

# Preferences in Interactive Systems: Technical Challenges and Case Studies

*Bart Peintner, Paolo Viappiani,  
and Neil Yorke-Smith*

■ *Interactive artificial intelligence systems employ preferences in both their reasoning and their interaction with the user. This survey considers preference handling in applications such as recommender systems, personal assistant agents, and personalized user interfaces. We survey the major questions and approaches, present illustrative examples, and give an outlook on potential benefits and challenges.*

The emergence of web-based applications, such as electronic commerce and internet media, has coincided with growing recognition that many of the tasks people perform with computers can be performed better when the application adapts to its user. A popular example is Netflix, which gives customized movie recommendations to each user, using past movie ratings of the user and his or her friends. Netflix and other preference-aware interactive systems share the common aim of aiding the user in carrying out tasks—from finding a product to editing a document—by eliciting preferences from the user, inferring a preference model, and using the model to decide when and how to take action.

The variety of preference-aware interactive applications includes recommender systems (such as Netflix or Amazon.com) that suggest items based on the user's similarity to other users or on previously viewed items, conversational systems that interact with the user in a simplified dialogue to perform a task, interfaces that adapt to the user's preferences and situation, and personal agents that can proactively support the user, modeling his or her needs and desires.

In this article we review characteristic examples of intelligent, preference-aware interactive systems, survey the major questions, challenges and approaches with respect to preference representation, reasoning, and explanation and with respect to interactivity and task performed, and give an outlook on the potential of these systems. Throughout, we illustrate that designing a successful, adoptable system requires more than simply selecting the best form of representation, explanation, and reasoning. It requires a cohesive design that ensures the perceived benefits from using preferences significantly exceed the perceived costs of eliciting them.

The interactivity of applications varies in nature, extent, modalities, and manifestation, as much as applications vary in their domains. Despite this diversity, the interactivity literature provides guidelines for designing an intelligent, interactive system (Horvitz 1999).

1. *Degree of interactivity* across the spectrum of user-system collaboration (Horvitz 1999), from fully manual through varying degrees of mixed initiative, to fully automated.
2. *Preference representation*: the type and fidelity of the model, for instance, a utility function.
3. *Preference elicitation*: the population of some or all of the model. Pertinent here are the methods used and the nature of the interactivity involved (Pu and Chen 2008).
4. *Nature of the reasoning with preferences*, which depends on the application's functionality, degree of interactivity, and preference representation.
5. Resulting output or behavior of the system, ranging from information to decisions to actions.

*Table 1. Assessing Interactive, Preference-Aware Applications.*

Preferences for interactive applications concern not only the *base task*—the characteristics of what the user wants achieved with the application—but also the interactive process—the characteristics of how the user wants it achieved. Lieberman and Selker (2000) discuss the notion of the context of an interactive application. Preferences constitute one aspect of context. In situations and for applications where preferences are pertinent—most notably, the preferences of the system's user or users—failure to design and build the application around these preferences risks leaving the application out of context.

Table 1 lists five aspects that collectively define a preference-aware interactive system. Associated with them is a set of technical challenges for the developer. We highlight four of these challenges that will be seen recurrently in the case studies to be discussed.

First, the degree and nature of interaction must be tailored to the specific need and the context. Would you use a navigation system that suggests the best route to your destination if you must first answer a battery of questions to seed a preference model? Personalization is of limited value if the application is difficult or tedious to use.

Second, the acquisition of preferences is a central challenge in an interactive system (Pu and Chen 2008). Preferences of sufficient breadth and fidelity must be acquired for the reasoning—or the reasoning must be robust to limited preference information—without degrading the user experience. However, the user may not be able to articulate, or may not be bothered to articulate, the base (task-level) or meta- (process-level) preferences. Moreover, whether the preference model is explicitly or implicitly represented, the system should provide a way to guide, understand, and correct its behavior.

Third, interface design should account for preferences. The design must respect requirements

stemming from the use of preferences (for example, it must provide a mechanism to express preferences). Further, the design should exploit the learned models, delivering to the user the benefits of customization. What are the most appropriate interaction modalities and interface elements for the user interface (UI)? For instance, Netflix asks users to rate movies and songs with an intuitive five-star scale. A further consideration for applications on mobile devices is the physical and computational limitations of the device. Finally, and not least, since all these elements are correlated, user studies are necessary in the iterative design and evaluation of an interactive application. In the case of preference elicitation, for instance, the elicitation strategy that is best theoretically may fail to elicit the model if the UI renders it unintelligible to the user.

These challenges cannot be solved in isolation. They are inseparably linked within the overarching cost/benefit equation in the design of interactive systems: what design delivers the maximum benefit to the user while keeping user effort to an acceptable level?

## Application Areas

In this article, we are going to explore systems and concepts from three broadly defined functional areas: recommender systems, personal agents, and personalized user interfaces. The systems we highlight collectively show the diversity of preference-aware applications, both in the dimensions discussed in table 1, and in the spectrum of research and commercial systems.

### Recommender Systems

Recommender systems represent a broad family of tools giving automatic personalized recommendations to a user, based on his or her preferences (Adomavicius and Tuzhilin 2005, Burke 2002).

In domains such as travel recommendations, an option is primarily specified by its attributes. In this case, the user's preference order over the possible options is almost fixed once the user preferences and tradeoffs are known. These domains are addressed by systems with explicit representations, where the system and the user interact in a conversation (by means of critiquing, for instance).

In domains such as film or book recommendations, attributes refer only to external characteristics. These domains are suited to systems that exploit auxiliary information such as from demographic and statistical sources.

*Table 2. Choosing an Appropriate Recommender for a Domain.*

The application domain plays an important role in determining which kind of recommender is more appropriate.

