

Last Minute Travel Application

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

TecInno GmbH
Sauerwiesen 2
D-67661 Kaiserslautern, Germany
{huebner,lenz}@tecinnno.com

check out Touristik GmbH
Friedrichstraße 112b
D-10117 Berlin, Germany
posthoff@transmedia.de

Abstract

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 is sold as last minute tours in Germany. It is impossible for a travel agent to keep track of all the offered tour packages. E-Commerce applications may help to 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 in the WWW. 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* (CRN), as internal data structure, proofed to be efficient to handle 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, like background information or the online car rental application.

Problem Description

The biggest share of the German travel market are 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 US and other countries.

It is common practice to offer tour packages that could not be sold until 4 weeks before departure day as *last minute tours*. Usually, a discount is given to the customer for those tour packages. Especially in recent years, the market for last minute packages grew by a vast amount and there are specific properties related to those products:

Property 1: Travel agents struggle with the *update* problem: Up to 6,000 of new packages are offered daily just by one of the major German tour providers. Traditionally, tour providers sent several dozen sheets of paper to their travel agents every day in order to inform them about available special offers.

Property 2: The above described way of informing agents via 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 for one of the offers, there is a high risk that this one is no longer available. This happens because there is no feedback about offers that have been brought to the market in recent days, i.e. it is not clear whether these are still available or not.

Property 3: Thirdly, *Last minute* offers (as provided by the tour providers) are tour packages: This means that the customer may accept an offer only as it is – there are no variations of it (except if it is stored as a separate offer). Consequently, there is no negotiation during the sales process.

Property 4 Finally, although there may 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 E-Commerce would be a good way to sell these travels. Some of the tour operators are offering database driven systems in the WWW to sell their tour packages. See TUI¹ or LTU² for example. 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 the tour. If the customer now specifies the request in detail, then most of the time a traditional system tells the customer "No hits found!", that means that no tour package in their database matches the request exactly (*no solution* situation). The customer himself 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

¹<http://www.tui.com/>

²<http://www.ltu.com/>

several thousand possible tour packages (*1,000 solutions* situation). These lists usually are not even ranked according to the request of the customer. So the customer himself has to look through all the presented offers to find the best matching tour package. This behavior of these conventional systems does not reflect a sales process at all. A good travel agent would never tell the customers that there is no tour package matching the request and send them away. Instead, the travel agent would propose some similar tour packages. So, an application was needed that implements a vague matching and ranks the found tour packages according to the customers request.

Application description

The last minute travel application is only one part of a complete virtual travel agency. As an E-Commerce application it is fully accessible via the WWW.

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 sending the given information to the web server of the virtual travel agency, a 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 systems 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 customers request. A template HTML page is loaded by the retrieval client and the features of the retrieved tour packages are inserted into that page before sending it back to the web browser of the customer. If exact matches are found, they will be displayed first, followed by tour packages that are most similar. The customers can decide how many offers are presented to them. Background information, like facts about the destination country, about 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 via email. The travel agency will contact the customer to verify the booking and information given. There is no real online payment implemented yet due to security concerns. The main component of the client server system is the retrieval server which implements a case based system.

The tour package data is provided daily by the tour operator as an ASCII file. This file is automatically downloaded via 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 started daily automatically via a cronjob, 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 Win NT PC is used to connect the last minute travel application to the WWW. The user interface is completely implemented in HTML, so no plug-in or special browser is needed by the customer. Highest possible compatibility was the prime reason for this.

In order to remain flexible with respect to changes in the layout of the various pages and also to 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 corresponds to the current session is placed in these templates. This also made the implementation of customized releases for other partners reasonably easy.

Uses of AI Technology

Case-Based Reasoning

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

- Some customers enter very detailed descriptions of their intended tour packages while others only have a rough idea. Consequently, the system has to be able to deal with vague as well as highly specific queries.
- Due to Property 4, the system has to be able to suggest appropriate alternatives if no offer completely fits the customers 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 (*1,000 solutions* situation).

Case-Based Reasoning, in general, is a technology that satisfies at least the last two criteria:

- 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 etc.
- The *no solution* situation is avoided by the previous point: consideration of alternatives. The *1,000 solutions* situation, on the other hand, is avoided by

