Enriching buyers' experiences: the SmartClient approach

Pearl Pu
Ergonomics of Intelligent Systems
ISR/DMT
Swiss Institute of Technology Lausanne
CH-1015 Ecublens EPFL, Switzerland
+41 21 693 6081
pearl.pu@epfl.ch

Boi Faltings
Artificial Intelligence Laboratory
Computer Science Department
Swiss Insitute of Technology Lausanne
CH-1015 Ecublens EPFL, Switzerland
+41 21 693 2738
boi.faltings@epfl.ch

ABSTRACT
In electronic commerce, a satisfying buyer experience is a key competitive element. We show new techniques for better adapting interaction with an electronic catalog system to actual buying behavior. Our model replaces the sequential separation of needs identification and product brokering with a conversation in which both processes occur simultaneously. This conversation supports the buyer in formulating his or her needs, and in deciding which criteria to apply in selecting a product to buy. We have experimented with this approach in the area of travel planning and developed a system called SmartClient Travel which supports this process. It includes tools for need identification, visualization of alternatives, and choosing the most suitable one. We describe the system and its implementation, and report on user studies showing its advantages for electronic catalogs.

Keywords
eCommerce, on-line travel planning systems, visual overview, client-server architecture, constraint solver

INTRODUCTION
A common assumption in electronic commerce is that buying starts from clearly identified needs that the buyer is able to articulate. According to [14], there are 6 stages of consumer buying behavior: need identification, product brokering, merchant brokering, negotiation and purchase and delivery. Activities migrate from one stage to another and some stages are iterative processes. However, most e-commerce user interaction separates the first two stages: first the buyer states her criteria, then an initial set of products is shown, followed by possibilities for comparison shopping, negotiation and placing an order.

We believe that in most cases, needs define themselves as a result of the products being offered. For example, originally we might have decided that a 300MHz processor was what we needed for our new PC. When we find out that we can get a 366Mhz and a CD-Rom drive at almost the same price, these might suddenly become part of our needs as well. Conversely, when we find out that the 19" screen we had asked for is very ugly, we might not want it anymore. Needs can end up completely redefined, and not just for irrational reasons: the design of the famous Sydney Opera house was selected in spite of violating *all* the criteria stated in the design competition.

Another important aspect in buyer decision making is that everyone wants to be convinced of getting a *good deal*. The buyer has to be convinced that among all products that can be obtained, what she is buying is an outstanding choice with respect to some criteria. One obvious criterion is price, but often it is also another feature or combination of features. Such a conviction cannot be achieved without comparing an item with its alternatives.

As a consequence, we believe that needs identification, product brokering and comparison should in fact be an iterative *conversation* where criteria and proposals are being exchanged. The buyer takes part in this dialogue by formulating needs and optimization criteria. The catalog should propose solutions and compare them to alternatives, and thus elicit refinement of the buyer's specifications.

We have explored such interfaces in the example of airline flight catalogs. In most current e-commerce sites, the traveler needs to enter dates and times and is then directly led to a small choice of flights. In our approach, the buyer initially specifies only a set of possible destinations and ranges of dates. The catalog then proposes a large set of possible products, using three different displays:

- an overview display which allows comparing the entire range of possibilities according to selected criteria,
- specific example products to elicit further constraints on their attributes, and
- a visualization comparing small sets of alternatives in their attributes.

In all these three displays, buyers identify their needs and evaluate them constantly until they reach an optimal

solution. Flexible interaction sequences are supported through the use of constraint satisfaction as a basic selection mechanism. This allows us to model a buyer's criteria accurately and explicitly. Constraints can be posted and retracted in any order, resulting in a flexible conversation that rapidly leads the buyer to refine his needs. Additionally, the constraint satisfaction paradigm provides support for visualizing and comparing the entire space of possibilities, thus ending up with a solution that appears to be the best deal and that he is thus ready to buy. Finally, it is the basis for our SmartClient architecture that implements this rich user experience with lightweight applets.

