Evaluating User-Adaptive Systems: Lessons from Experiences with a Personalized Meeting Scheduling Assistant

Pauline M. Berry, Thierry Donneau-Golencer, Khang Duong
Melinda Gervasio, Bart Peintner, Neil Yorke-Smith
SRI International, 333 Ravenswood Ave., Menlo Park, California 94025, USA
{firstname.lastname}@sri.com

Abstract

We discuss experiences from evaluating the learning performance of a user-adaptive personal assistant agent. We discuss the challenge of designing adequate evaluation and the tension of collecting adequate data without a fully functional, deployed system. Reflections on negative and positive experiences point to the challenges of evaluating user-adaptive AI systems. Lessons learned concern early consideration of evaluation and deployment, characteristics of AI technology and domains that make controlled evaluations appropriate or not, holistic experimental design, implications of "in the wild" evaluation, and the effect of AI-enabled functionality and its impact upon existing tools and work practices.

Introduction

Artificial Intelligence (AI) technology has much to contribute to personal software tools. Learning and adaptation, for instance, offer prospects for new functionality, together with increased personalization and ease of use. However, development and infusion of AI technology must be supported by adequate evaluation of its efficacy. Without adequate evaluation, reassessment, and redesign, AI risks hindering rather than aiding users. For instance, an adaptive system that behaves erratically, in a way unpredictable to the user, will likely be rejected out of hand. However, designing and conducting suitable and adequate evaluation itself poses key challenges in the domain of personalized systems.

We discuss experiences from evaluating the learning performance of an adaptive personal assistant agent. The problem domain is personal time management: in particular, the problem of arranging meetings and managing one's calendar over time. This paper primarily concerns the *evaluation* of an interactive, adaptive system that learns preferences over an extended period; the system itself, and its performance, are secondary topics. The results from the evaluation are mixed. Even negative results, however, produce lessons that can aid the evaluation of such user-adaptive AI systems. These lessons are the primary contribution of the paper.

The AI technology in our *Personalized Time Management* (PTIME) calendaring assistant is designed not only to help with scheduling meetings, but also to avoid protracted negotiations, simplify the reservation of resources, and advocate for the user's time needs. The enabling AI disciplines are preference modelling and machine learning to capture user preferences, natural language understanding to facilitate elicitation of constraints, and constraint-based reasoning to generate candidate schedules. Human-computer interaction plays a central role.

PTIME is part of a larger project, *Cognitive Assistant that Learns and Organizes* (CALO), aimed at exploring learning in a personalized cognitive assistant. Thus, the primary assessment of PTIME is in terms of its adaptive capabilities, although the system clearly must also be able to assist with time management tasks to provide a context for learning. At the commencement of the project, however, the degree of robustness and usability required to support evaluation was not immediately obvious. Evaluation was focused almost exclusively on the AI, designed to measure performance improvements due to learning within a controlled test environment simulating a period of real-life use. As technologists, we are trained primarily to conduct such "in-the-lab" evaluations but, as we argue in this paper, there are situations where evaluation requires placing the technology into actual use with real users in order to provide a meaningful assessment of the technology.

We introduce the problem domain and summarizing the current design and capability of PTIME, highlighting the potential value of AI technology. We then discuss in detail the design, execution, and outcome of the evaluation of PTIME and relate our experiences in attempting to assess the system's learning capability over several years of the project. We describe the factors that led to the current multi-faceted evaluation with real users in a deployed setting, and the challenges in maturing the research technology sufficiently for that deployment. We conclude with an analysis of our experiences and the lessons learned pertaining to the adequate evaluation of user-adaptive AI-based assistant technology.

Personalized Meeting Scheduling

The vision behind the CALO project was a personal assistant for a busy decision-maker in an office environment, akin to an administrative assistant who can help facilitate routine tasks (SRI International 2009; Myers et al. 2007). Time management—in particular, scheduling meetings in an over-constrained setting—was a natural problem to address. All too often, protracted negotiations occur as potential participants communicate in the semi-transparent world of their own and others' calendars. Valuable time could be saved if

