

Last-Minute Travel Application

André Hübner, Mario Lenz, Roman Borch, and Michael Posthoff

■ In this article, we present a last-minute travel application as part of a complete VIRTUAL TRAVEL AGENCY. Each year, a significant amount of tour packages are sold as last minute tours in Germany. It is impossible for a travel agent to keep track of all the offered tour packages. Electronic-commerce applications might present the best possible tour package for a specific customer request. Traditional database-driven applications, as used by most of the tour operators, are not sufficient enough to implement a sales process with consultation on the World Wide Web. The last-minute travel application presented here uses case-based reasoning to bridge this gap and simulate the sales assistance of a human travel agent. A case retrieval net, as an internal data structure, proved to be efficient in handling the large amount of data. Important for the acceptance by customers is also the integration into the VIRTUAL TRAVEL AGENCY and the interconnections to other parts of this system, such as background information or the online car rental application.

The biggest share of the German travel market is tour packages. A usual tour package contains the flight to the destination and back, transfers from the airport to the hotel and back, board, and lodging. The market share for individual tours is much smaller than in the United States and other countries. It is common practice to offer tour packages that could not be sold until four weeks before departure as last-minute tours. Usually, a discount is given to the customer for these tour packages. Especially in recent years, the market for last-minute packages has grown by a vast amount, and there are specific properties related to these products:

Property 1: Travel agents struggle with the update problem. As many as 6000 new packages are offered daily by just one of the major German tour providers. Traditionally, tour providers send several dozen sheets of paper to their travel agents every day to inform them about available special offers.

Property 2: The method of informing agents by stacks of papers implies an availability problem. Often, the amount of places that can be booked on a specific offer is highly limited. Hence, when customers decide on one of the offers, there is a high risk that it is no longer available because there is no feedback about offers that have been brought to the market in recent days; that is, it is not clear whether these offers are still available or not.

Property 3: Last-minute offers (as provided by the tour providers) are tour packages. Thus, the customer can accept an offer only as it is; there are no variations to it (except if it is stored as a separate offer). Consequently, there is no negotiation during the sales process.

Property 4: Although there might be a huge number of offers, it is unlikely that the desires of a customer can be fulfilled all at once. Rather, it is often the case that alternative departure dates, neighboring airports, or even other destinations need to be suggested. In contrast to people having planned and prepared their holiday carefully, customers looking for last-minute vacations expect such variations. It seems that electronic commerce would be a good way to sell these travel packages. Some of the tour operators are offering database-driven systems on the web to sell their tour packages.¹ A shortcoming of these applications is that ordinary search requests (usually internally realized with some database query language) are not sufficient enough. Customers usually have a certain idea of what kind of tour they would like to book. They do have a preferred country, departure date, and length of tour. If the customer now specifies the request in detail, then most of the time, a traditional system tells the customer "No hits found!" meaning that no tour package in their database matches the request exactly (no solution situation). The customer then has to widen the request by leaving some of the input fields blank. The result is that the system probably will present the customer with several

