14 DieToRecs: A Case-based Travel Advisory System

FRANCESCO RICCI, DANIEL R. FESENMAIER, NADER MIRZADEH, HILDEGARD RUMETSHOFER, ERWIN SCHAUMLECHNER, ADRIANO VENTURINI, KARL W. WÖBER AND ANDREAS H. ZINS

1. Introduction

There is a growing number of websites that support a traveller in the selection of travel destinations or travel products (e.g. flight or hotel). Typically, the user is required to input product constraints or preferences, which are matched by the system in an electronic catalogue. Major e-commerce websites dedicated to tourism such as Expedia, Travelocity and TISCover have started to cope with travel planning by incorporating recommender systems, i.e. applications that provide advice to users about products (Schafer et al., 2001). Recommender systems for travel planning try to mimic the interactivity observed in traditional counselling sessions with travel agents (Delgado and Davidson, 2002). The current generation of travel recommender systems focuses on destination selection and does not support the user through a personalized interaction in bundling a tailor-made trip comprising one or more locations to visit, an accommodation and additional attractions (museum, theatre, etc.).

The DieToRecs10 system extends current recommender systems by incorporating a human choice model extracted from both the literature and the empirical analysis of the traveller’s behaviour. DieToRecs supports the selection of travel products (e.g. a hotel or a visit to a museum or a climbing school) and building a ‘travel bag’, i.e. a coherent (from the user point of view) bundling of products. DieToRecs also supports multiple decision styles by letting the user ‘enter’ the system through three main ‘doors’: iterative single-item selection, complete travel selection and inspiration-driven selection. The first

---

10 This work has been partially funded by the European Union’s Fifth RTD Framework Programme (under contract DIETORECS IST–2000–29474).
door enables the most experienced user to efficiently navigate the potentially overwhelming information provided by the two integrated databases (TISCover and APT Trentino). The user is allowed to select whatever products he or she likes and in the preferred order using the selections done up to a certain point (and in the past) to personalize the next stage. The second ‘door’ enables the user to select a personalized trip that bundles together items available in the catalogue. The personalized plan is constructed by ‘reusing’ the structure and main content of trips either built by other users or available from some provider. The third door allows an inspiration-seeking user to choose a complete trip by exploiting a simpler user interface (icon-based) as well as an interaction, which is kept at the minimum length as possible. It must be stressed that all these decision styles are supported in a uniform and seamless way by means of a graphical user interface. Hence, switching from one style to another is always possible and easy to do.

Sections 2 and 3 describe our conceptual approach to travel planning and illustrate how the notion of decision styles emerged from the research. Then, we describe the fundamental element of the designed application and its technological implementation. We end the chapter by summarizing the results of this project and discussing some still open issues to recommendation system development.

2. Travel Decision Styles

Ideally, the system must enable a (new) user entering the travel destination recommendation system to be classified quickly in order to provide him or her the optimal navigational path and mode of presentation (type and sequence of questions, graphical widgets selection, length of interaction, etc.). The user should be able to influence the dialogue management component by explicitly volunteering information that is useful to determine his or her decision style. This can be achieved by self-selection of the decision style, for instance, by presenting iconic descriptions of the styles. Although Grabler and Zins (Chapter 12, this volume) recommend that information on a user’s decision style should be acquired at a very early stage in the session, the system should provide the possibility of switching between different interface styles. Again, this can be achieved by self-selection of the user or derived from a pattern of user interaction with the computer. Information presented at a later stage of the user session should be structured differently according to the requirements of the respective decision styles. However, the following three stages are used to decompose the dialogue:

1. Filtering: The user must be able to enter the primary variables or constraints that describe his or her decision style; however, not all information categories are required at the beginning of the dialogue.
2. Specification: Additional information related to the responses provided in the first stage are presented to the user. An important aspect of this stage is that the user believes that only personally ‘important’ features are asked.
3. Selection or sorting: The user must be able to make his final decision
based on a small number of alternatives presented. This list may be sorted by one (or more) of the key factors associated with the identified decision style and/or specified during the decision process by the user.

