Electron Commerce Res (2008) 8: 1–27
DOI 10.1007/s10660-008-9015-z

Evaluating product search and recommender systems for E-commerce environments

Pearl Pu · Li Chen · Pratyush Kumar

Published online: 7 May 2008
© Springer Science+Business Media, LLC 2008

Abstract Online systems that help users select the most preferential item from a large electronic catalog are known as product search and recommender systems. Evaluation of various proposed technologies is essential for further development in this area. This paper describes the design and implementation of two user studies in which a particular product search tool, known as *example critiquing*, was evaluated against a chosen baseline model. The results confirm that example critiquing significantly reduces users' task time and error rate while increasing decision accuracy. Additionally, the results of the second user study show that a particular implementation of example critiquing also made users more confident about their choices. The main contribution is that through these two user studies, an evaluation framework of three criteria was successfully identified, which can be used for evaluating general product search and recommender systems in E-commerce environments. These two experiments and the actual procedures also shed light on some of the most important issues which need to be considered for evaluating such tools, such as the preparation of materials for evaluation, user task design, the context of evaluation, the criteria, the measures and the methodology of result analyses.

This research was supported by the Swiss National Science Foundation.

P. Pu (✉) · L. Chen
Human Computer Interaction Group, School of Computer and Communication Sciences, Swiss Federal Institute of Technology in Lausanne (EPFL), 1015 Lausanne, Switzerland
e-mail: pearl.pu@epfl.ch

L. Chen
e-mail: li.chen@epfl.ch

P. Kumar
Business Administration, Darden Graduate School of Business, University of Virginia, Charlottesville, USA
e-mail: KumarP08@darden.virginia.edu

Keywords Preference-based search · Product recommender systems · Example critiquing interfaces · Decision technology · Electronic product catalog · Tradeoff analysis · Fisheye view interfaces

1 Introduction

Whether choosing the right job candidate to hire among a dozen applicants, selecting a software contractor to outsource your IT work, or determining which digital camera to purchase, every decision maker is likely to employ one of two approaches [22]. In the first case, we try to achieve high decision accuracy, but we face the very time consuming task of sifting through all k options and trading off the pros and cons between multiple attributes. Alternatively, we can adopt heuristic decision strategies and process information more selectively. Although we expend less effort, these heuristic strategies can lead to decision errors and are likely to cause decision regrets. Clearly, neither approach is ideal, since adopting one or the other implies a compromise on either decision accuracy or effort. According to [22], the tradeoff between accuracy and effort is an inherent dilemma in decision-making that cannot be easily reconciled.

We investigate the problem of users searching for the most preferred item in an online catalog. This problem is generally known as preference-based multi-criteria product search. Systems that address this problem are recently being developed and employed in E-commerce environments to help users select and decide about their choices.[1] Even though many of them allow users to narrow down a large choice of products to a smaller set of k interesting candidates, making the final selection where k can be a very high number, especially in an E-commerce environment, remains a daunting task. The effort in processing all the product information in order to make accurate decisions can potentially be tremendous. Most of these systems do not satisfactorily address the effort and accuracy dilemma that decision makers face in the search process.

We propose a system, called *example critiquing*, which aims at assisting the user to not only find the product that she is looking for, but also help her process tradeoff information in order to achieve better decision accuracy. While looking for ways to find out if this tool indeed achieves high decision accuracy while lowering the barrier of effort for decision makers, we realized that this research domain lacks a coherent framework of evaluation techniques. Therefore, identifying the right criteria to evaluate the real benefits of these systems has become a priority. Indeed, evaluation of various proposed technologies is essential for further development in this area. Current methods such as precision and recall were adopted from the information retrieval domain, which only address the performance issues, not the interaction aspect, of the underlying search algorithm. More recently, researchers started developing more user-centric evaluation framework, known as the mean average error (MAE) mechanism, to measure the performance of rating based collaborative filtering recommender systems [14]. However, neither framework captures the accuracy vs. effort tradeoff that underlies the usability issue of product search: will users be

[1] See for example http://shopping.yahoo.com/.

more likely engaged in interacting with the tool, or will they turn away because the interaction effort demanded of them is too high.

This paper describes the design and implementation of two user studies in which our tool, the *example critiquing*, was evaluated against a chosen baseline model. Results show that example critiquing significantly reduces users' task time and error rate while requiring a lower level of interaction effort than the baseline model. Results of the second user study showed that a particular implementation of example critiquing also made users more confident about their choices.

The main contribution of this paper is that through these two user studies, an evaluation framework of three criteria was successfully identified: (1) the relative decision accuracy that a user is able to achieve via the use of the tool compared with the baseline model, (2) the effort that she expends on interacting with the tool also compared with the baseline model, and finally (3) decision confidence, which measures whether users are convinced of the products that were recommended to them. These three criteria and the evaluation methodology, which will be reported in detail later in this article, can serve as a general evaluation framework for product search and recommender systems used in E-commerce environments.

Another contribution of this paper is the identification of a baseline model to measure relative accuracy and effort. For a model to serve as the usability baseline we decided to look for a tool that is easy to use and is the current norm used in E-commerce websites. Our final selection went to the ranked list product search tool where a user search for her most preferred item by sorting on the list of products based on a set of criteria judged to be important to her. In the search for a digital camera case, the ranked list tool allows users to browse the entire catalog based on the price, the optical zoom, the weight, etc. In addition, users can browse a subspace of the catalog by specifying a range of values of a criterion. This model implements the lexicographical ordering decision strategy, which is known to be a low effort requiring and non-accurate heuristic strategy [22]. We reasoned that if a tool achieves higher accuracy, but requires less or the same amount of effort as the ranked list, it is likely to offer significant benefits to consumers in terms of decision accuracy and effort and therefore will motivate users to adopt the tool.

The first user study was reported in the ACM E-Commerce conference in 2004 [26]; a follow up user study, reported for the first time here, points out new developments relevant to the evaluation of recommender tools. A fisheye view interface capable of displaying more recommended items was able to significantly improve users' subjective confidence of their choices compared to the earlier example critiquing interface. Putting these two related user studies into a single publication is therefore essential to the coherent reporting of our work. Together, we learned that one way of designing a successful product search and recommender tool is to combine the use of critiquing features with a fisheye view interface in which more results are displayed and the recommended items are emphasized. These two experiments and the actual procedures also shed light on some of the most important issues which need to be considered for evaluating such tools, such as the preparation of materials for evaluation, user task design, the context of evaluation, the criteria, the measures and the methodology of result analyses.

This article is organized as follows: Sect. 2 discusses related work, Sect. 3 describes in detail our tool, the SmartClient system, based on the example critiquing

paradigm, Sect. 4 presents the results of the first user study, Sect. 5 then introduces the fisheye interface, Sect. 6 presents the user study of this interface as well as the analysis of results, and Sect. 7 concludes the article followed by acknowledgements.

2 Related work

We focus our attention on buyers using an electronic product catalog to find items that best suit their needs in an E-commerce environment, although these techniques can be applied in other decision and choice scenarios as well. We primarily consider the final product selection stage where k items have been selected and displayed by a recommender system. We briefly describe the background of multi-attribute product search and recommender systems, specify how they display search results, and compare related works in terms of ways to evaluate such systems and user benefits that have been established by previous evaluations.

2.1 Multi-attribute product search tools

An electronic product catalog is an online database containing well organized information about products and their features. Most E-commerce websites, such as Amazon (www.amazon.com), Expedia (www.expedia.com), or eBay (www.ebay.com), use such catalogs. A crucial element of these electronic catalogs is a search function that takes the customer's needs and preferences as input and returns a set of matched items. When products can be represented by the same set of attributes, a search tool often uses a utility model to determine the attractiveness (or utility) of an item based users' preference specification [16]. Items matching or partially matching the preference specification are displayed in a descending order of the utility scores. A commercial tool of this type can be found at www.activedecisions.com. These systems are also known as content-based recommendation systems [3], decision support interface systems [34], product search with personalized recommendation systems, and utility based product ranking systems [35, 40]. We will refer to them as multi-attribute product search tools (MAPST).