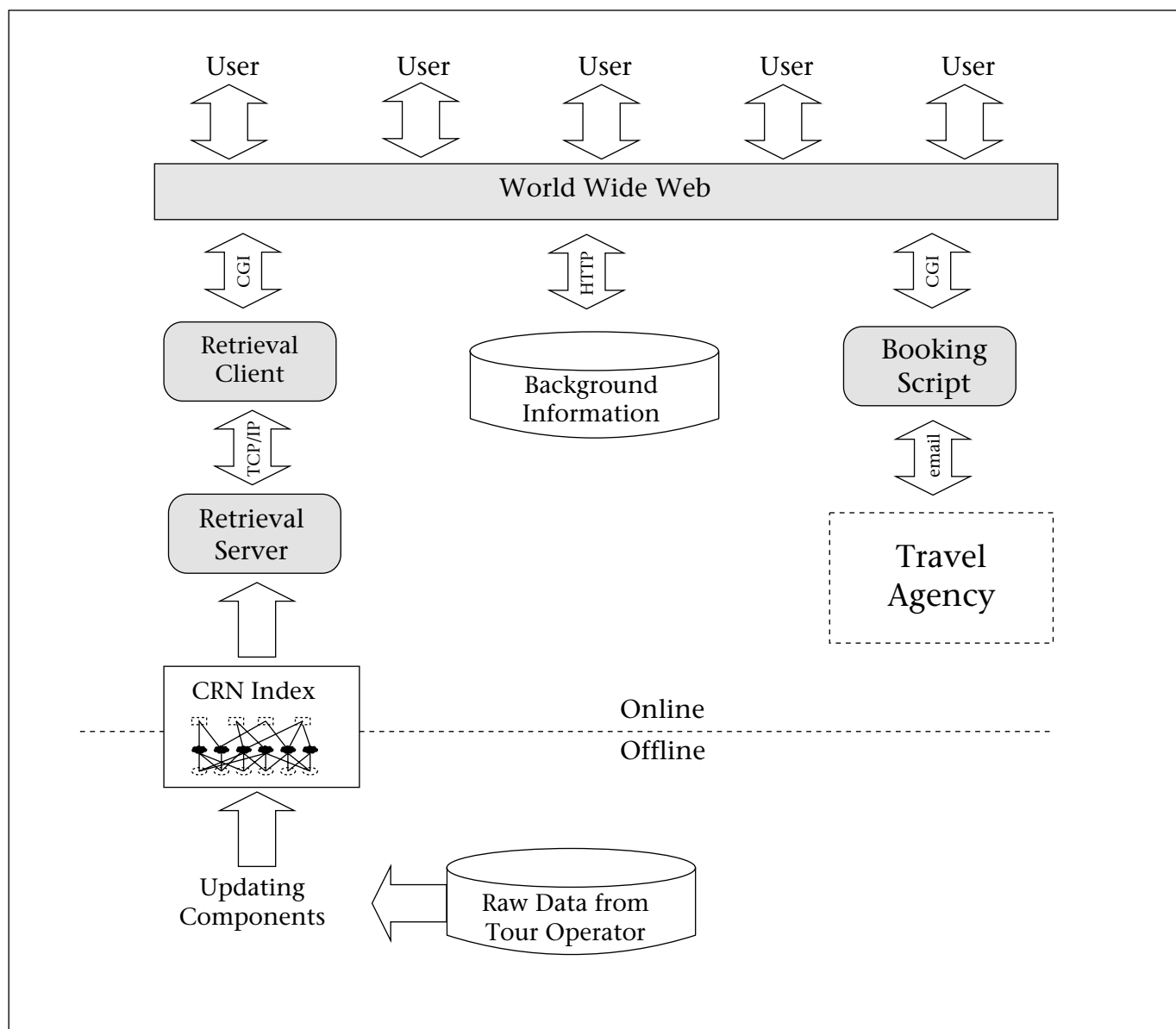


Figure 1. The Last-Minute Travel Application.

thousand possible tour packages (1000-solutions situation). These lists are usually not even ranked according to the request of the customer; so, the customer has to look through all the presented offers to find the best-matching tour package. The behavior of these conventional systems does not reflect a sales process at all. A good travel agent would never tell a customer that there is no tour package matching the request and send him/her away. Instead, the travel agent would propose some similar tour packages. Thus, an application was needed that implements a vague matching and ranks the found tour packages according to the customer's request.

Application Description

The last-minute travel application (figure 1) is only one part of a complete VIRTUAL TRAVEL AGENCY. As an electronic-commerce application, it is fully accessible by the web. The customer usually connects to the last-minute travel application by following a link from the web pages of the VIRTUAL TRAVEL AGENCY. The presented web page contains a form where information about the destination, the travel date, the kind of hotel room, and some other data can be entered. Note that all these input fields can also be left blank, which means that the customer does not care about these features. After send-

... an application was needed that implements a vague matching and ranks the found tour packages according to the customer's request.

