User-Involved Preference Elicitation for Product Search and Recommender Systems

Pearl Pu
and Li Chen

■ *We address user system interaction issues in product search and recommender systems: how to help users select the most preferential item from a large collection of alternatives. As such* systems *must crucially rely on an accurate and complete model of user preferences, the acquisition of this model becomes the central subject of this article. Many tools used today do not satisfactorily assist users to establish this model because they do not adequately focus on fundamental decision objectives, help them reveal hidden preferences, revise conflicting preferences, or explicitly reason about trade-offs. As a result, users fail to find the outcomes that best satisfy their needs and preferences. In this article, we provide some analyses of common areas of design pitfalls and derive a set of design guidelines that assist the user in avoiding these problems in three important areas: user preference elicitation, preference revision, and explanation interfaces. For each area, we describe the state of the art of the developed techniques and discuss concrete scenarios where they have been applied and tested.*

To perform complex tasks, such as searching the web for suitable products or services, planning a trip, or scheduling resources, people increasingly rely on computerized product recommender systems (also called product search tools) to find outcomes that best satisfy their needs and preferences. However, automated decision systems cannot effectively search the space of possible solutions without an accurate model of a user's preferences. Preference *acquisition* is therefore a fundamental problem of growing importance.

Without an adequate interaction model and system guidance, it is difficult for users to establish a complete and accurate model of their preferences. More specifically, we face the following difficulties:

First, inadequate elicitation tools can easily mislead users to focus on *means objectives* rather than fundamental decision objectives and force them to state preferences in the wrong order. For example, a user who commits to the choice of minivans (*means objective*) for spacious baggage space (fundamental) is not focusing on the values and could risk missing alternatives offered by station wagons. In value-focus thinking, Keeney (1992) suggests that the specification and clarification of values should not be overtaken by the set of alternatives too rapidly. This theory has a direct implication on the order in which the system initially elicits user preferences.

Second, users are not aware of all preferences until they see them violated. For example, a user does not think of stating a preference for the intermediate airport until a solution proposes an airplane change in a place the user dislikes. This observation sheds light on the interaction design guideline on how to help users discover their hidden preferences.

Finally, preferences can be inconsistent. Users can state preference val-

 ISSN 0738-4602

ues that are potentially in conflict with values stated earlier (for example, a rather tight budget conflicted with the user's preference on a business trip). This suggests that preferences must be maintained for their consistency.

In addition, there seems to be a dichotomy between what is required of a decision maker (such as being an expert of a domain and fluent with preference expressions) and the nature of his or her task (for example, finding complex products in unfamiliar domains). The main problem in this dichotomy results from uncertainties in the user's decision goals and the user's lack of information on product characteristics. Therefore, the acquisition process between the system and the user must be incremental and adaptive in nature. Explanation interfaces are therefore crucial in convincing users of the recommended alternatives in this incremental process.

This article surveys existing work that addresses user interaction issues in the domain of preference elicitation and compiles them into a coherent set of best-practice guidelines. Novel user-system interaction methods that take into account human decision behaviors are discussed and illustrated as examples for implementing the guidelines. They are further accompanied by empirical user studies whenever possible that demonstrate how they indeed produce significantly better results than earlier techniques.

Our article will first address guidelines regarding how to help users state complete and sound preferences with recommended examples and then describe strategies to help users resolve conflicting preferences and perform trade-off decisions. We also present explanation principles in terms of how to explain the recommended examples so as to increase the system's transparency and build user trust in it.

Stimulate Preference Expression with Examples

Incorrect *means objectives* arise mainly due to users' unfamiliarity with the available options. It has been frequently observed that people find it easier to construct a model of their preferences when considering examples of actual options (Payne, Bettman, and Schkade 1999). This constructive view of human decision making also applies to experts. According to Tversky (1996), people do not maximize a precomputed preference order but construct their choices in light of the available options. Therefore, to educate users about the domain knowledge and help them construct complete and sound preferences, we propose the following guideline:

Guideline 1: Consider showing example options to help users gain preference fluency.

We call such an interaction model *example critiquing* since users build their preferences by critiquing the example products that are shown. This allows users to understand their preferences in the context of available options. Example critiquing was first mentioned in Williams and Tou (1982) as a new interface paradigm for database access, especially for novice users to specify queries. Recently, example critiquing has been used in two principal forms by several researchers: those supporting product catalog navigation and those supporting product search based on an explicit preference model.

In the first type of systems, as were used in the FindMe systems (Burke, Hammond, and Cooper 1996; Burke, Hammond, and Young 1997), product search is described as a combination of search and browsing called *assisted browsing*. The system first retrieves and displays the best matching product from the database based on a user's initial query. It then retrieves other products based on the user's critiques of the current best item. The interface implementing the critiquing model is called *tweaking*, a technique that allows users to express preferences with respect to a current example, such as "look for an apartment similar to this, but with a better ambience." According to this concept, a user navigates in the space of available products by tweaking the current best option to find his or her target choice. The preference model is implicitly represented by the current best product, that is, what a user chooses reflects his or her preference of the attribute values. Reilly and colleagues have proposed *dynamic critiquing* (Reilly et al. 2004) based on some improvements of the tweaking model. In addition to the unit-value tweaking operators, compound critiques allow users to choose products that differ from the current best item in two or more attribute values. For example, the system would suggest a digital camera based on the initial query. It also recommends cameras produced by different manufacturers, with less optical zoom, but with more storage. Compound critiques are generated by the Apriori algorithm (Agrawal, Imielinski, and Swami 1993) and allow users to navigate to their target choice in bigger steps. In fact, users who more frequently used the compound critiques were able to reduce their interaction cycles from 29 to 6 in a study involving real users (McCarthy et al. 2005).

In the second type of example-critiquing systems, an explicit preference model is maintained. Each user feedback in the form of a critique is added to the model to refine the original preference model. An example of a system with explicit preference models is the SmartClient system used for travel planning (Pu and Faltings 2000; Torrens, Faltings, and Pu 2002). It shows up to 30 examples of travel itineraries as soon as a set of initial preferences have been established. By critiquing the examples, users state additional preferences. These preferences are accumulated in a model that is visible to the user through the interface (Torrens, Faltings, and Pu 2002) and can be revised at any time. Other tools that work in a similar way are ATA (Linden, Hanks, and Lesh 1997), ExpertClerk (Shimazu 2001), the Adaptive Place Advisor (Goker and Thompson 2000), and the incremental dynamic-critiquing systems (McCarthy et al. 2005b). One advantage of maintaining an explicit model is to avoid recommending products that have already been ruled out by the users. Another advantage is that a system can suggest products whose preferences are still missing in the stated model.

