Effective Interaction Principles for Online Product Search Environments

Pearl Pu
Human Computer Interaction Group
Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland
Pearl.Pu@epfl.ch

Boi Faltings and Marc Torrens
Artificial Intelligence Laboratory
Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland
Boi.Faltings@epfl.ch, Marc.Torrens@epfl.ch

1. Introduction

To find products in online environments, people increasingly rely on computerized search tools. The performance of such tools depends crucially on an accurate model of their users' preferences. Obtaining such models requires an adequate interaction model and system guidance.

Utility theory provides a solid mathematical foundation for optimal decision support. However, it assumes complex preference models that cannot be obtained in e-commerce scenarios: people are not willing to go through complex preference elicitation processes, and furthermore are not capable to articulate their preferences with the required precision. Thus, electronic catalogs have to work with partial and inaccurate models of users' preferences. Unfortunately, this can easily cause users to finish with a suboptimal solution.

We now examine the problems in detail and propose interaction principles that address them. We illustrate how the principles can be applied using a tool for the travel planning domain, Isy-travel, based on SmartClient technology ([9, 12]). Isy-travel allows users to search for a flight itinerary according to their personal preferences and constraints.

2. Providing Domain Knowledge

In many cases, users of an online catalog are not very familiar with the available products and their characteristics. Thus, their preferences are not well established, but *constructed* while learning about the available products ([8]). To allow such construction to take place, we have our first principle:

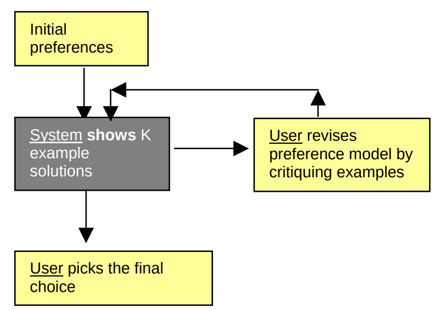

Figure 1. *Example critiquing interaction. The dark box is the computer's action, the other boxes show actions of the user.*

Principle 1 Elicit preferences within context. *A search tool should ask questions with reference to a complete and realistic context, not in an abstract way.*

A good way to follow this principle is to implement an *example critiquing* interaction (see Figure 1). It shows examples of complete solutions and invites users to state their critique of this solution. This allows users to better understand their preferences.

Figure 2 shows how example-critiquing and the principle of eliciting preferences within context is implemented in Isy-travel, a search tool for travel itineraries ([9]).

Another problem arises when users state preferences that cannot all be satisfied. Many existing e-commerce sites simply return no answers, leaving the user in the

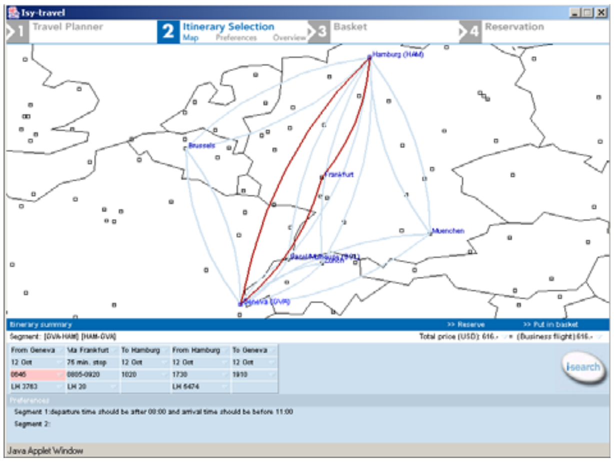

Figure 2. *A critiquing context is established by showing a solution that violates the user's preferences.*

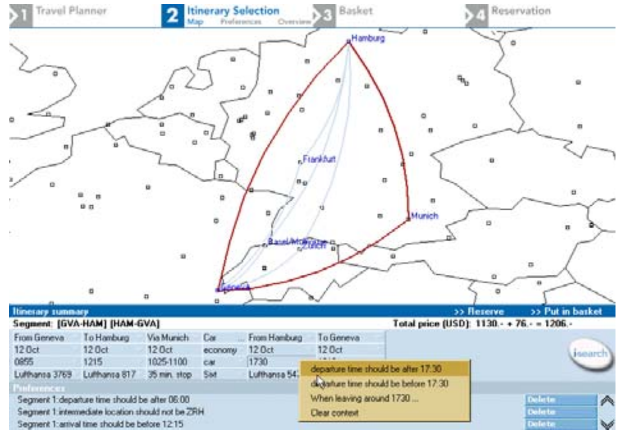

Figure 3. *Users can state preferences on any attribute, in any order they choose.*

dark about how to reformulate his preference. Hence the following:

Principle 2 Allow partial satisfaction *of user preferences: when no solutions exist that satisfy all preferences, show solutions that satisfy a maximal subset.*

Figure 2 shows an implication of this principle: the shown solution violates one of the preferences posted by the user, and this violated preference is highlighted in red. In Isy-travel, we generate a set of solutions that maximally satisfy the stated preferences and thus give the user an idea of how well they can be satisfied.