Determining a good match between a product and a user's product needs requires accurate information on users' preferences, known as the preference model. Thus, a crucial element in MAPST is a preference elicitation tool. Users' participation of the elicitation process varies depending on the effort expected of the user. In the simplest case, a system will try to recognize a new user as a member of a particular community of users and use the preference information previously gathered of that community to infer the needs and preferences of the newcomer [12, 30]. The most successful of these methods is called collaborative filtering. Herlocker et al. [14] has proposed an excellent evaluation framework for these so called community-based recommender systems. While very effective for low-risk products such as films and books, users are much less likely to accept an inferred preference structure from a community-based recommender system for choosing products which present high financial and emotional risks [6, 22]. For these so-called high involvement products, more refined preference models are favored which involve asking users to state their

needs and preferences up-front or interactively with varying degrees of effort. The non-incremental elicitation methods were found to be too difficult for novice users and often force them to focus on wrong decision objectives [25], and they will not be further discussed here. There are three types of methods which elicit users' preference models incrementally and interactively: the single-item critiquing-based, the k-item non critiquing based, and the k-item critiquing-based systems. A critiquing interface offers not only product recommendations, but also the possibility for users to provide critiquing feedback to improve the recommendation. Spiekermann and Paraschiv [34] offered a similar classification of MAPST based on the number of required search criteria. However, they did not include the treatment of critiquing-based systems. The feedback offered by the critiquing-based systems was in fact described as a major drawback lacking in the DSISs that they reviewed. Results presented here thus indicate a major development in the design and evaluation of MAPST.

2.1.1 Single-item critiquing-based MAPST

The FindMe [4] system was the first known single-item critiquing-based MAPST. It uses knowledge about the product domain to help users navigate through the multi-dimensional space by recommending one product at a time. An important interface element in FindMe is called *tweaking*, which enables users to navigate from an item to its tradeoff alternative and compare them. Via tweaking, a user trades off more of one valued attribute for less of another valued attribute. For example, by tweaking the current suggested restaurant on its price for a lower value, a user gets a recommendation of a cheaper restaurant that possibly has a less desirable ambiance. However, the evaluation of FindMe systems to quantify the benefits did not occur until 2004 [20, 21, 28, 29, 32].

Researchers evaluated several dynamic critiquing systems whose interface was based on the tweaking philosophy originally used in the FindMe systems. Dynamic critiquing systems continue to recommend items one at a time. However, for each recommended item, several sets of compound critiques are suggested to the user in addition to the unit critiques. For example, in the case of recommending a digital camera, the system suggests that "We have more matching cameras with the following." Then a compound critique such as "more memory and larger and heavier (134)" is displayed, suggesting that 134 digital cameras with more memory, but are larger and heavier are available in the catalog. A compound critique is therefore an item with improved values on one attribute and compromised values on two attributes. The evaluation of dynamic critiquing systems has mainly focused on the session length criterion, which is the number of times recommendations are given to users. One user study showed that participants who selected the compound critiques more frequently were able to reduce the interaction session length from an average of 28 recommendation cycles to 8.53 cycles.

According to Spiekermann and Paraschiv [34], minimizing consumer's time should not be the only design goal for MAPST. Minimizing purchase risk is equally important. Since decision accuracy is important in minimizing purchase risk, it should be measured at the same time. Therefore, evaluating dynamic critiquing systems based on session length alone may not indicate the fundamental user benefits.

2.1.2 k-item non critiquing-based MAPST

The simplest and most commonly used interface to show k-item in multi-attribute product search tools is a ranked list. Items are sorted and displayed in descending order of their relevance to users' current preference specification. The interfaces used by PersonaLogic [19], ScoreCat [35], Active Decisions (www.activedecisions.com), and shopping.yahoo.com (http://shopping.yahoo.com/) are some examples based on this approach. Notice that there are no critiquing components in these systems. Selecting the final winner requires users to exhaustively examine all potentially interesting candidates and perform tradeoffs manually.

More recently, researchers have shown the benefits of including a tool, called the comparison matrix (as is available on CompareNet website), in the ranked list interface [13]. This component facilitates users' tradeoff examination of products in the final stage of their search. As users view the recommended k items, they can manually select a set of finalists and compare them in a side-by-side matrix based on attribute values. Evaluation of such tools found that the comparison matrix augments the quality of the consideration sets and the quality of purchase decisions [13]. These benefits are crucial to the final selection process of product search. Compared to their work, we provide two pieces of additional knowledge. We identify the critiquing feature as yet another decision aid component which can be used on top of the comparison matrix. The critiquing interface (Fig. 2 and explained subsequently) assists users in identifying the relevant items to be included in the comparison matrix and therefore *lessons* the burden of completing this task manually. Secondly, we show that the critiquing component provides additional benefits to improve users' decision quality. The decision accuracy criterion that we use measures the amount of information a user processes to resolve tradeoffs among a set of options, whereas the decision quality used in [13] estimates the likelihood that users select non-dominated products. Therefore the accuracy criterion is an additional desirable property of decision quality. Other researchers have also argued for decision accuracy as a quality measure. Jedetski et al. [11] showed that the use of a side-by-side comparison matrix increases the likelihood that a user will employ more compensatory (i.e., accurate) decision strategies. However, they did not identify the component which *assists* users in selecting the items.

2.2 k-item critiquing-based MAPST

The ATA (Automated Travel Assistant) system [18] and our system, SmartClient, were examples of MAPST which provide all three components: a recommender agent that provides a set of k items that best match users' current preference model, a critiquing component that provides feedback to the recommender system and allows users to actively identify and confront a set of tradeoff alternatives, and a comparison matrix in the form of a multi-attribute basket to compare the finalists. ATA displays three items at a time, whereas SmartClient displayed 7 items in the most recent version. Users can select any of the displayed items and navigate to products that offer tradeoff potentials. Any interesting items viewed along the way can be kept in a basket for further comparison and final selection.

To our knowledge, no prior work has evaluated the k-item critiquing or the k-item non-critiquing based MAPST in terms of decision accuracy and effort. Since the ranked list interface implements the k-item non-critiquing based MAPST, and is the current accepted norm in E-commerce websites, we will use that as the baseline model, and measure the performance increase offered by the k-item critiquing based MAPST. More precisely we measure the benefits offered by the *critiquing* component.

3 Example critiquing interface

The example critiquing interaction paradigm, initially used in ATP [38], was developed around the same time as FindMe. Later on, ATP became SmartClient, an online product catalog for finding flights [24]. This method was subsequently applied to catalogs of vacation packages, insurance policies, and apartments.

The example critiquing interface (see Fig. 1) contains a critiquing module and a basket. It is powered by SmartClient's search engine which helps users narrow the product space down to a smaller consideration set. The critiquing module assists users to encounter and resolve tradeoff decisions, and provides feedback to improve the system's recommendation. The basket helps users memorize potentially interesting products and compare them side-by-side based on their attribute values. The

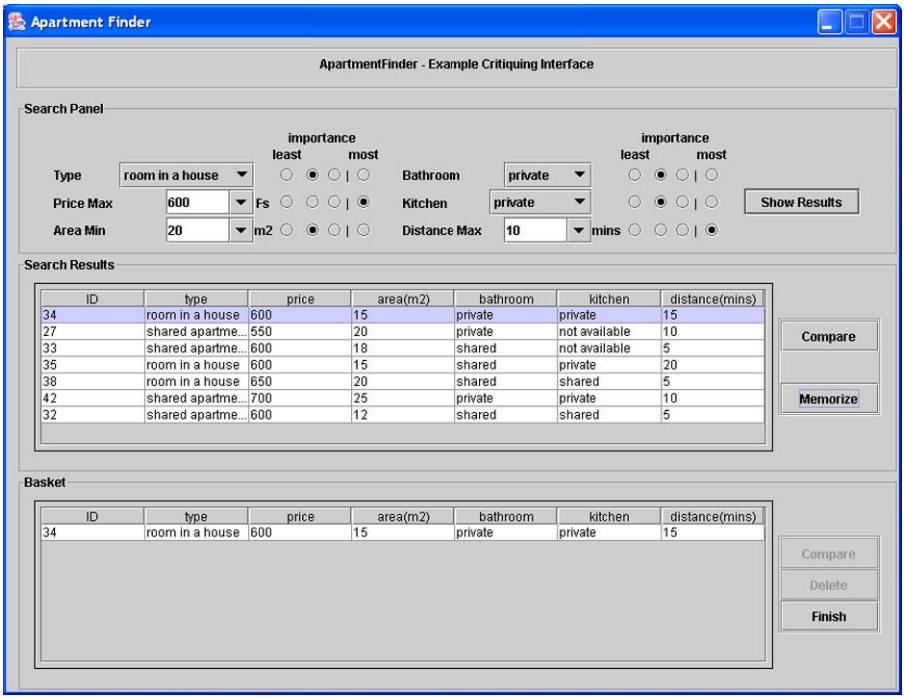

Fig. 1 Step one in example critiquing: system showing a set of 7 results after a user query

search engine can be a simple ranking function for simple products based on multiple attribute utility theory [16] as in the case for finding apartments. For configurable products, SmartClient employs more sophisticated constraint satisfaction algorithms and models user preferences as soft constraints [25]. More detail on SmartClient's architecture and the modeling of preferences for configurable products was provided in Torrens and Faltings [36] and Torrens et al. [37]. We focus our attention on the design and evaluation of the example critiquing interface in this paper.