How Many Examples to Show

Two issues are critical in designing effective example-based interfaces: how many examples to show in one display and which examples should be included. Faltings el al. (2004) investigated the minimum number of examples to display so that the target choice is included even when the preference model is inaccurate. Various preference models were analyzed. If preferences are expressed by numerical penalty functions and they are combined using either the weighted sum or the min-max rule, then

$$t = \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^d \quad (1)$$

where d is the maximum number of stated preferences, and t is the number of displayed items so that the target solution is guaranteed to be included. The error of the preference function is bounded by a factor of epsilon (ε) above or below. Since this number is independent of the total number of available items, this technique of compensating inaccurate preferences by showing a sufficient amount of solutions scales to very large collections. For a moderate number of preferences (up to 5), the correct amount of display items typically falls between 5 and 20. When the preference model becomes more complex, inaccuracies have much larger effects. A much larger number of examples are required to cover the model inaccuracy.

What Examples to Show

The most obvious examples to include in the display are those that best match the users' current preferences. However, this strategy proves to be insufficient to guarantee optimality. Since most users are often uncertain about their preferences and they are more likely to construct them as options are shown to them, it becomes important for a decision system to guide the user to develop a preference model that is as complete and accurate as possible. However, it is important to keep the initiative to state more preferences on the user's side. Therefore we call examples chosen to stimulate uses to state preferences *suggestions*. We present two suggestion strategies: diversity and model-based techniques.

The ATA system was the first to show suggestions (Linden, Hanks, and Lesh 1997), which were extreme-valued examples where some attributes, for example departure time or price, took extreme values such as earliest or cheapest. However, a problem with such a technique is that extreme options are not likely to appeal to many users. For example, a user looking for a digital camera with good resolution might not want to consider a camera that offers four times the usual resolution but also has four times the usual weight and price. In fact, a tool that suggests such an option will discourage the user from even asking for such a feature, since it implies that high resolution can only be at the expense of many other disadvantages.

Thus, it is better to select the suggestions among examples that are already good, given the currently known preferences, and focus on showing diverse rather than extreme examples. Bradley and Smyth (2001) were the first to recognize the need to recommend diverse examples, especially in the early stage of using a recommender tool. They proposed the bounded greedy algorithm for retrieving the set of cases most similar to a user's query but at the same time most diverse among themselves. Thus, instead of picking the k best examples according to the preference ranking $r(x)$, a measure $d(x,Y)$ is used to calculate the relative diversity of an example x from the already selected set Y according to a weighted sum

$$s(x,Y) = \alpha\, r(x) + (1-\alpha)\, d(x,Y) \quad (2)$$

where α can be varied to account for varying importance of optimality and diversity. For example, as users approach the final target, α can be set to a higher value (such as 0.75 in their experiment setup) so that the system privileges the display set's similarity rather than diversity. In their implementations, the ranking $r(x)$ is the similarity $sim(x,t)$ of x to an ideal example t on a scale of 0 to 1, and the relative diversity is derived as

$$d(x,Y) = 1 - \sum_{y \in Y} sim(x,y)$$

The performance of diversity generation was evaluated in simulations in terms of its relative benefit, that is, the maximum gain in diversity achieved by giving up similarity (Smyth and McClave 2001). Subsequently, David McSherry (2002) has shown that diversity can often be increased without sacrificing similarity. A threshold t was fixed on the ranking function, and then a maximally diverse subset among all products x for which $r(x) > t$ was selected. When k options are shown, the threshold might be chosen as the value of the k-th best option, thus allowing no decrease in similarity, or at some value that does allow a certain decrease.

We thus propose the following guideline:

Guideline 2: Consider showing diverse examples to stimulate preference expression, especially when users are still uncertain about their final preferences.

The adaptive search algorithm used in McGinty and Smyth (2003) alternates between a strategy that privileges similarity and one that privileges diversity to implement the interaction "show me more like this" by varying the α in the ranking measure. At each point, a set of example products is displayed, and the user is instructed to choose his or her most preferred option among them. Whenever the user chooses the same option twice consecutively, the system considers diversity when proposing the next examples in order to refocus the search. Otherwise, the system assumes that the user is making progress and it continues to suggest new options based on optimality. Evaluations with simulated users show that this technique is likely to reduce the length of the recommendation cycles by up to 76 percent compared to the pure similarity-based recommender.

More recent work on diversity was motivated by the desire to compensate for users' preference uncertainty (Price and Messinger 2005) and to cover different topic interests in collaborative filtering recommenders (Ziegler et al. 2005). For general preference models, it is less clear how to define a diversity measure. Pu, Viappiani, and Faltings (2006) considered the user's motivation to state additional preferences when a suggestion is displayed. A user is likely to be opportunistic and will only bother to formulate new preferences if she or he believes that this might lead to a better choice. Thus, they propose the following *look-ahead principle* (Pu, Viappiani, and Faltings 2006):

Guideline 3: Consider suggesting options that may not be optimal under the current preference model but have a high likelihood of optimality when additional preferences are added.

The look-ahead principle can be applied to constructing model-based suggestions by explicitly computing, for each attribute a_i, a difference measure $diff(a_i, x)$ that corresponds to the probability that a preference on this attribute would make option x most preferred. Items are then ranked according to the expected difference measure over all possible attributes:

$$F_a(x) = \sum_{a_i \in A} P_{a_i} diff(a_i, x) \tag{3}$$

where P_{ai} is the probability that the user is motivated to state a preference on attribute a_i. Such probabilities are summed over all attributes for which the user has not yet expressed a preference. The best suggestions to display are therefore those items possessing the highest probability of becoming optimal after considering hidden preferences.

