe, and N. Yorke-Smith. Deploying a personalized time management agent. In *Proc. of AAMAS'06 Industrial Track*, pages 1564–1571, Hakodate, Japan, May 2006.

P. M. Berry, M. Gervasio, B. Peintner, and N. Yorke-Smith. Balancing the needs of personalization and reasoning in a user-centric scheduling assistant. Technical Note 561, Artificial Intelligence Center, SRI International, February 2007.

J. L. Bresina and P. H. Morris. Explanations and recommendations for temporal inconsistencies. In *Proc. of 5th Intl. Workshop on Planning and Scheduling for Space (IWPSS'06)*, Baltimore, MD, October 2006.

M. Brzozowski, K. Carattini, S. R. Klemmer, P. Mihelich, J. Hu, and A. Y. Ng. groupTime: preference-based group scheduling. In *Proc. of CHI'06*, pages 1047–1056, 2006.

A. Faulring and B. A. Myers. Enabling rich human-agent interaction for a calendar scheduling agent. In *Proc. of CHI'05*, pages 1367–1370, 2005.

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

P. J. Modi, M. M. Veloso, S. F. Smith, and J. Oh. CMRadar: A personal assistant agent for calendar management. In *Proc. of Agent-Oriented Information Systems (AOIS'04)*, pages 169–181, 2004.

M. D. Moffitt, B. Peintner, and N. Yorke-Smith. Multicriteria optimization of temporal preferences. In *Proc. of CP'06 Workshop on Preferences and Soft Constraints (Soft'06)*, Nantes, France, October 2006.

K. Myers, P. M. Berry, J. Blythe, K. Conley, M. Gervasio, D. McGuinness, D. Morley, A. Pfeffer, M. Pollack, and Milind Tambe. An intelligent personal assistant for task and time management. *AI Magazine*, 28(2):47–61, 2007.

L. Palen. Social, individual and technological issues for groupware calendar systems. In *Proc. of CHI'99*, pages 17–24, Pittsburgh, PA, May 1999.

PeopleCube, Inc. Meeting Maker `www.peoplecube.com/products/meetingmaker/`, 2008.

I. Refanidis and A. Alexiadis. SELFPLANNER: Planning your time! In *Proc. of ICAPS'08 Scheduling and Planning Applications woRKshop (SPARK'08)*, Sydney, Australia, September 2008.

S. Sen, T. Hayes, and N. Arora. Satisfying user preferences while negotiating meetings. *Intl. Journal of Human Computer Studies*, 47:407–427, 1997.

J. Weber and N. Yorke-Smith. Time management with adaptive reminders: Two studies and their design implications. In *Working notes of CHI'08 Workshop: Usable Artificial Intelligence*, Florence, Italy, April 2008.