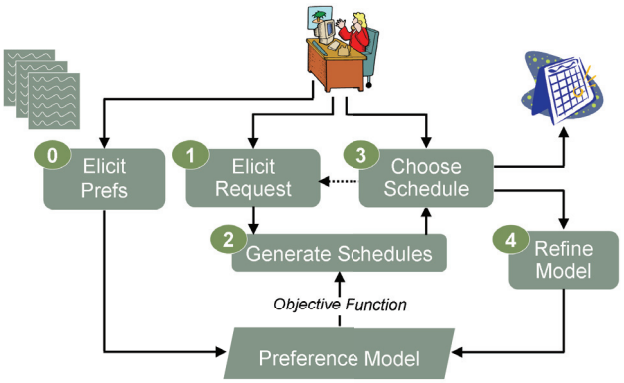

Figure 1: User–system interaction

the effort spent organizing meetings could be reduced. In a world of electronic calendars and advanced AI technology, the prospect of scheduling assistance seems a plausible and desirable application, yet the most common practice for negotiating meetings remains extended email, phone, or instant messaging interactions.

Both commercial (e.g., Microsoft Outlook) and open source (e.g., Yahoo! Zimbra) calendaring systems abound to support centralized solutions within an institution. However, they leave the task of identifying and choosing a meeting option to the user. At best, more advanced tools like PeopleCube's *Meeting Maker* support the user in selecting a time by graphically depicting the availability of participants. Such group calendaring systems are strong in integration with institutional workflow, but they remain tools rather than providing intelligent assistance.

Intelligent, personalized calendaring assistance is difficult because people seldom have a realistic understanding of the preferences they use in practice (Viappiani, Faltings, and Pu 2006), few are willing to keep providing explicit feedback without seeing tangible benefit relatively quickly (Carroll and Rosson 1987), and most are unwilling to spend time training a calendaring system (Weber and Yorke-Smith 2008). As a result, we designed PTIME to be an adaptive system that would learn its user's preferences reasonably quickly and unobtrusively through implicit feedback.

There exist scheduling systems that readily solve the multi-participant, distributed or centralized meeting scheduling problem. However, they suffer from low adoption rates because they fail to account for the intensely personal nature of scheduling, or demand too much control of an important aspect of the individual's life (Palen 1999). The process of managing one's time, the tools employed, and the preferences over events all exhibit considerable variation between individuals. For example, the appropriate solution for an overconstrained meeting request usually depends on the user and the situation: it may be reschedule an existing meeting, choose another time, or omit a participant. We designed PTIME to support multiple means of calendar management while being adaptive to individual preferences.

Providing intelligent scheduling assistance required that we either augment an existing calendaring system or develop a fully-functional, stand-alone assistant. We initially attempted to use Outlook as our calendaring interface, but eventually set aside that paradigm when CALO developed its own calendaring interface, in line with the vision of CALO as a separate, cognitive entity. Following further evolution, PTIME became a lightweight, self-contained system that provides its own user interface.

The PTIME System

PTIME comprises four main components: the user interface, calendar proxy, constraint reasoner, and preference learner. PTIME's *user interface* (UI) lets the user enter scheduling constraints and details using restricted natural language and direct manipulation. The *calendar proxy* provides the ability to connect to a variety of calendar servers, supporting PTIME's ability to manage calendars from multiple sources (e.g., personal and work calendars). The constraint reasoner generates candidate schedules using the current preference model while the preference learner updates this model based on user feedback on the generated options. Berry et al. (2007) provide a detailed system description.

Figure 1 depicts the interactions between the user and the PTIME system during scheduling. Optionally, PTIME (0) may elicit scheduling preferences. Then PTIME (1) elicits an event request, (2) computes preferred candidate schedules (possibly relaxations) in response to the request and presents a subset to the user, (3) accepts the user's selection, and (4) updates the preference model accordingly. Steps 2 and 3 are repeated as necessary, with the system presenting new options after each new detail is entered by the user. The updated model is used in the subsequent scheduling session.

Underlying the constraint reasoner and the learner is a preference model designed to be expressive enough to capture the user's preferences while being simple enough to support tractable reasoning and efficient learning. The *constraint reasoner* generates scheduling options in response to new or revised details and constraints from the user, using the current preference model to generate preferred options. The reasoner translates requests such as *"next fri afternoon with wayne and kim"* into a set of soft constraints and solves a soft constraint problem with preferences. Soft constraints allow all aspects of the user's request, including start time, location, participants and duration, to be relaxed in the case where the request cannot be satisfied.

