

# Flexible User Interfaces for B2C Systems, Using Booking as a Case Study

Olsen, K.A.

Molde College  
N-6402 Molde Norway  
[kai.olsen@himolde.no](mailto:kai.olsen@himolde.no)

Malizia, A.

Dip. Scienze dell'Informazione, Universita' di Roma,  
Via Salaria 113, 001895 Roma (Italy)  
[malizia@di.uniroma1.it](mailto:malizia@di.uniroma1.it)

## Abstract

With the Internet and the Web the “terminal” can be moved into our homes, allowing us to access databases of any kind directly without going through intermediates. In practice these business to consumer applications have been implemented by building front ends to the old legacy systems. In this paper we question if this is enough? Many customers going to the Web to make a booking may be flexible with regard to dates and times, where to go and even if to go. The interfaces that are offered today expect detailed and specific data and do therefore not account for this flexibility. In this paper we discuss the possibilities of retaining some of this functionality offered by the human travel agent by enhancing current Web interfaces.

After presenting a background study, we will suggest interfaces that to a better degree can aid the customer in performing a booking. While we use booking as an example, many of the ideas presented here can be applied to other systems as well.

## 1 Background

With the Internet in our homes we can take intermediates out of the loop and do the job ourselves, interfacing directly to the airline booking system, the bank system, the online shop. Many companies have implemented these B2C (Business to Consumer) systems by building a Web interface on top of their legacy systems. Thus customers are now using the same systems as the intermediates used earlier (Lightner, 2004).

This works fine in many cases. Take booking as an example. With some enhancement of the interface, letting users select from lists, choose a date from a calendar, offer a credit card number, and click on buttons the booking process can be performed by anyone with a minimum of computer and Internet experience. This works for simple closed request, i.e., request that can be mapped directly into formalized terms such as dates, airports, flights, etc. However, they break down for the more complex closed requests, i.e., where the customer is flexible with regard to attributes such as destination and dates. Further, current systems cannot handle the more open requests that cannot be mapped directly into the formalized terms offered by the Web interface. These are the cases where the intermediate earlier used her knowledge and experience in order to aid the customer.

For example, a request to the intermediate, the travel agent, may be to find an “inexpensive weekend trip from Pittsburgh to New York”. Initially the agent can expand this open request by asking the customer what type of hotel he is interested in, both with regard to quality and location, and on which dates he may wish to go. The agent may use her experience to suggest times of the year when there may be a possibility of getting bargain tickets, also to suggest airlines that offer discount fares to discount airports. She may even ask what he wants to see in the city, and may even recommend Washington DC as better alternative if he is interested in taking the kids to museums and avoiding expensive hotels.