These systems map explicit or implicit user preferences to options that are likely to fit these preferences. They range from applications that require relatively little user input (such as Amazon.com) to more mixed-initiative systems such as conversational recommenders. The resulting recommendations are given based on a user profile derived from different information sources. Preference elicitation can vary between question answering, rating options, critiquing, and conversational interaction.

Some systems generate recommendations with as little effort as possible from the user. An example of such a technique is Amazon.com's "People who bought this item also bought...." These systems provide recommendations to the user based on demographics, content (for example, viewed items), or similar users' data. The system uses this information to provide the recommendations to the passive user; thus preferences are implicitly and indirectly elicited.

In contrast, *conversational* recommender systems involve the user elaborating his or her requirements and desiderata over the course of an extended dialogue. They are often used in situations where the user is actively searching for a particular item but he or she has only partial knowledge of the domain, a task sometimes called a *preference-based search*.

Recommender systems are a wide area of research and a detailed survey is not our aim in this article; we refer to the existing surveys in literature (Adomavicius and Tuzhilin 2005, Burke 2002). In the following paragraphs, we describe the popular class of collaborative filtering systems, and two conversational recommenders, SmartClient (a tool based on example-critiquing interaction) and Teaching Salesman (a tool that explicitly represents user needs along with preferences), to highlight the different facets of preference elicitation, recommendation quality, and interface design and usability.

**Collaborative Filtering Systems.** The idea that similar people like similar things is at the core of

collaborative filtering techniques (Konstan et al. 1997). Here, an implicit model of a user's preferences is inferred from items that he or she has rated. Reasoning is based on statistical predictions to estimate the missing ratings. Collaborative filtering has obtained attention because it can uncover serendipitous associations. It is applied by websites such as Amazon.com (for example, books, videos) and Last.fm and Mystrands (music), and by services such as Netflix. The interaction is asynchronous—the user can provide new feedback and obtain new recommendations at any time—and the recommendation engine works inside the framework of an online user community.

Social context has become an important element of the equation. Users of Mystrands, for example, in addition to personalized recommendations, can discover what songs other members of the community are listening to. This idea has been exploited further in systems such as TaRS (Massa and Avesani 2007) where recommendations are based on explicitly trusted neighbors. The quality of recommendations depends on the number of users participating in the community and the amount of feedback that each of them is willing to give. Preference fidelity is necessary for recommendation quality, motivating research about how to stimulate participation. Rashid et al. (2006), for example, found that users of social recommenders are surprisingly altruistic: they are more willing to provide more feedback if the uniqueness of their contribution to the community is made salient.

**SmartClient.** A tool for planning travel arrangements, SmartClient (Pu and Faltings 2000) is based on critiquing examples of possible solutions. For instance, "the arrival time of this flight leg is too late." SmartClient has been commercialized for business travelers as Reality, shown in figure 1. The interaction is cyclical: (1) the system offers example solutions, (2) the user examines any of them and may state a critique on any aspect of it, (3) the critique becomes an additional preference in the



Figure 1. SmartClient, Commercialized as Reality, Is an Example-Critiquing Tool for Arranging Business Travel.

model, and (4) the system refines the solution set. Ricci and Nguyen (2007) propose a similar critiquing interaction to provide recommendations of restaurants in a mobile context.

As discussed in Pu and Chen (2008), the motivation for this methodology is that people usually cannot state preferences up front but construct their preferences as they see the available options. However, because the critiques come from the user in response to the shown examples, the current solutions can hinder the user from refocusing the search in another direction (the *anchoring effect*). A complete preference model can be acquired only if the tool is able to stimulate the user by showing diverse examples.

A refinement is to employ model-based suggestions, such as in FlatFinder (Viappiani, Faltings, and Pu 2006), to teach the user about available options, based on an analysis of the user's current preference model and his or her potential hidden preferences. For instance, a student looking for a low-price accommodation could realize that, by paying just a bit more, he or she can find an apartment with nearby subway access. Suggestions can profit from prior knowledge (for example, most people prefer unfurnished apartments) and can be adapted online, observing user reactions to the displayed examples.

Even so, the user still takes an active role in the construction of the preference model; this requires some effort. Instead of suggesting complete outcomes, incremental dynamic critiquing (Reilly et al. 2005) makes use of data mining algorithms to propose predefined multifeature critiques such as "lighter but more expensive" to guide the exploration of the search space, thus reducing the burden on the user.

An important challenge is scalability. For instance, since travel planning involves configuration, the number of possible solutions is exponential, yet custom solutions have to be computed specifically for each user according to his or her preference model. SmartClient addresses this challenge by distributing the search tasks to lightweight problem-solving clients based on constraint satisfaction.

**Teaching Salesman.** Teaching Salesman (Stolze and Ströbel 2004) helps the user choose an item based on his or her needs: for example, a camera to take holiday pictures, as shown in figure 2. The options are characterized at the use layer (for example, outdoor use compared to night use), characterized by features (for example, optics, power, and weight) that are abstractions of basic attributes (for example, consumption and battery size). The system evaluates the alternatives by mapping the marginal score of each feature to each specific purpose.

The interaction is divided into three phases.

First, preference discovery is based on questions about needs and intended uses of the product. Next, preference optimization allows the user to tweak the model: users can directly assess preferences for specific features, and feature values (for example, optical zoom, flash), and their importance. Finally, in the preference debugging phase, in addition to the top-score recommendation, two other choices are shown, labeled "Upgrade" (a more expensive option with more features) and "Alternative" (an option with different features).

A limitation of this approach is that features satisfying specific needs have to be defined beforehand and hard-coded in the domain model. Unlike SmartClient's example critiquing, Teaching Salesman directly exposes features and goals to the user; this is meaningful in domains where complex preference statements are digestible to the user.

### Personal Agents

Personal agents extend intelligent agent-based systems with personalized assistance. A significant aspect of the personalization comes from the agent acquiring knowledge of the user and using it to direct its activity. This knowledge can include models of the user's preferences over what to do and how to do it, preferred ways of working, and degrees of autonomy, as well as models of the user's knowledge and skills, intentions, and so forth.