3.1 How example critiquing interface works

A user starts the search by specifying one or any number of preferences in the query area. Based on this initial preference model, the search engine will find and display a set of matching results (see [7] for the optimal number of displayed solutions). He/she is able to revise his/her preferences if the displayed results are not satisfactory.

However, when a user is ready to select an apartment to put in the basket, the example critiquing interface will first show a pop-up window (see Fig. 2) where he/she can compare his/her current selection with others and perform tradeoff analysis. For example, suppose that the current selection is apartment 34. In the comparison window, the user can specify his/her desire for a bigger apartment by clicking on the checkbox

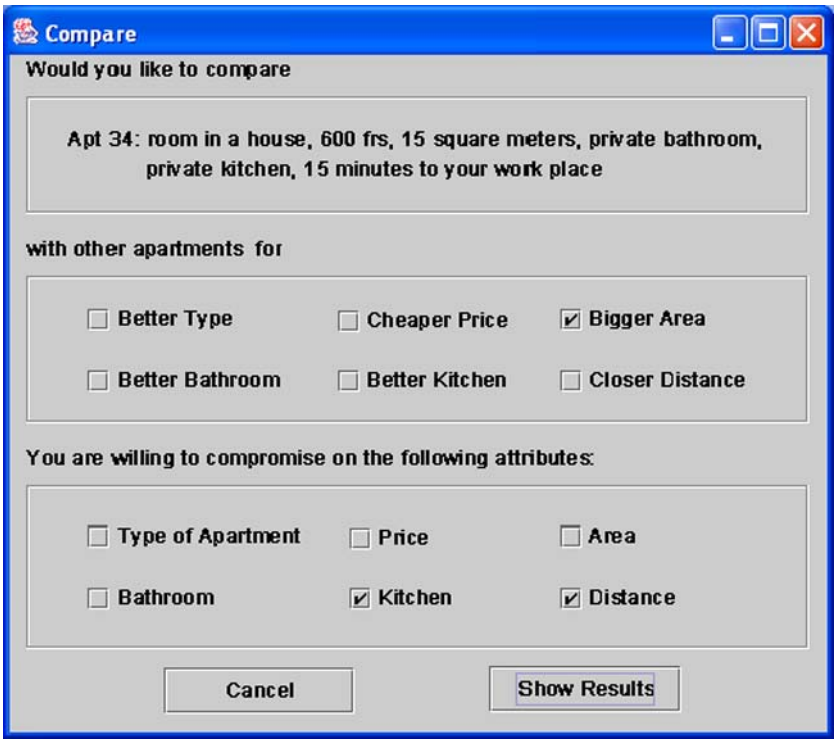

Fig. 2 Step two in example critiquing: guiding users to find tradeoff alternatives in the product comparison pop-up window

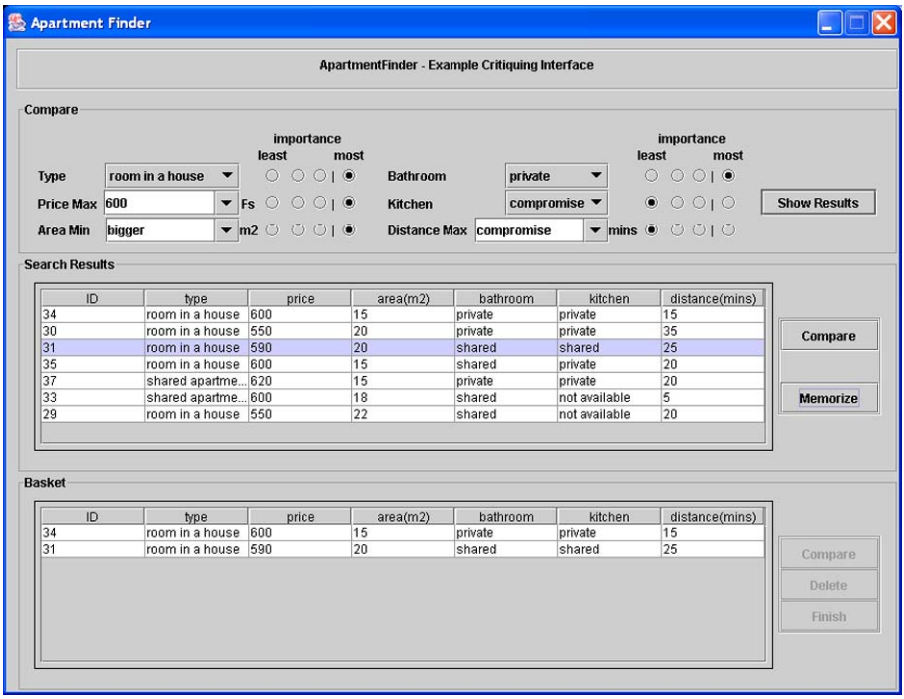

Fig. 3 Step three in example critiquing: the system showing tradeoff alternatives

next to the "bigger area" label. However, knowing that he/she may sacrifice something for a bigger apartment, he/she specifies "compromise" for both the distance and kitchen attributes by clicking on the checkboxes next to these two attributes. Compromise means that a user is willing to accept a lesser value of the respective attribute. Once a set of critiques has been composed, the system will show another set of matching examples (see Fig. 3). Apartment 31 seems quite interesting, since it is around the same price, but 5 square meters bigger, although it is 10 minutes more commuting time and the bathroom is shared. The system does not resolve tradeoffs for the user, but provides relevant information for him/her to understand the decision context. The final choice is left to the user.

This query/critiquing completes one cycle of interaction, which can continue if users change their preference structures. In our general observation, users perform on average 3 to 4 cycles. The "Compare" pop window will become accessible by clicking the "Compare" button and will not be forced on users when they put items in the basket.

4 Evaluating example critiquing interfaces

4.1 User task design

We compare the performance of the example critiquing interface against the ranked list interface focusing on decision accuracy and effort. We ask all of the participants

to find simple and complex tradeoff alternatives of a given item while using both interfaces that are to be compared. The design of these identification tasks is based on the fact that decision makers undertaking these tasks use compensatory strategies [15, 22] and are more likely to achieve higher decision accuracy. Therefore, the more complex tradeoff alternatives a user can identify, the more accurate his or her decision is; and the faster a user finds answers to tradeoff alternatives, the less effort he or she makes. We demonstrate that example critiquing significantly reduces users' task time and error compared to the ranked list when complex tradeoff tasks were performed. Before describing the procedure for conducting such a comparative user study, we define more precisely the notion of tradeoff tasks.

4.2 Tradeoff navigation

In finding tradeoff alternatives, a user explores the product space by navigating from one product to others that provide tradeoff scenarios. As explained before, the more tradeoff scenarios a user examines, the more accurate his/her decisions are. In the example critiquing interface, he/she starts the navigation from a recommended item after the initial search, and then posts a critique (e.g., a cheaper apartment) in order to see a new set of products. We call this process the tradeoff navigation process. More precisely, tradeoff navigation refers to a user navigating from an item to its tradeoff counterparts which offer improved values on one or several attributes, and compromised values on other attributes. This type of tradeoff is known as attribute value tradeoff. Pu and Faltings [25] discussed other types of decision tradeoffs in product search.

As the number of attributes becomes larger, the complexity of the tradeoff navigation task increases. Let us define each tradeoff navigation task as having two variables: (optimize, compromise), where *optimize* represents the set of attributes to be optimized, and *compromise* the set of attributes to be compromised. So ({price}, {size of room}) denotes that a user wants to get a better price by sacrificing the size of his/her room. ({price}, {size of room, distance to work}) denotes that the user wants to get a better price by sacrificing the size of his/her room, the distance to work, or both. Furthermore, we use pairs (x, y) to specify the complexity of tradeoff navigation tasks. $(1, 1)$ denotes that one attribute is being optimized, while at the same time another attributed is being compromised. $(1, 2)$ denotes the participation of two attributes for the compromising process, and one attribute for the optimization process. It is clear that $(1, 1)$ entails one single tradeoff scenario, while there are three scenarios for the $(1, 2)$ case because there are three ways to compromise the values of two attributes (two times on one attribute alone, and one time on both attributes). As the number of variables participating in a tradeoff scenarios increases, the optimize/compromise scenario pairs increase exponentially. For the case of $(1, 3)$, there are 7 optimize/compromise pairs. That is, there are 7 different ways to compromise the values of the three chosen attributes, thus 7 different ways to navigate.

4.3 Tradeoff identification tasks

Tradeoff navigation takes place most likely in the final stage of product selection where users are comparing and examining in detail a set of candidates [13]. Therefore

we assume that a user has narrowed down the search space to a smaller set. This provides the reasoning for having only 50 products in the test catalog.

