

# Personalized Navigation of Heterogeneous Product Spaces using SmartClient

**Pearl Pu**

Ergonomics of Intelligent Systems and Design  
Database Laboratory  
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 Institute of Technology Lausanne  
CH-1015 Ecublens EPFL, Switzerland  
+41 21 693 2738  
boi.faltings@epfl.ch

## ABSTRACT

Personalization in e-commerce has so far been server-centric, allowing users to create individual profiles on servers that can be later accessed. However, as more product information is coming from multiple and heterogeneous sources, it is impossible for users to create many profiles. We present SmartClient, a technology based on constraint programming where a thin but intelligent client provides a personalized information access for its users. As the process can run on the user's side, it allows much stronger filtering and visualization support with a wider range of personalization options than existing tools. It also eliminates the need to personalize many sites individually with different parameters, and supports product configuration and integration of different information sources in the same framework. We illustrate the technology using an application in travel e-commerce, which is currently under commercial deployment.

## Keywords

Personalization, electronic catalogs, intelligent user interfaces, multivariate information seeking.

## INTRODUCTION

Assisting a shopper in finding the best product for his or her needs, also called product brokering, is a key requirement in e-commerce. Products have multiple attributes, represented by a vector of predefined features. We distinguish simple products and configurable products. The former, e.g., restaurants, cameras, or books, does not contain parts that can be substituted by others in a customization process, whereas the latter, such as travel products, personal computers or insurance policies, includes several constituent parts which must be composed together according to users' needs before it is offered. Product brokering in either cases requires assisting users in

finding information in a complex multidimensional space.

Several comparative shopping tools, based on technology such as Jango [5] or Junglee [13], have already been introduced and are in widespread use. These tools work on servers connected to a central product database, or on an infomediary such as a portal. They generally assume that the data source can be easily accessed and that data is delivered rapidly and reliably.

Recent developments in electronic commerce show the following trend. Since more and more merchants offer services and products via the Internet, product brokering requires gathering information services in a dynamic and distributed environment. Data come from multiple sources, they are heterogeneous (not every product has the same features), and products are configurable rather than simple. For example, in the travel industry, a portal (such as Expedia [6] or Travelocity [21]) is much more likely to offer products of a variety of different companies and integrated products (e.g., packages including air, car rental, and event-related parts) for coordinated planning tasks. While there is often a database of constituent parts (such as segments of flights in flight databases), the size of all possible combinations of these parts (i.e., customized trips) does not justify the deployment of a database. Furthermore, the data comes from multiple sources and usually cannot be integrated in a central database. Information must often be transmitted through network connections that are transient, limited in capacity, brittle, and costly. In addition, travel e-commerce has to allow travelers to modify trips while away from their offices and to interact with the travel site via mobile computing devices. These features are difficult to implement with an architecture that relies on a central server, but become easier in a distributed architecture.

Another shortcoming of current product brokering tools is that while they understand the top-level purchasing need of a user (e.g., buy a personal computer), they do not take into consideration secondary and contextual preferences, and offer few methods for product integration. Information services offered by these tools cater to a large population of users, focusing on serving the most prominent groups in a uniform way. For example, it is generally assumed that the overriding criterion of choice is price. However, this may

*LEAVE BLANK THE LAST 2.5 cm (1") OF THE LEFT  
COLUMN ON THE FIRST PAGE FOR THE  
COPYRIGHT NOTICE.*

not fit everyone: if I need to get a certain book in a hurry, price may matter much less than the delivery conditions. Unfortunately, there is no comparison shopping service that finds me the shortest delivery time to my residence. Even when services exist, these tools may not find them for a user because he has not stated the right criteria in the initial query. In buying a PC for instance, a user may ask for a simple configuration with low budget as the only criterion. However, upon viewing the products being offered, he may decide to consider flat (LCD) screens because they are now offered at much lower prices than he thought. However, because this criterion was not explicit in the initial query, the comparison shopping has to be invoked again so that all the offers coming back take this new feature into account. Finally, the server-centric architecture makes it difficult to support visualization of alternatives.