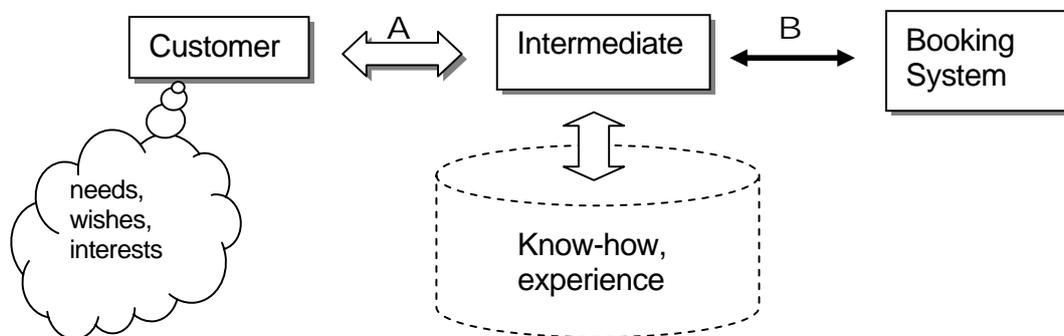


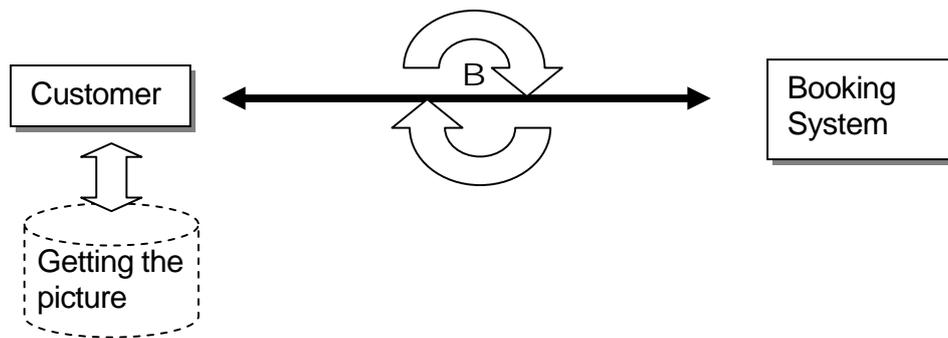
Figure 1: Booking tickets through a human intermediate (a travel agent)

What the agent does is to formalize the customer's open request. This is seen from Figure 1 above. While the first part (A), between the customer and the intermediate can be open or closed, the second part (B) must be closed, formalized to the level of the B2C system. In this process the intermediate will use her know-how and experience, here illustrated as a "database", to aid the user. Thus the A = "inexpensive weekend trip from Pittsburgh to New York" may be replaced by B = "Search flights 04/12, return 04/14, Pittsburgh-Washington DC; Washington DC Hotels 04/12-14".



**Figure 2:** Booking tickets on the Web

Today's B2C systems only support the case where all customer data are closed, and most systems only support the simplest of these closed requests. That is, the situation is as described in Figure 2 above. We see that the open part has disappeared, the customer himself must formalize the request into the terms required. When the customer does not know exactly what he is interested in, the limited flexibility of the B-part becomes a serious drawback. Note that the limitations of the B-part were seldom a problem for the human travel agent. At this point in the process she would have used her experience to pinpoint the alternatives. She then uses the system to check out availability, to get detailed prices and to perform the actual booking. The limitations first become apparent when we remove the more open part that went on between the customer and the travel agent.



**Figure 3:** Getting the picture by repeated searches

The customer can work around some of these limitations by trying all alternatives, and then choosing the "best" alternative. Through extensive searching it may be possible to accumulate some of the know-how and experience of the agent, at least with regard to the customer's special needs and wishes (Figure 3). However, without the experience of the travel agent this may be a tedious job without any guarantee of a satisfactory result. The customer will not only have to check out various combinations of locations and dates, but will also have the job of organizing the data returned.

We could try to replace the travel agent by a "virtual intermediate", an "intelligent" system that could embody the knowledge of the travel agent. Here, however, we shall see that by viewing the booking system also as an *information* system, many of the problems that ordinary travelers meet on the Web may be resolved. The task of the information system will be to provide the user with a general picture that can be an aid in taking decisions. This overview may be offered based on data from the customer. By allowing him to select a range of alternatives, e.g., a range of cities as destination, a range of dates for departure and to limit the scope by setting constraints, e.g., "latest return Sunday night", "any weekend except...", the system can by extensive searching find the best alternatives. Or, if there are no capacity for performing these searches, to let the system collect statistics based on previous searches.

In order to open for broader queries we will have to formalize the often informal knowledge that the intermediary had before. For example, in order to answer a request as to "a place suitable for children" we need a value for this attribute for all locations. This could be set manually, e.g., with a star ranking, but will be a major task. In this paper we explore alternative ways of retrieving these data. Efforts to formalize open data have been tried with moderate success by natural language systems and search engines, but in our case there are many ways of ensuring correctness of data, especially when the information on the Web can be backed up by statistical data.