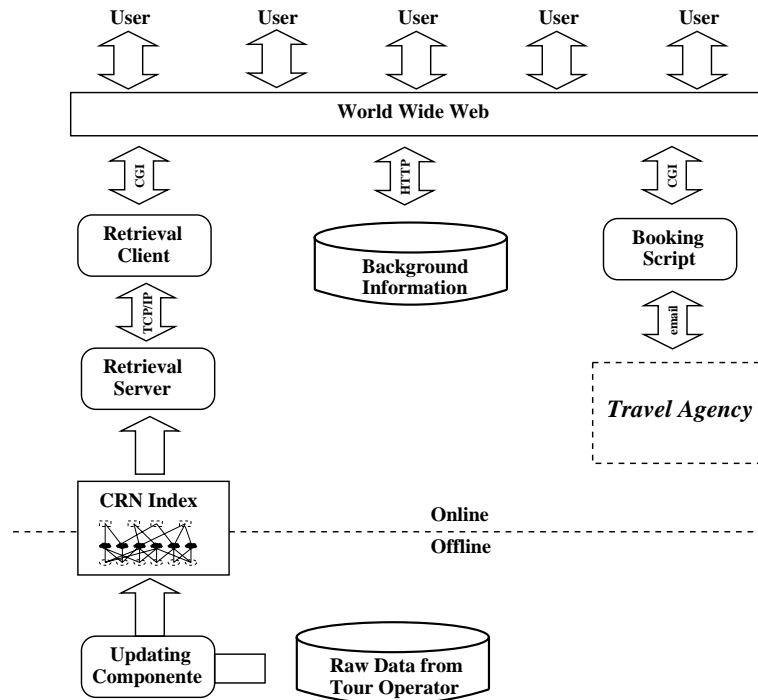


Figure 1: The last minute travel application

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 the above 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*:

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.

Efficient Retrieval: Customers are not willing to wait long for the result of a search request. Even the delay caused by the WWW often causes severe problems. Thus, the retrieval method has to be highly efficient due to the large amount of data that has to be managed.

Easy Updates: Since 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* (Burkhard & Lenz 1996) (CRNs) satisfies all three criteria. For the purpose of this paper, it suffices to think of CRNs as a specific index of the case base that can be built offline and that 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 very 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 information about the destination, the offered hotel, or the climate.



Figure 2: Snapshot of the result web page

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 while for other programs this information is provided.

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 customer, and if so it will send an email 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 quite simple and could be built without much effort. Much more time was required to find appropriate mechanisms to handle this huge amount of data (up to 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 towards a lean implementation of the system that allowed to realize a working (and profitable) tool in 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 straight-forward similarity mea-

sures. On the other hand, the system is generic in so far as 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 environment of the university to test the system, but it was already fully integrated into the *Virtual Travel Agency*. In July 1998, it was relaunched at the commercial server of the *Virtual Travel Agency*. Since then the last minute travel application was steadily extended as much as the virtual travel agency itself. This evolutionary development will be continued in the future.

During the first year approx. 300.000 requests were measured, in peak seasons several thousands a day. The average number of requests was constantly raising 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 1.000 customer used the last minute travel application to book a tour package and an estimated turnover of 1 million German marks a year was achieved. Indirectly it was even more, because a lot customer did not use the system for buying the tour packages but for looking for the best of-

fer and then buying it in the real travel agency, or via phone. Nevertheless so far E-Commerce is not as much deployed in Germany as it is in US. E-Commerce is still a new way of business especially for the end consumer. We expect the growth to be according growth to the whole E-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 as an *additional* branch of the real travel agency, with its own business processes.

Application Development and Deployment

The *Virtual Travel Agency* as the overall system was first started with a flight database in 1996. From this, experiences with the WWW and cgi programming were collected and used in later applications. Travel related information was added all the time and additional applications, like renting a car online, evolved. The last minute travel application was developed by 2 people (one student, one university staff) within 3 months. The estimated effort was 2 man months.

The system was designed to be clearly subdivided into 2 parts: **the backend**, and **the frontend**.

The backend was written by the above mentioned developers in C++ and includes the cgi script, the retrieval server and the update script.

The frontend including the web pages (written in standard HTML) and therewith the layout was developed by the travel agents staff.

The overall development costs were low due to this strict separation and the university setting. This was very important because financing was a major problem, as the travel agency did not have enough money for letting such a system develop by software houses. The tour operators were not (and still aren't) interested in this system for different reasons:

1. Usually there is already a web presentation of these tour operators which they consider as sufficient enough.
2. The main business of the tour operator is generated by the travel agencies and only a small part directly via E-Commerce. The traditional travel agents are not interested in getting competitors from E-Commerce. So the tour operator can not support such a system without displeasing the travel agencies, their major business partners.

We will not go into further details about these and related problems, they are discussed more broadly in (Lenz 1998; 1999).

Maintenance

Two aspects have to be considered concerning the maintenance of the system: data update (corresponds to the

case base maintenance at CBR) and model update (corresponds to similarity update at CBR).

Data update is done automatically using the update component, usually once a day, when new data from the tour provider arrives. It can also be done manually if required.

Model update is done by the travel agency staff themselves and hardly ever needed. In case that a new destinations appears the system prints a log message and the travel agency staff has to insert the new destination into the similarity model. The same procedure would be done with new values at all the other features of a tour package. With the next updating process these new values are considered.

References

- Burkhard, H.-D., and Lenz, M. 1996. Case Retrieval Nets: Basic ideas and extensions. Informatik-Berichte, 103-110. Berlin: 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. Springer Verlag.
- Lenz, M. 1998. Experiences from Deploying Electronic Commerce Applications. Techn. report, Humboldt University, Berlin.
- Lenz, M. 1999. Experiences from deploying cbr applications in electronic commerce. In *accepted for GWCBR-99*.