ing the given information to the web server of the VIRTUAL TRAVEL AGENCY, a common gateway interface (CGI) script is started, the retrieval client. This retrieval client will then contact the retrieval server, which is running in the background. The retrieval server is running 24 hours a day and can handle requests from several CGI scripts at once. The number of simultaneous retrieval clients is only limited by the system hardware. The retrieval is performed using case-based reasoning (CBR) (Lenz et al. 1998) in the server, and the result, a sorted list of tour packages, is sent back to the retrieval client. The offered tour packages are ranked according to how well they fit the customer's request. A template page in HTML format is loaded by the retrieval client, and the features of the retrieved tour packages are inserted into the page before sending it back to the customer's web browser. If exact matches are found, they will be displayed first, followed by tour packages that are most similar. The customer can decide how many offers are presented to him/her. Background information, such as facts about the destination country or a single tour package, are provided by clicking on the appropriate link. If the customer decides to book a tour package, another form has to be filled out, and this information is then sent to the real travel agency by e-mail. The travel agency contacts the customer to verify the booking and information given. No real online payment is implemented yet because of security concerns. The main component of the client-server system is the retrieval server, which implements a case-based system.

The tour package data are provided daily by the tour operator as an ASCII file. This file is automatically downloaded by FTP. The update component is now used to transfer the raw data into a more appropriate structure. Information about the offered tour packages is gathered during the updating process. This information and the tour package data are then loaded by the retrieval server. This updating process is automatically started daily by a "CRONJOB" (a specific kind of batch job that is automatically started and scheduled by the operating system) but can also be triggered manually if needed. The whole system was developed in the UNIX environment and is now running on a LINUX PC. All components have been coded in C++. The entire VIRTUAL TRAVEL AGENCY consists of 20,000 lines of code, including all additional components. A Microsoft web server on a WINDOWS NT PC is used to connect the last-minute travel application to the web. The user interface is completely implemented in HTML, so no plug in or special browser is needed by the customer. The highest

possible compatibility was the prime reason for this. To remain flexible with respect to changes in the layout of the various pages and not overload the programs with too much information about the HTML layout, we implemented a strategy in which template HTML pages are provided by the travel agents. These are then used within the system, and the data that correspond to the current session are placed in these templates. Thus, the implementation of customized releases for other partners was reasonably easy.

Uses of AI Technology

In this section, we discuss case-based reasoning, integration, and insights.

Case-Based Reasoning

As already discussed, simple database approaches would not be sufficient for implementing this type of application because they cannot provide intelligent sales support. In particular, the following features are essential:

First, some customers enter very detailed descriptions of their intended tour packages, but others only have a rough idea. Consequently, the system has to be able to deal with vague, as well as highly specific, queries.

Second, because of property 4, the system has to be able to suggest appropriate alternatives if no offer completely fits the customer's requirements. The system has to present the available offers to the customer in a reasonable manner. In particular, it should definitely avoid situations in which no offer is made to the customer (no solution situation) as well as those where the customer is left alone with pages and pages of possible offers (1000-solutions situation).

CBR, in general, is a technology that satisfies at least the last two criteria:

First, alternatives are suggested by considering similar offers where the similarity measure takes into account information about departure dates, geographic locations of departure airports and destinations, climatic conditions, and so on.

Second, the no-solution situation is avoided by the previous point: consideration of alternatives. The 1000-solutions situation, however, is avoided by establishing preference orderings and presenting the best-suited offers first. Hence, the result of a retrieval process is not a set of all applicable offers (as in databases) but a ranking of the most suitable offers.

Fulfillment of the first of these criteria very much depends on the particular technique used for implementing the system. As should be clear from the description so far, a CBR system has to deal with three crucial problems in the VIRTUAL

TRAVEL AGENCY: (1) *flexible retrieval*: some customers enter very specific requests; others only express vague intentions. Hence, a CBR system needs a flexible retrieval method that can cope with both situations. (2) *efficient retrieval*: customers are not willing to wait long for the result of a search request. Even the delay caused by the web often causes severe problems. Thus, the retrieval method has to be highly efficient because of the large amount of data that have to be managed. (3) *easy updates*: Because regular updates of the data are required, building the internal structures of the CBR system has to be manageable in reasonable time and possibly while the system is still running.