Interface agents (Maes 1994) actively assist a user in operating an interactive interface (for example, an e-mail client); collaborative agents partner with the user, acting and dialoguing together to complete a shared task; autonomous agents can act without user intervention, concurrently with the user. Personal agents may have any or all of these purposes, in addition to being adapted to an individual.

Personal agents have been built for a diverse range of domains, from time management to trip planning, with a diverse spectrum of interactivity and proactivity. Interaction modalities range from speech on mobile phones to animated avatars (Jacko and Sears 2003). Effective interaction is enhanced by honoring user preferences over interaction modalities: for instance whether or not a tutoring agent is manifest as an animated guide, and the extent to which it uses natural language. The employed models of preferences, the methods of elicitation, and the use of preferences in reasoning likewise span many approaches.

The ability to act autonomously or proactively on behalf of the user is one distinguishing aspect of agent-based applications. Concerns arise about trust, expectations, safety, privacy, and control (Norman 1994), increasing the importance of preferences over how a task is accomplished and introducing the need for adjustable autonomy prefer-

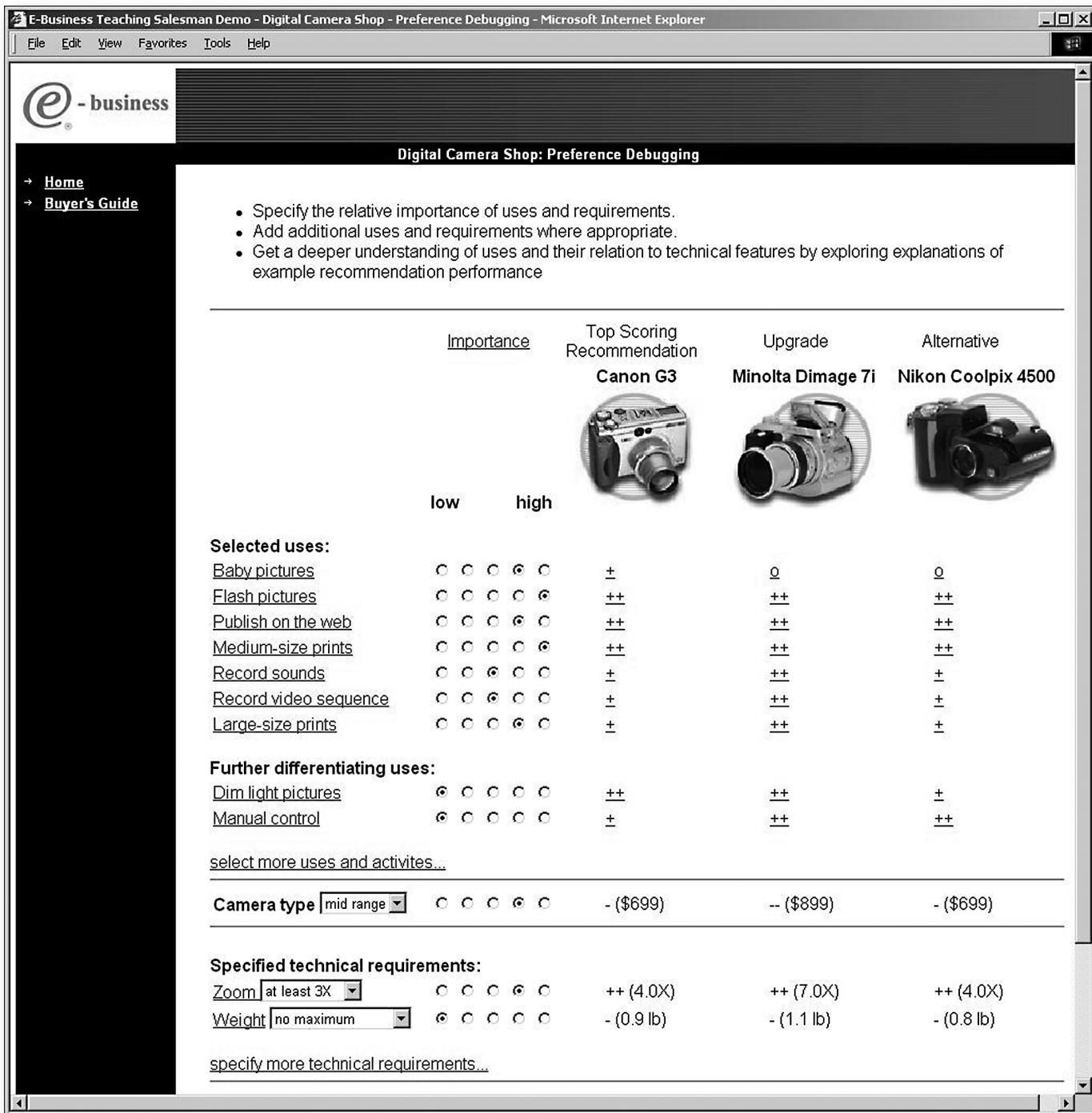


Figure 2. *Teaching Salesman Helps the User Compare Different Products Based on How They Match His or Her Needs.*

ences that govern the agent's operation in a mixed-initiative setting. Failure to properly account for these aspects risks only to cause frustration to the user, as in the infamous "Clippy" of Microsoft Office (Schiaffino and Amandi 2004).

With the breadth of personal agents deserving a

survey in its own right, we describe two characteristic systems and a growing class of personal agents.

**CAP: Calendar Apprentice.** Alongside e-mail assistance, time management is a domain receiv-

ing sustained research interest. The concept of a personal agent that assists in managing your calendar has been pursued since the early days of electronic calendars. Studies have shown that time management is an intensely personal matter; hence understanding and fulfilling the user's individual preferences is essential if people are to value this type of personal agent (Palen 1999). A representative and influential example is CAP (calendar apprentice) (Mitchell et al. 1994). A CAP agent is an interactive assistant that observes a user operating with his or her electronic calendar and provides advice. This advice consists of suggested autocompletions of parameters when the user is creating a new meeting entry.