BUYER EXPERIENCE IN CURRENT TRAVEL E-COMMERCE

Jurca surveyed 10 commercial on-line flight reservation systems [10]. Almost all of them impose a fixed decision-making sequence on the user. As an example, consider Travelocity [19], a popular site for air travel. It requires a user to first fill in a form with the following information:

- Itinerary
- Dates and times of travel
- Class of travel, number of seats required
- Preferences of airline companies

Then it returns a list of possibilities for the first leg of the trip, of which we have to choose one, then possibilities for the second leg, etc.. When all flights have been selected, the systems shows the total price and offers to reserve. Alternatively, we can have a display of 9 different complete itineraries, or an entirely different model where we start by selecting a fare and then choose dates and flights that fit it. In all cases, the buyer has to fit a particular sequence of decision-making in which needs specification and browsing the product offering are two distinct stages.

We think that this fixed sequence makes the site cumbersome to use. If we discover that there are no good connections for the return date we have chosen, we have to restart the entire process from the beginning. If we find that business class is terribly expensive, we have to plan the trip all over again to see what the price in economy would be. As a consequence, it is not uncommon to spend more than 1 hour planning a trip using this site! The process would be much more efficient if it were possible to specify and compare criteria while browsing the available flights.

Furthermore, for longer trips there are many criteria that are important to the traveler but for which the site offers no optimization possibilities: total flying time, departure time, transfer airports, total ground time at transfer airports, etc.

RELATED WORK ON PRODUCT BROKERING

Electronic catalogs are examples of decision-making problems: the task is to make a decision among a set of alternatives. Decisions are made as a result of constraints, preferences, or optimization criteria. In most systems, constraints, preferences and criteria are not modeled explicitly, but remain implicit in selections the user makes.

In most existing systems for e-commerce, buyers have to commit to their needs early on in the buying process, and information about products can only be displayed in response to this. As a result, when buyers' needs change, it results in frequent information exchanges between buyer and seller, with the associated delays and server load. This problem is not uncommon in other practices of e-commerce. The dilemma encountered by system designers is how to best support decision making with sufficient information while guaranteeing a reasonable speed to download it. So far, few solutions have been proposed to implement a rich user experience in an efficient way. Most of the existing electronic catalogs fall into four types: using hierarchies, filtering, preferences, or configuration.

In hierarchically organized catalogs (e.g., PC-Zone [15]), buyers first answer a fixed sequence of questions corresponding to how databases are organized. This questionnaire form is then sent to the seller's catalog server. Product information is retrieved and sent to the buyer. Two problems can be identified: 1) this model is inadequate for natural interaction between buyers and product servers since buyers do not identify their needs in any particular order; 2) only information on a few products is given to the buyers, leaving them wondering what other alternatives are available.

Filtering catalogs (e.g., Personalogic[16], Automated Travel Assistant[12]) allow users to explicitly formulate constraints on what acceptable products are. These act as a filter: the catalog only shows products that satisfy these constraints. Rather than distinguishing acceptable from unacceptable products, constraints could also return a number that reflects the degree to which they are satisfied. It then models users' criteria for optimization. Many catalogs allow optimizing such criteria or combinations of them.

Soft constraint techniques, that is expressing users' criteria as a scale of preferences using weights [11], are more flexible for navigating in large product catalogs [17,18]. There is no sequential order to define criteria. At the same time, soft constraints allow sub-optimal solutions to remain in the navigation space, thus making a large enough portion of the catalog available. Our work is based on the similar observation that larger product space encourages serendipitous buying opportunity. The difference is that we use partial constraint satisfaction techniques to handle soft constraints instead of weights. In domains such as travel, it is hard to attach importance to a

criterion before hand without knowing what are the influences of other criteria in the same problem.

Finally for complex products, the approach is very different. Instead of representing every data item in the catalog, configuration techniques [7] are used to propose products to customers according to his current preferences and constraints. Some of the configurable electronic catalogues are currently employed by e-commerce sites at Dell [6] and Cisco[5], and several e-commerce solutions are offered by Calico Commerce[3] and by ILOG [8]. Most of the configuration catalogs, however, do not provide enough interaction techniques for browsing alternatives, nor supporting tradeoff analysis.

As we can see from the above examples, providing a rich user experience by integrating needs identification, product brokering, and product comparison poses challenges for the system architecture and interaction design.