Several new techniques have recently been developed for personalized information seeking in complex multidimensional spaces. A few leading examples include dynamic queries for finding films [1], the FindMe approach for assisted browsing in selecting restaurants among a range of other things ([3,4]), and Lieberman's work on finding apartments ([16]). They present important advances in user experience and personalization.

We build on these advances to accommodate the requirements of travel e-commerce: many complex and often unsatisfiable criteria, configurable products, and distributed and heterogeneous data sources. To satisfy these requirements, both user experience and computation architecture have to be addressed. Additionally, this type of tools must also provide quality services to mobile network users, thus further constraining the size of the tools. In this paper, we describe our approach, called **SmartClient**, a distributed agent-based architecture for gathering information. It implements navigational features that can be tailored to the exact needs of each user. It offers solutions to

- capture the initial large quantity of "crude information" into a temporary data store,
- use constraint satisfaction problem solving techniques to model the data without full deployment of databases,
- help users to browse in this complex data space
- and assist them to choose the best solutions that fit their profile and dynamic criteria.

SmartClient uses on the order of 200K bytes of Java code to deliver all the intelligence that a user requires to perform information seeking and searching. Each application of SmartClient for a particular domain requires slightly different implementation of the graphical user interface (GUI). Travel planning using SmartClient, *Isy-travel*, employs some of the most sophisticated GUIs called *contextual navigation maps* (discussed in detail in a later section on user experiences). It requires additional 300K bytes for the GUI part, which includes a real-time zoomable

world map. On the other hand, we succeeded in implementing a version of *Isy-travel* for the PalmPilot using no more than 250K bytes of code. The following discussions of our approach will be in the domain of travel planning.

**SMARTCLIENT: not fat, not thin, but agile**

The idea of SmartClient is shown schematically in Figure 1. It is based on decentralizing the comparison processes so that they take advantage of the processing power available at each end user's computer. Rather than pre-classifying data at the server side, as would be done in common web services technology, data from different sources is sent to the client in compressed form and filtered according to the user's own combination of criteria at the client side.

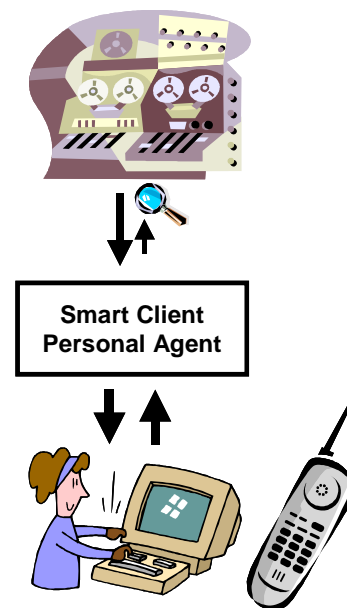


Figure 1: SmartClient brings intelligent behavior to the end user.

Criteria are modeled as constraints. The main mechanism for obtaining them is by letting the user express critiques on solutions. The results are then arranged in a variety of graphical forms, including tables, Starfield displays, and data maps. This allows a user to navigate in them while being aware of the context.

**OUR APPLICATION DOMAIN: TRAVEL AND TOUR PLANNING**

Travel is an example of products having a large range of choices and criteria. In spite of the fact that travel appears to be an ideal product for e-commerce - there is no need to keep warehouses of inventory - the current tools do not handle this complexity very well and travel e-commerce has consistently lagged predictions. The travel planning problem is to assemble a travel plan, including transportation (air, train, etc.), rentals (in particular cars and

hotels) and events such as theaters. Options for each component are available from various information systems; each airline, car rental agency and hotel operates its own reservation system and usually makes it accessible through the World Wide Web (WWW) and other specialized interfaces.

A particular difficulty in travel is that for each option, only a limited number of seats, cars, or rooms are available. Thus, data is highly dynamic and must be obtained specifically for each request. In turn, once a suitable plan has been made, it must be turned into reservations. The job of providing schedule, pricing and inventory data as well as accepting reservations on a worldwide basis is being handled by Global Distribution Systems (GDS) (Sabre, Galileo, Amadeus, Worldspan). These operate data centers that periodically duplicate the contents of most airline, car and hotel reservation systems, and provide travel agents with access to this data.