To confirm the importance of suggestions in producing accurate decisions, several user studies were carried out (Pu, Viappiani, and Faltings 2006; Viappiani et al. 2005). One was conducted in an unsupervised setting, where the user's behavior was monitored on a publicly accessible online system. They collected logs from 63 active users who went through several cycles of preference revision. Another study was conducted in a supervised setting. Forty volunteers were recruited and divided into two groups. One group evaluated the interface with model-based suggestions, and another group evaluated the one without. Both user studies showed that users who used the suggestion interfaces stated significantly more preferences than those who did not and also reached significantly higher decision accuracy.

Preference Revision

After the system recommends a set of example products to a user, the process of preference revision is to change one or more desired characteristics of a product that a user has stated previously, the degree to which such characteristics should be satisfied, or any combination of the two. According to user studies reported in Pu and Chen (2005), every user changes at least one initial preference during the entire search process for finding a product. Many users change preferences because there is rarely an outcome that satisfies all of the initial preferences. Two frequently encountered cases often require preference revision: (1) when a user cannot find an outcome that satisfies all of his or her stated preferences and must choose a partially satisfiying one, or (2) when a user has too many possibilities and must further narrow down the space of solutions. Even though both activities can be treated as the process of query refinement, the real challenge is to help users specify the correct query in order to find the target item. Here we present a unified framework of treating both cases as a *trade-off process* because finding an acceptable solution requires choosing an outcome that is desirable in some respects but perhaps not so attractive in others.

Preference Conflicts and Partial Satisfaction

A user who inputs a query for cameras with high resolution and a low price range and obtains "nothing found" as a reply learns very little about how to state more suitable preferences.

The current industry practice manages preference conflicts by browsing-based interaction techniques. A user is only allowed to enter preferences one at a time starting from the point where the entirety of the product space is available. As the user specifies more preferences, she or he essentially drills down to a subproduct space until either target is selected in the displayed options or no more product space is left. This interaction style has become very popular in comparison shopping websites.[1, 2, 3] Although the system designers have promptly prevented users from specifying conflicting preferences, this interaction style is very limited. Users are unable to specify contextual preferences and especially trade-offs among several attributes. If a user enters the set of preferences successively for each attribute, the space of matching products could suddenly become null with the message "no matching products can be found." At this point, the user may not know which attribute value to revise among the set of values that she or he has specified so far, except backtracking several steps and trying different combinations of preference values on the concerned attributes.

A more sensible method, such as the one used in SmartClient (Pu and Faltings 2000; Torrens, Faltings, and Pu 2002), manages a user's preference conflicts by first allowing the user to state all of his or her preferences and then showing the options that maximally satisfy subsets of the stated preferences based on partial constraint satisfaction techniques (Freuder and Wallace 1992). These maximally satisfied products educate users about available options and facilitate them in specifying more reasonable preferences. In the same spirit, McCarthy and colleagues propose to educate users about product knowledge by explaining the products that do exist instead of justifying why the system failed to produce a satisfactory outcome (McCarthy et al. 2004). FindMe systems rely on the background information from the product catalog and explain the preference conflicts at a higher level (Burke, Hammone, and Cooper 1996; Burke, Hammond, and Young 1997). In the case of a user wanting both a fuel-efficient and high-powered car, FindMe attempts to illustrate the trade-off between horsepower and fuel efficiency. This method of showing partially satisfied solutions is also called soft navigation by Stolze (2000). More recently, Binshtok and colleagues (2007) addressed the problem of computing an optimal subset given a preference specification through a search-over-CSPs algorithm. DesJardins and Wagstaff (2005) proposed to identify the best subset based on a so-called DD-PREF language by which users can specify feature-based preferences over sets of objects.

To convince users of the partially satisfied results, we can also adopt the explanation approach by clarifying in detail how the system satisfies some of the users' preferences and not others. A qualitative user survey about such explanation mechanisms was conducted in the form of a carefully constructed questionnaire, based on a series of hypotheses and corresponding applicable questions. Fifty-three subjects answered the survey, and most of them highly agreed that the explanation components are more likely to inspire their trust in the recommended solutions (Chen and Pu 2005). In addition, an alternative explanation technique, the organization interface where partially satisfied products are grouped into a set of categories (figure 1), was much preferred by most subjects compared to the traditional method, where each item is displayed along with an explanation construct (more details about explanation survey is in section "Explanation Interfaces").

Guideline 4: Consider resolving preference conflicts by showing partially satisfied results with compromises clearly explained to the user.

Trade-off Assistance

As catalogs grow in size, it becomes increasingly difficult to find the target item. Users may achieve relatively low decision accuracy unless a tool helps them efficiently view and compare many potentially interesting products. Even though a recommender agent is able to improve the decision quality by providing filtering and comparison matrix components (Haubl and Trifts 2000), a user can still face the bewildering task of selecting the right items to include in the consideration set.

Researchers found that online tools could increase the level of decision accuracy by up to 57 percent by helping users select and compare options that share trade-off properties (Pu and Chen 2005). Twenty-eight subjects attended the experiment, each of whom was first asked to make a choice and then use the decision aid tool to perform a set of trade-off navigation tasks. The results showed that after a user considers an item to be the final candidate, the tool can help him or her reach higher decision accuracy by prompting the user to see a set of trade-off alternatives. The same example-critiquing interfaces can be used to assist users to view trade-off alternatives, for example, "I like this camera, but can I find something lighter?" This style of interaction is called *trade-off navigation.*

Tweaking (used in FindMe (Burke et al. 1996, 1997)) was the first tool to implement this trade-off assistance. It was originally designed to help users navigate to their targets by modifying stated preferences one at a time. Reilly and colleagues (2004) introduced another style of trade-off support with *dynamic critiquing* methods. Critiques are directional feedback at the attribute level that users can select in order to improve a system's recommendation accuracy. For example, after recommending a model of the Canon digital cameras, the system may display "we have more matching cameras