The *preference learner* interacts with both the UI and the constraint reasoner. In addressing a meeting request, the reasoner queries the learner for the current preference model, and uses the model to generate candidate solutions. After the user selects an option, the UI sends the candidate set to the learner, providing it with the feedback that the selected option is preferred to all the other candidates. Although users have the option to provide explicit ratings on any of the candidates, the learner is designed primarily to learn unobtrusively from implicit feedback.

Evaluating Learning in PTIME

Aspects of PTIME have been under development over the past five years as part of the CALO project (SRI International 2009). During that time, we have attempted a number of evaluations of the technology focused on its adaptive ca-

pabilities, since learning is the major thrust of the CALO project. For various reasons we discuss next, this turned out to be a much harder task than originally anticipated, requiring a rethinking not only of our evaluation methodology but also of the design of PTIME itself and the scheduling assistance capabilities it provides.

Evaluating PTIME as Part of CALO

A primary objective of the CALO project was to stimulate the development of learning technology to support autonomous adaptation in an intelligent assistant. The annual *CALO Test*, started in Year 2, was designed to assess the effects of learning on CALO's performance as a personalized assistant. Patterned after standardized tests such as the SAT for U.S. college admissions, each year's test drew from a library of parametrized questions developed by an independent evaluator. In Years 2 and 3, we relied on the CALO Test to provide an evaluation of PTIME. However, this turned out to be problematic for a number of reasons.

First and foremost, the CALO Test was designed specifically to evaluate CALO as a whole rather than any of the learning modules in isolation. Given the large collection of modules, each year's test could include only a small number of questions pertaining to time management. Second, the data gathered during the week-long critical learning period (CLP) was insufficient for PTIME's learning. In Year 2, participants simply did not schedule enough meetings. In Year 3, they were directed to schedule at least a minimum number of meetings but they ended up scheduling mostly underconstrained meetings. As a result, PTIME did not get data on how they managed trade-offs, which are most indicative of user preferences. Finally, usability and stability issues—with PTIME and with CALO overall—significantly affected user interaction, impacting our ability to gather data.

Between Years 2 and 3, as we began to recognize the importance of usability, we conducted our first user study to investigate calendaring needs and relevant decision factors in the CALO target population of knowledge workers. Eleven subjects participated in the study, based on an in-situ diarying exercise and semi-structured interviews. The results indicated that users desired scheduling capability beyond that of Outlook and perceived the concept of PTIME to be desirable; they also helped identify additional factors affecting users' scheduling decisions, thereby guiding our development of a richer preference model (Berry et al. 2007).

A More Focused PTIME Evaluation

In Year 4, we sought to address the problems with the earlier evaluations using a three-pronged strategy. First, we augmented the CLP with a dedicated training phase during which participants were presented with carefully selected overconstrained scheduling scenarios to force them to think through difficult trade-offs (e.g., for a given scenario, would a shorter or later meeting be more desirable?) and provide PTIME with more meaningful training instances. For the CLP itself, participants were given stronger guidance regarding the number and variety of meetings they should try to schedule. Second, we administered a PTIME-specific questionnaire after the CLP to gather subjective opinion of PTIME functionality, performance, and—the first time this type of data was gathered for any CALO component—usability. One-on-one semi-structured interviews sought to probe trust and acceptance of the system (Glass, McGuinness, and Wolverton 2008). Finally, we attempted a limited deployment of PTIME within our department.

Our changes to the experimental setup were successful in that a significant portion of the data collected now involved users making trade-offs in overconstrained scenarios. However, the learned models arguably still did not truly reflect the participants' scheduling preferences, again for a number of reasons. In the dedicated training phase, although the details of each scheduling scenario were spelled out, it remained up to the participants to internalize the details and pretend as if they had created that scenario. In reality, participants were often forced to make trade-offs in scenarios they would have never created. For example, one scenario specified a meeting to be scheduled for the following Wednesday or Thursday with two other people, and presented the options of choosing a time that was inconvenient for one participant or a Friday meeting time instead. However, since the scenario description did not explain why it required "Wednesday or Thursday" and it involved a hypothetical meeting, it was often difficult for users to decide which option they preferred. Meanwhile, in the CLP, the fact that the participants knew they would not have to actually *attend* the meetings they were scheduling sometimes had a dramatic influence on their perception of what was acceptable (e.g., "Sure, I am perfectly happy with four back-to-back meetings!").