SMARTCLIENT ARCHITECTURE

We have patented a technique that formulates travel planning as a constraint satisfaction problem (CSP [13,20]). It allows transferring product information between product server and buyers through a skinny data connection. At the buyer's side, information is assembled into product configurations according to their constraints and preferences. In travel planning, when a buyer contacts the flight server, he only defines a range of destinations and possible dates. From this information, the system constructs a CSP model which gets shipped to the customer computer. In addition, a constraint solver and a graphical user interface are also included. These generate solutions according to the customer's stated needs and preferences. An advantage of our technique is that the code is very lightweight and can be efficiently packaged into small Java applets. Thus, the download of a typical travel example requires about 500 Kbytes, corresponding to the size of 8 average web pages of Travelocity. Interaction with the visualizations is then instantaneous, providing a rich and satisfying user experience.

We call this architecture SmartClient because it involves clients that are both thin (smart) and intelligent (smart) at the same time. The possible space of products offered at the client sites can go up to thousands.

INTERACTION DESIGN FOR SMARTCLIENT

Buyers enter the interaction with often very vague ideas of what their actual needs are. Thus, in air travel, they usually know where and roughly when they want to travel, but they do not think of many other secondary criteria, such as departure times, airports and airlines they like to avoid, etc. Only the destination airports and ranges of departure and arrival dates are required.

Defining initial needs

We use the world map as a metaphor for defining origination and destination airports.

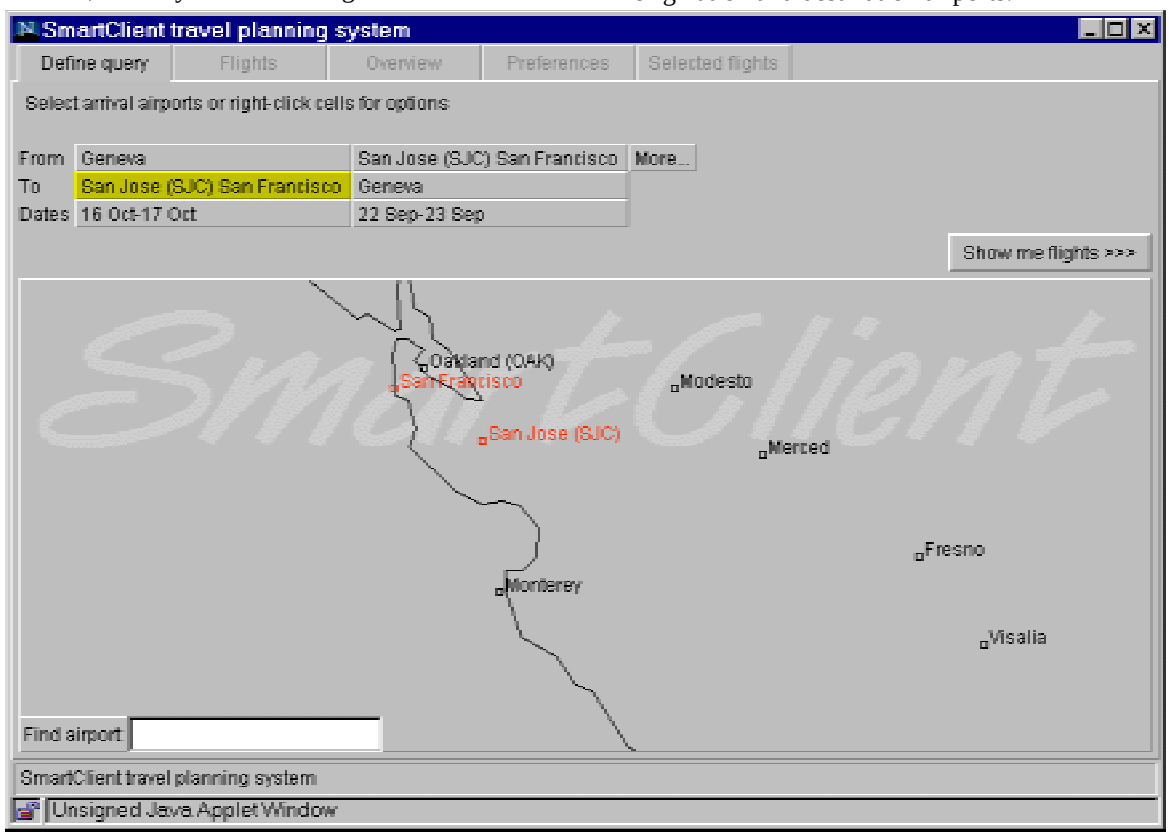

Figure 1: Query display with world map.