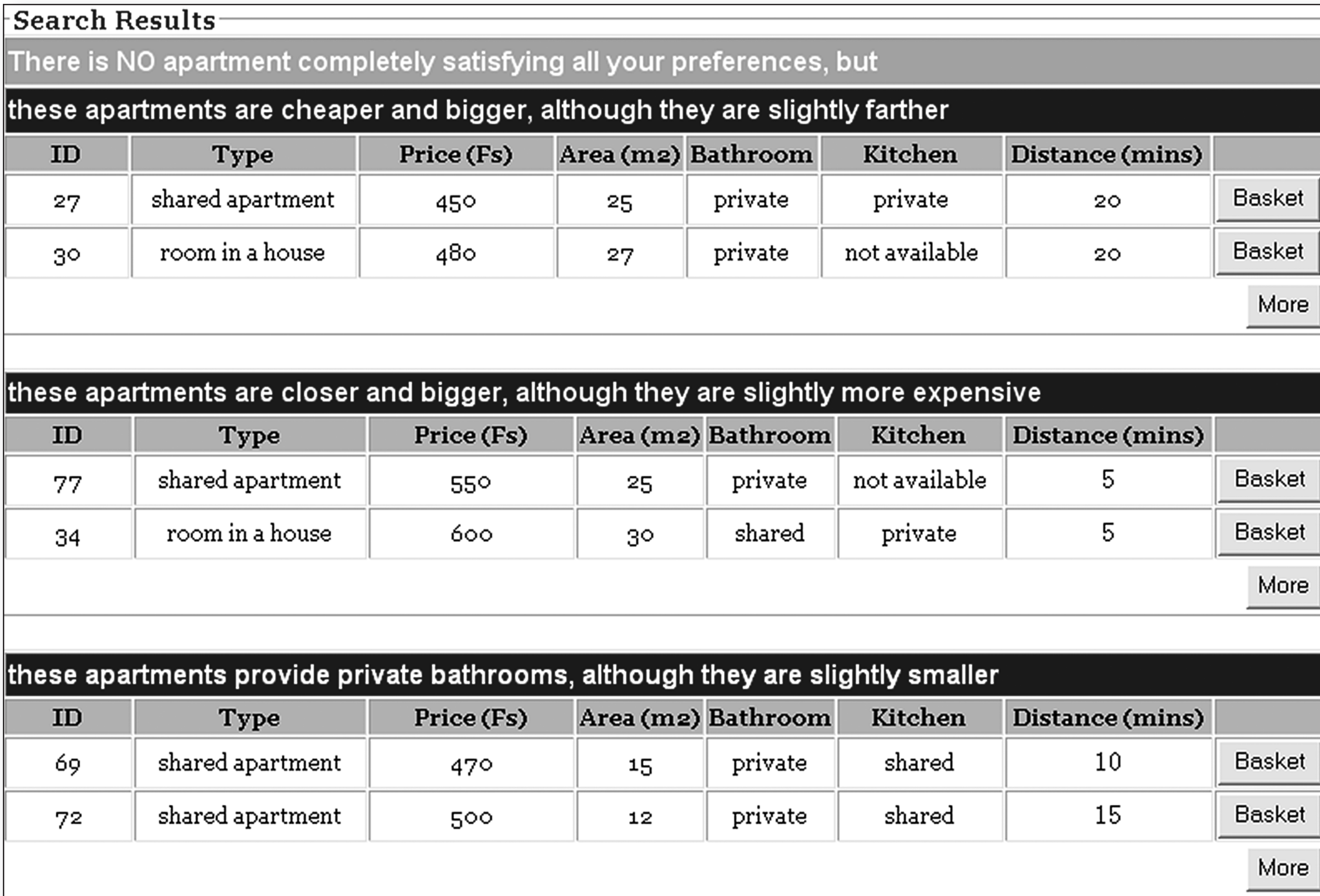

Figure 1. Partially Satisfied Products in an Organization Interface.

with the following: (1) less optimal zoom and thinner and lighter weight; (2) different manufacturer and lower resolution and cheaper; and (3) larger screen size and more memory and heavier."

Although originally designed to support navigation in recommender systems, the unit and compound critiques described in Reilly et al. (2004) correspond to the simple and complex trade-offs defined in Pu and Kumar (2004). They are both mechanisms to help users compare and evaluate the recommended item with a set of trade-off alternatives. However, the dynamic-critiquing method provides system-proposed trade-off support because it is the system that produces and suggests the trade-off categories (see figure 2), whereas example critiquing provides a mechanism for users to initiate their own trade-off navigation (called *user motivated critiques* in Chen and Pu [2006]; see figure 3).

A recent study compared the performance of user-motivated versus system-proposed approaches (Chen and Pu 2006). A total of 36 volunteers participated in the experiment. It was performed in a within-subjects design, and each participant was asked to evaluate two interfaces with the respective two approaches one after the other. Three evaluation criteria were used: decision accuracy, user interaction effort, and user confidence. The results indicate that the user-motivated trade-off method enables users to achieve a higher level of decision accuracy with less cognitive effort mainly due to its flexibility in allowing users to freely combine unit and compound critiques. In addition, the confidence in choice made with the user-motivated critique method is higher, resulting in users' increased intention to purchase the product they have found and return to the agent in the future. We thus propose:

> *Guideline 5:* In addition to providing the search function, consider providing users with trade-off assistance in the interface using either system-proposed or user-motivated approaches. The latter approach is likely to provide users with more flexibility in choosing their trade-off desires and thus enable them to achieve higher decision accuracy and confidence.