The limited deployment of PTIME was almost entirely unsuccessful for three reasons. First, although much improved over Year 3, the UI, coupled with slow performance and stability issues, continued to hamper usability. Second, the installation was still quite heavyweight, requiring some memory-intensive CALO components, such as the knowledge base. Finally, while PTIME provided advanced scheduling features, it did not provide sufficient advantage in users' simpler, more common calendaring needs to entice them to use it in conjunction with their existing tools.

Toward a Comprehensive Evaluation of PTIME

The evaluations conducted in Years 2, 3, and 4 of the CALO project gave some insights into PTIME's performance and value but yielded insufficient data to evaluate the usefulness of PTIME's adaptive capabilities. In Year 5, the annual CALO Test was not conducted, so we were able to step back and create an evaluation plan specifically for PTIME. Learning from our earlier experiences, we designed a three-stage evaluation with the objective of answering these questions:

- **Is PTIME a usable calendaring assistant?** (user experience)
- **Does PTIME increases task effectiveness?** (objective performance measures, including measures of learning)
- **Is PTIME a potentially useful assistant?** (subjective appraisal and performance measures)

The first stage involved cognitive walkthroughs, think-aloud exercises, and prototyping, and the second stage comprised of a set of structured evaluation sessions. Together,

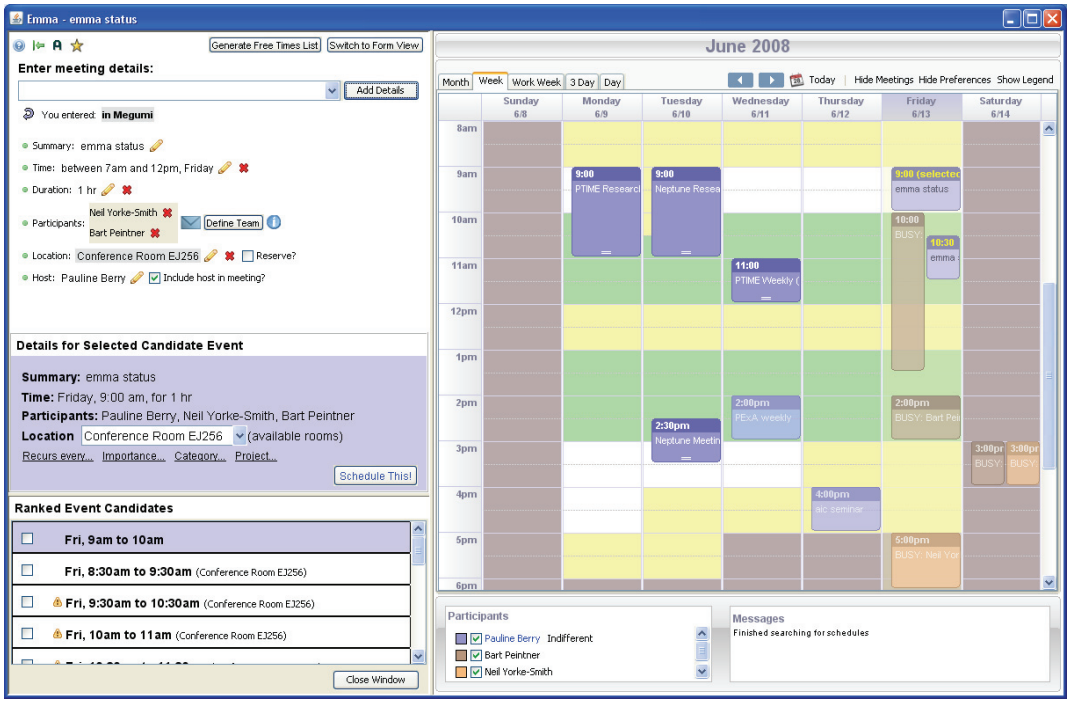

Figure 2: The PTIME interface

these studies focus on gathering data to assess usability and perceived usefulness. Deliberately, we did not attempt to gather quantitative data to assess preference modelling or learning by means of these structured studies. The third stage is a longitudinal study "in the wild". This study, ongoing at the time of writing, is gathering data to assess learning performance as well as long-term perceived usefulness.