3. Avoiding Means Objectives

When users have to formulate preferences in a particular order or using an attribute that does not correspond to their actual thinking, they can fall prey to *means objectives* ([4]) because they don't have the catalog knowledge to relate this to their true objectives. For example, if someone who is looking for a car to fit 3 children and baggage is asked to first state whether he wants a car or a van, he would probably choose van, and miss a lot of interesting station wagon options. Thus, we require:

Principle 3 Allow partial preference models*: do not force the user to provide any specific preferences.*

In travel planning, if someone whose true objective is to *arrive* at his destination at 15:00 is asked to choose flights by departure time, he is forced to establish a means objective - departure time - rather than his true objective - arrival time. This is a serious problem because it can cause tools to find a wrong solution. For example, if the user assumes that it takes 5 hours to get to his destination, he might enter a departure time of 10:00 and receive a connection with a plan change that indeed takes 5 hours. If there are a direct flight leaving at 13:00 and taking only 2 hours, this much better solution would be missed by the search. This leads us to the following principle:

Principle 4 Any preference*: allow users to state their preferences on any attribute rather than a fixed subset.*

Another way for means objectives to arise is to ask for preferences in the wrong order. For example, a user who has the goal of minimzing price but is asked first to state his airline preference might state an airline that he believes will give him the best price, again establishing a means objective that may turn out to be wrong. Thus, we also propose:

Principle 5 Any order*: allow users to state their preferences in any order they choose.*

These principles are hard to follow if the search tool is built directly on top of a database that allows access only through a certain schema. An elegant way to satisfy them is to use constraint satisfaction where every preference is modelled as a soft constraint. For example, in Isy-Travel we model every preference as a constraint on the solution. Any partial preference model can be evaluated and used to select the iterneraries that satisfy the constraints best. Constraints can be formulated on any attribute or combination of attributes in the solution, as shown in the example in Figure 3. Furthermore, the result is independent of the order in which preferences are posted.

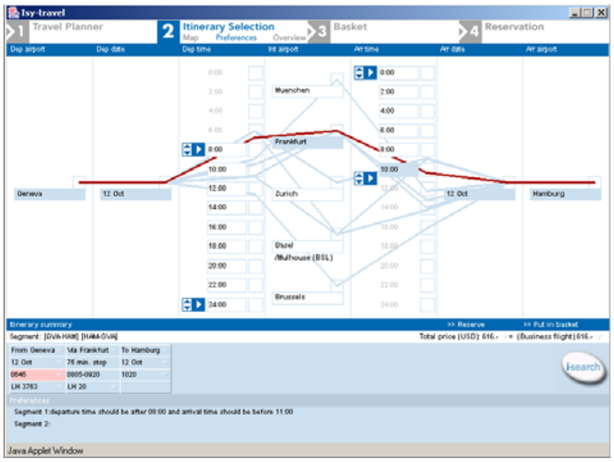

Figure 4. *Attribute ranges and solution space in one display.*

4. Convincing the User

If the system just comes up with one solution that is supposed to be the best, users might not be convinced. They need to compare possibilities and examine different possible tradeoffs to find their true best answer. The search tool should support such navigation effectively. Thus, we propose:

Principle 6 Support tradeoff navigation*: the search tool should provide active tradeoff support for the user to compare examples shown.*

Figure 4 shows one solution display available in Isy-Travel that allows users to compare solutions using parallel coordinates ([5]. In this display, the different possible values for each attribute of a solution are points distributed along vertical lines. Each solution is a line connecting its different attribute values. Users can rapidly see the available choices and compare solutions.

Example critiquing can also be used to provide active support for tradeoff, as analyzed in [10]. The paper shows an example-based navigation tool, also based on constraint satisfaction, and proves empirically that it allows users to make better tradeoffs on complex products, in this case apartments.

5. SmartClient: A Technology for Implementing the Principles

The most common technique in use today for electronic catalogs is to store products in a database and map user preferences to database queries through a web interface. While conceptually simple and easy to implement, such a technique violates most of the principles we have outlined here.

As shown in Section 3, a good way to model user preferences is as a set of *constraints*. There can be both hard and soft constraints. Hard constraints are filters that rule out certain attribute values or value combinations. Soft constraints are functions that map attribute values or value combinations to numerical values that indicate the degree to which a preference is satisfied. Soft constraints can be combined using either a *utilitarian* approach, where the sum of the numbers is optimized, or an *egalitarian* approach, where the least satisfied preference is optimized.

Constraint satisfaction combines well with example critiquing, an efficient technique for providing users with knowledge about the domain. Using well-understood constraint optimization techniques, for example the branch-and-bound algorithm, the preference model can be used to generate a set of k best solutions according to the model. These can be the examples that users can critique to refine their preference model. In [2], we have shown that by displaying a set of k best solutions, such a mechanism can compensate for the inaccuracy of the preference model and guarantee that users will find their best solution among the selected examples with certainty.

In order to make users aware of their preferences, it may further be desirable to stimulate them by showing extreme examples that may become optimal if additional preferences existed. In [3], we have shown several techniques that allow us to select such examples. In experiments, it can be seen that these techniques lead to significantly more precise preference models.