Other systems have offered assistance for tasks such as negotiating meetings. For instance, Sen, Hayes, and Arora (1997) reach consensus through a distributed negotiation process, ranking meeting options with a voting scheme. The agent votes automatically as the user's proxy, based on a preference model. The model is composed of values of each feature (for example, topic, time of day, participants, location), summed with weights on feature importance.

A challenge for assistive agents is the tension between helpfulness "out of the box" and customized assistance after preferences have been acquired. Similar to the cold start problem in recommender systems, users are rarely willing to spend time training the system, beyond stating a few crude preferences. Is the mandated preference specification of Sen, Hayes, and Arora (1997) too cumbersome (for example, each user is asked to rate all other known users), despite default values; is a window full of UI sliders to set weights between features too intimidating?

CAP unobtrusively learns a preference model from user actions. Its model is based on a decision tree of rules; rules can be viewed and edited by the user and form the basis of explanation of the agent's actions. Other systems commonly augment automated acquisition of their preference model with circumscribed direct preference elicitation, such as of preferred meeting times (*ceteris paribus*), to seed the model and make the system more useful from the beginning.

An additional challenge for agents in these domains is how to expose the agent's assistance without degrading the user experience with familiar application UIs, as discussed in Jacko and Sears (2003). As an earlier example of the genre, CAP takes the simplest but suboptimal approach of replacing the calendar UI with its own.

**PExA: Project Execution Assistant.** PExA (Myers et al. 2007) aids a busy knowledge worker in managing time commitments and performing tasks. The system was designed to help (1) relieve the user of routine tasks, thus allowing the user to

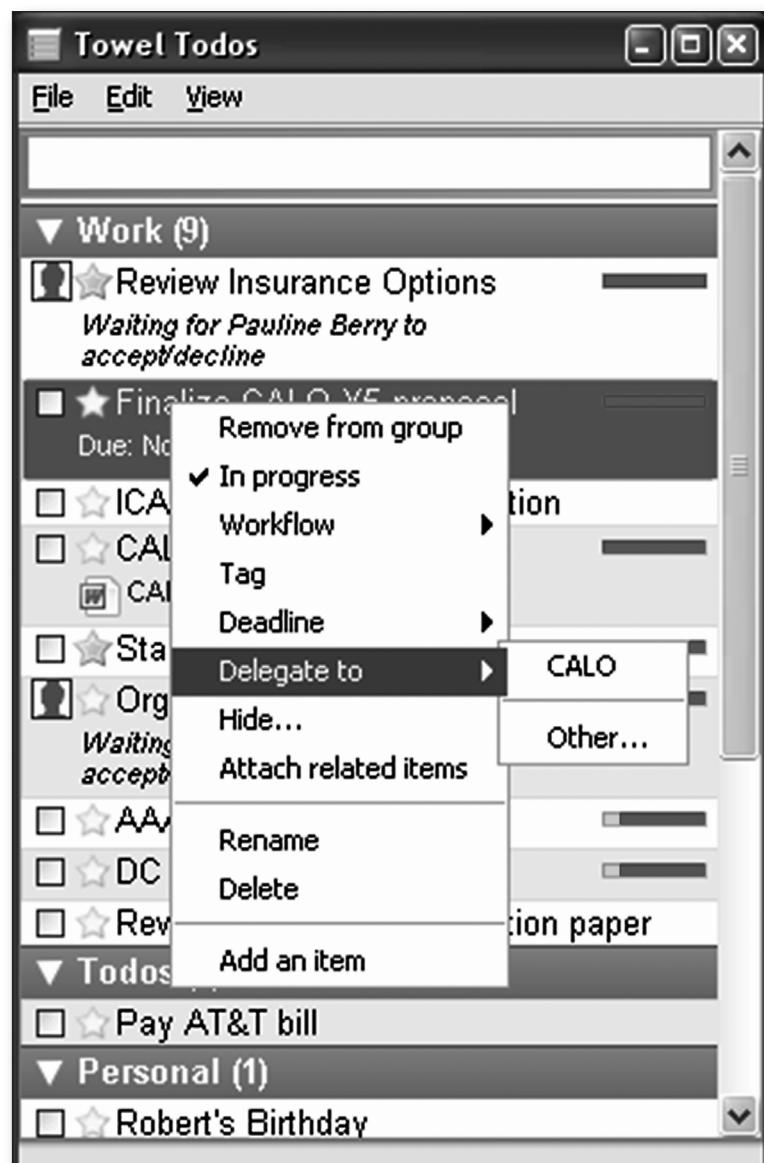


Figure 3. Task Delegation with the PExA Personal Agent.

focus on tasks that critically require human problem-solving skills, and (2) intervene in situations where cognitive overload leads to oversights or mistakes by the user.

Akin to an adept personal assistant, PExA's utility to the user critically depends on personalization. The agent must have an understanding of the user's tasks and preferred working style. Since PExA is part of a larger effort to explore machine learning, known as the Personal Assistant that Learns (PAL), an emphasis is placed on learning techniques to populate the multiple preference models employed by components of the system, to govern the interaction of the system with the user, and to direct and review its behavior.

A PExA agent comprises a task manager, and a time manager described in Berry et al. (2006). PExA can perform tasks on behalf of its user at his or her request, such as purchasing an item of equipment, applying for expense reimbursement, planning travel, and registering for a conference. It uses models of the user's base-level preferences, such as his or her preferred airline and preferred meeting locations. Besides execution of delegated tasks, PExA's task management capabilities include explanation of its activities, learning of task models (for instance, those obtained from PLOW (Allen et al. 2007)), and execution monitoring. Based on user activities, the agent can infer the state of shared user-system tasks and proactively offer suggestions, such as performing the next step of a task or delegating a task to another user (as shown in figure 3).