As a user zooms in, detailed information appears, such as each country's contours and the names of available airports. Clicking any name will enter the corresponding airport into the itinerary definition panel located on the upper-left corner. An example shown in Figure 1 shows the itinerary data of the following example:

A professor living in Geneva, Switzerland, who wants to spend a week in Silicon Valley to meet with his colleagues. The best airport for his trip is in San Jose.

In SmartClient travel, one can also type the names of the initial airports. This leads to displays of the regional map that shows the selected city among its neighbors (see Figure 1). When selecting the destination, it thus becomes obvious that flying into San Francisco could also be a good alternative, so this city gets selected as well.

The "show me flights" button generates a solution space (not solutions themselves) which is then shipped to the customer's side, whose constraints and preferences are used as guidelines to define an initial focus on the solution space.

Getting an overview of the available products

Often, the range of choices that a buyer might consider is bewildering. When there are many competing and possibly conflicting evaluation criteria, there can be a huge number of relevant choices, each optimal for some of the criteria and suboptimal for others. For example, the cheapest flight may require three plane changes. A non-stop flight, on the other hand, is expensive. In trip planning, the complexity and richness of such decision problems are further compounded by the conditional nature of users' criteria. That is for certain flights, they'd prefer the cheapest, while for others, they'd prefer the non-stop feature.

Some catalog systems attempt to solve this situation by requiring a numerical weighting of the criteria and using the weighted sum of the different criteria to rank the solutions. When criteria depend on context, it is easy to find situations where no weighting can accurately model the correct preference structure. For example, for a flight leaving from Geneva, our professor might like a departure as early as 8 o'clock, while with a departure from Zurich this should be 11 o'clock to account for the train ride there – but this interaction cannot be modeled by feature weights alone. For this reason, it is not very realistic to expect buyers to quantify tradeoffs in this way. More importantly, it leaves the buyer with the uneasy feeling of choosing a product without knowing why.

We believe that a better approach is to help the buyer find the criterion in which one choice clearly stands out as the best one. For example, if all flights leave between 8 and 9 am, this is not a useful criterion for comparing them. On the other hand, if the price varies between 300 and 1500 Francs, this could be a much more important attribute to look at. A buyer who chooses a flight because it costs only half of similar alternatives feels that he is getting a good deal, whereas if he should take a flight because it leaves at 8:45 instead of 8:30 he might not be very sure of his choice. This is an important element in convincing the buyer to actually go ahead with the purchase.

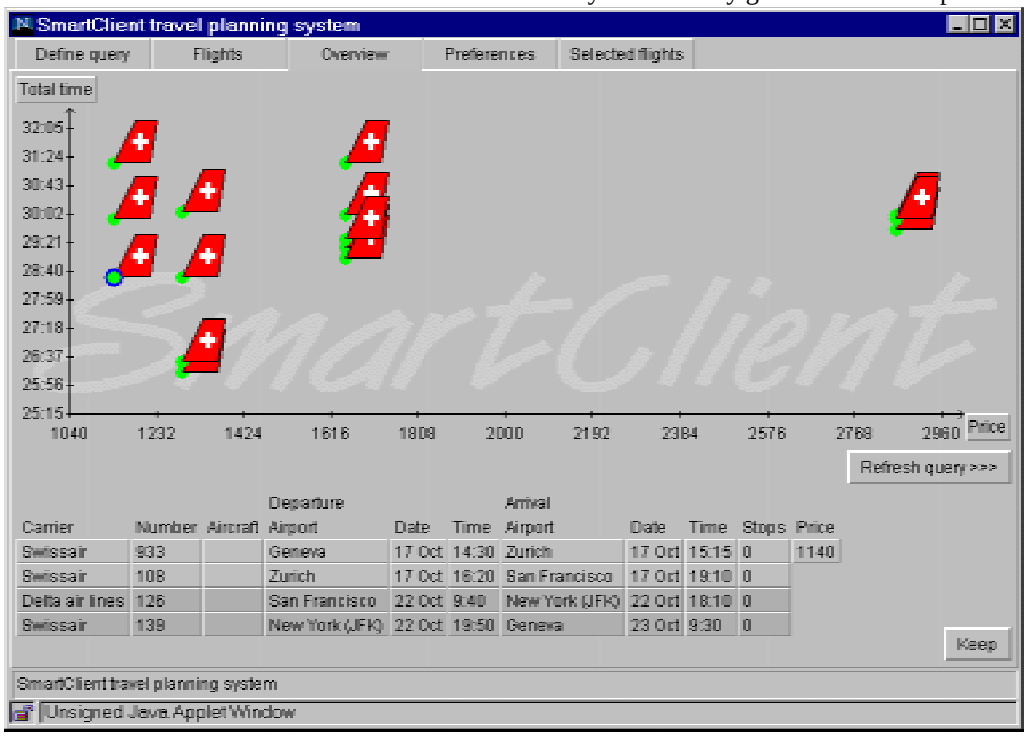

Figure 2: Tradeoff between price and total flying time in Overview.