Our goals were to instruct all of our users to perform tradeoff navigations using the example critiquing and ranked list interfaces respectively and to measure their task time and error rates. We started with a rather specific decision goal by asking them to identify their most preferred apartment encountered so far (the first choice). Then we asked everyone to improve various attributes of that apartment and evaluated how quickly they found tradeoff alternatives in relation to the first choice. The specific tasks given to our users were as follows:

1. Find your most preferred apartment.
2. Can you find something closer? You can compromise on one and only one attribute.
3. Can you find something bigger than what you found for question #1? You can compromise on one and only one attribute.
4. Find something which is roughly 100 Swiss francs less than the answer to question #1. You can compromise on up to two attributes, but not more.
5. Find an apartment which is 5 square meters bigger than the answer to question #1. You can compromise on up to two attributes but not more.

The questions can be broadly divided into three categories. The first question is a simple search task of finding a multi-attribute product from a list of products. This question on one hand ensures that we get an idea of the user's comfort level with the interfaces; it also gives us a starting point for answering subsequent tradeoff questions. The second category of questions (Questions 2, 3) deals with simple multi attribute tradeoff tasks with one attribute being optimized and the other compromised, i.e., the $(1, 1)$ tradeoff case. The third category of questions (Questions 4, 5) deals with complex multi attribute tradeoff tasks where the user gains on one attribute, and compromise on two attributes, i.e., the $(1, 2)$ case.

The entire user study was carried out in experiments scheduled in three phases, with 11, 5, and 6 participants involved in each of the phases respectively. The task completion time was defined to be the amount of time a participant took to answer each of the questions. The error rate was defined to be the total number of wrong answers a participant gave over the total number of questions.

4.4 Data set and participants

The data set originally used in SmartClient dealt with multi-attribute and configurable products in the travel industry. However, we chose to evaluate SmartClient for apartment searches in this study. First, it is easier for our participants to relate to task scenarios used in apartment searches rather than finding flights because they are not likely to be frequent travelers. Second, travel data (price, intermediate airports, flying routes) undergo frequent changes and therefore cannot remain relevant throughout the duration of a research project (in this case two years). On the other hand, apartment data are relevant for up to three years, especially in countries where rent control is tight. Furthermore, the evaluation of SmartClient for apartment searches could demonstrate that the example critiquing interface is useful not only for travel planning, but also for other domains.

We used two sets of 50 rental properties (student apartments) located in the vicinity of our university. Each set is used for evaluating the ranked list (called RankedList henceforth) or the example critiquing interface (called EC). The entries in these two sets are not identical to avoid any learning effects when users compare and evaluate the two tools. However, they are equivalent with respect to user tasks. That is, each data set contained at least a correct and similar answer for each of the user questions. The rental properties used in the experiment were based on real data with slight modifications. For instance, each property, regardless of its type, was normalized for the purpose of accommodating one person only.

The 22 participants (8 females) were graduate students, research assistants and administrative personnel recruited from our university (the Swiss Federal Institute of Technology or EPFL). Since EPFL does not provide sufficient dormitory rooms for our graduate students, most of them are likely to be familiar with apartment searching tasks in either online or offline environments. To make the group as diverse as possible, participants were selected from a variety of nationalities. They were Swiss, Algerian, American, Indian, Vietnamese, Chinese, and Mexican.

Participants were given adequate time to familiarize themselves with the interfaces. The data set used for this warm-up exercise was different from those used for the real experiments. To help them learn how to use the interfaces, users were instructed to perform a test search, for example finding an apartment for the price of 550 Swiss Francs and an area of 20 square meters.

4.5 User debriefing and experiment procedure

Before each experiment session, we informed each participant of the experiment's objectives, explained the meaning of labels on each of the interfaces, and told them that we would be recording their task performances. We then gave them 5–10 minutes to try out the interfaces with test scenarios.

We used a within-subjects design for the experiment procedure. Each of the 22 users was asked to perform the 5 tasks (described in Sect. 4.3) using Interface 1 (RankedList) and Interface 2 (Example Critiquing or EC). The order of the interfaces evaluated alternated for every two consecutive users. This was to counterbalance any biases that users may develop while evaluating one interface and carrying these biases to the evaluation of the other one.

4.6 Post session questionnaire

To understand our participants' perception of the two interfaces, we conducted a semi-structured interview after each user study. Besides collecting their comments and opinions regarding these two interfaces, we asked participants to answer the following two questions:

1. Which of the two interfaces do you prefer? Why?
2. Which one of the interfaces makes you feel more certain that you have found the correct answers? Why?

Table 1 Statistics of task time and error rate for task 1

Condition		Group A: RL first ($n = 11$)				Group B: EC first ($n = 11$)			
		Min	Max	Mean	SD	Min	Max	Mean	SD
RL	Time	20.00	111.00	69.60	35.58	50.00	150.00	102.20	34.84
	Error	–	–	–	–	–	–	–	–
EC	Time	41.00	84.00	62.20	15.03	30.00	180.00	90.60	46.65
	Error	–	–	–	–	–	–	–	–

4.7 Results

The recorded average task completion time and error rate were smaller for the example critiquing interface (61.7 versus 73.9 seconds for the ranked list, and 0.27 versus 0.67 errors per task for the ranked list). To further understand the significance of these results and to verify if the counter balance measure had worked, we performed the repeated measures Analysis of Variance (RM-ANOVA) on the collected data. We also divided the tasks into three categories in order to know where the significance had occurred.

4.7.1 Multi attribute searching task

The first task required users to find an apartment of their choice. No errors can be defined for such personal choices. A number of individuals took more time to find the answer while using EC than RankedList, especially when EC was given first to evaluate (see Table 1). From observing how the participants' worked with the interfaces, we believe that this was largely due to the unfamiliarity of the example critiquing interface compared to the RankedList interface. Participants tended to take more time to learn to use EC, especially under testing conditions. A RM-ANOVA test yielded no statistical significant differences in task time due to interfaces used or due to any interaction effects between interfaces and order, at the 0.05 level of significance (see Table 2).

4.7.2 Trade-off with 2 attributes

This category of tradeoff tasks (#2 and #3) are simple tradeoff tasks and required the participants to find alternatives that improve the value on one identified attribute while compromising the values of only one of the four remaining attributes. Although participants took less time on average to do these two tasks with EC (see Table 3), a repeated measure analysis of variance (RM-ANOVA) yielded no statistical significant difference for task time at the 0.05 level of significance. However, the same analysis showed statistical significances for a higher error rate when RankedList was used (see Table 4). The significance was further maintained for within group analysis regardless whether RL or EC was used first. We concur that the relatively high error rate was due to the fact that participants had to do a significant amount of visual search

Table 2 RM-ANOVA of differences for task time and error rate for task 1

Type of comparison		RM-ANOVA results		
		df	*F*	*P*
Between group				
RL vs. EC	Time	1	1.245	0.278
	Error	–	–	–
RL first vs. EC first	Time	1	0.036	0.850
	Error	–	–	–
Within group				
Group A:	Time	1	0.372	0.557
First RL, then EC	Error	–	–	–
Group B:	Time	1	0.852	0.380
First EC, then RL	Error	–	–	–

Table 3 Statistics of task time and error rate for simple tradeoff tasks

Condition		Group A: RL first ($n = 11$)				Group B: EC first ($n = 11$)			
		Min	Max	Mean	SD	Min	Max	Mean	SD
RL	Time	20.00	113.00	54.40	30.45	43.00	168.50	84.95	36.88
	Error	0.00	2.00	1.10	0.74	0.00	2.00	1.30	0.67
EC	Time	38.50	97.50	54.65	20.07	28.50	172.50	65.80	40.82
	Error	0.00	2.00	0.20	0.42	0.00	1.00	0.60	0.52

Table 4 RM-ANOVA of differences for task time and error rate for simple tradeoff tasks

Type of comparison		RM-ANOVA results		
		df	*F*	*P*
Between group				
RL vs. EC	Time	1	2.016	0.172
	Error	1	26.435	**5.8e−05**
RL first vs. EC first	Time	1	0.864	0.359
	Error	1	0.277	0.602
Within group				
Group A:	Time	1	0.001	0.972
First RL, then EC	Error	1	14.878	**0.004**
Group B:	Time	1	3.098	0.112
First EC, then RL	Error	1	10.756	**0.010**

using RankedList, and as a result they were more susceptible to making mistakes. Furthermore, the order of interfaces did not seem to have an effect on task time, or

error rate, thus confirming that the counterbalanced measure was rather successfully implemented.

4.7.3 Trade-off with more than 2 attributes

This category of questions (#4 and #5), called complex tradeoff tasks, increased the task complexity by requiring participants to perform tradeoffs on more than two attributes. They have to improve the value on one identified attribute while compromising the values of two of the four remaining attributes. A repeated measure analysis of variance (RM-ANOVA) yielded statistical significant differences not only for error rates, but also for task time due to interfaces used at the 0.05–0.08 level of significance (see Tables 5 and 6). Furthermore, the order of interfaces used did not seem to have an effect on task time or error rate, again confirming that the counterbalanced experiment was successfully designed. However, the within group analysis showed that the significance was not maintained for the user group where EC was used before RL. That is, the difference in mean error rates, 0.8 for RL and 0.5 for EC, did not reach significance, although the other p values were relatively small (between 0.028–0.093) for the within group analysis.