Before PTIME could be evaluated "in the wild", the architectural, deployment, and usability issues that hindered previous evaluations had to be addressed. We undertook an iterative development process alongside the first two stages of evaluation, using the results of the user studies, think-aloud exercises, and prototyping to stage alpha, beta, and release deployments (Weber and Yorke-Smith 2008). The result was a lightweight, user-centric version of PTIME called *Emma* (*Event Management Assistant*).

Emma operates independently of the full CALO system; it operates cross-platform on Mac and Windows; and it interfaces with standard calendar servers, such as Yahoo! Zimbra and Google Calendar. Although the separation from CALO limited some functionality (e.g., Emma lacks knowledge of organizational hierarchy), it was clear that PTIME would remain unused if deployment required the full CALO system to be installed. We redesigned the calendar view for usability (Figure 2) and added support for typical calendar functions to address the "easy" cases of scheduling. Learning from the think-aloud exercises, we made the user–system interaction more incremental and redesigned the request specification and option viewing interface to make less cumbersome the "hard" cases of scheduling.

The objective of the structured evaluation sessions was to evaluate Emma empirically in terms of ease of use and perceived usefulness, factors crucial for eventual user acceptance. We recruited 24 participants of varied demographics from our organization and gave them tasks to perform with Emma in a controlled environment. A pre-session questionnaire gathered demographic, habitual, and experiential background. Post-session questionnaires and interviews (1) gathered subjective ratings of mental effort, overall effort, performance success, system understanding, and system trust; (2) identified which types of calendaring tasks users would like help with and whether an assistant along the lines of Emma would help; and (3) gathered subjective ratings of specific Emma capabilities.

The majority of participants thought that Emma was a potentially useful tool, helping them to manage their calendar more quickly and with superior outcome. Almost all subjects liked the concept of "a tool like Emma" (mean 0.92 on a scale $[-3, 3]$, significant by a one-sided t-test against indifference (mean 0) at $p < 0.01$) and "would use it if it was stable" (mean 2.25, significant at $p < 0.001$). Emma was found to be extremely significant in increasing task effectiveness in general (mean 0.85 on a scale $[-2, 2]$), and in terms of perceived speed (mean 0.76) and quality (mean 0.71). It was likewise found easy to use (mean 0.72). However, it is found to be neither easy nor hard to learn; likewise with perceived understanding and control of the system. As expected from the short duration of usage, participants did not have a sense of whether the system was adapting to their preferences. Finally, the participants who had used an earlier version of PTIME expressed strong agreement that Emma was "superior to PTIME Year 4" (mean 1.33).

While these studies on usability and perceived usefulness do not address the issue of adaptation, we firmly believe they are critical to the evaluation of user-adaptive assistants such as PTIME. As we discovered in our earlier technology-

centric evaluations, it is difficult to synthesize scenarios that will effectively elicit real-world behaviour from users within a controlled setting. But evaluating a system "in the wild" requires that a system be usable and useful enough for users to actually want to use it. These studies thus served as prerequisites to our current longitudinal study that will evaluate PTIME's adaptive capabilities.

Evaluating User-Adaptive Systems

Given our experience, it is natural to ask what others have done in similar contexts. Perhaps the largest body of work on user-adaptive systems is in the field of recommender systems (Adomavicius 2005). We discuss how particular characteristics of recommendation tasks affect evaluation, then focus on previous work on using machine learning for adaptive calendaring assistance. We then discuss alternative user studies that we might have conducted and our rationale for PTIME's Year 5 evaluation strategy.

Recommender Systems

To understand what makes some recommender systems easier to evaluate than others, we highlight key dimensions along which their tasks differ and the ramifications on evaluation. The first dimension is the nature of the recommendation task. Most recommender systems focus on the task of *information filtering* (Montaner, Lopez, and de la Rosa 2003): i.e., identifying a preferred subset from a large set of items such as books, music, movies, or news articles, or more complex items such as vacations. In contrast, *generative* tasks (Langley 1999), involves creating candidates in response to each new problem, for example, driving routes (Fiechter and Rogers 2000) or job-shop scheduling repair actions (Miyashita and Sycara 1995). Generative tasks impose the requirement of having a system with which users can interact: in our case, the PTIME calendaring assistant.

Second, in some systems the problem request is static (e.g., a one-shot request to recommend a book). In others, the request requires incremental refinement through interaction with the user and/or other agents (e.g., scheduling a multi-person meeting). However, as we discovered, the complex social and personal factors affecting such interactions cannot always be captured in a controlled environment.