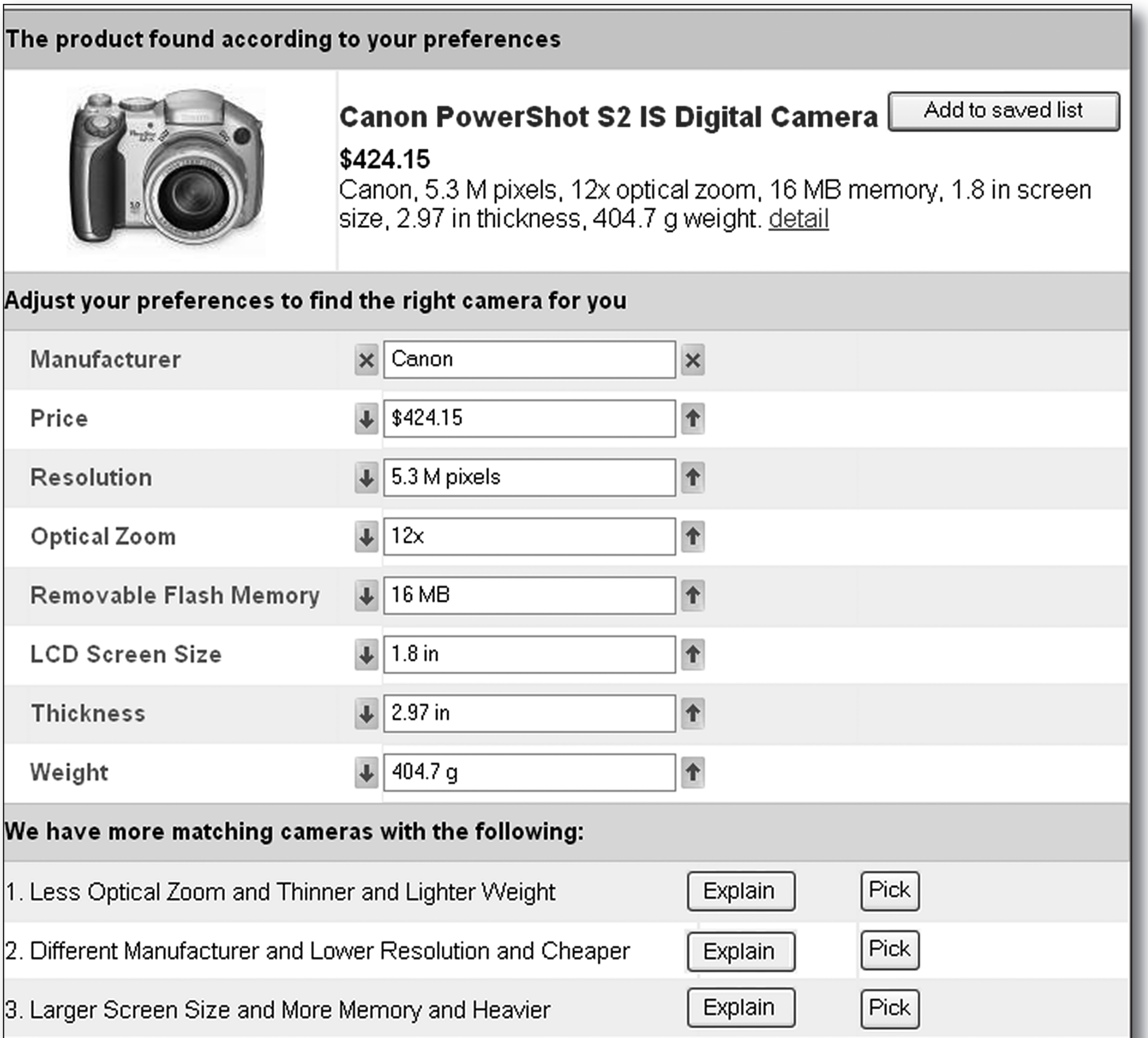

Figure 2. The Dynamic Critiquing Interface That Suggests a Set of Critiques for Users to Choose.

Explanation Interfaces

Being able to effectively explain results is essential for product recommender systems. When users face the difficulty of choosing the right product, the ability to convince them to consider a proposed item is an important goal of any recommender system in e-commerce environments. A number of researchers have started exploring the potential benefits of explanation interfaces in a number of directions.

Case-based reasoning recommender systems that can explain their recommendations include ExpertClerk (Shimazu 2001), dynamic critiquing systems (Reilly et al. 2004), and FirstCase and TopCase (McSherry 2003, 2004). ExpertClerk explained the selling point of each sample in terms of its difference from two other contrasting samples. In a similar way, FirstCase can explain why one case is more highly recommended than another by highlighting the benefits it offers and also the compromises it involves with respect to the user's preferences. In TopCase, the relevance of any question the user is asked can be explained in terms of its ability to discriminate between competing cases. Some consumer decision support systems with explanation interfaces can be found on commercial websites such as Logical Decisions[4] Active Decisions (see figure 4) and Yahoo SmartSort.[5]

Researchers also reported results from evaluating explanation interfaces with real users. Herlocker, Konstan, and Reidl (2000) addressed explanation interfaces for recommender systems using ACF (automated collaborative filtering) algorithms and demonstrated that a histogram by grouping neighbor ratings was the most compelling explanation component among the studied users. They maintain that providing explanations can improve the