However, the information contained in GDS is incomplete in various ways. In particular, they do not deliver all possible fare options, and data regarding car rentals and hotels is often inaccurate. Therefore, most travel agents also access additional data sources, in particular fare and hotel databases. To provide plans of the same quality, e-commerce systems must also integrate this information. We have developed a travel planning system, *Isy-travel*, based on SmartClient technology. Below, we describe how *Isy-travel* supports the user's travel planning through intelligent interfaces.

#### **ACQUIRING USER CRITERIA**

The main challenge in building a personalized filtering tool is to obtain an accurate model of the user's criteria and preferences. This is difficult for two reasons:

- 1) users cannot state all their criteria up-front. For example, I might hate to change planes in Heathrow, but I would not know enough to state this requirement initially.

- 2) criteria are highly variable. For example, the same user may be planning a business trip in the morning, where time constraints are the most important, and a trip for his family in the evening, with price as the overriding criterion. We have adopted a solution, similar to [17], based on the principle that people find it easy to critique proposed solutions - if there is nothing to critique, they have already found what they were looking for. We implement this using a mixed-initiative dialog: the system generates 30 solutions that are good according to the known criteria. The user examines these and states additional constraints, generally expressing aspects that he dislikes about one or several of these solutions.

The system again generates the 30 best solutions according to the revised criteria, and the cycle continues. We start the process with a user profile that contains invariant criteria such as preferred airlines, general preferences for short

flights and few stops, etc. Criteria are added incrementally as planning proceeds.

An example of interaction with *Isy-travel* is shown in Figure 2. The user is planning a trip from Geneva to Hamburg. Considering the available flights, he adds a constraint that he would not like to change planes in Brussels, an airport often plagued by strikes. At any time, the user can ask the system to generate the 30 best solutions according to the current criteria. Thanks to an efficient search process, this usually takes less than 0.1 seconds, even when the system has millions of potential combinations. When it is not possible to find a solution that satisfies all criteria, the system generates the solutions that respect them as much as possible. Violations of criteria are acknowledged by coloring the corresponding cells in the display in red, such as shown for the arrival time in the solution shown in Figure 2.

Users can inspect the criteria that are currently being used for generating solutions in a separate display. This display also allows deleting or changing the weights of preferences. By incrementally manipulating his or her set of criteria and observing the effect on the resulting selection, the user eventually converges on a small set of solutions that best fit the criteria, and manually selects the right one for his trip.

In comparison with commonly used travel planning systems, *Isy-travel* has the advantage of letting the user indicate his decisions in any order. For example, if a user cannot find flights with suitable schedules in economy class, he can relax his preference so that business-class becomes an available choice. In conventional systems, he would have to start the planning process all over from the beginning, since the choice of class is forced to be stated in the beginning. Similarly, it is possible to modify airline, time, and date preferences in any order in *Isy-travel*.

Another advantage of the constraint-based mechanism is that the criteria gathered accurately represent the state of the planning process at that point in time. It is possible to leave the planning and come back to it at a later time simply by storing the criteria in a database. In travel planning, such incrementality is an important usability feature since people often plan travel over time before committing to a particular plan. It also simplifies changing travel plans, since earlier criteria can remain active and only changes have to be communicated. Our earlier paper at CHI2000 [19] provides a more detailed discussion on the criteria solicitation process.

#### **Generating personalized solutions**

Since we allow users to freely state whatever criteria come to their mind, it soon happens that they become contradictory and no longer allow any solutions that satisfy all of them. We deal with this problem by a process of *partial* constraint satisfaction [8], where we generate solutions that go the farthest in satisfying the user's criteria. Each constraint is given one of four different weights. The

lowest weight is carried by a large set of common sense constraints, such as short flight time, low numbers of stops, not changing airlines. The next higher weight is applied to criteria taken from the user's profile, such as preferred airlines, hotels, or car rental companies. The third weight category is applied to criteria specifically stated during the current interaction. Finally, the highest weight is given to hard constraints, for example, that you cannot leave from somewhere before you arrive.

Travel plans are assembled from the available components using a branch-and-bound search process that generates the 30 best solutions. Solutions are ranked according to a sum of penalties for violated constraints, where each constraint violation generates a penalty equal to the weight associated with that constraint. For constraints on continuous quantities such as price and time, we use a nonlinear function that generates penalties according to the degree of violation.