The model of case retrieval nets (CRNs) (Burkhard and Lenz 1996) satisfies all three criteria. For the purpose of this article, it suffices to think of CRNs as a specific index of the case base that can be built offline and supports both efficient and flexible case retrieval by means of a spreading activation process in an associative memory.

A *case* is, of course, a representation of a single tour package as a vector of attribute-value pairs according to the following features: the departure airport; the destination of the desired holiday trip; the departure date; the duration of the trip; the type of accommodation; the type of catering; the number of participating persons; and, of course, the price.

A *case base* is the set of all cases in a single data set. A *query* is represented similarly to cases except that the feature price has not been included as a searchable parameter. This decision was based on the observation that all offers with otherwise similar parameters will, in general, be similar priced in last-minute data sets.

Integration

As already mentioned, the last-minute travel application is one part of a VIRTUAL TRAVEL AGENCY. A BACKGROUND INFORMATION MODULE is used to display additional information about a selected tour package, such as the destination, the offered hotel, or the climate (figure 2).

The type of information shown depends on the data set the offer originates from. For example, there are last-minute programs for which the name of the hotel is intentionally not available, but this information is provided for other programs.

Another very tight connected application is called INFORMER. The INFORMER can be used as an agent to specify a request for a last-minute tour package some time before tour packages for the desired departure time will be offered. The INFORMER will check each day if there is a tour package that meets the requirements of the

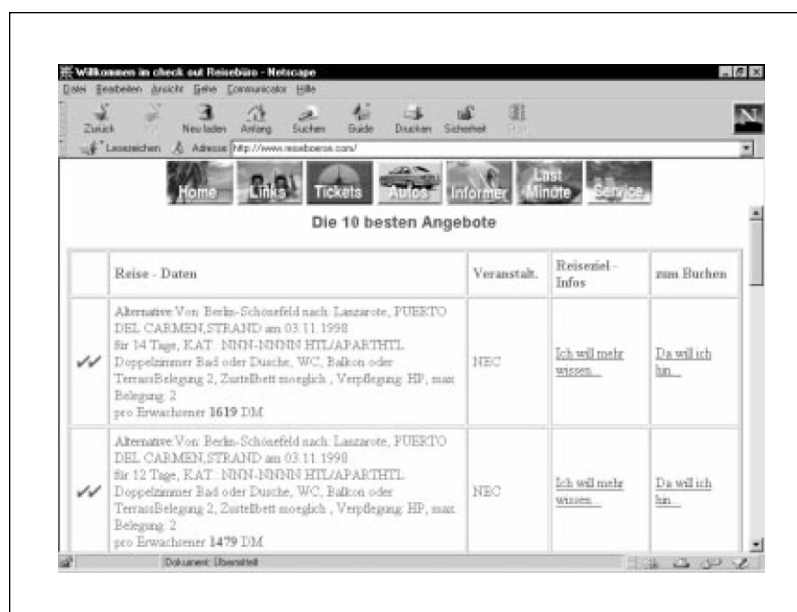


Figure 2. Snapshot of the Result Web Page.

customer, and if so, it will send an e-mail to this customer.

Insights

The last-minute travel application differs from typical research applications in some important points. The model of the domain (travel) is simple and could be built without much effort. Much more time was required to find appropriate mechanisms to handle this huge amount of data (as much as a quarter of a million). The response time of the system was crucial for the success of the last-minute travel application.

Consequently, strict design decisions have been made toward a lean implementation of the system that allowed us to realize a working (and profitable) tool in a reasonable time. The resulting system is limited in several respects. For example, it assumes a relational data model as known from traditional database technologies and is based in fairly straightforward similarity measures. However, the system is generic in that it can be used to create solutions for other e-commerce scenarios. The system itself is based on the CBR-SELLS product by TecInno.

Application Use and Payoff

The first version of the last-minute travel application went online in March 1997. It was considered a trial version using the university environment to test the system, but it was already fully integrated into the VIRTUAL TRAVEL AGENCY.