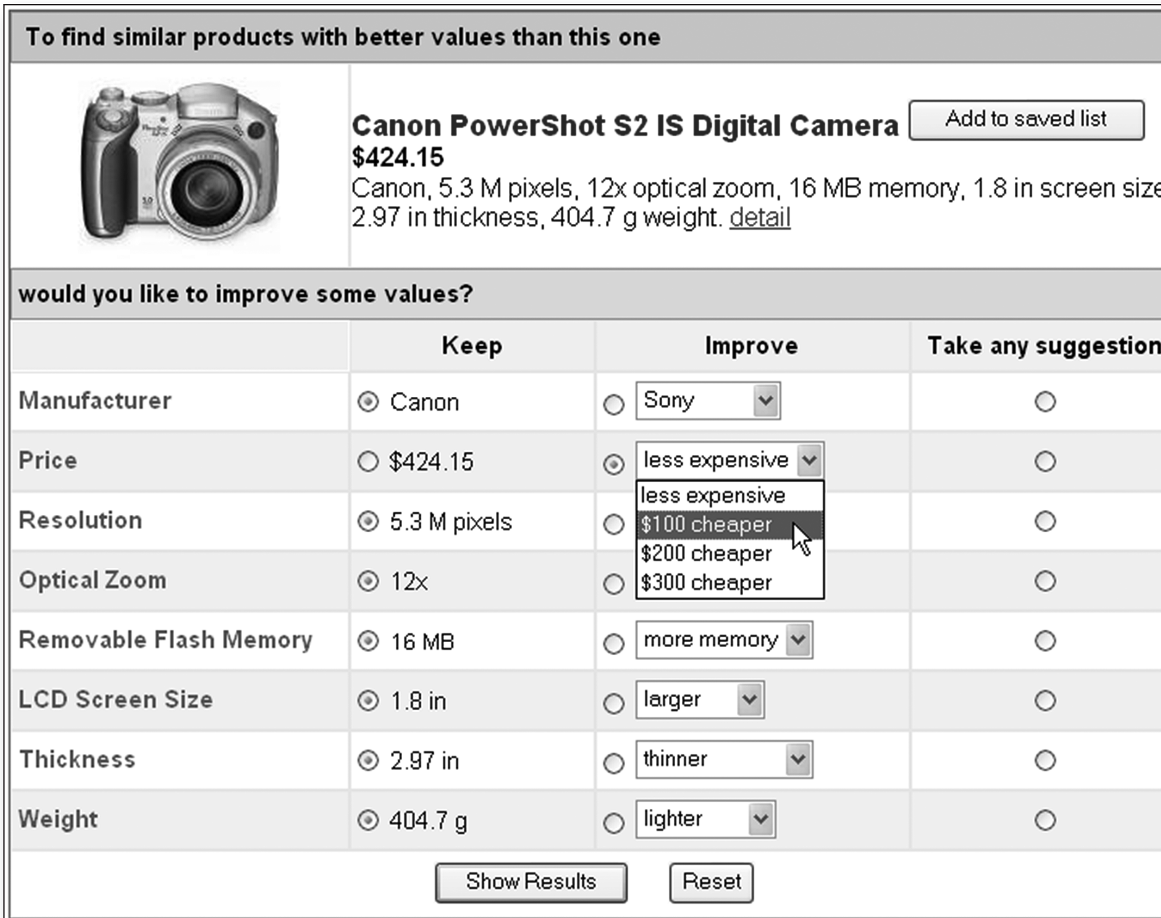

Figure 3. The Example Critiquing Interface Where Users Specify Critiques on Their Own.

acceptance of ACF systems and potentially improve users' filtering performance. Sinha and Swearingen (2002) found that users like and feel more confident about recommendations that they perceive as transparent.

A recent significant-scale empirical study (72 subjects) further evaluated the ability of an organization-based explanation interface, where recommended products are grouped into multiple categories and each category is labeled with a title explaining its contained products' similar characteristics (Pu and Chen 2006). The study revealed that the organization technique can significantly more effectively inspire users' trust and enhance their intention to save cognitive effort and use the interface again in the future, compared to traditional explanation methods (a "why" component along with each recommendation). Moreover, the study found that the actual time spent looking for a product did not have a significant impact on users' subjective perceptions. This indicates that less time spent on the interface, while very important in reducing decision effort, cannot be used alone in predicting what users may subjectively experience. Effective design principles for the organization interface were established, and an algorithm was presented for generating the content of such interfaces. Here we propose:

> *Guideline 6:* Consider designing interfaces that are capable of explaining how ranking scores are computed, because they are likely to inspire user trust.

Conclusion

A number of researchers from both behavioral and qualitative decision theory have pointed out the advantage of eliciting user preferences as they construct them, thus suggesting an incremental and interactive user system interaction process for product search and recommender systems. We have shown a detailed analysis of this process and developed a set of interaction design guidelines aimed at enabling users to state hidden preferences, revise conflicting preferences, and gain a better understanding of the available options and the recommended products through explanation interfaces. We have selected techniques, most of which have been validated through empirical studies, to demonstrate how to implement the guide-

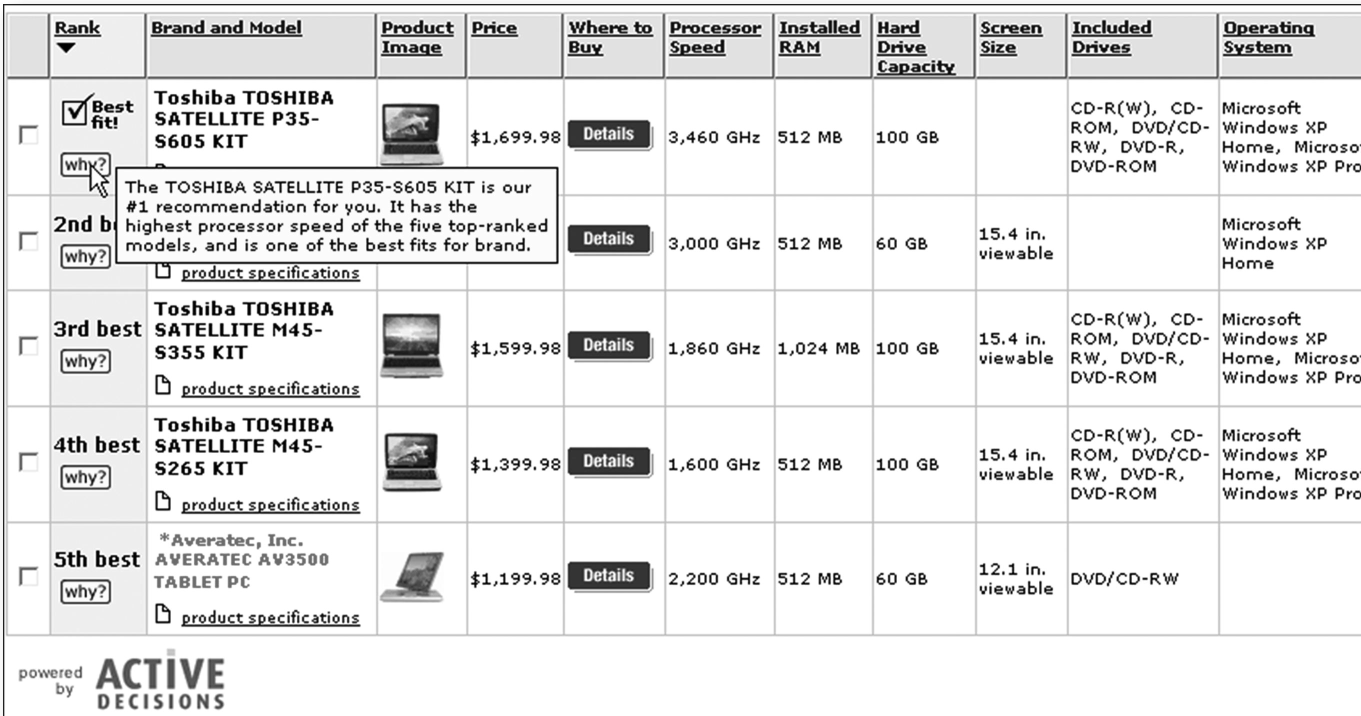

Figure 4. The Recommendation Interface with a "Why" Component for Each Displayed Product (powered by Active Decisions).

lines. Emphasis was given to those techniques that achieve a good balance on three criteria: increased decision accuracy, low user interaction effort, and high user confidence. Adopting these guidelines and the design approaches should significantly increase the benefits for users such as increased preference certainty and decision confidence, and a significantly reduced effort in searching for their preferred products. Moreover, we believe that most of these guidelines will be applicable for other application domains that involve an explicit procedure to acquire user preferences elicitation and refinement, including critiquing collaborative filtering-based recommender systems, adaptive user interfaces, and personal assistant agents (as surveyed by Peintner, Viappiani, and Yorke-Smith in this issue).