### USER EXPERIENCE AND CONTEXT MAPS

User experiences in an information seeking and searching environment consists of the following requirements:

1. Users must be able to evaluate how well the information gathered is meeting their task needs. Because this has to be done in an incremental way, a visual reification of the task would help reduce their cognitive load and augment his problem solving capabilities.
2. It is important to convince users that they have found the right solution for their tasks. This requires giving them an overview of the context in which the solution is placed.
3. Selection of a product is highly dependent on what is being offered. Serendipitous behavior can be supported only if more than one product is offered and compared.
4. Finally, when initial criteria need to be modified in big quantities, users should be able to manipulate them without having to do too many small tweaks.

In *Isy-travel*, we explore the use of contextual navigation maps to fulfill the information seeking and searching criteria stated above. Maps, according to Boden [2], allow people to navigate through spaces and provide a visual structure regarding the boundary of spaces. A dynamic map also helps users evaluate his current path against his destination, provides possible alternatives, and allows him to express his navigation task in a visual way. Context maps also allow users to focus on a subspace of exploration, while keeping some semantic contexts in the peripheral perception.

#### Task context

The information seeking environment should visually represent users' tasks (visual reification [10]) so that they are aware of the status of each task, the resources each requires, and most importantly are stimulated to act upon

them. There are two types of task maps employed in *Isy-travel*. The first one (reported in [19]) consists of an interactive real-time world map where users can select their origination and destination by clicking on the names of airports. This tool enables the discovery of alternative airports, is useful to users who are not familiar with the geography of their destinations, and simply fun for the others. The second one visually overviews offered products using b-splined curves outlining the different travel routes (Figure 2). By clicking on those lines, users view the detail of each proposed itinerary in the table at the bottom. At the same time, undesirable features can be excluded by clicking on the dropdown menu on cells where such constraints can be specified. This task context map thus allows users to verify how products from different sources are satisfying their travel needs.

In the case of vacation packages, users enter a set of criteria. The available packages returned from the search are grouped depending on its geographical location. They are further characterized by the degree they match the users' criteria. At this point, each cluster of geographically related packages is displayed in a pie chart (Figure 3). Colors from green to red render the degree of matching. This task context map allows users to identify products that are outside of his initial search criteria. For instance, someone who lives in Europe might find packages in Spain more attractive because potential savings on air travel, even though he originally wanted to go to the Caribbean islands.

#### Overview context

Buyers shopping for products like to be convinced that they are getting the best deal. Depending on the context, a best deal can be the cheapest, the quickest in time (for travel), the most comfortable, the product from the most favorite brand, and etc. In short, this is a task of optimization in a multivariate space. Our solution (Figure 3) is a 2D scatter plot where we display the 30 best solutions with two default criteria: the total flying time and the total cost. Other criteria, such as the number of stops and the total time spent in connection, can be selected when the axes are changed.

Even though we can implement this interface using more advanced techniques similar to those used in [15], a 2D scatter plot is more familiar to most users and is easier to manipulate. It is easy to identify solutions that perform the best in all given criteria, called dominant solutions in [18], which lie at the extreme end of the space of the map. Further, since the overview is a tradeoff space, when dominant solutions do not exist, users automatically perform tradeoff analysis using the available criteria. For instance, a user can choose a flight that is shorter in flying time, but is slightly more costly. In most travel portals, which only consider cost as the dominant criterion, a flight with good connection time but costing 50 dollars more might not show up until 3 pages later.

### **Parameter context**

Configurable products offer a large range of possible values for each of its constituent part. It is especially the case in travel. Planning a flight from Geneva to San Francisco with a stop in Boston involves configuring a handful of transatlantic segments with a large number of domestic flights within the US.

Parallel coordinate [12] is a visualization technique originally designed to display data sets in high dimensions. It is a very effective way to manipulate parameters largely due to its simple visual affordance. All parameters describing a solution will be represented by a vertical line notched to display the data value as well as a slider for manipulation. When a solution is selected, a lateral line tracing all points lying on the vertical lines is displayed, showing the individual characteristics of that solution as well as giving a global view of how that solution compares with the whole set.