For multidomain, multifunction agents such as PExA, transparency and controllability are demanded for user comfort and system adoption. Would you trust an agent with your credit card number? Preferences and adjustable autonomy advice from the user govern PExA's reasoning over when to act, when to suggest, when to ask for clarification, and when to do nothing. Preferences likewise govern the modality of the interaction the agent employs (Horvitz 1999); for instance, a pop-up dialogue box versus an unobtrusive notification in a sidebar.

**Cognitive Assistants.** A cognitive assistant trains, augments, or restores the cognitive abilities of its user. These assistants help students learn new knowledge and skills, help the elderly remember to carry out daily activities, and help people with head injuries regain lost abilities—all activities that require in-depth personalization. These systems cover the entire range of interactivity. Forms of tutoring, at one extreme, consist of systems that give direct instruction. For instance, adaptive hypermedia systems alter the content or appearance of the presented information on the basis of a dynamic understanding of the individual user, so as to adapt the content or presentation to certain characteristics of the user (Eklund and Sinclair 2000). These systems dynamically choose which text to turn into hyperlinks and dynamically change the target of existing links, such as the "Next" button.

Domains of adaptive tutoring agents include mainstream educational pedagogy (Graesser et al. 2001), practicing a musical instrument,<sup>1</sup> learning to program (Selker 1994), and touring a city or museum (a form of recommender system) (Cheverst et al. 2000).

At the other extreme of interactivity, cognitive assistants can operate transparently to the user, intervening only when necessary. Systems such as Autominder (Pollack et al. 2003) and Opportunity

Knocks (Patterson et al. 2004) are designed to augment memory skills in older adults with mild cognitive impairment. For instance, Opportunity Knocks takes the form of a smart mobile phone that helps plan your route with public transport. Two critical design goals are to require no explicit programming by user or caregiver and to be robust in the real world. The agent learns the user's routine and does not require the user to explicitly provide information about preferred modes of transit or the starting or ending points of his or her journeys. It provides proactive assistance with route planning, changing mode of transport, and recognizing and recovering from user errors.<sup>2</sup> In the middle ground, systems such as COACH (Selker 1994) tutor users with instructional content as they learn and perform. When operating in conjunction with another application, this middle class of system can be seen as a form of interface agent; in contrast to CAP, here the agent is the "expert" and the user is the "apprentice."

The primary role of preferences in these agents concerns the user's mode of learning and the nature of the help desired. The agents personalize to, for instance, whether the users are expert or novice at a task, whether their learning style is more inductive or deductive, their emotional state, and whether they like continuous context-sensitive hints or prefer more focused but attention-demanding help after they have explored on their own. Often, these preferences are acquired—or refined after initial explicit elicitation—unobtrusively and automatically as the user works. The goal of the agent is to foster the learning or aid the performance of the user; the user's attention should be on the matter at hand, and not on the agent.

## Personalized User Interfaces

Personalized user interfaces (PUI) move beyond the pervasive "one size fits all" paradigm of UI design. By learning and employing models of the user and his or her environment, personalized UIs customize the interface to the user's tastes and the context of the interaction. We have already seen the value of interface personalization in personal agents.

PUIs form a subfield within the adaptive user interface community, which collectively adapts UIs to a broad range of context, including characteristics of the user (for example, identity, current task, level of ability, emotional state, preferences), the device (for example, screen size, bandwidth, available input methods), and the environment (for example, noise level, presence of other agents). For each application, a different subset of the full context is relevant to adapting the interface. PUIs focus on systems whose relevant context includes a user's preferences and current task. What differ-

Active elicitation, such as asking the user to compare two outcomes, can converge to the cost function quite rapidly, as indicated by ARNAULD. However a comparative evaluation of FlatFinder and questions-answering (a form of active elicitation) showed that active elicitation can produce incorrect preference models when the user does not know the domain. Thus there is no simple rulebook for determining the best form of elicitation — but there are guidelines (Pu and Chen 2008).

*Table 3. Which Type of Elicitation Is Best?*

entiates PUIs from its siblings within the adaptive UI community is the need to focus on preference elicitation and reasoning.

Given general and possibly also case-specific preferences, an interface can be personalized in one of several ways. First, the components of the interface can be organized in a way that chooses, highlights, or modifies visual elements relevant to the user's current task or consistent with his or her past behavior. Second, user actions in the UI can be automated if patterns of use match previous patterns (for example, display a help menu or finish formatting a table). Third, the modality of the interaction or presentation can be changed (for example, show concept as text or as a graphic) (Billsus et al. 2002; Zhou and Aggarwal 2004; Brafman and Friedman 2006).

**ARNAULD/SUPPLE.** The ARNAULD (Gajos and Weld 2005) and SUPPLE (Gajos and Weld 2004) systems collectively generate and lay out graphical user interfaces using descriptions of UI elements, a specification of the target device, and a user model, which is obtained through inference over user interaction traces and elicited preferences. ARNAULD interacts with the user to elicit preferences over UI widgets and locations, while SUPPLE searches for interfaces that are optimal with respect to the "cost" of using the interface to perform a set of user interface tasks.

ARNAULD employs two interaction methods for eliciting preferences over interface layout. The first is example-critiquing (shown in figure 4), where the user can add, remove, and reposition elements in the UI based on his or her preferences. The second method is active elicitation, in which users are asked to choose the best widget for a given situation. From these elicitations, the system infers the cost of performing an action on each element and the effort in making the transition to the next element in the interaction sequence.