Notes

1. www.shopping.com.
2. www.pricegrabber.com.
3. www.yahoo.shopping.com.
4. www.logicaldecisions.com.
5. shopping.yahoo.com/smartsort.

References

Agrawal, R.; Imielinski, T.; and Swami, A. 1993. Mining Association Rules between Sets of Items in Large Databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, 207–216. New York: ACM Press.

Binshtok, M.; Brafman, R. I.; Shimony, S. E.; Mani, A.; and Boutilier, C. 2007. Computing Optimal Subsets. In *Proceedings of the 22nd National Conference on Artificial Intelligence* (AAAI-07), 1231–1236. Menlo Park, CA: AAAI Press.

Bradley, K., and Smyth, B. 2001. Improving Recommendation Diversity. In *Proceedings of the 12th Irish Conference on Artificial Intelligence and Cognitive Science*, 85–94. Berlin: Springer-Verlag.

Burke, R.; Hammond, K.; and Cooper, E. 1996. Knowledge-Based Navigation of Complex Information Spaces. In *Proceedings of the 13th National Conference on Artificial Intelligence* (AAAI-96), 462–468. Menlo Park, CA: AAAI Press.

Burke, R.; Hammond, K.; and Young, B. 1997. The Find-Me Approach to Assisted Browsing. *IEEE Expert: Intelligent Systems and Their Applications* 12(4): 32–40.

Chen, L., and Pu, P. 2005. Trust Building in Recommender Agents. In *Proceedings of the First Workshop on Web Personalization, Recommender Systems, and Intelligent User Interfaces* (WPRSIUI'05), in conjunction with *the 2nd International Conference on E-Business and Telecommunication Networks* (ICETE'05), 135–145. Portugal: INSTICC Press.

Chen, L., and Pu, P. 2006. Evaluating Critiquing-Based Recommender Agents. In *Proceedings of the 21st National Conference on Artificial Intelligence* (AAAI-06), 157–162. Menlo Park, CA: AAAI Press.

DesJardins, M., and Wagstaff, K. 2005. DD-PREF: A Language for Expressing Preferences over Sets. In *Proceedings of the 20th National Conference on Artificial Intelligence* (AAAI-05), 620–626. Menlo Park, CA: AAAI Press.

Faltings, B.; Pu, P.; Torrens, M.; and Viappiani, P. 2004.

Designing Example-Critiquing Interaction. In *Proceedings of the 9th International Conference on Intelligent User Interfaces* (IUI'04), 22–29. New York: ACM Press.

Freuder, E. C., and Wallace, R. J. 1992. Partial Constraint Satisfaction. *Artificial Intelligence* 58(1–3): 21–70.

Goker, M., and Thompson, C. 2000. The Adaptive Place Advisor: a Conversational Recommendation System. Paper presented at the 8th German Workshop on Case Based Reasoning, March 2–3, 2000, Lammerbuckel, Germany.

Haubl, G., and Trifts, V. 2000. Consumer Decision Making in Online Shopping Environments: The Effects of Interactive Decision Aids. *Marketing Science* 19(1): 4–21.

Herlocker, J. L.; Konstan, J. A.; and Riedl, J. 2000. Explaining Collaborative Filtering Recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work* (CSCW'00), 241–250. New York: ACM Press.

Keeney, R. 1992. *Value-Focused Thinking: A Path to Creative Decision Making*. Cambridge, MA: Harvard University Press.

Linden, G.; Hanks, S.; and Lesh, N. 1997. Interactive Assessment of User Preference Models: The Automated Travel Assistant. In *Proceedings of the 6th International Conference on User Modeling* (UM'97), 67–78. New York: Springer.

McCarthy, K.; McGinty, L.; Smyth, B.; and Reilly, J. 2005a. A Live-User Evaluation of Incremental Dynamic Critiquing. In *Proceedings of the 6th International Conference on Case-Based Reasoning* (ICCBR'05), 339–352. Berlin: Springer.

McCarthy, K.; Reilly, J.; McGinty, L.; and Smyth, B. 2005b. Experiments in Dynamic Critiquing. In *Proceedings of the 10th International Conference on Intelligent User Interfaces* (IUI'05), 175–182. New York: ACM Press.

McCarthy, K.; Reilly, J.; McGinty, L.; and Smyth, B. 2004. Thinking Positively—Explanatory Feedback for Conversational Recommender Systems. In *Proceedings of the Workshop on Explanation in CBR at the 7th European Conference on Case-Based Reasoning* (ECCBR'04), 115–124. Berlin: Springer-Verlag.

McCarthy, K.; Reilly, J.; McGinty, L.; and Smyth, B. 2005. Experiments in Dynamic Critiquing. In *Proceedings of the 10th International Conference on Intelligent User Interfaces* (IUI'05), 175–182. New York: ACM Press.

McGinty, L., and Smyth, B. 2003. On the Role of Diversity in Conversational Recommender Systems. In *Proceedings of the 5th International Conference on Case-Based Reasoning* (ICCBR'03), 276–290. Berlin: Springer.

McSherry, D. 2002. Diversity-Conscious Retrieval. In *Proceedings of the 6th European Conference on Advances in Case-Based Reasoning*, 219–233. London: Springer-Verlag.

McSherry, D. 2003. Similarity and Compromise. In *Proceedings of International Conference on Case-Based Reasoning Research and Development*, 291–305. Berlin: Springer-Verlag.