The high number of parameters involved in travel plans prohibits the display of one parallel coordinate for the entire trip. Thus only one segment at a time is displayed. It is especially effective in showing bursts of a solution space where parameters cover a wide range. As shown in Figure 5, the portion between Denver and San Francisco has many similar solutions. With the help of sliders on each vertical coordinate, users can quickly put constraints on arrival, departure time, or both.

### **RELATED WORK**

#### **Information seeking systems**

The query-based information retrieval paradigm no longer suffices for the information seeking task in multivariate spaces. Modern tools emphasize assisting users to formulate their goals [3,4], helping them solicit hidden criteria [17, 19], comparing a large collection of items in context, and providing information about the structure of the information space. A differentiating factor of our work is that we have severe constraints on users' interaction with data sources, and the heterogeneous and configurable nature of data sources. All of this provides additional challenges for the underlying architecture as well as ensuring quality user experiences.

#### **Related visualization techniques**

The dynamic query method [1] combined with Starfield displays employs 2D scatter plots to visualize an entire data collection. Users are able to first perceive the structure of the entire collection (overview) before focusing on a subspace by manipulating the attribute sliders. A limitation is that such a method can be applied to quantitative data, excluding nominal and categorical data. Our overview context map has been inspired by the StarField display. However, we use it for performing tradeoff analysis, rather than as an overview of a database.

Parallel coordinates were traditionally used to display and manipulate data sets in terms of attributes. We use them to

allow users to quickly navigate in densely related product spaces. One problem with our implementation of parallel coordinates is that the number of vertical axes quickly grows and manipulating non-quantitative data has not been trivial. A possible future work is to employ tabular methods such as TableLens [20] or InfoZoom [11]. They overcome the space limit on the display problem by integrating fisheye view techniques [9]. Thus an area of interest will be provided sufficient space for exploration and manipulation, while the context stays in the background. Symbolic and quantitative data are mixed in the same display, but in different tabular cells, thus avoiding visual confusion of the two types of data.

#### **Travel planning**

Our group has done a survey of ten on-line travel sites [14]. The conclusion was that while most users can get through the process, a number of usability issues were to be addressed. Most noticeably users do not like the fixed sequence of question answering procedure before they can view offered products. Worse, when they change their criteria during product offering, an entire search process has to be launched. We first reported our solution to a more user-centric online travel planning at CHI2000 [19].

A more recent work described a mixed-initiative system for travel planning [7]. The system takes users requests and gathers information from different information sources. It tries to automatically fill some of the fields at the planning stage by using calendars, address books, and etc. Furthermore, the interface also prompts users to help with filling in missing information by using visual cues.

#### **CONCLUSIONS**

The world-wide web has been developed with a particular emphasis on server-centric functions. This was driven largely by an e-commerce business model where the seller is responsible for properly advertising his product, and he can do this only by providing the servers. However, in travel the model is now changing in that suppliers are no longer willing to pay commissions that would pay for customer-specific services. At the same time, travelers are overwhelmed by the complexity of the planning task. The result is that travel agents can levy per-ticket charges to cover their consulting services. Applying this same model to e-commerce, it becomes imaginable that end users will invest in better tools to help them plan trips using the Internet.

We have shown how a user-centric approach using personalized software, *Isy-travel*, can significantly improve usability of travel e-commerce. Using SmartClient and the contextual navigation maps, customers can rapidly get an overview of available trips and obtain better solutions than those found by a travel agent who is less aware of their criteria.

The approach is based on constraint programming as a basic paradigm, which provides for a combination of advantages:

1. A simple computational mechanism provides for lightweight yet intelligent code that can be easily deployed through a network.
2. Constraints provide a uniform framework for expressing both data items and user preferences, allowing smooth integration of user interaction and problem solving.
3. Constraint programming is already the leading technology for product configuration and thus allows the search process to be easily applied to configurable products with different sources.
4. Search allows to easily generate context maps that support task reification, tradeoff analysis, and mixed-initiative variation of preferences and parameters.

This combination makes SmartClient a highly practical approach for brokering complex products. We have developed applications in travel – now in deployment as a commercial product – insurance and electronic catalogs.

We are currently working on extending the approach to more general personal agents, combining the user experience with proactive behavior that supports on-going decision processes over time.