## 2 Scenarios – flexible Web user interfaces

We shall describe the requirements for more flexible Web interfaces by describing some scenarios. Each shows how a more flexible user interface can aid the user in the searching and booking process:

- Joan lives in London and has a friend in Berlin. In her last letter her friend invited her to come over for an extended weekend. Joan has some flexibility in her job and can leave London on any plane later than 4 PM on Thursday as long as she can return Sunday night, alternatively leaving on Friday and returning Monday. She tells the system that she can go any weekend in the next six months. Based on these data the system should immediately give her an idea on fares (minimum fare, average economy fare), departure times, arrival times, etc. for different weekends through the next six months. This information tells Joan that she will get the best offer if she takes a Thursday to Sunday trip in October, and that fares are then within her limit.
- Per lives in Oslo, Norway and wants to take his 2 kids to London during the school holiday. He provides the system with the necessary information (destination, date range, one week with bargain hotel) and finds that the minimum price is far higher than what he is willing to pay. He therefore gives the system a range of alternative cities, and from the data returned he finds several interesting destinations that are within his price range.
- Luigi lives in Rome and wants to attend a wedding in Milan next Saturday. He tells the booking system that he has to be in Milan from noon Saturday until Sunday morning. Based on this information the system will give him an overview of travel times and minimum fares for buses, trains and planes.
- William and Maria are retired and live in Minneapolis. This winter they are interested in going on a cruise for a week. They can go at any time during the winter months, and do not have any special requirements. They are, however, not sure if they can afford such an expensive holiday. Choosing cruise, Minneapolis as the departing city and a length of one week, the system will give them an overview of prices for various cruise companies. Each price is offered as an example only, but with the possibility of getting more information (destinations, cabin data, etc.).

Such a system will make bookings performed by ordinary travelers more efficient, and will also ensure that the users find what they need. As we shall see in the next section, it should be quite simple to implement the basic functionality.

## 3 User needs, handling flexibility

Today's interfaces satisfy the needs of the customer with specific requests. Booking a flight between Pittsburgh and New York on the first flight tomorrow is quite simple, especially if one is willing to pay full price. However, as we have seen the problems arise when the traveler is flexible with regard to times, dates, destinations, hotels, etc. The request may also be sensitive with regard to prices and flight times. For example, we may only take the trip if the costs are reasonable, if we get a morning flight, if we can return Sunday night, if we avoid stop-overs. The challenge is to develop interfaces that can accommodate this flexibility.

We shall present this discussion in four parts:

1. Providing data on user needs, requirements and interests.
2. Giving the user the overall picture
3. Implementation
4. The actual booking

In this first discussion we shall assume that the data provided by the user and the data in the database are formalized to the same level, so that matches can be performed directly by the system (e.g., on destinations, dates, times). The handling of more open data will be discussed in the next section.

### 3.1 Providing specifications

In the most open case the user may go to the Web system with travel as his only "need". However, even the human travel agent would be astonished if this was the only specification that the user could give. In most cases the customer will be able to specify more, on destinations, dates and price range.

For example, destination may be given as the name of a city, but may also be more broadly specified as a set of cities, or "a place in the sun", "the Caribbean" a "ski resort". In some cases there may be no destination, e.g., the users need may be specified as a "cruise". Further, dates may be specified by day/month/year or by names of days "leaving on Friday",

“returning Sunday” or may be left out altogether (“a one week holiday”). Accommodation may be specified as the name of a hotel, a hotel chain, a class of hotel (“three star”) or by a maximum price (“we are willing to spend up to \$100 a night”).

We suggest that the specification of these requirements and wishes are given up-front. Since many of the needs would fall into clear categories this entry process could be performed by a simple form, or by a wizard. Here the users could indicate places that they wanted to see, for example by checking out the various locations. To simplify, the system should also be able to group locations, such as major cities, historic sites and resort locations. In addition, or as an alternative, the users can select what they are interested in seeing or doing from an alternative list. Here a user may check out “theaters” and “art exhibitions,” while another is interested in “swimming” and “scuba diving”.