SUPPLE's cost function is based on sequences of elements manipulated by the user. When evaluating an interface, the system totals the user's expected effort in performing the sequence on that interface. The trace is both device- and inter-

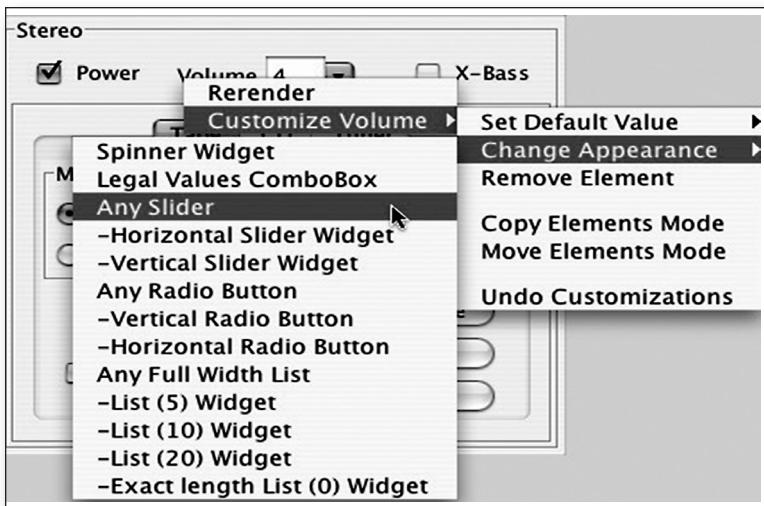
face-independent, making it applicable for transferring to new devices and situations.

Which elicitation method is best, in general and for a given application? The authors found that example-critiquing is most effective when the cost function is almost correct, but that it requires an interface allowing its customization by the user. Active elicitation, conversely, is convenient during the early stages of the elicitation process and both is easy for the user and puts little demand on the UI. However, it suffers from the fact that the user does not direct the learning process, burdening the system to choose elicitation questions that will most quickly train the model (see table 3).

**Gaze-X.** In addition to the choice of widget and its location, the Gaze-X system (Maat and Pantic 2007) modifies the widgets themselves, changing font and component sizes in response to changes in user focus or task. Gaze-X is more dynamic and more proactive than ARNAULD/SUPPLE, showing contextual help when it notices the user looking at a particular element for a long time. The systems share the goal of altering the display in response to inferred user preference, but differ (1) in the complexity of user model—Gaze-X's is more shallow and broad—and (2) in the way they use the model—ARNAULD/SUPPLE attempts to optimize a layout before use, while Gaze-X tries to constantly adapt to the current situation.

A key challenge for Gaze-X is designing the output of the system. Changing the interface too quickly, too often, or at the wrong time will doom an interactive system to a spot in the AI graveyard. Gaze-X takes a conservative approach, making changes one at a time and at a slow pace.

**Adaptive Rich Media Presentations.** Presenting rich media, such as video, Flash presentations, and animated text, requires the interface designer to respect constraints not encountered in PUI layout systems such as Gaze-X and SUPPLE. Consider a promotion for a sporting event that contains multiple video segments, Flash and video ads, and a diverse audience viewing the promotion from a diverse set of devices (Brafman and Friedman 2006). Temporal constraints arise from video (a



*Figure 4. Incorporation of Example-Critiquing in the ARNAULD/SUPPLE System.*

Example-critiquing put significant demands on the interface, but the user benefits in the short term (better interface now) and in the long term (improved models).

particular ordering of the clips); device constraints abound, in terms of both dimensions and software; and respecting user preferences is critical or the user will not “stay tuned” to hear the entire message.

Brafman and Friedman’s approach to generating rich media presentations uses TCP-nets (Rossi, Walsh, and Venable 2008) to reason over these constraints and select a qualitatively optimal presentation of the media. The approach allows the author of the presentation to focus purely on the content.

In contrast to the problems addressed by SUPPLE and Gaze-X, the user model is relatively simple—it is based on demographic information only—and so preference elicitation is straightforward. Reasoning about the preference model and associated constraints, by contrast, is much more complicated than it is in the other systems described, showing that while preference-aware interactive systems share a common set of challenges, the challenges manifest themselves differently in each application.

## Summary and Outlook

This article has surveyed preference-aware interactive systems, enumerating common challenges in their design and highlighting case studies in the domain of recommender systems, personal agents, and personalized user interfaces. The borders between these application areas are somehow

fuzzy or illusory: for example, highly interactive and persistent recommender systems assume the characteristics of personal agents, while adaptive hypermedia agents that simply change text to hyperlinks could be considered as a PUI. We presented several technical challenges in designing these systems and argued that designers cannot attack these challenges independently: the best preference model for a given application may require more elicitation time than a user cares to devote, for instance, or the available reasoning time may not afford the interaction style desired. Instead, the designer must trade off the design of all components in such a way that the user’s perceived and actual benefits will significantly exceed his or her perceived costs. The case studies presented reinforce this basic idea. All the systems addressed these challenges in diverse, yet non-competing ways. We do not see that critiquing-based systems are superior to direct elicitation; rather, we begin to understand the scenarios appropriate for each.

Commercialized preference-aware systems tend to adopt low effort elicitation strategies. Does this imply that the public is not familiar with the benefit of preference-aware systems? Or that only the lightest weight elicitation of preference is acceptable in the typical situation? Or that research has not produced medium-to-high effort elicitation strategies whose return justifies their effort? Commercial vendors are pragmatic: their aim is not the best technical solution but the system that maximizes profit. For instance, a recommender system more customized to the user might make the customer happier but lead the user to purchase a lower-end item.

The study of interface design and usability are outside the scope of this article, but, as demonstrated with the PEACH museum guide assistant (Goren-Bar et al. 2005), poor judgment in UI design can significantly affect the use and adoption of a system, regardless of the preference-handling mechanisms. If the user experience is lacking, a preference model of low accuracy may result.

## Emerging Trends and Domains

Interactive systems are a dynamic domain of research. Several practical applications in industry have already been successfully deployed, and we can expect that the emergence of new domains, new sources of data, and technical advances will bring even more attention to the field.