#### ACKNOWLEDGMENTS

This work was supported by the Swiss National Science Foundation within its competence center on e-commerce technologies. We also thank Marc Torrens, Sebastian Gerlach, and the team of Iconomic Systems for their implementation on the architecture and graphical user interfaces.

#### REFERENCES

1. Christopher Ahlberg and Ben Shneiderman. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays, in *Proceedings of ACM CHI'94*, ACM Press, 1994.
2. Boden, M. *The Creative Mind. Basic Books*, 1990.
3. Burke, R., Hammond, K., and Cooper, E. Knowledge-based navigation of complex information spaces. In *Proceedings of the 13<sup>th</sup> National Conference on V. Pareto*. (1896). Cours d'économie politique, Technical report, Rouge, Lausanne, Switzerland. *Artificial Intelligence*, AAAI, 1996.
4. Burke, D., Hammond, K.J., and Young, B.C. The FindMe Approach to Assisted Browsing. *IEEE Expert*, pp. 32-40, July-August 1997.
5. Excite Shopping. Available at: <http://www.jango.com/>. Accessed September 27, 2001.
6. Expedia. Available at: <http://www.expedia.com/>.

7. Frank, M., Muslea, M., Oh, J., Minton, S., and Knoblock, C. An Intelligent User Interface for Mixed-Initiative Multi-Source Travel Planning, in *Proceedings of Conference on Intelligent User Interfaces*, ACM Press, 2001.
8. E. Freuder and R. Wallace. Partial Constraint Satisfaction. *Artificial Intelligence*, 58:21--70, 1992.
9. Furnas, G. W. Generalized Fisheye Views. In *Proceedings of ACM SIGCHI Conference*, pp. 16-23, 1986.
10. Furnas, G. W. and Rauch, S. J. Considerations for Information Environments and the NaviQue Workspace. In *Proceedings of International Conference on Digital Libraries (DL 98)*, ACM Press, pp. 79-88. 1998.
11. InfoZoom. Available at: <http://www.humanit.com/en/products/infozoom/>. Accessed September 27, 2001.
12. Inselberg, A. The plane with parallel coordinates. *The Visual Computer*, 1, New York, Springer Verlag, pp. 69--91, 1985.
13. A. Rajaraman and P. Norvig. Virtual Database Technology: Transforming the Internet into a Database, in *IEEE Internet Computing*, Vol. 2, pp. 55-58, 1998,.
14. Jurca, A. Survey of Online Travel Planning Systems. Technical Report (No. 99-01), ISR/DMT, Swiss Federal Institute of Technology, Lausanne, 1999.
15. D. Keim and H-P. Kriegel. VisDB: Database Exploration Using Multidimensional Visualization. *IEEE Computer Graphics and Applications*, 14(5):40--49, 1994.
16. Lieberman, H. Autonomous Interface Agents. In *Proceedings of CHI'97*, March 1997.
17. Shearin, S. and Lieberman, H. Intelligent Profiling by Example, in *Proceedings of Conference on Intelligent User Interfaces*, ACM Press, 2001.
18. V. Pareto. Cours d'économie politique, Technical report, Rouge, Lausanne, Switzerland, 1896.
19. Pu, P. and Faltings, B. Enriching buyers' experiences: the SmartClient approach, in *Proceedings of ACM CHI'2000*, ACM Press, ???
20. R. Rao and S. K. Card. The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information, in *Proceedings of ACM CHI'94*, ACM Press, pages 318--482.
21. Shearin, S., and Lieberman. Intelligent Profiling by Example. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI 2001)*, Sante Fe, NM, January 14-17, 2001.
22. Travelocity. Available at: <http://www.travelocity.com/>.