Table 5 Statistics of task time and error rate for complex tradeoff tasks

Condition		Group A: RL first ($n = 10$)				Group B: EC first ($n = 10$)			
		Min	Max	Mean	SD	Min	Max	Mean	SD
RL	Time	30.50	137.50	69.15	33.56	38.00	92.50	63.10	19.11
	Error	0.00	1.00	0.80	0.42	0.00	2.00	0.80	0.79
EC	Time	26.00	60.00	46.05	12.57	21.50	109.50	50.90	26.09
	Error	0.00	1.00	0.30	0.48	0.00	2.00	0.50	0.71

Table 6 RM-ANOVA of differences of task time and error rate for complex tradeoff tasks

Type of comparison		RM-ANOVA results		
		df	F	P
Between group				
RL vs. EC	Time	1	10.362	**0.005**
	Error	1	3.234	0.088
RL first vs. EC first	Time	1	0.510	0.479
	Error	1	0.261	0.612
Within group				
Group A:	Time	1	6.836	**0.028**
First RL, then EC	Error	1	5.000	0.052
Group B:	Time	1	3.520	0.093
First EC, then RL	Error	1	0.574	0.468

4.7.4 Overall performances of EC compared to RankedList

The overall performance of these two interfaces indicates that users took increasingly less time and made fewer errors when performing tradeoff tasks using EC even though the task complexities increased (see Fig. 4). On the other hand, the difference in task completion time for RankedList compared to EC increased as tradeoff tasks became more complex (Fig. 4). While the error rate for the second and third categories of tasks decreased both for EC and RankedList, the average of errors committed for RankedList remained high (see Fig. 5). The obtained data suggests that EC provides a time efficient tool for making multi-attribute tradeoff analysis, especially as the complexity of tasks increases. In addition, by observing participants interacting with the interfaces, we noticed that the tradeoff tasks were made significantly easier in EC because users could just set an attribute to the "compromise" value when they were willing to sacrifice it and concentrate on improving the values of those attributes which are important to them.

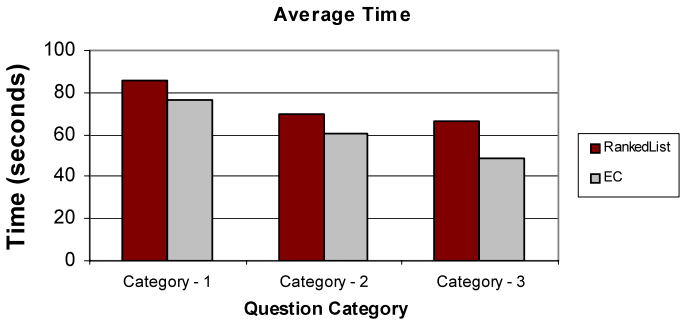

Fig. 4 Average task completion times in seconds for the three categories of tasks when evaluating RankedList and EC respectively

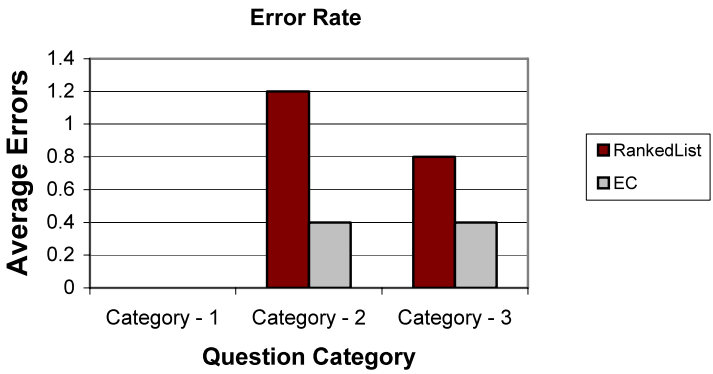

Fig. 5 Average error rates for the three categories of tasks when evaluating RankedList and EC respectively

4.7.5 User preference and confidence level

10 out of 22 participants preferred the RankedList interface over EC despite the fact that many of them took more time to perform the tasks and made more errors. In addition to preference, we also asked participants to comment on their confidence levels with respect to the two interfaces. By confidence level, we mean a user's level of certainty in believing that a correct answer has been found with respect to an interface. After compiling the responses, we were somewhat surprised to find that more participants felt higher confidence levels when RankedList was used.

We decided to further analyze users' feedback regarding confidence level by analyzing in detail participants' comments recorded during the user study. Many of them said that "the search engine used in EC hides something from me," whereas "I can see everything in the ranked list." Some also expressed concern about the fact that the results returned by the EC interface did not correspond to their ranking of decision outcomes. That apparently increased the doubt as well. To summarize, there were two main reasons for the preference for RankedList: (a) users would like to see more search results than the 7-result set, even though they may not want to examine all of the details; and (b) they sometimes have difficulties in accepting the outcomes ranked by the machine, thus wondering if other (better) outcomes existed beyond the 7-result set. We then did a trial user study where we used the scroll bar to display all 50 items, 7 results at a time. Many users scrolled down briefly, only few stopped to carefully examine a result. Only one user actually selected a result which was ranked outside of the initial 7-result set in the EC interface.

We further extended our study by searching for answers in decision behavior theory to understand why it is not always easy for people to accept the ranking of decision outcomes calculated by the multi-attribute utility theory. It turned out that human decision strategies can be very different from the utility model, also called the weighted added sum strategy, which underlies many decision support systems [22]. Largely due to effort limitations, a human cannot be as detailed as a machine in summing up the utilities of each of the attributes towards ranking the set of outcomes. Instead, they are likely to develop biases. For example, there is a phenomenon called the prominence effect [8, 39]. When a set of choices were given to users, they were more likely to select an alternative that maximizes the value of a single attribute. However, if they were given a matching problem (which is equivalent to the choice problem), they would give different values to the prominent attribute. If the search engine uses a utility model to calculate the ranking of each outcome, it is possible that humans may have trouble accepting the machine's scores. This is especially true when outcomes' utility values only differ by a small amount. For these outcomes, which we call the gray area, a linear display imposes an implied ranking despite the fact that displayed outcomes do not have a clear order. In other words, people's judgment of certain results cannot conform to such a strict assignment of order. We believe that this is the cause of the discomfort expressed by our subjects when they claimed a low confidence level for the example critiquing interface.

4.7.6 Conclusion: the real benefits of example critiquing

The user study described in this section reports an interesting evaluation of example critiquing. Contrary to the common belief that EC is naturally better than the ranked

list, results here show that EC performed comparably as the RankedList for simple tradeoff tasks. However, when complex tradeoff tasks were given, EC not only reduced users' task time significantly, but also the likelihood of making mistakes in choosing the right item. Therefore, we conclude that EC is likely to attract more attention in the next generation E-commerce sites where products become more complex. In those circumstances, users are likely to rely more on the critiquing component to perform tradeoff to achieve high accuracy in making their final decisions.

4.8 Follow-up interface design

Based on the trial user study, we began designing a new interface for the example critiquing interaction paradigm. We focused our efforts on enlarging the displayed result set and began looking for a display strategy which is less likely to allow users to dispute the ordering of results. One display technique, fisheye views, seems to provide solutions to both objectives (see Fig. 6). In the new interface, the basic principles of example critiquing interaction model stay the same. However, the number of displayed results increases from 7 to 50. The 7 recommended results are emphasized by a significant distortion effect so that their text stands out, while the other 43 results are displayed with much smaller font sizes. The distortion effects are achieved by row heights which are proportional to the utility scores of each of the recommended results. Furthermore, the natural ordering in the display follows the price attribute, or any other quantitative attributes (e.g., the size of the rooms or the distance to the university).

Because of the fisheye views, the 50 results can still be displayed in a single overview, which provides an overall context for users to see the recommended results. Additionally, it seemed that the human eyes' imprecision in judging precisely the row heights provided an advantage to display the ranking of the decision outcomes, especially for the gray areas. The new design, therefore, resulted in a less disputable display in terms of ranking order, while at the same time providing much more room for showing search results.

5 Implementing fisheye views for example critiquing interfaces

Fisheye view techniques, already described in Spence and Apperly in 1982 [33], were developed by Furnas [9] into a general framework called the Generalized Fisheye View. The principle idea is that information should be displayed according to its importance.