Such multi-criteria analysis can be performed in the overview display (Figure 2), showing a scatter-plot of a sample set of solutions according to fare (horizontal axis) and total travel time (vertical axis). This technique is similar to the starfield display used in data base query systems described in [1]. However, only a focused set of solutions are displayed. As users change their criteria and preferences, this overview shifts its viewing area to other solutions. Therefore, it implements a type of semantic fish-eye, as opposed to a normal fish-eye view [2]. This scatterplot is useful to see that:

- There is quite a variation in fares, so fare should be a criterion we check for.
- Paying a higher fare does not seem to allow us much shorter flying times.

It is possible to inspect each of the possibilities in the display below, and use this to make an initial choice which we then further inspect in other displays. Here, we select the shortest flight but with the lowest possible fare. This flight is then shown in detail below, and provides a good starting point for further selection. Solutions can be selected using the "Keep" button and are then stored in the "Selected flights" folder.

At any moment during solution space navigation and browsing, users can go to the overview area to further compare trips. Overviews can be provided for any combination of price, total travel time, number of intermediate stops and solution quality regarding to users' criteria. A trip ranks low on solution quality if it violates many of the criteria. Additional information such as the main carrier's flags are denoted by the graphical forms of each node.

Eliciting further needs and constraints

A typical buyer has many constraints that are not stated up front. He becomes aware of these only when solutions are proposed that violate them. In our example, the most cost-effective solution in fact has several problems:

- It leaves too early to allow finishing up the last breakfast meeting on Oct. 22nd.
- It transits in New York JFK airport, which the customer would like to avoid.