Some of the current Web interfaces allow for some flexibility with regard to dates, for example with the possibility of trying the week ahead or the week after. Ideally the user should be able to specify any possible pattern. In the interface this can be specified by indicating interesting dates in a calendar, by specifying an interval of interests (from-to, or next 3 months) and by setting departure and return days, as we discussed above.

By filling out such a form or running through a wizard we try to simulate the conversation that goes on between the customer and the human travel agent. That is, the phase where the agent tries to get an idea of what the customers interests really are. As we have seen, this can imply that the agent offers different alternatives to what the user had initially thought of (e.g., Washington DC as an alternative to New York). This is, of course, more difficult to do in the more formal setting of a computer system. But this can be partially implemented by letting the user distinguish between requirements and wishes. For example, the user may offer departure, arrival date and a major city in Europe as requirements, but that the specification of London is only a wish. That is, the user may accept other destinations than London as long as the requirements are followed.

### **3.2 Getting the overall picture**

An important task for all information systems, for example the Execute Information Systems (EIS) used by many companies, is to offer an overview of the data in the system. This overview may be presented as graphical trend charts, reports, statistics, tables or as status data. With more and more data gathered electronically these systems can offer more reliable and more updated information. For example, with IT systems at the gate and electronic tickets the number of passengers on each flight, who they are and what they have paid will be known by the airline’s EIS system before the plane leaves the ground. This gives important data back to the executives, to the marketing department, and to flight planners. Similar, the executives of the online bookstore, auction site, music store, etc. will enjoy a similar feeling of being up-to-date — to have the background information they need in order to make decisions.

A customer going to a Web site does not need as much information as the travel agency or the airline executives. However, the Web system should be available to give statistical information on:

- availability
- prices
- departure and arrival times

based on the requirements and wishes that the user provided in the initial phase.

The important factor here is to give users an overview with a minimum of effort on their side. As we see, by allowing some flexibility in the input data the Web system should be able to give the overall picture. Based on this the users can go on to a more detailed booking process or use their flexibility to look for other destinations, dates, etc. if the offers did not suit their needs. Of course, this overall picture can also be created by using today’s systems, but then the information has to be gathered piece by piece and put together by the customers themselves. This is a time consuming and quite unnecessary process.

What we propose here is an information system that can provide the user with all the necessary data, just by a button click after the initial data has been provided.

### 3.3 Implementation

There are several methods that can be used to let the system retrieve the data needed to give the user this overall picture:

1. Extensive online searching.
2. Statistics
3. Reports based on historical data

Each method offers different levels of information quality and set different requirements with regard to the resources that are needed to perform the operations. We shall give a brief discussion of each method below.

#### Extensive online searching

The system will perform a search based on the initial data provided by the user. Since this may encompass a high degree of flexibility, in the worst case all destinations and all dates, this will require major resources in the form of fast servers and efficient databases. This may seem to be a major undertaking today, but already search engines such as Google are able to do this on the Web (Zhang et al, 2003). The trick is, of course, to gather as much information as possible ahead of time and store the indexes (the inverted Web) on fast servers. Compared to Web engines the booking system has the advantage that it searches in more formalized databases, but the disadvantage that updated information is needed. However, in any query there exist cut-off values that can be used to simplify the searching.

We should also suppose that with a flexible system in place that users will spend less time on the Web system. That is, if we can give them an overall picture right away there is no need for to try out many other alternatives.

#### Statistics

Alternatively, to save computer resources, the system could collect statistics based on all the searches that are performed (Shim et al, 1999). For example, if a result set to a query shows that the cheapest flight between New York and London are \$120 in November, the system could retain this as a minimum fare for this month on this destination, at least until another query give a different value. The idea will be to use ordinary user queries as a basis for gathering general information. In addition, the system itself could use any idle time to update its tables. With such a system a majority of user requests can be answered by a simple lookup, albeit with the drawback that the information is not updated to the last minute. In practice, the minimum price may no longer be available. However, since the main goal is to give the user an overview over the situation this may be satisfactory in an initial phase.