A rich emerging domain is personalized medicine. Many predictions have identified the aging population of developed countries as a grand social and economic challenge. The rising cost of health care can be met in part by personalized assistance to the elderly, keeping track of their needs, tasks, preferences, and medical treatments. In this and

similar contexts (such as providing assistance to disabled and impaired people) the system should be able to monitor and understand the different cognitive abilities of the user.

We foresee the continued emergence of systems that cross-fertilize between the application areas that we have treated in this article. Further, an important development will be to acquire and reason with broader preference models, adding not only a user's location, but also the user's personality, emotional states, and cognitive load.

New sources of data will dramatically change the cost/benefit equation in the design of interactive systems. With the growing popularity of social websites and web 2.0 and beyond applications, system designers can plan to leverage preferences, interests, and other user information readily available without additional user effort. The increased prevalence of mobile and location-aware devices (such as Apple's iPhone) will allow, for example, a traveler to obtain recommendations for restaurants based on his or her current location at lunchtime, a personalized city tour in the afternoon, recommendations for cinema in the evening, and assistance in navigating around town either with public transportation or by car during the full day.

It is possible that society will warm to appropriate roles of carefully designed personal agents that can act proactively (Norman 1994). Even for strictly reactive, more easily understood and controlled systems, however, a greater adoption of personalized applications is possible only if users trust the systems (Konstan, Riedl, and Smyth 2007). At the same time, users have a right to expect that their personal data is used only for the purposes delegated to the system.

## Acknowledgements

We thank Ronen Brafman, Boi Faltings, and the anonymous reviewers for their suggestions. The work of the two authors at SRI International was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. Any opinions, findings, and conclusions or recommendations are those of the authors and do not necessarily reflect the view of DARPA or the Department of Interior NBC. Paolo Viappiani has been supported by the Swiss National Science Foundation under Contract No. 200020-111862.

## Notes

1. SmartMusic. [www.smartmusic.com](http://www.smartmusic.com).
2. Although the initially tested system did not feature an explicit user preference model, subsequent work underscored the importance of adapting the guidance strategies and interface modalities of such devices to the preferences of the user (Liu et al. 2006).

## References

- Adomavicius, G., and Tuzhilin, A. 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.
- Allen, J. F.; Chambers, N.; Ferguson, G.; Galescu, L.; Jung, H.; Swift, M.; and Taysom, W. 2007. PLOW: A Collaborative Task Learning Agent. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, 1514–1519. Menlo Park, CA: AAAI Press.
- Berry, P.; Conley, K.; Gervasio, M.; Peintner, B.; Uribe, T.; and Yorke-Smith, N. 2006. Deploying a Personalized Time Management Agent. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi-agent Systems* (AAMAS-06), 1564–1571. New York: Association for Computing Machinery.
- Billsus, D.; Brunk, C. A.; Evans, C.; Gladish, B.; and Pazzani, M. 2002. Adaptive Interfaces for Ubiquitous Web Access. *Communications of the ACM* 45(5): 34–38.
- Brafman, R. I., and Friedman, D. 2006. Adaptive Rich Media Presentations Via Preference-Based Constrained Optimization. In *Preferences: Specification, Inference, Applications*, No. 04271 in Dagstuhl Seminar Proceedings. Internationales Begegnungs und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany.
- Burke, R. D. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction* 12(4): 331–370.
- Cheverst, K.; Davies, N.; Mitchell, K.; Friday, A.; and Efratouli, C. 2000. Developing a Context-Aware Electronic Tourist Guide: Some Issues and Experiences. In *Proceedings of the 2000 SIGCHI Conference on Human Factors in Computing Systems* (CHI'00), 17–24. New York: Association for Computing Machinery.
- Eklund, J., and Sinclair, K. 2000. An Empirical Appraisal of the Effectiveness of Adaptive Interfaces for Instructional Systems. *Educational Technology and Society* 3(4).
- Gajos, K., and Weld, D. S. 2004. SUPPLE: Automatically Generating User Interfaces. In *Proceedings of the 8th International Conference on Intelligent User Interfaces* (IUI-04), 93–100. New York: Association for Computing Machinery.
- Gajos, K., and Weld, D. S. 2005. Preference Elicitation for Interface Optimization. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (UIST-05), 173–182. New York: Association for Computing Machinery.
- Goren-Bar, D.; Graziola, I.; Kuflik, T.; Pianesi, F.; Rocchi, C.; Stock, O.; and Zancanaro, M. 2005. I Like It—An Affective Interface for a Multimodal Museum Guide. Paper presented at the Workshop on Affective Interactions at the 9th International Conference on Intelligent User Interfaces (IUI-05). San Diego, CA, January 10–13.
- Graesser, A. C.; VanLehn, K.; Rose, C. P.; Jordan, P. W.; and Harter, D. 2001. Intelligent Tutoring Systems with Conversational Dialogue. *AI Magazine* 22(4): 39–52.
- Horvitz, E. 1999. Principles of Mixed-Initiative User Interfaces. In *Proceedings of the 1999 SIGCHI Conference on Human Factors in Computing Systems* (CHI-99), 159–166. New York: Association for Computing Machinery.
- Jacko, J. A., and Sears, A., eds. 2003. *The Human-Computer Interaction Handbook*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