The solution display, shown in Figure 3, allows posting

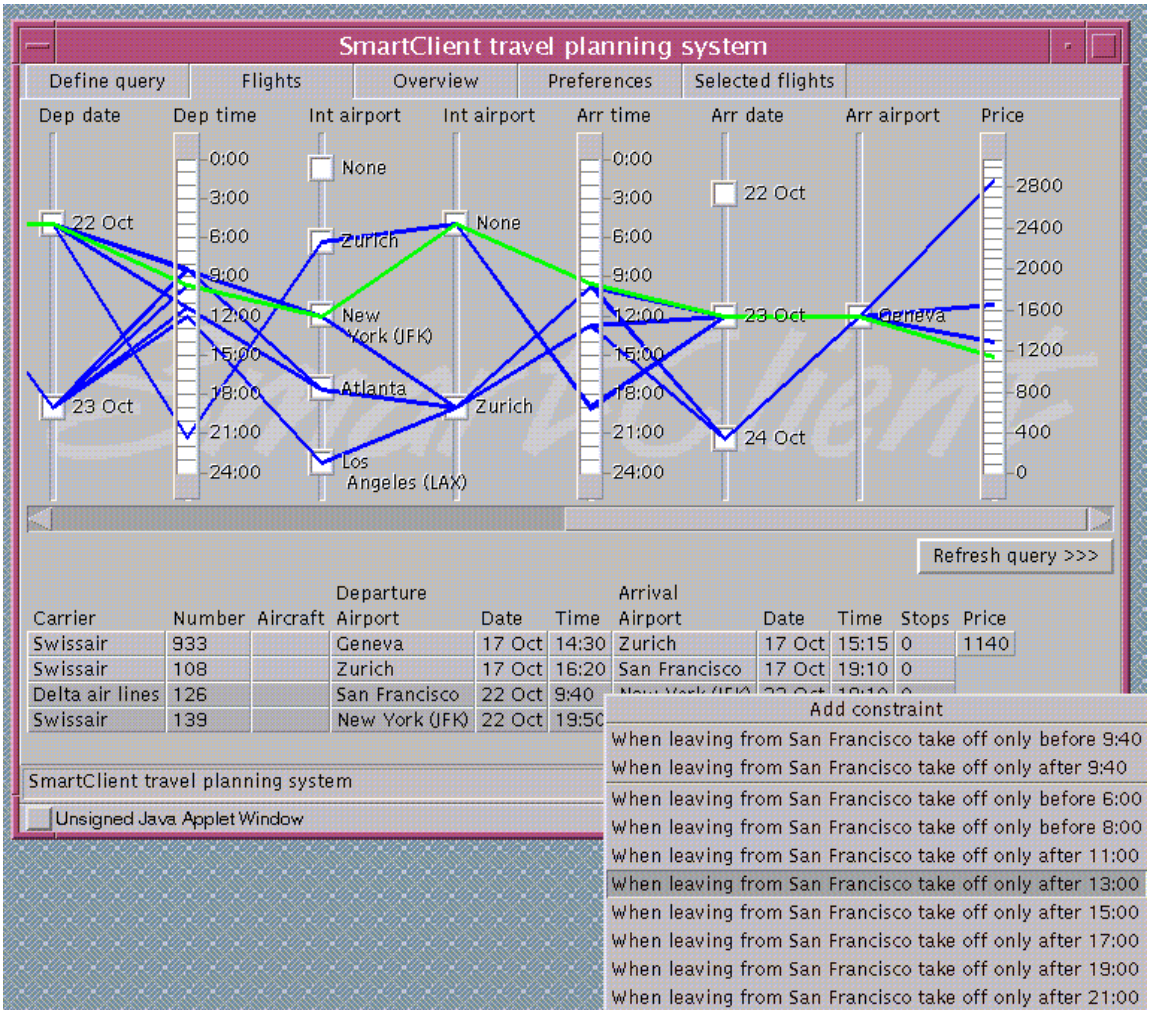

Figure 3: Posting constraints

constraints on any item in the display. In the textual display, they can be posted by clicking on the respective cell and thus activating a menu, as shown in the figure. Similarly it is possible to post constraints on any of the following attributes:

- Price
- Airlines
- Aircraft types
- Departure and arrival dates
- Departure and arrival times
- Intermediate airports
- Direct or non-direct flights

When constraints are posted in this way, they are automatically restricted to the context in which they were posted. For example, if I post a constraint on a departure time, it will by default be applied to flights for that particular leg and leaving from that particular airport only. Applicability can be further restricted by selecting cells as a context, for example only when leaving from San Francisco airport because of the longer driving time.

Constraints can also be posted using sliders in the graphical tracer display above the textual display, which is discussed in more detail later.

Posting constraints in this manner eliminates one major difficulty with conversational interfaces: it makes it impossible for the user to input constraints that cannot be understood. Since the display does not show attributes or values that do not exist, it is not possible to post constraints on them. Thus, we cannot post a constraint on the type of food served on the airline, nor on the size of the seats (unfortunately), since these attributes are not available in airline reservation systems.

At any time, the user can request the system to compute a new set of solutions that satisfy all constraints posted so far, and will usually obtain immediate response. In our example, this shifts the most advantageous trip to one where the return is now from San Jose and through Chicago.

Comparing alternatives in detail

Our users studies found that once a buyer is familiar with such navigation techniques in product space, he'll want faster interaction. When he has narrowed down the choices by posting constraints, he is ready to select particular solutions. We applied the parallel coordinate display method [4,9] to the travel domain. A tracer display (Figure 4) shows each solution as a trace through the set of flight attributes comprising a trip itinerary. For each attribute, there is one vertical bar with its possible values. A solution is a trace that links the values of the different attributes. An individual solution is selected whenever the mouse is moved over it and is then displayed in detail at the bottom of the display.

The tracer display quickly makes apparent the differences among a set of possibilities. Here, we can see in particular that given our constraints, there is a choice of returning via a variety of airports: Chicago, Los Angeles, New York and Zurich, and a variety of times (morning to afternoon).

The tracer display also allows posting additional constraints by using sliders on each attribute. This gives users who prefer to work with more graphical abstractions another way of declaring needs and criteria to the system.