#### Reports based on historical data

The human travel agent will know from her experience that winter is the high season in Hawaii, and that there may be good offers in August, that hotels in New York, Paris and London are expensive, especially during summer. She uses this information to assist her customer. This information is, of course, also available for the booking system – as explained above. The idea here is to use this information to give the user a general report on the destinations and times that are provided. The report can include data such as price ranges, availability and when a booking is normally needed. It will, of course, be customized with regard to the initial customer requirements and wishes. A good example of the analysis of historical data is provided in Riedewald et al, 2002.

### 3.4 Performing the booking

Let us assume that the user is satisfied with the information that was received from the overall information system. The task is now to perform the actual booking. However, now the user can have a very specific idea of what he wants with regard to locations, dates, etc. If the overall data is based on statistics the user may not have all the necessary information, so some sort of flexibility may still be needed, for example, by trying the week ahead or behind the date offered if there is no seat available. In principle, however, today's Web interfaces will do the job.

## 4 Unformalized data

In the examples above we worked with formalized data, expressed as locations, dates, etc. However, if we return to the travel agency situation the customer can also provide unformalized or open data. For example, asking for a "good hotel," "a place suitable for families," "close to the beach", "suitable for the elderly"... In these cases the human travel agent will utilize her knowledge and experience to find accommodation that fulfills user needs.

On the Web such request needs to be formalized, such that the system can match request with the information in the databases. The formalization can be performed by the customer (“not more than 200 yards from the beach”), by the providers (“family hotel”) or by an unofficial or official organization (“3 star hotel,” “lifts, handicap access”). Ideally, these attributes should be stored in a record-based structure, making searching fast, accurate and efficient. However, in practice it may be difficult to get such data for all different types of hotels. An alternative is to use free text searching. This requires a lower formalization level on the database side, but results will be flavored by the pitfalls of text searching (Olsen et al, 1998). For example, a simple search engine looking for places suitable for the elderly may react positively on the words “lift” and “suitable for the elderly” in the sentence “the hotels has no lift, and is therefore not suitable for the elderly.”

Customer’s reviews are another option. A Web based travel agency may invite customers to evaluate trips they have been on, similar to the way online bookstores ask readers to evaluate books. While such review systems open for the possibility of fraud, i.e., that a hotel owner activates family and friends to give excellent reviews, this problem may be reduced by only allowing persons that have booked a trip to submit a review.

In addition we can get interesting statistics from the system. Where do families with small children go, to which locations and to which hotels? Where do the elderly go? To which places do travelers return most often? This is valuable data that can be provided to the customer as additional information to close the open case. Such a system would be similar to the “customers that bought this book also bought...” functionality of Internet bookstores (see for example Cumby et al, 2004).

## 5 Summing up – the need for flexible user interfaces

Requirements Phase		
Functional Requirements	Description	Implementation
Finding travel packages meeting user specifications	Providing data on user needs, requirements and interests	Extensive online search based on data provided by the user
Describe travel packages meeting user specifications	Giving the user an overall picture of the results (could be used for refining)	Statistical information on: availability, prices, dept./arrival dates, ...
Suggest complementary products	Predicting customer needs based on customer who bought similar packages	Cross-sell recommendation
Suggest complimentary products	Recommendation based on past purchases	Mining and summarising customers reviews
Provide communication with other customers	Customer reviews, customer comments and discussion board	e-mail, chat, collaborative tools
Access travel package literature and news	Special combinations and discount, travel guides, ...	Press releases, online search, updated statistics
Compare travel packages	User interests and wishes comparison mechanism	Efficient integration and aggregation of historical data

**Table 1.** Requirements phase

Ives and Learmonth (1984) introduced the Customer Service Life Cycle (CSLC) as a guide for differentiating a service throughout a buying cycle. The CLSC consists of four stages: Requirements, Acquisition, Ownership and Retirement. We have here applied the first two stages (the last two, alternative forms of payment, shipping and tracking, are well-known today) to summarize the requirements for an innovative booking system (Table 1 and Table 2, respectively). The Requirements stage establishes the functionality needed for a booking service and determines the service attributes. The Acquisition stage consists of the selecting, assisting, suggesting, ordering and acceptance of a travel package by the consumer.