For supporting the final tradeoff analysis, we can use a visualization of the k best solutions. As an alternative method, we can use tradeoff navigation based on examples, as shown in [10]. In this method, users can navigate through the space of examples by *tweaking*: moving from one example to the next by changing the value of different attributes. It has been shown that this method significantly outperforms simple rankings on complex products.

Finally, an important issue is the complexity of implementing such a personalized search. An important advantage of using constraint programming to represent preferences is that even very powerful constraint-based search algorithms are very compact. Thus, they can be coded in no more than 100kB of Java code, and downloaded as an applet. This also allows distributing the computational load of the search to the users' computers, thus alleviating scalability problems of many conventional web servers. The architecture and implementation of SmartClient is described in [12]. Isy-travel is a commercial product that has been further developed into a business travel planning tool marketed by i:FAO, the leading provider of business travel software in Europe.

6. Related Work

Several authors have proposed example critiquing as an interaction technique for product search. FindMe ([1]) provides knowledge support to users navigating in a large information space, and has been applied in various online product search tools for renting an apartment, choosing restaurants, finding cars, selecting videos, and others. An important element in FindMe is tweaking, an interaction model that enables users to navigate to alternatives based on examples. This is particularly useful for tradeoffs and similar to the technique analyzed in ([10]).

Linden et al. ([6]) describe a tool for finding flights. Initially only few user preferences need to be expressed. The ATA system (automated travel assistant) uses a constraint solver to obtain several optimal solutions. Five of them are shown to the user, three optimal ones in addition to two extreme solutions (least expensive and shortest flying time). User preferences are modeled as soft constraints in the CSP formalism. ATA observes most of the principles give in this paper. However, on more detailed analysis ([2]), it becomes clear that showing 5 examples is by far not enough for this problem. This shows the importance of the deeper analysis we are carrying out.

Apt Decision ([11]) is an apartment search and decision support tool. It uses learning techniques to synthesize a user's preference model by observing their critiques of apartment features. Users identify hidden features by browsing through the shown examples to discover new features of interest. The system then revises their preference model accordingly.

Even though a number of example-based query search tools have been proposed, no principles have been established for building these interfaces. We hope that our analysis leads to a deeper understanding of the issues underlying such interaction.

7. Conclusions

The internet has made an amazing amount of choices available to anyone. Searching for the right ones now requires computerized tools. The most common techniques for searching product catalogs are based on traditional database approaches and ask users to provide a query that is then executed. While easy to implement, it turns out that such systems frequently mislead users. For example, in the travel industry a survey ([7]) found that only 18% of users of travel sites felt that they found the right product, and less than 50% would consider buying it.

We have shown that there are good reasons why such tools fail, and given several design principles that serve to avoid these problems. We have illustrated the technique, example-critiquing, on the domain of travel planning, but it is broadly applicable to search in general. We have already applied it to vacation packages, insurance and apartments, and are studying how to apply it to document search. In further work, we have analyzed the example-critiquing approach and shown how to configure it to the needs of an application ([2, 3]).

References

[1] R. Burke, K. Hammond, and B. Young. The findme approach to assisted browsing. *IEEE Expert*, 12(4):32–40, 1997.

[2] B. Faltings, M. Torrens and P. Pu. Solution Generation with Qualitative Models of Preferences. Computational Intelligence **20**(2), pp. 246-263, 2004

[3] B. Faltings, P. Pu, M. Torrens and P. Viappiani. Designing Example-critiquing Interaction. International Conference on Intelligent User Interfaces, 2004, pp. 22-29

[4] R. Keeney. Value-Focused Thinking: A Path to Creative Decision Making. 1992: Harvard University Press.

[5] A. Inselberg and B. Dimsdale. Parallel Coordinates: a Tool for Visualizing Multidimensional Geometry, in Proceedings of IEEE Visualization '90, IEEE Computer Society, p. 361-378, 1990.

[6] G. Linden, S. Hanks, and N. Lesh. Interactive assessment of user preference models: The automated travel assistant. In Proceedings of User Modeling '97, 1997.

[7] Morgan Stanley Dean Witter Equity Research Europe. Transportation E-Commerce and the Task of Fulfilment, March 2000

[8] J.W. Payne, J.R. Bettman and E.J. Johnson. The Adaptive Decision Maker. Cambridge University Press, 1993

[9] P. Pu and B. Faltings. Enriching Buyers' experiences: the SmartClient Approach. Proceedings of The ACM SIGCHI Conference on Human Factors in Computing Systems, 2000.

[10] P. Pu and P. Kumar. Evaluating Example-based Search Tools. Proceedings of the 5th ACM Conference on Electronic Commerce, pp. 208-215, 2004.

[11] Shearin, S. and Lieberman, H. Intelligent Profiling by Example. in Proceedings of the Conference on Intelligent User Interfaces, ACM Press, 2001.

[12] Torrens, M., Faltings, B., and Pu, P., SmartClients: Constraint Satisfaction as a Paradigm for Scaleable Intelligent Information Systems. International Journal of Constraints, 2002. 7: p. 49-69.