The name is an analogy to how a fisheye lens distorts an image. That is, it gives more details about what is being observed (the focus) and less to elements which are distant from the focus. Fisheye views therefore increase the detail or visual emphasis of a focalized set of displayed items, while filtering or deemphasizing less important results to maintain context. Fisheye views have been used to visualize a range of data structures: tables [27], trees [17], graphs [31], menus [1], and semantic information [10]. User studies of comparing fisheye views with non-distortion displays were also carried out: Pirolli et al. [23], Callahan and Koenemann [5], and Bederson et al. [2].

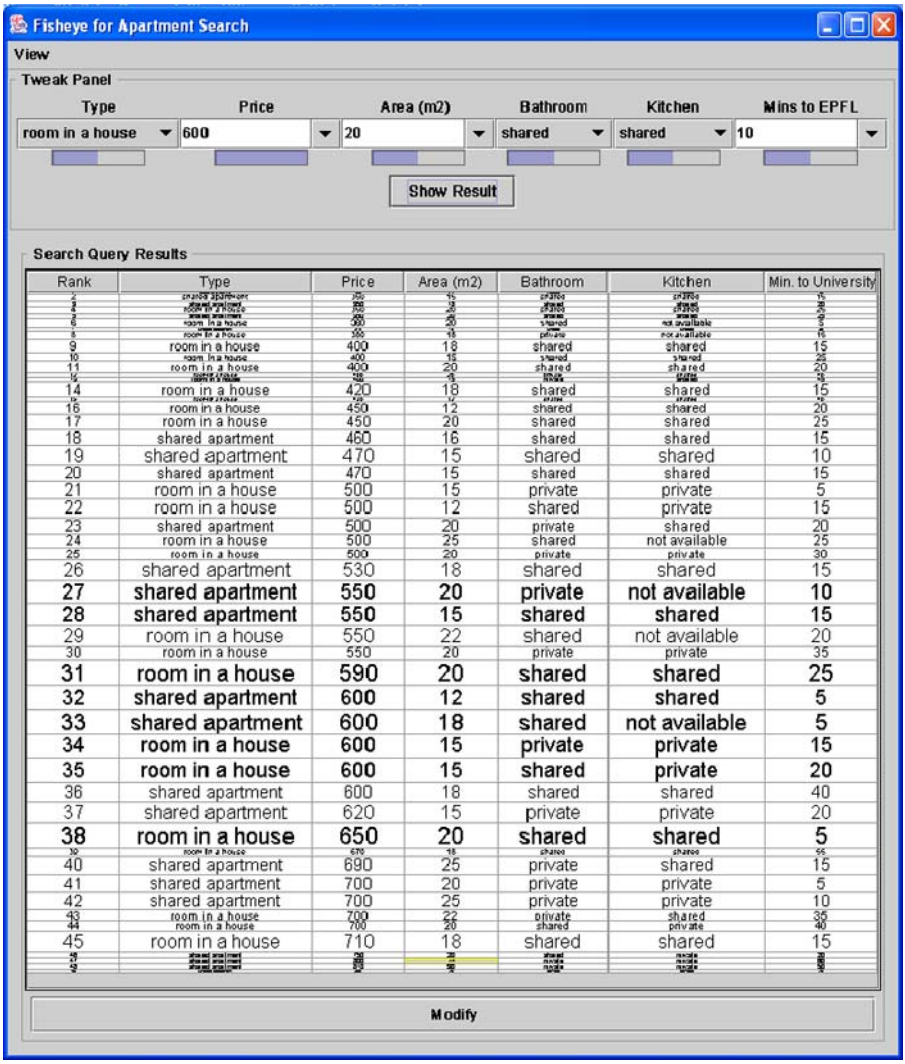

Fig. 6 Example critiquing interface using fisheye views

Of particular interest to this work was the study by Callahan and Koenemann which showed that users learned the content of a product catalog more quickly using the InfoZoom system (which implements fisheye views) than a regular electronic product catalog system.

According to Funas, the degree of interest function (DOI) determines the relevance of each item in the information space. The DOI is then used to calculate the size and visibility of an item. In our case, the DOI of each decision outcome is simply the utility score of that outcome. However, there is a high density of low utility scores of outcomes and a wide gap between low and high scores. Thus, we must use a non-linear function for calculating the DOI of each outcome so that their visibility is well

maintained. That is, a wide range of utility values must be modeled, while at the same time the mapping results in groups of 10 distinct visual ranges. After experimenting with several functions, we opted for the slow-in slow-out method developed by Sarkar and Brown [31].

6 Evaluating example critiquing with fisheye view techniques

Our objective in the second experiment was to compare two example critiquing interface designs (the original and the one with fisheye views) and see if the later design significantly improves a user's confidence level. Our hypothesis was that users would give a higher confidence score when using the example critiquing interface with fisheye views (henceforth EC + FE) than the original example critiquing interface (henceforth EC). On the other hand, because more results are displayed in FE, we expected the users to take more time to complete tasks.

Twenty participants (6 females) took part in this second user study. 16 of them were undergraduate students from our university, who were in their early 20s, and four were regular employees between the ages of 30 and 80 years old. The same data sets, where each set consists of 50 rental properties, were used. We again chose the within-subjects design for the experiment procedure. Participants were first debriefed about the goal of the experiment, and then given a period of 5–10 minutes to familiarize themselves with each of the interfaces. The user tasks, consisting of five questions, were also identical to those used in the first experiment. The first task was to find the user's favorite apartment, the second two tasks were to find $(1, 1)$ tradeoff alternatives (gain and sacrifice on one attribute respectively), and the last two tasks were to find $(1, 2)$ tradeoff alternatives (gain on one attribute, but sacrifice on up to two attributes). To measure the confidence level that participants associated with each of the interfaces, we asked the following question after each task was completed: Are you sure that you have found the best answer using this interface? (100% very sure—0% not at all.)

In addition, we asked each participant to indicate which interface he or she preferred after each session (post experiment questionnaire). Task completion time and error rate were defined exactly in the same way as in the first experiment.

6.1 Analysis of experimental results

6.1.1 Task time

On average, participants took more time to answer the questions using FE than EC, especially for the first two categories of questions (see Fig. 7). However, upon closer examination, ten participants took less time using FE, and ten took more time using it. The difference did not reach significance at the level of 0.05 via a RM-ANOVA test (see Table 8) when data for all five tasks were combined. At the same time, we found no dependency of task completion time on the order of interfaces presented in the evaluation procedure (Table 8). That is, there was no interaction between the task completion time and the order in which the user evaluated the two interfaces.

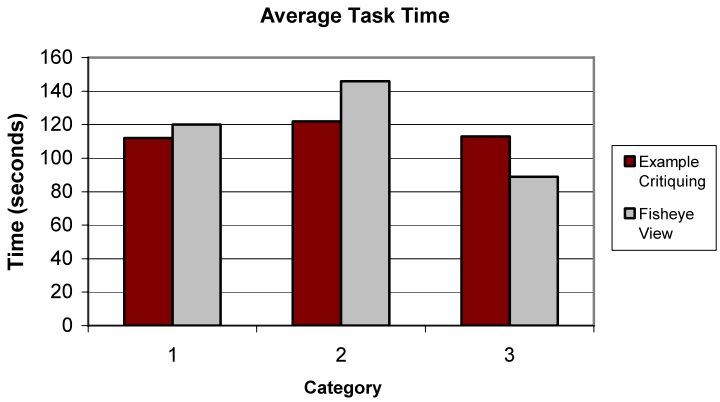

Fig. 7 Average task completion times in seconds for the three categories of tasks when evaluating EC and FE respectively

It is quite surprising to see that users did not take on average much more time with FE than EC for completing the tasks, which is contrary to what we had expected. In fact, when they got to the third category of tasks, they performed faster with FE than EC. The difference in task time actually reached significance at the level of 0.05 via a RM-ANOVA test (see Tables 9 and 10). The result confirmed our observation that users increasingly relied on the outcomes recommended by the search engine for answering the questions as they became more familiarized with the interface. Only one of them spent some time browsing data that was not emphasized. There seemed to be, however, dependencies between the task time and the order in which participants evaluated the interfaces for the complex tradeoff tasks. For both within-group and between-group comparisons, users performed faster with FE if EC was given first to evaluate. That is, there seems to be some learning effect carried over from EC to FE, but not from FE to EC. We are unable to explain the general meaning of this result.

6.1.2 Error rate

Figure 8 shows that 20 participants made a total of 8 errors using FE versus a total of 10 while using EC. Although the error rate is slightly higher for EC, the difference did not reach significance at the level of 0.05 via the RM-ANOVA analysis (see Table 8).

6.1.3 Confidence level

Figure 9 shows the average confidence levels that users expressed while using EC and FE respectively to answer the five questions. While both were very high, only three participants out of twenty expressed higher confidence levels for EC than FE. These differences reached significance at the level of 0.05 via a RM-ANOVA test (see Tables 7 and 8). Furthermore, 12 out of 20 users preferred the FE interface over the EC interface.