As the number of possible offers should be counted and provided to the user at any time, it is possible that a specific user session does not require all three stages as the number of matched offers may be limited. Additional information, which appears to be important in describing the characteristics of the decision process and therefore needs to be monitored and stored during a session, includes: usage type, sequence of sub-decisions, flexibility or rigidity of trip characteristics, degree of pre-specification, number of alternatives, decision style and experiential proneness.

3. The Proposed Approach

The theoretical considerations provided above set the requirements for a travel-planning recommender system that must support, by means of an adaptive behaviour, rather different decision styles and must personalize the suggestions on the base of both personal and travel characteristics. In this section we shall describe the basic design choices of the DieToRecs recommender system and illustrate a typical man–machine interaction. The DieToRecs system is based on the following elements:

- **Bundling a mix-and-match travel.** DieToRecs basically supports the user in building a personalized travel plan that can either comprise a pre-packaged offer or can be obtained by iteratively selecting tourist products (travel items) such as locations to visit, accommodations and activities. Item selection dialogue is driven by the personal and travel characteristics that are structurally decomposed into what are referred to as ‘general travel wishes’ and ‘detailed travel wishes’. General travel wishes provide basic information about the nature of the travel the user is going to plan, like its duration, the travel party and the budget. Detailed travel wishes are preferences and constraints that the user expresses on features of the specific products (destinations, accommodations and activities) to be included in the travel ‘bag’.

- **Allow the user to enter through three functional doors.** Users can build (configure) their travel plan by means of three top system functions, which act as different doors to enter the system. These doors enable users to provide in whatever order and amount they like general and detailed travel wishes. The first door allows the users to select whatever products they like and in the preferred order (Single-item Recommendation). The second door enables the users to select a personalized trip that bundles together items available in the catalogues (Complete Travel). The third door enables inspiration-seeking users to choose a complete trip exploiting (selecting and modifying) examples of travels shown by means of a user interface, which is strongly based on images and which minimizes the interaction length (Seeking for Inspiration). At any
moment users can switch from one system function to the other preserving the choices already done, allowing the users to build their travel by combining the three approaches; for example, the user can bundle his personalized travel first by selecting a complete trip using the second or third door and then by completing it with additional products from the catalogues using the first door.

- **Decision styles and functional doors.** Decision styles are initially mapped (probabilistically) to these functional doors. This means that DieToRecs is bootstrapped with a default assignment of decision styles to doors (e.g. the ‘highly predefined user’ to the first door), but user activity logs, stored as cases, will provide data for training the DieToRecs classifier to: (i) identify the decision style (ii) suggesting the user switch to another door.

- **Wizard-like GUI approach.** The system drives the users through the logical steps needed to define the travel. At any step of the decisional process, the system displays the next alternative steps that can be followed to complete the task. The sequence of steps depends on the functional door chosen. We will now describe this approach by means of a sample user session.

- **Register the user interaction session as a case.** The adaptive behaviour of DieToRecs is based on a structured representation of the interaction session that is stored as a case in a case-based reasoning system (Ricci et al., 2002a). A case includes general travel and detailed travel wishes acquired during the interaction, items in the ‘travel bag’, feedback provided by the user on the items selected and an ordered list of the system functions called during the interaction (activity log).

- **Personalize the questions posed to the user using cases and catalogue analysis.** DieToRecs exploits Intelligent Query Management techniques to help the user identify his query. After having acquired some travel wishes from the user, DieToRecs poses in-context questions, trying either to further specify the travel wishes, whose effect is to tighten the search, or to relax conditions that cannot be satisfied. The identification of those travel wishes that could be asked or should be relaxed relies on: (i) the analysis of the users behaviours stored in the cases (statistics over user explicit preferences); (ii) constraint relaxation techniques; and (iii) information theory indicators, such as entropy, computed on the catalogues of products (Ricci et al., 2002b).