Acquisition Phase		
Functional Requirements	Description	Implementation
Assist in understanding the package selection process	Give the users an overall picture	Extensive online search, updated statistics
Assist in performing the selection	Providing data on user needs and wishes	Report based on historical data
Assist in package specification	Input user needs, requirements and interests	Forms, wizards, ...
Customise travel package to individual	Specifying purchasing categories such as: facilities for children, suitable for elderly	Record-based structures provided by travel agencies or free text search
Accumulate packages of interests for possible purchase	Base on customer needs and customers who also bought the same package	Cross-sell techniques and clustering
Review product selection	Customer can specify reviews with free text or star ranking systems	Mining and summarising customer reviews
Notify customer of travel package availability	Customer can specify parameters such as date or price	e-mail based on historical data

**Table 2.** Acquisition phase

Functional Requirements provide the baseline functionality expected from the final system, thus addressing the question “what” the system does and reduce the tendency to address the “how”. For example, a functional requirement developed with implementation in mind could be: “The system will have the ability to perform free textual searches”, instead of “The system will have the ability to explore all possible alternatives based on keywords and constrains provided by the user in order to give an overall picture of the data”.

The tables above could be used as a first approach to specifications for a more flexible user interface for booking, an interface that accommodated the flexibility inherent in user needs and wishes.

## 6 Conclusion

In this article we have shown that current Web user interfaces for booking do not fulfill the needs of the user in the cases where the user is flexible with regard to locations, dates, accommodation and other factors, neither do they accept more open data. By removing the human travel agent from the loop an important source of information has been lost. Current interfaces allow the user to retrieve overview data by extensive searching, i.e., try many destinations, many dates, etc., but this may be both difficult and time-consuming.

We have presented the features needed to enhanced interfaces so that they can aid the customers also in the more open and flexible situations. By letting the user provide requirements and wishes up-front, it is quite easy to offer overview data. These data may then help the user to decide where to go, when to go or if to go. As we have shown, such an information system may be implemented in today’s booking system without requiring much additional resources.

Flexibility and some methods to handle the more open data are especially important in booking systems, as users may not know exactly what they want. However, the same functionality and the same methods that have been described here will be useful in any other situation where the customer is flexible. That could be anything from online shopping to queries to real estate systems.

## References

- Cumby, C.M., Fano, A.E., Ghani, R., Krema, M. (2004) Predicting customer shopping lists from point-of-sale purchase data. *KDD*, 402-409
- Ives, B., Learmonth, G. (1984) The Information System as competitive weapon, *Communications of ACM* 27, 12, 1193-1201.
- Lightner, N.L. (2004) Evaluating e-commerce functionality with a focus on customer service. *Communications of the ACM* 47(10), 88-92.
- Olsen, K.A., Sochats, K.M., and Williams, J.G. (1998). Full text information retrieval and Information overload, *The International Information and Library Review*, No. 30, 105-122.
- Riedewald, M., Agrawal, D., El Abbadi. A. (2002) Efficient integration and aggregation of historical information. *SIGMOD Conference*, 13-24
- Shim J., Scheuermann, P., Vingralek, R. (1999). Dynamic Caching of Query Results for Decision Support Systems, 254-263.
- Zhang, L., Sridharan, B Kinshuk (2003). On-Line Knowledge Management Search Engine. *ICALT*, 304-305.