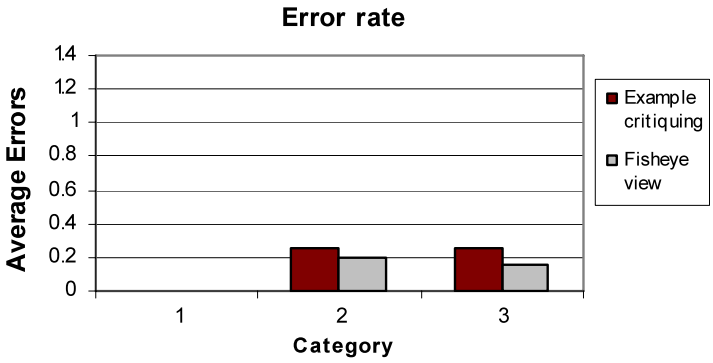

Fig. 8 Average error rates for the three categories of tasks when evaluating EC and FE interfaces respectively

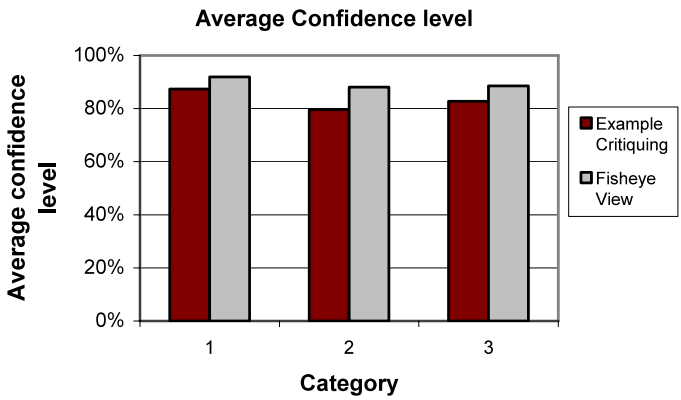

Fig. 9 Average confidence levels associated with EC and FE respectively in three task categories

6.2 Conclusion of the second experiment

Analysis of the second user study indicated that the design of a new example critiquing interface using fisheye views is a viable solution to combine the benefits of a ranked list and a critiquing interface. On one hand, the suggested solutions from a decision aid tool enabled users to quickly find tradeoff alternatives. On the other hand, the enlarged set provided them with a high level of certainty for their resulting choice.

We have introduced two new elements into the new EC + FE interface: a longer result list and the fisheye view effect. Measuring confidence may not provide a clear answer as to whether the improvement is due to the longer list or the fisheye view effect. However, the trial study reported in Sect. 4.7.5 did not indicate that users actively pursued the items outside of the 7 recommended set. We therefore assume that while a longer result could lead to more confidence, the items outside of the 7 set did not provide much recommendation power. Thus, the benefit of higher confidence levels is due to a combined effect of a longer display list and fisheye view effect.

Table 7 Statistics of task time, error rate, and confidence level for all tasks

Condition		Group A: EC first ($n = 10$)				Group B: FE first ($n = 10$)			
		Min	Max	Mean	SD	Min	Max	Mean	SD
EC	Time	72.00	167.67	126.05	35.88	64.33	163.17	113.88	36.57
	Error	0.00	0.67	0.17	0.24	0.00	1.00	0.17	0.32
	Conf.	0.75	0.98	0.85	0.07	0.73	1.00	0.90	0.10
FE	Time	70.50	216.67	115.05	41.90	58.17	225.17	133.23	47.78
	Error	0.00	0.33	0.13	0.17	0.00	0.33	0.10	0.16
	Conf.	0.87	1.00	0.93	0.04	0.82	1.00	0.95	0.06

Table 8 RM-ANOVA of differences of task time, error rate, and confidence level for all tasks

Type of comparison		RM-ANOVA results		
		df	F	P
Between group				
EC vs. FE	Time	1	0.184	0.673
	Error	1	0.681	0.419
	Conf.	1	17.377	**0.0005**
EC first vs. FE first	Time	1	1.383	0.247
	Error	1	0.051	0.822
	Conf.	1	0.452	0.506
Within group				
Group A: First EC, then FE	Time	1	0.537	0.482
	Error	1	0.184	0.678
	Conf.	1	11.783	**0.007**
Group B: First FE, then EC	Time	1	3.009	0.117
	Error	1	0.474	0.509
	Conf.	1	5.759	**0.040**

Table 9 Statistics of task time, error rate, and confidence level for complex tradeoff tasks

Condition		Group A: EC first ($n = 10$)				Group B: FE first ($n = 10$)			
		Min	Max	Mean	SD	Min	Max	Mean	SD
EC	Time	54.50	187.00	127.70	39.49	44.00	186.00	105.50	54.24
	Error	0.00	1.00	0.30	0.48	0.00	1.00	0.20	0.42
	Conf.	0.75	0.95	0.84	0.07	0.65	1.00	0.90	0.12
FE	Time	45.00	119.50	76.15	21.82	54.50	199.00	110.05	47.20
	Error	0.00	1.00	0.20	0.42	0.00	1.00	0.10	0.32
	Conf.	0.75	1.00	0.92	0.10	0.70	1.00	0.95	0.10

Table 10 RM-ANOVA of differences of task time, error rate, and confidence level for complex tradeoff tasks

Type of comparison		RM-ANOVA results		
		df	*F*	*P*
Between group				
EC vs. FE	Time	1	4.942	**0.039**
	Error	1	2.111	0.163
	Conf.	1	4.888	**0.040**
EC first vs. FE first	Time	1	4.367	**0.044**
	Error	1	0.000	1.000
	Conf.	1	0.136	0.714
Within group				
Group A: First EC, then FE	Time	1	15.376	**0.004**
	Error	1	1.000	0.343
	Conf.	1	3.950	0.078
Group B: First FE, then EC	Time	1	0.167	0.693
	Error	1	1.000	0.343
	Conf.	1	1.324	0.280

We therefore conclude that example critiquing, together with a fisheye-view display technique, enables users to accurately select their final products and convinces them of their choices.

7 Conclusion

We reported the experimental procedures and data analysis of two related user studies, comparing the performance of the example critiquing (EC) system that we have developed with the baseline model. In the first one, we found that the example critiquing (EC) paradigm enables users to achieve higher decision accuracy more effectively while requiring less effort compared to the ranked list model. This is because the effort required for using EC corresponds to the correct manipulation of the interface elements and is less than the effort needed to manually process tradeoff information in order to achieve high decision accuracy. The surprise finding from this evaluation was that, under simple task conditions, EC performed comparably to the ranked list in terms of task time and error rate. However, when task conditions became more complex, EC significantly reduces a user's task time and likelihood of making mistakes. Results of the second user study showed that a particular implementation of example critiquing also made users more confident about their choices.

Traditionally, people could not reach their full potential of decision accuracy due to limited cognitive resources. They use heuristic decision strategies that save time and effort, although these strategies often lead to serious errors that can cause emotional pain and financial loss.

We have shown in this article that interface technology and computerized decision support tools *could* make it easier for people to make accurate decisions with a modest level of effort, thus overcoming the classical dilemma. In addition, through

the two user studies, we have identified three criteria to evaluate advanced search and recommendation tools: decision accuracy and effort, and user confidence. Tools that achieve these three objectives simultaneously provide significant benefits to users. Earlier research has indicated a cost-benefit framework to explain online users' willingness to make efforts based on perceived benefits [34]. We therefore predict that consume are more likely to adopt example critiquing and similar tools in large scales in the so-called second generation E-commerce environments, where products are becoming more complex and vendors are eager to offer more product information. Such tools will also make consumers likely to buy with confidence, therefore increasing the overall conversion rates of E-commerce sites.

Acknowledgements This research work would not have been possible without the financial support of the Swiss National Science Foundation. We also like to thank Yves Dubugnon for his help in conducting the fisheye view experiment, and Jennifer Lott, Paul Janecek, and Jiyong Zhang for their insightful suggestions to improve the paper in numerous ways. Finally, we would like to thank Boi Faltings for the constant morale support during the submission, revision, and the publication of this article and his long-standing collaboration with us on the design and implementation of various decision theoretic interfaces.