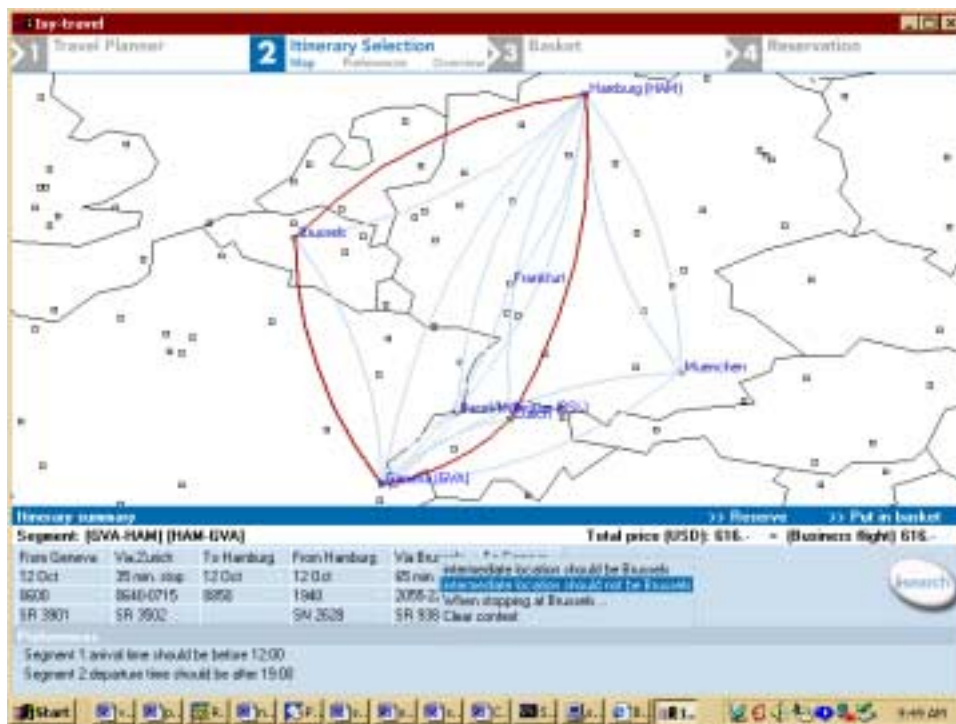


Figure 2: Task context: users perceive how offered products meet their travel needs visually.

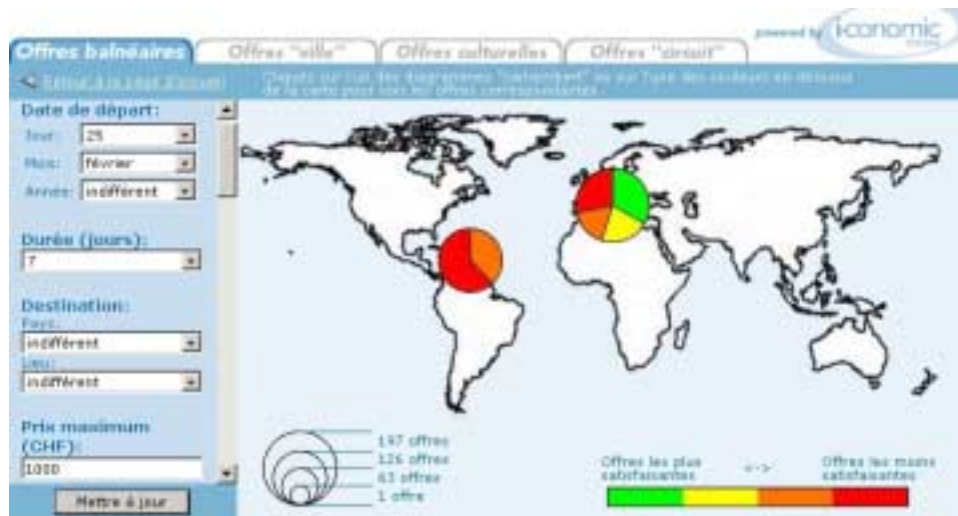


Figure 3: Users perceive the distribution of matched and semi-matched vacation packages in a world map.