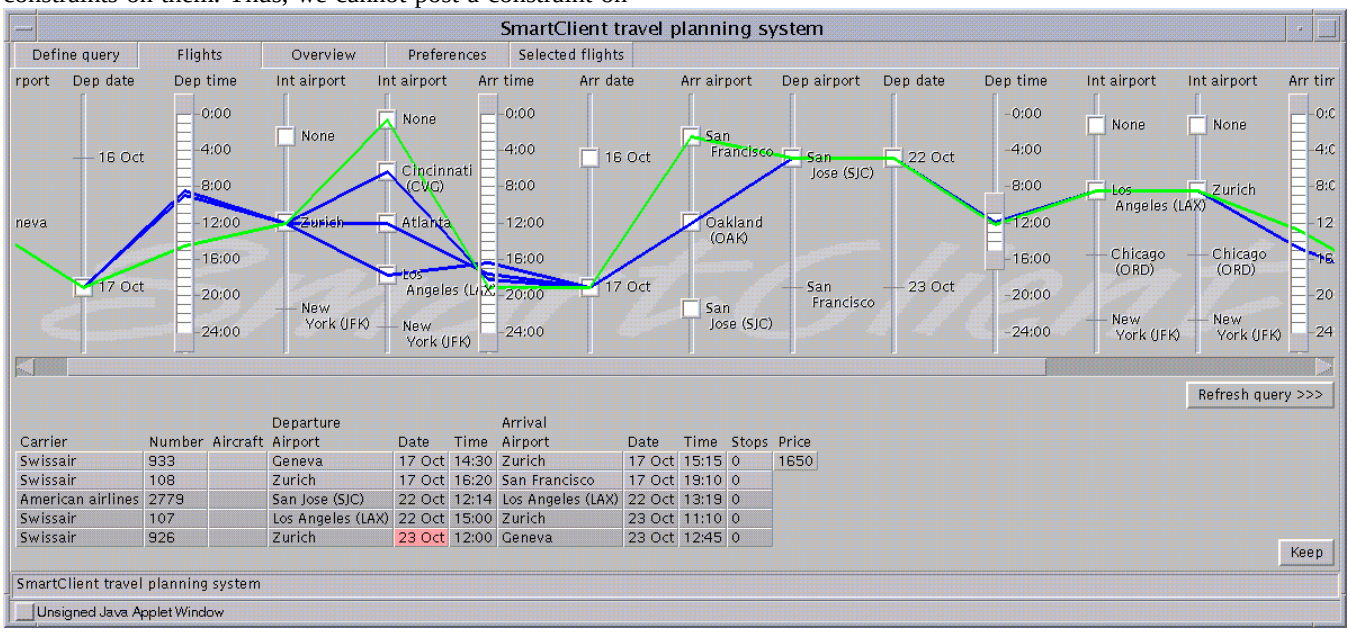

Figure 4: Tracer display with final solution.

Sliders on attribute bars can further provide rapid specification of ranges of data for dates and time. Clicking on the other hand allows easy interaction for choices, such as whether someone wants to stop in Zurich or not. Even though many users are first shocked to see what usually looks like stock market graphs, they appreciate the power of visualizing many options in a compact display area.

Figure 4 displays 17 possibilities as curves mapped onto the multi-value graphs for our example trip. The canvas area for the detailed flight information is sensitive to the current mouse position on the curves. Thus, each solution is compared with the rest and viewed in detail at the same time.

In our example, the user may be particularly sensitive to transfer airports, and inspect solutions based on that criterion. This will lead to further narrowing of the space by:

- Disallowing transfer in Chicago
- Constraining departure time to be after 11 am, from San Jose only

In the final display, shown in Figure 4, the customer now only sees solutions that satisfy all the posted constraints, and can manually compare them to find the truly best one.

As an alternative way, this interaction design is particularly useful when travel is less constrained, allowing buyers to quickly decide that they can only expect small differences in cost, but potentially large gains in travel time. This can help buyers define criteria to find the "good deal".

All of the above can be done without further contacts with the server. Using a common travel site such as Travelocity, the same trip requires more than 10 client-server contacts, at the end of which the customer is still not sure how good the solution he is getting really is.

USER STUDIES

We have evaluated SmartClient with 43 users, all of them students of our university. Being in the 20-23 years old range, this youth group is one of the most targeted groups for airline eCommerce in Europe. The students range from computer science, electrical engineering, industrial engineering, to civil engineering majors. They formed into a team of two students and each team was asked to use SmartClient Travel to complete three specified trips and one trip of their own choice. They were to compare the experience of SmartClient Travel to a commercially available system. One person was to work on the problem using the computer, while the other recorded the time needed for each trip planning, the usability of the software, and the usability of the constraint editing features. Based on the findings, we have the following conclusions:

- None of them had trouble discovering the criteria editing feature offered by SmartClient Travel.
- All of them agreed that SmartClient Travel allows them to examine a much larger space of solutions than other tools.
- Most of them complained that the speed in getting the initial data from SmartClient Travel is a problem (this was due to the low quality connection to the airline information system that was available for the study).
- When trips are simple and needs are known up front, they noted that SmartClient Travel and others are more or less equally powerful.