Third, just as the request may be interactive, the environment in which the system exists may also be dynamic. When selecting a book or news article to recommend, the set of choices is relatively static, but schedules and vacation availability inhabit an ever-changing world. While in the former, experiments can be performed in relative isolation, the latter implies a more complicated experimental setup to properly capture the context within which users make decisions.

Fourth, some recommendations directly address the user's immediate task (e.g., driving routes to the destination), while others are peripherally related (e.g., items users might like based on browsing history). Experiments to evaluate the former require the development of problem scenarios to which users can relate in order to provide appropriate feedback on the recommendations. Our experience with such synthetic problems in the Year 4 evaluation of PTIME indicates that it is easy to underestimate this task.

Adaptive Calendaring Assistants

Of the several previous explorations around developing user-adaptive calendaring assistants, few have involved actual user studies. As with recommender systems, there are distinct differences between these calendaring assistants that determine what evaluation is appropriate.

The *Calendar Apprentice* (CAP) (Mitchell et al. 1994) and *SmartCal* (Krzysicki and Wobcke 2008) learned user preferences for predicting individual meeting attributes (e.g., meeting time, location). Because the learning task was focused on predicting attribute values rather than generating entire schedules, learning could be performed independently of the assistance task, simplifying evaluation. In contrast, PTIME uses its learned preference model to generate candidate schedules and it learns from user selections over alternatives, so learning cannot be evaluated independently of the scheduling assistance context.

The *Learning Interface Agent* (LIA) (Kozierok and Maes 1993), *groupTime* (Brzozowski et al. 2006), and *CMRadar* (Modi et al. 2004) focused on learning user preferences over time slots. As in our earlier evaluations of PTIME (Gervasio et al. 2005), these systems have been evaluated for their adaptive capabilities using synthetic data and, in the case of LIA and CMRadar, using synthetic target user models as well. Learning was shown to improve performance under these conditions but, as we have argued, there is a difference between evaluating learning performance and evaluating the usefulness of an adaptive assistant. Brzozowski et al. (2006) did conduct a user study evaluating scheduling with groupTime vs. scheduling by email and found promising results w.r.t. the usefulness of the tool. However, the user study did not evaluate the usefulness of the adaptive capability itself.

Ethnographic and Comparative Studies

The Year 2 evaluation of PTIME and the recognition of the need to address usability led to our first user study of scheduling needs and preferences reported earlier. We also considered conducting an evaluation of user experience of existing, non-adaptive commercial calendaring systems (e.g., Outlook), but excellent studies of that nature have already been reported in the literature (Palen 1999), with lessons correlating to those gleaned from our own first study.

With the primary focus of CALO being machine learning, it was not until Year 5 that we had the opportunity to conduct evaluations targeted at usability and usefulness. Given that we now had to evaluate a human–computer pair, a possibility would have been to conduct a comparative A-B study comparing performance with PTIME against a suitable baseline or comparable system. For example, RADAR measured the performance of the human-agent pair, with and without learning, against that of humans using commercial, off-the-shelf tools (Freed et al. 2008; Steinfeld et al. 2006).

There were a number of reasons we decided against such a study. First, expert opinion from the HCI community advised us that an A-B study of Emma versus a commercial tool such as Outlook would be unlikely to yield meaningfully reliable data on task effectiveness, due to the difficulty of ensuring comparable tasks between the tools, control of multiple parameters, equivalent user populations,

and paradigms of use (Cohen 1995; Greenberg and Buxton 2008). Other user-adaptive calendaring assistants do not provide comparable baselines either, since they address distinctly different tasks. Second, in PTIME, the preference model is integral to the creation and exploration of the search space. Without a preference model, the performance of the constraint reasoner is greatly affected, rendering insignificant any comparison against PTIME without any learning. Third, our goal was not to prove the superiority of a research prototype compared to commercial systems, but to validate the promise of the PTIME concept. We wanted to assess whether PTIME is effective at its purpose, whether it is usable for naive users, and what are the subjective opinions of the AI-empowered concept it embeds. Although we did not perform comparative studies, our structured evaluation did include subjective assessment of the perceived usefulness of Emma in relation to existing tools.

Lessons Learned

Through our experience with PTIME and analysis of previous work, we have explored challenges in evaluating a user-adaptive AI-based personal assistant. Five lessons stand out from this case study in the time management domain.