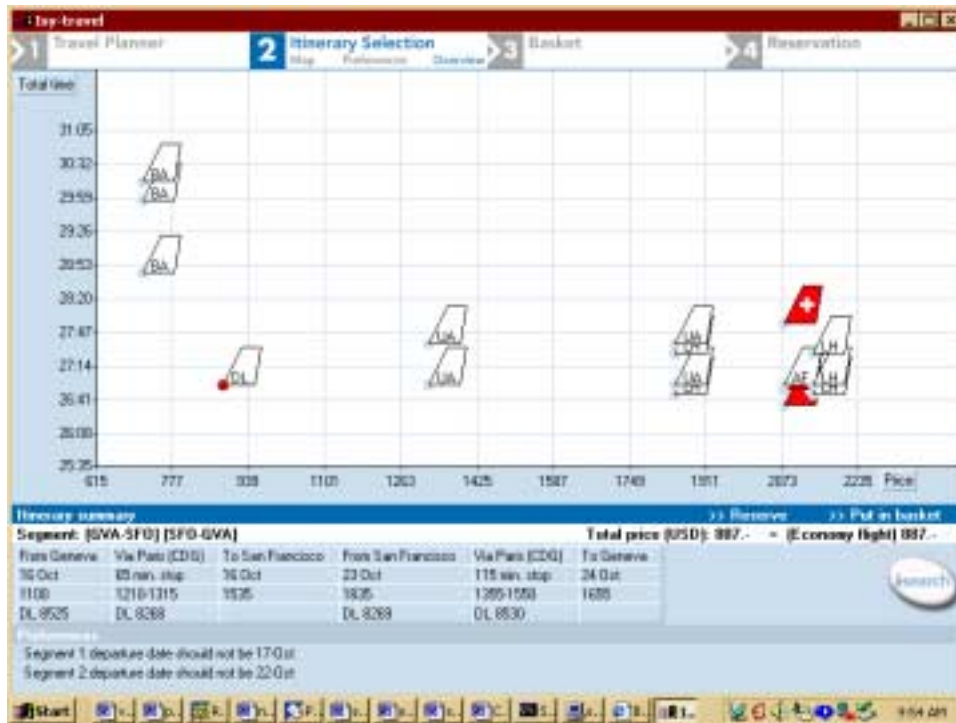


Figure 4: Overview context: supporting serendipitous discovery when selected solutions are compared with others.

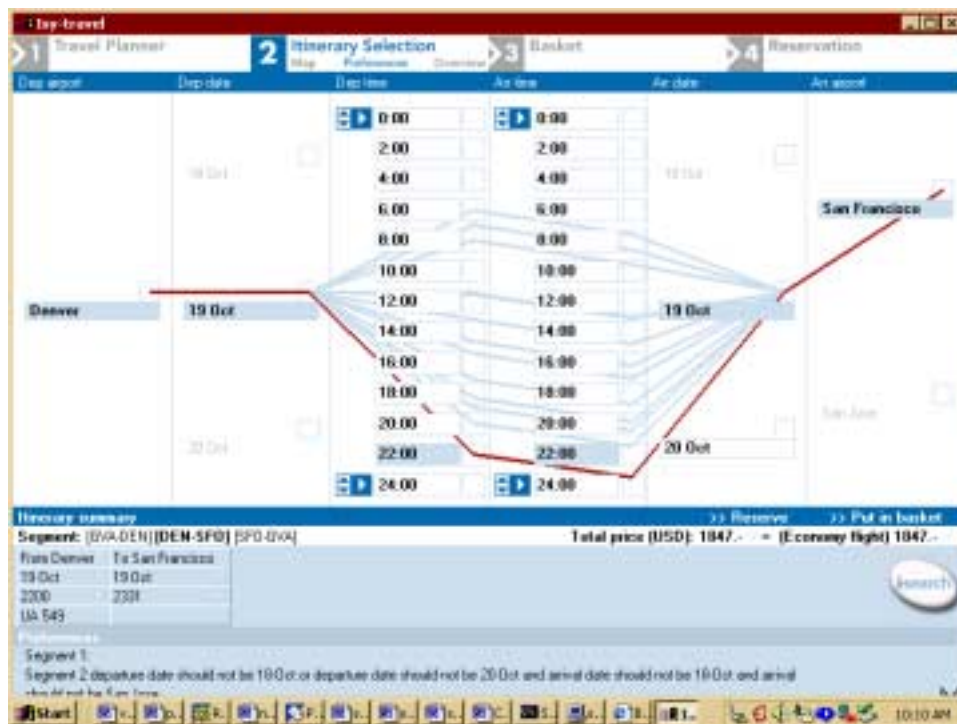


Figure 5: Parameter context: parameters are uniformly presented allow modification in large quantities.