CONCLUSION

We have shown an approach to electronic catalogs where user criteria and preferences are explicitly modeled as constraint satisfaction. This simple and general formalism is the basis for SmartClients, lightweight applets that allow browsing a space of solutions in an intelligent way. This offers important practical advantages for electronic catalogs:

- Criteria can be given and modified in any order, rather than following a predefined dialogue model. Product selection can become a flexible conversation where customers discover their criteria through inspection on the available choices..
- Using overview displays, users can get a quick idea of the importance of different criteria, and understand tradeoffs between them.
- Different solutions can be compared in a single framework using the tracer display. This lets users make a final their choice that they are confident about and ready to buy.

The SmartClient approach has been implemented as a prototype system. We are currently working on a system to be put in practical use. We are also exploring application of the same technique for electronic commerce of insurance products.

ACKNOWLEDGEMENT

We thank the undergraduate students in our university for participating in our user studies. We also thank Marc Torrens for implementing the SmartClient architecture, Adriana Jurca for implementing the first generation interfaces used in SmartClient, and Sebastian Gerlach for the second generation interfaces (World Map, sliders, overview, etc.).

REFERENCES

1. Christopher Ahlberg and Ben Shneiderman, Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays, Human Factors in Computing Systems. Conference Proceedings CHI'94, pp. 313-317, 1994.
2. Marc H. Brown and James R. Meehan and Manojit Sarkar Browsing Graphs Using a Fisheye View, in *Proceedings of ACM INTERCHI'93 Conference on*

Human Factors in Computing Systems, Formal Video Programme: Visualisation, p. 516, 1993.

3. Calico Commerce, http://www.calicotech.com
4. Card, S. Eick, S., and Nahum G., Information Visualization Tutorial, in *CHI'99 tutorial notes*, 1999.
5. Cisco Connection Online, //www.cisco.com
6. Dell Inc., http://www.dell.com
7. Faltings, B. and Freuder, E., Configuration, in *IEEE Intelligent Systems and their applications*, as guest editors' introduction, July/August 1998.
8. ILOG S.A., http://www.ilog.com
9. Inselburg, A. Dimsdale, B., "Parallel Coordinates: A Tool for Visualizing Multi- Dimensional Geometry," in *Proceedings of the First IEEE Conference on Visualization*, 1990.
10. Jurca, A.., Survey of Online Travel Planning Systems, Technical Report (No. 99-01), ISR/DMT, Swiss Federal Institute of Technology Lausanne, 1999.
11. Keeney, R.L., Faiffa H., Decision Making with Multiple Objectives: Preferences and Value Tradeoofs. Cambridge University Press, Cambridge, UK, 1993.
12. G. Linden and S. Hanks and N. Lesh, Interactive Assessment of User Preference Models: The Automated Travel Assistant, in Proceedings of the 6th International Conference on User Modeling (UM-97), CISM, Vol. 383, pp. 67-78, Springer, June 02-05 1997.
13. Mackworth, A., Constraint Satisfaction, *Encyclopedia of Artificial Intelligence*, pp. 205-211, John Wiley and Sons, 1987.
14. Pattie Maes and Robert H. Guttman and Alexandros G. Moukas, Agents that buy and sell, Communications of the ACM, 42(3), pp. 81-91, March 1999.
15. PC-Zone, http://www.pc-zone.com
16. Personalogic/AOL, http://www.personalogic.com
17. Stolze, M. Soft Navigation in Product Catalogs, in *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*. C.Nikolaou and C. Stephanidis. Heraklion, GR, Springer, Berlin: 385-396, 1998
18. Stolze, M. Comparative Study of Analytical Product Selection Support Mechanisms, in *Proceedings INTERACT 99*, August 1999,
19. Travelocity, http://www.travelocity.com
20. Edward Tsang, Foundations of Constraint Satisfaction, In *Academic Press*, 1993.