The context of the AI technology must be a primary focus in designing an evaluation strategy. Outside the arena of pure academic research, the overarching question is usually whether a system is more or less beneficial with the AI technology. There is often a practical question at the heart of the evaluation. Will the product sell? Will customer satisfaction be increased? Will the resulting system reduce errors or increase user productivity? Therefore, in designing the evaluation of user-adaptive systems, it is paramount to consider the context in which the AI technology is to be used and, consequently, the questions the evaluation must answer (Greenberg and Buxton 2008). Is it to quantify the contributions of academic research? Support a decision to fund product development? Justify deployment? An adequate evaluation must consider context, whether the primary concern is AI research or system development/deployment.

User-adaptive systems require distinct evaluation strategies. Researchers are trained to evaluate algorithms and technological approaches, primarily to show that they improve performance in some way (e.g., increased predictive accuracy, efficiency, precision/recall). However, when the objective is to use AI technology to assist users by adapting to their preferences, theoretical learning curves do not suffice. In user-adaptive assistants such as PTIME, users respond both to the assistance and to the adaptation; it is moot to evaluate learning performance without thought to overall system usefulness.

In-the-wild evaluation is necessary when factors affecting user behaviour cannot be replicated in a controlled environment. A key question that must be resolved early in the design process is where on the continuum from controlled experiment to evaluation "in the wild" is most appropriate. With AI systems, it is easy to be deceived into thinking a highly controlled study will suffice. Controlled evaluations are easier to perform and often yield a greater amount of data in a shorter period of time. However, our experience is that it is easy to miss critical complex social and personal factors surrounding the use of a personalized system. In PTIME, we found that users behave differently when their decisions do not impact their actual time commitments, requiring our evaluation be conducted "in the wild". That said, when the recommendation task is separable from the learning task, or training data can be effectively gathered offline, meaningful, isolated evaluation of the adaptive capabilities can be conducted.

In-the-wild evaluation implies significant additional development costs. To prove return on investment, the realism of "in the wild" evaluation is particularly compelling, but conducting such an evaluation is significantly more demanding. Besides care in the selection of the test subject population, attention must be given to system usability, stability, training, and support. Even large, well-funded, AI projects such as CALO (Myers et al. 2007), RADAR (Freed et al. 2008), and Electric Elves (Chalupsky et al. 2001) have conducted evaluations that are arguably only partially "in the wild". CALO relied on a dedicated critical learning period during which participants conducted very loosely scripted office activities with their CALOs. RADAR devised an artificial conference organization task within which to evaluate the system's ability to help users cope with email overload. Electric Elves was evaluated through actual use over several months—but by its own researchers and with assessment only in terms of indirect metrics, such as the reduction in the number of emails exchanged regarding meeting delays. In PTIME, we saw how lack of robustness and poor usability can hinder data collection even in a controlled setting. A strategy is needed to ensure that the system being developed is (1) sufficiently robust to work reliably for the duration of the evaluation; (2) usable and effective enough to be accepted by the subjects; and (3) integrated into their working environment.

Ease of adoption of the system by users will determine the success or failure of a deployed evaluation strategy. Users who happily download a new tool will as quickly discard it the first time that they encounter serious bugs or difficulty in accomplishing "easy" tasks. Unless the assistance is sufficient (i.e., enough capability), adaptation is irrelevant. Beyond the maturity and usability of a system, another barrier to adoption may be the paradigm shift implied by new capabilities. A strategy to help avoid this problem is to augment an existing tool instead of building an entire replacement. However, augmenting existing tools has challenges, too, such as adequate access to their internals, project constraints, engineering effort, and the suitability of the interaction paradigm. As discussed earlier, PTIME was initially implemented as an add-on to Outlook, but project, integration, and evaluation constraints required a PTIME-specific interface to be built.

When building new tools, consideration must be given to users' current work practices as change to familiar mission-critical tools can be costly and difficult. In PTIME, we found that even motivated volunteers who disliked their current tools found it hard to remember to use PTIME regularly. Even though a majority of the users agreed that scheduling using constraints is preferable to manually finding times that work for all participants, it was difficult to break the established social practice of sending an email requesting

available times as the first step to scheduling a meeting. Whether the intent is for shorter-term evaluation or long-term use, ensuring adoption requires the value of change to be demonstrated to stakeholders. While technology—or data-gathering—push is simple, user pull can be elusive.