- Konstan, J. A.; Miller, B. N.; Maltz, D.; Herlocker, J. L.; Gordon, L. R.; and Riedl, J. 1997. GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM* 40(3): 77–87.
- Konstan, J. A.; Riedl, J.; and Smyth, B., eds. 2007. *Proceedings of the 2007 ACM Conference on Recommender Systems*. New York: Association for Computing Machinery.
- Lieberman, H., and Selker, T. 2000. Out of Context: Computer Systems That Adapt to, and Learn from, Context. *IBM Systems Journal* 39(3–4): 617–632.
- Liu, A. L.; Hile, H.; Kautz, H.; Borriello, G.; Brown, P. A.; Harniss, M.; and Johnson, K. 2006. Indoor Wayfinding: Developing a Functional Interface for Individuals with Cognitive Impairments. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS-06), 95–102. Portland, OR, October 23–25. New York: Association for Computing Machinery.
- Maat, L., and Pantic, M. 2007. Gaze-X: Adaptive, Affective, Multimodal Interface for Single-User Office Scenarios. In *Artificial Intelligence for Human Computing*, Lecture Notes in Computer Science, Vol. 4451, 251–271. Berlin: Springer.
- Maes, P. 1994. Agents That Reduce Work and Information Overload. *Communications of the ACM* 37(7): 30–40.
- Massa, P., and Avesani, P. 2007. Trust-Aware Recommender Systems. In *Proceedings of the 2007 ACM Conference on Recommender Systems* (RecSys-07), 17–24. New York: Association for Computing Machinery.
- Mitchell, T.; Caruana, R.; Freitag, D.; McDermott, J.; and Zabowski, D. 1994. Experience with a Learning Personal Assistant. *Communications of the ACM* 37(7): 80–91.
- Myers, K.; Berry, P.; 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.
- Norman, D. A. 1994. How Might People Interact with Agents. *Communications of the ACM* 37(7): 68–71.
- Palen, L. 1999. Social, Individual, and Technological Issues for Groupware Calendar Systems. In *Proceedings of the 1999 SIGCHI Conference on Human Factors in Computing Systems* (CHI-99), 17–24. New York: Association for Computing Machinery.
- Patterson, D. J.; Liao, L.; Gajos, K.; Collier, M.; Livic, N.; Olson, K.; Wang, S.; Fox, D.; and Kautz, H. 2004. Opportunity Knock. In *Proceedings of the 6th International Conference on Ubiquitous Computing* (Ubi-Comp-04), 433–450. Berlin: Springer.
- Pollack, M. E.; Brown, L. E.; Colbry, D.; McCarthy, C. E.; Orosz, C.; Peintner, B.; Ramakrishnan, S.; and Tsamardinos, I. 2003. Autominder: An Intelligent Cognitive Orthotic System for People with Memory Impairment. *Robotics and Autonomous Systems* 44(3–4): 273–282.
- Pu, P., and Chen, L. 2008. User-Involved Preference Elicitation for Product Search and Recommender Systems. *AI Magazine* 29(4): 93–103.
- Pu, P., and Faltings, B. 2000. Enriching Buyers' Experiences: The SmartClient Approach. In *Proceedings of the 2000 SIGCHI Conference on Human Factors in Computing Systems* (CHI-00), 289–296. New York: Association for Computing Machinery.
- Rashid, A. M.; Ling, K. S.; Tassone, R. D.; Resnick, P.; Kraut, R. E.; and Riedl, J. 2006. Motivating Participation by Displaying the Value of Contribution. In *Proceedings of the 2006 Conference on Human Factors in Computing Systems* (CHI-06), 955–958. New York: Association for Computing Machinery.
- Reilly, J.; McCarthy, K.; McGinty, L.; and Smyth, B. 2005. Incremental Critiquing. *Knowledge-Based Systems* 18(4–5): 143–151.
- Ricci, F., and Nguyen, Q. N. 2007. Acquiring and Revising Preferences in a Critique-Based Mobile Recommender System. *IEEE Intelligent Systems* 22(3): 22–29.
- Rossi, F., Walsh, T., and Venable, K. B. 2008. Preferences in Constraint Satisfaction and Optimization. *AI Magazine* 29(4): 58–68.
- Schiaffino, S., and Amandi, A. 2004. User-Interface Agent Interaction: Personalization Issues. *International Journal of Human-Computer Studies* 60(1): 129–148.
- Selker, T. 1994. Coach: A Teaching Agent That Learns. *Communications of the ACM* 37(7): 92–99.
- Sen, S.; Hayes, T.; and Arora, N. 1997. Satisfying User Preferences While Negotiating Meetings. *International Journal of Human-Computer Studies* 47(3): 407–427.
- Stolze, M., and Ströbel, M. 2004. Recommending as Personalized Teaching. In *Designing Personalized User Experiences in eCommerce*, 293–313. Norwell, MA: Kluwer Academic Publishers.
- Viappiani, P.; Faltings, B.; and Pu, P. 2006. Preference-Based Search Using Example-Critiquing with Suggestions. *Journal of Artificial Intelligence Research* 27: 465–503.
- Zhou, M. X., and Aggarwal, V. 2004. An Optimization-Based Approach to Dynamic Data Content Selection in Intelligent Multimedia Interfaces. In *Proceeding of the 17th Annual ACM Symposium on User Interface Software and Technology* (UIST-04), 227–236. New York: Association for Computing Machinery.
- Bart Peintner** is a computer scientist at SRI International's Artificial Intelligence Center. His most recent work has used his expertise in scheduling, planning, constraint optimization, and probabilistic reasoning to develop techniques for use in cognitive assistants, both for office workers and elderly persons. He holds a Ph.D. from the University of Michigan.
- Paolo Viappiani** is a postdoctoral researcher at the University of Toronto's Department of Computer Science. His research interests include preference elicitation, intelligent interfaces, and recommender systems. He holds a Ph.D. in artificial intelligence from the Ecole Polytechnique Fédérale de Lausanne (EPFL) with a thesis on personalized search tools.
- Neil Yorke-Smith** is a computer scientist at SRI International's Artificial Intelligence Center. His research aims to help people make decisions in complex situations, with interests including temporal reasoning, planning and scheduling, preferences, advisable agents, intelligent interfaces, and constraint programming, and their real-world applications. He holds a Ph.D. from Imperial College London.