In July 1998, it was relaunched as the commercial server of the VIRTUAL TRAVEL AGENCY. Since then, the last-minute travel application has steadily been extended as much as the VIRTUAL TRAVEL AGENCY itself. This evolutionary development will be continued in the future.

During the first year, approximately 300,000 requests were measured, in peak seasons, several thousands a day. The average number of requests was constantly raised during the whole year. There was no special customer group to identify, and despite the fact that the VIRTUAL TRAVEL AGENCY is operated by a local Berlin travel agent, the requests came from all over Germany and even abroad.

During the first season, almost 1000 customers used the last-minute travel application to book a tour package, and an estimated turnover of 1 million German marks (US \$500,000) a year was achieved. Indirectly, it was even more because a lot of customers did not use the system for buying the tour packages but for looking for the best offer and then buying it at a real travel agency or by telephone. Nevertheless, to date, electronic commerce is not deployed as much in Germany as it is in the United States. Electronic commerce is still a new way of business, especially for the end consumer. We expect the growth of system use to be in accordance with the growth of the whole electronic-commerce market. The system does not generate enough business yet to make a living for the operating travel agency. The traditional way of selling tour packages is still the major income source. The VIRTUAL TRAVEL AGENCY can be considered an additional branch of the real travel agency, with its own business processes.

Application Development and Deployment

The VIRTUAL TRAVEL AGENCY was first started with a flight database in 1996. From this, experiences with the World Wide Web and CGI programming were collected and used in later applications. Travel-related information was added all the time, and additional applications, such as renting a car online, evolved. The last-minute travel application was developed by two people (one student, one university staff member) within three months. The estimated effort was two-person months.

The system was designed to be clearly subdivided into two parts: (1) the back end and (2) the front end. The back end was written by the previously mentioned developers in C++ and includes the CGI script, the retrieval server, and the update script. The front end, including the web pages (written in standard HTML) and the layout, was developed by the travel agent staff.

The overall development costs were low because of this strict separation and the university setting. The low costs were important because financing was a major problem because the travel agency did not have enough money to have such a system developed by a software house. The tour operators were not (and still aren't) interested in this system for different reasons: First, usually there is already a web presentation of these tour operators, which they consider sufficient enough. Second, the main business of the tour operator is generated by the travel agencies and only a small part directly by electronic commerce. The traditional travel agents are not interested in getting competitors from electronic commerce; so, the tour operator cannot support such a system without displeasing the travel agencies, their major business partners. These and related problems are discussed more broadly in Lenz (1999, 1998).

Maintenance

Two aspects have to be considered concerning the maintenance of the system: (1) data update (corresponds to the case base maintenance of CBR) and (2) model update (corresponds to similarity update of CBR). Data update is done automatically using the update component, usually once a day, when new data from the tour provider arrive. It can also be done manually if required. Model update is done by the travel agency staff themselves and is hardly ever needed. In case that a new destination appears, the system prints a log message, and the travel agency staff inserts the new destination into the similarity model. The same procedure would be done with new values for all the other features of a tour package. With the next updating process, these new values will be considered.

Note

1. See TUI (www.tui.com) or LTU (www.ltu.com) for example.

References

- Burkhard, H.-D., and Lenz, M. 1996. Case Retrieval Nets: Basic Ideas and Extensions. *Informatik-Berichte*, 103(110), Humboldt University.
- Lenz, M. 1999. Experiences from Deploying CBR Applications in Electronic Commerce. In *Proceedings of the Seventh German Workshop on Case-Based Reasoning*, 3–5 March, Würzburg, Germany.
- Lenz, M. 1998. Experiences from Deploying Electronic Commerce Applications. Technical report, Humboldt University.
- Lenz, M.; Burkhard, H.-D.; Bartsch-Spörl, B.; and Wess, S. 1998. *Case-Based Reasoning Technology from Foundations to Applications*. Lecture Notes in Artificial Intelligence 1400. New York: Springer Verlag.