McSherry, D. 2004. Explanation in Recommender Systems. In *Workshop Proceedings of the 7th European Conference on Case-Based Reasoning* (ECCBR'04), 125–134. Berlin: Springer-Verlag.

Payne, J. W.; Bettman, J. R.; and Schkade, D. A. 1999. Measuring Constructed Preferences: Towards a Building Code. *Journal of Risk and Uncertainty* 19(1–3): 243–270.

Peinter, Viappiani, Yorke-Smith. 2008. Preferences in Interactive Systems: Technical Challenges and Case Studies. *AI Magazine* 29(4): 13–24.

Price, B., and Messinger, P. R. 2005. Optimal Recommendation Sets: Covering Uncertainty over User Preferences. In *Proceedings of the 20th National Conference on Artificial Intelligence* (AAAI'05), 541–548. Menlo Park, CA: AAAI Press.

Pu, P., and Chen, L. 2005. Integrating Tradeoff Support in Product Search Tools for E-Commerce Sites. In *Proceeding of the 6th ACM Conference on Electronic Commerce* (EC'05), 269–278. New York: ACM Press.

Pu, P., and Chen, L. 2006. Trust Building with Explanation Interfaces. In *Proceedings of the 11th International Conference on Intelligent User Interface* (IUI'06), 93–100. New York: ACM Press.

Pu, P., and Faltings, B. 2000. Enriching Buyers' Experiences: the SmartClient Approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI'00), 289–296. New York: ACM Press.

Pu, P., and Kumar, P. 2004. Evaluating Example-Based Search Tools. In *Proceedings of the Fifth ACM Conference on Electronic Commerce* (EC'04), 208–217. New York: ACM Press.

Pu, P.; Viappiani, P.; and Faltings, B. 2006. Stimulating Decision Accuracy Using Suggestions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI'06), 121–130. New York: ACM Press.

Reilly, J.; McCarthy, K.; McGinty, L.; and Smyth, B. 2004. Dynamic Critiquing. In *Proceedings of the Seventh European Conference on Case-Based Reasoning* (ECCBR'04), 763–777. Berlin: Springer-Verlag.

Shimazu, H. 2001. ExpertClerk: Navigating Shoppers' Buying Process with the Combination of Asking and Proposing. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence* (IJCAI'01), 1443–1448. San Francisco: Morgan Kaufmann Publishers.

Sinha, R., and Swearingen, K. 2002. The Role of Transparency in Recommender Systems. In *Extended Abstracts of the SIGCHI Conference on Human Factors in Computing Systems* (CHI'02), 830–831. New York: ACM Press.

Smyth, B., and McClave, P. 2001. Similarity vs. Diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning* (ICCBR'01), 347–361. Berlin: Springer.

Stolze, M. 2000. Soft Navigation in Electronic Product Catalogs. *International Journal on Digital Libraries* 3(1): 60–66.

Torrens, M.; Faltings, B.; and Pu, P. 2002. SmartClients: Constraint Satisfaction as a Paradigm for Scalable Intelligent Information Systems. *International Journal of Constraints* 7(1): 49–69.

Tversky, A. 1996. Contrasting Rational and Psychological Principles in Choice. In *Wise Choices: Decisions, Games, and Negotiations,* ed. R. Richard, J. Zeckhauser, and J. Sebenius, 5–21. Boston, MA: Harvard Business School Press.

Viappiani, P.; Faltings, B.; Schickel-Zuber, V.; and Pu, P. 2005. Stimulating Preference Expression Using Suggestions. *Mixed-Initiative Problem-Solving Assistants, AAAI Fall Symposium Series*, 2005, 128–133. Menlo Park, CA: AAAI Press.

Williams, M. D., and Tou, F. N. 1982. RABBIT: An Inter-

face for Database Access. In *Proceedings of the ACM '82 Conference*, 83–87. New York: ACM Press.

Ziegler, C. N.; McNee, S. M.; Konstan, J. A.; and Lausen, G. 2005. Improving Recommendation Lists through Topic Diversification. In *Proceedings of the 14th International World Wide Web Conference* (WWW'05), 22–32. New York: ACM Press.

Pearl Pu obtained her master's and Ph.D. degrees from the University of Pennsylvania in artificial intelligence and computer graphics. She was a visiting scholar at Stanford University in 2001, both in the database and HCI groups. She currently leads the HCI Group in the School of Computer and Communication Sciences at the Swiss Federal Institute of Technology in Lausanne (EPFL). Her research is multidisciplinary and focuses on issues in the intersection of human computer interaction, artificial intelligence, and behavioral decision theories. She works on preference-based search in online environments. She was among the first to show why and how existing product search tools, such as those used in e-commerce websites, do not always allow users to find their most preferred items. Her other research interests are concerned with decision support systems, electronic commerce, online consumer decision behavior, product recommender systems, content-based product search, travel planning tools, trust-inspiring interfaces for recommender agent, music recommenders, and user technology adoption. She is the associate editor for the *IEEE Transactions on Multimedia,* the general chair of the 2008 ACM Conference on Recommender Systems, program cochair of the 2008 International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, publicity and tutorial chairs of the ACM Conference on Intelligent User Interfaces (IUI) in 2008 and 2007, respectively, and program committee member of a number of highly regarded conferences such as the ACM Conference on E-Commerce, AAAI Conference on Artificial Intelligence, Intelligent User Interface conference, and European AI conference.

Li Chen obtained her Ph.D. degree in the Human Computer Interaction (HCI) Group, School of Computer and Communication Sciences at Swiss Federal Institute of Technology in Lausanne (EPFL). She has participated in a Swiss National Science Foundation project and worked on user issues for online decision systems in e-commerce environments, and coauthored a number of publications in domains of e-commerce, artificial intelligence, intelligent user interfaces, user modeling (UM'07), and recommender systems. She won the Best Student Paper award in 2007 at the International Conference on User Modeling. She has been also invited as a reviewer for a number of journals and conferences (IJEC, TOCHI, IEEE IS, UM, and IUI). Before joining the HCI group in EPFL, she obtained her bachelor's and master's degrees in computer science at Peking University. During her graduate study, she worked as a research assistant in the database lab, actively participating in two national key and fundamental projects. Both were related to the development of advanced database technologies to serve for mobile devices and online massive content integration and retrieval.