Concluding Remarks

Our experience with evaluation has been interwoven with difficulties caused by both the user-adaptive nature of the AI technology being developed and the dynamic nature of the application domain. We have presented our opinion on the continuum of study types and their trade-offs, from fully controlled to real-world. Although we chose to evaluate our AI technology within a stand-alone system, delivery of AI technology as embedded within or augmenting existing tools can provide a baseline for evaluation that offers advantages over evaluating a new tool. We propose that thorough evaluation of an AI-enabled system be multi-faceted and assess the system in terms of usability, usefulness, acceptance, trust, and adaptiveness. It should also be contextualized within the broad problem or research question at hand. Such an approach provides inherent benefits: the resulting tools will be more mature and user needs better understood.

Acknowledgments We thank Daniel Shapiro, Karen Myers, and the anonymous reviewers for their constructive comments; and Aaron Spaulding, Julie Weber, and Mark Plascencia for help with the user studies and evaluations. We also gratefully acknowledge the many participants in our various studies, especially the PExA team. 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

Adomavicius, G. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17(6):734–749.

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

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

Carroll, J. M., and Rosson, M. B. 1987. *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. Cambridge, MA: MIT Press. 80–111.

Chalupsky, H.; Gil, Y.; Knoblock, C. A.; Lerman, K.; Oh, J.; Pynadath, D. V.; Russ, T. A.; and Tambe, M. 2001. Electric Elves: Applying agent technology to support human organizations. In *Proc. of IAAI-01*.

Cohen, P. 1995. *Empirical Methods for Artificial Intelligence*. Cambridge, MA: MIT Press.

Fiechter, C.-N., and Rogers, S. 2000. Learning subjective functions with large margins. In *Proc. of ICML-2000*, 287–294.

Freed, M.; Carbonell, J.; Gordon, G.; Hayes, J.; Myers, B.; Siewiorek, D.; Smith, S.; Steinfeld, A.; and Tomasic, A. 2008. RADAR: A personal assistant that learns to reduce email overload. In *Proc. of AAAI'08*, 1287–1293.

Gervasio, M. T.; Moffitt, M. D.; Pollack, M. E.; Taylor, J. M.; and Uribe, T. E. 2005. Active preference learning for personalized calendar scheduling assistance. In *Proc. of IUI'05*, 90–97.

Glass, A.; McGuinness, D. L.; and Wolverton, M. 2008. Toward establishing trust in adaptive agents. In *Proc. of IUI'08*, 227–236.

Greenberg, S., and Buxton, B. 2008. Usability evaluation considered harmful (some of the time). In *Proc. of CHI'08*, 111–120.

Kozierok, R., and Maes, P. 1993. A learning interface agent for scheduling meetings. In *Proc. of IUI'93*, 81–88.

Krzysicki, A., and Wobcke, W. 2008. Closed pattern mining for the discovery of user preferences in a calendar assistant. *New Challenges in Applied Intelligence Technologies* 67–76.

Langley, P. 1999. User modeling in adaptive interfaces. In *Proc. of UM'99*, 357–370.

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

Miyashita, K., and Sycara, K. 1995. CABINS: A framework of knowledge acquisition and iterative revision for schedule improvement and reactive repair. *Artificial Intelligence* 76(1–2):377–526.

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

Montaner, M.; Lopez, B.; and de la Rosa, J. L. 2003. A taxonomy of recommender agents on the internet. *Artificial Intelligence Review* 19:285–330.

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

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

SRI International. 2009. CALO: Cognitive Assistant that Learns and Organizes. `http://caloproject.sri.com/`.

Steinfeld, A.; Bennett, R.; Cunningham, K.; Lahut, M.; Quinones, P.; Wexler, D.; Siewiorek, D.; Hayes, J.; Cohen, P.; Fitzgerald, J.; Hansson, O.; Pool, M.; and Drummond, M. 2006. The RADAR test methodology: Evaluating a multi-task machine learning system with humans in the loop. Report CMU-CS-06-125, Carnegie Mellon University.

Viappiani, P.; Faltings, B.; and Pu, P. 2006. Preference-based search using example-critiquing with suggestions. *J. Artificial Intelligence Research* 27:465–503.

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