References

1. Bederson, B. B. (2000). Fisheye Menus. In *Proceedings of ACM conference on user interface software and technology (UIST 2000)* (pp. 217–226). New York: Assoc. Comput. Mach.
2. Bederson, B. B., et al. (2004). DateLens: a fisheye calendar interface for PDAs. *ACM Transactions on Computer-Human Interaction*, *11*(1), 90–119.
3. Burke, R. (2002). Hybrid recommender systems: survey and experiments. *User Modeling and User-Adapted Interaction*, *12*(4), 331–370.
4. Burke, R., Hammond, K., & Young, B. (1997). The findme approach to assisted browsing. *IEEE Expert: Intelligent Systems and Their Applications*, *12*(4), 32–40.
5. Callahan, E., & Koenemann, J. (2000). A comparative usability evaluation of user interfaces for online product catalogs. In *Proceedings of ACM electronic commerce conference* (pp. 197–206). New York: Assoc. Comput. Mach.
6. Carenini, G., & Poole, D. (2002). Constructed preferences and value-focused thinking: implications for AI research on preference elicitation. In *AAAI-02 workshop on preferences in AI and CP: symbolic approaches*, Edmonton, Canada.
7. Faltings, B., Torrens, M., & Pu, P. (2004). Solution generation with qualitative models of preferences. *International Journal of Computational Intelligence and Applications*, *20*(2), 246–263.
8. Fischer, G. W., & Hawkins, S. A. (1993). Strategy compatibility, scale compatibility, and the prominence effect. *Journal of Experimental Psychology: Human Perception and Performance*, *19*, 580–597.
9. Furnas, G. W. (1986). Generalized fisheye views. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 16–23). New York: Assoc. Comput. Mach.
10. Janecek, P., & Pu, P. (2002). A framework for designing fisheye views to support multiple semantic contexts. In *Proceedings of the international conference on advanced visual interfaces (AVI'02)*. New York: Assoc. Comput. Mach.
11. Jedetski, J., Adelman, L., & Yeo, C. (2002). How web site decision technology affects consumers. *Internet Computing*, *6*(2), 72–79.
12. Ha, V., & Haddawy, P. (2003). Similarity of personal preferences: theoretical foundations and empirical analysis. *Artificial Intelligence*, *46*(2), 9–173.
13. Haubl, G., & Trifts, V. (2000). Consumer decision making in online shopping environments: the effects of interactive decision aids. *Marketing Science*, *19*(1), 4–21.
14. Herlocker, J., Konstan, J., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, *22*(1), 5–53.
15. Hogarth, R. (1987). *Judgment and choice*. New York: Wiley.

16. Keeney, R. L., & Raiffa, H. (1976). *Decisions with multiple objectives: preferences and value tradeoffs*. New York: Wiley.
17. Lamping, J., Rao, R., & Pirolli, P. (1995). A focus + context technique based on hyperbolic geometry for visualizing large hierarchies. In *CHI'95, ACM conference on human factors in computing systems* (pp. 401–408).
18. Linden, G., Hanks, S., & Lesh, N. (1997). Interactive assessment of user preference models: the automated travel assistant. In *Proceedings of user modeling'97* (pp. 67–78).
19. Maes, P., Guttman, R., & Moukas, A. (1999). Agents that buy and sell: transforming commerce as we know it. *Communications of the ACM*, *42*(3), 81–91.
20. McCarthy, K. et al. (2004). On the dynamic generation of compound critiques in conversational recommender systems. In *Proceedings of the third international conference on adaptive hypermedia and adaptive web-based systems*.
21. McCarthy, K., et al. (2005). Experiments in dynamic critiquing. In *Proceedings of the international conference on intelligent user interfaces (IUI05)* (pp. 175–182).
22. Payne, J. W., Bettman, J. R., & Johnson, E. J. (1993). *The adaptive decision maker*. Cambridge: Cambridge University Press.
23. Pirolli, P., Card, S. K., & Wege, M. M. (2003). The effects of information scent on visual search in the hyperbolic tree browser. *ACM Transactions on Computer-Human Interaction*, *10*(1), 20–53.
24. Pu, P., & Faltings, B. (2000). Enriching buyers' experiences: the SmartClient approach. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 289–296). New York: Assoc. Comput. Mach.
25. Pu, P., & Faltings, B. (2004). Interface technologies for decision tradeoff problems using CSP. *International Journal of Constraints*, *9*(4), 289–310.
26. Pu, P., & Kumar, P. (2004). Evaluating example-based search tools. In *Proceedings of the ACM conference on electronic commerce (EC'04)* (pp. 208–217). New York: Assoc. Comput. Mach.
27. Rao, R., & Card, S. K. (1994). The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceeding of Human factors in computing systems (CHI'94)* (pp. 318–322). New York: Assoc. Comput. Mach.
28. Reilly, J., et al. (2004). Dynamic critiquing. In *Proceedings of the seventh European conference on case-based reasoning (ECCBR-04)*.
29. Reilly, J., et al. (2004). Incremental critiquing. In *Proceedings of the 24th SGAI international conference on innovative techniques and applications of artificial intelligence* (AI-04).
30. Resnick, P. et al. (1994). Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the ACM conference on computer supported cooperative work* (pp. 130–137). New York: Assoc. Comput. Mach.
31. Sarkar, M., & Brown, M. H. (1992). Graphical fisheye views of graphs. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 83–91). New York: Assoc. Comput. Mach.
32. Smyth, B., et al. (2004). Compound critiques feedback for conversational recommender systems. In *Proceedings of the IEEE/WIC/ACM international conference on web intelligence* (WI-04).
33. Spence, R., & Apperly, M. (1982). Data base navigation: an office environment for the professional. *Behaviour and Information Technology*, *1*(1), 43–54.
34. Spiekermann, S., & Paraschiv, C. (2002). Motivating human-agent interaction: transferring insights from behavioral marketing to interface design. *Journal of Electronic Commerce Research*, *2*, 255–285.
35. Stolze, M. (2000). Soft navigation in electronic product catalogs. *International Journal on Digital Libraries*, *3*(1), 60–66.
36. Torrens, M., & Faltings, B. (1999). SmartClients: constraint satisfaction as a paradigm for scaleable intelligent information systems. In *Workshop notes, artificial intelligence for electronic commerce, the sixteenth national conference on artificial intelligence (AAAI'99)* (pp. 10–15). Merlo Park: AAAI Press.
37. Torrens, M., Faltings, B., & Pu, P. (2002). SmartClients: constraint satisfaction as a paradigm for scaleable intelligent information systems. *International Journal of Constraints*, *7*(1), 49–69.
38. Torrens, M., Weigel, R., & Faltings, B. (1997). Java constraint library: bringing constraints technology on the internet using the Java language. In *Workshop notes, constraints and agents, the fourteenth national conference on artificial intelligence (AAAI'97)*. Merlo Park: AAAI Press.
39. Tversky, A., Sattath, S., & Slovic, P. (1988). Contigent weighting in judgement and choice. *Psychology Review*, *95*, 371–384.

40. Zhang, J., & Pu, P. (2004). *Survey of solving multi-attribute decision problems* (Technical Report No. IC/200454). Swiss Federal Institute of Technology in Lausanne (EPFL), Lausanne, Switzerland, June, 2004.

Pearl Pu obtained her Master 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 associate editor for the IEEE Transactions on Multimedia, the general chair of the 2008 ACM Conference on Recommender Systems, program co-chair 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 is currently finishing her Ph.D. degree in Human Computer Interaction (HCI) group, School of Computer and Communication Sciences at Swiss Federal Institute of Technology in Lausanne (EPFL). She participates in a Swiss National Science Foundation project and works on user issues for online decision systems in E-commerce environments. The research goal is to design and develop intelligent and adaptive user interfaces to improve users' decision quality, save their effort and increase subjective perceptions such as decision confidence and trust. She has co-authored a number of publications in leading journals (Knowledge-based Systems Journal 2007) and conference proceedings on E-commerce (ACM EC'05), artificial intelligence (AAAI'06), intelligent user interfaces (IUI'06 and IUI'07), user modeling (UM'07) and recommender systems (ACM RecSys'07). 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 and Master degrees in Computer Science at Peking University, one of the top universities in China. 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. Her papers were published in 2002 China National Databases conference and China Journal of Computer Science.

Pratyush Kumar is currently pursuing his MBA from the University of Virginia's Darden Graduate school of Business Administration. He is keenly interested in entrepreneurship and online businesses. While at Darden, Pratyush has been awarded the Batten Entrepreneurial scholarship for 'demonstrating experience in past entrepreneurial ventures, an entrepreneurial spirit, strong leadership skills, and a drive for innovation'.
While still an undergraduate student, Pratyush co-founded uCertify (www.ucertify.com), an internet startup developing online test preparation software for certification exams such as Microsoft's MCSE, SUN's SCJP, etc. Pratyush has a deep experience in business and computer science research. Prior to joining business school, he was involved with the R&D division of Adobe Systems, working on their flagship product Adobe Acrobat. As part of the HCI research team at the Ecole Polytechnique Fédérale de Lausanne, Pratyush studied consumer buying behavior in online catalog stores and published his findings in reputed journals such as the ACM Conference on Electronic Commerce under the guidance of Dr. Pearl Pu. Pratyush holds a bachelors degree in Information Technology from the Indian Institute of Information Technology, a leading engineering institute for the study of Information Technology, established as a center of excellence by the Govt. of India. He was a recipient of the institute's Meritorious Student Scholarship for the entire duration of his four year course.