- **Personalize the recommendations using collaboration filtering through case similarity.** The items (and the complete trips) suggested by DieToRecs are ranked according to a collaboration-via-content-based approach (Pazzani, 1999). In fact, items filtered according to the user travel item preferences are then sorted such that those contained in (or more similar to) similar recommendation sessions (cases) are scored best. In this respect DieToRecs is a hybrid recommender system that overcomes classical problems of pure collaborative-based approaches such as huge amounts of registered user logs data needed to deliver recommendations.
Details about the case structure, the recommendation methodology and the
Intelligent Query Management technique can be found elsewhere in this
book. The rest of this section describes how tourists can bundle their trip
by using DieToRecs. When starting to use the system, the users must first
decide how to search through the travel and product catalogues managed by
the system. If the users want to build their trip by selecting each travel item
(destinations, accommodations, activities) matching specific detailed travel
wishes, they should choose the ‘Single-item Recommendation’ function. If
the users want to select a complete trip already including a set of products by
specifying general travel wishes, they should choose the ‘Complete Travel’
function. If they want to find their travel by browsing complete trips sug-
gested by the system, they should choose the ‘Seeking for Inspiration’ func-
tion. Sections 3.1 and 3.2 will describe sample sessions for the ‘Single-item
Recommendation’ and the ‘Seeking for Inspiration’ functions. The interac-
tion supported by the Complete Travel is not explicitly described since it is
quite similar to the ‘Single-item Recommendation’.

3.1 Single-item Recommendation

When tourists have highly detailed wishes about the trip (destinations,
accommodations, activities), they normally use the Single-item Recommen-
dation function (first door). The wizard-like interface drives them through
the sequence of steps normally needed to self-bundle a complete travel. The
typical sequence of steps is:

- **General Travel Preference Specification.** In this first step (Fig. 14.1), the
  system asks the user the most important features characterizing his or
  her travel (destination, travel party, budget, duration). The (optionally)
  provided data will be exploited by the system to provide personalized
  recommendations. On the right frame (Fig. 14.1), the system suggests the
  possible next alternative steps: further specifying travel preferences (ad-
  vanced travel preferences), searching for a destination, an accommoda-
  tion or activities. Let us assume that the user first decides to search for the
  main destination.

- **Destination Preferences.** The user is prompted for specific preferences
  about the sought destination. Two kinds of searches are provided: the
geographical search, in case the user wants to select a village in a certain
  tourist area or region; or the search by activity, if the user wants to find
destinations that allow practising his or her preferred activities (Fig. 14.2).
  After having specified the interests, the user can ask for the recommended
destinations by following the ‘Search for Destination’ link. Two cases may
  occur. If the preferences specified by the user allow to select a reasonable
  number of results (neither zero nor too many), the destinations matching
  the criteria are ranked and shown to the user (Destination Recommen-
dation step). Otherwise, the system initiates an interaction with the user to
  better clarify his or her needs (Relaxation and Tightening step).
Fig. 14.1. Single-item Recommendation: General Travel Preference.

Fig. 14.2. Single-item Recommendation: Search by Activity.
• **Relaxation and Tightening.** This step is initiated by the system when it is not able to select a reasonable number of products. In this case, the system suggests to the user how he or she can change the preferences to improve the result set. Two situations may take place. If too many destinations are selected, the system suggests additional preferences that could be set to reduce the result set (tightening). If no destination matches the preferences specified by the user, the system suggests which preferences should be changed (relaxed) to get results. After the user revises the preferences, the system either recommends some destinations or, if it is not able to select a reasonable number of products, provides further relaxation or tightening suggestions.

• **Destination Recommendation.** When the specified destination preferences enable the system to select a reasonable number of destinations, they are ranked and shown to the user (Fig. 14.3). Specifically, the system ranks the selected destinations exploiting all the information acquired from the user during the interaction and the past travels built by other users in similar situations. The user can obtain an explanation about why the recommended destination fits his preferences (‘Explain Why’ link), browse the next recommended destinations and when he or she finds the one suited to his or her needs, adds it to the travel bag.

![Fig. 14.3. Single-item Recommendation: Destination Recommendation.](image-url)
With a similar interaction, the user can then select and add to his or her personal travel bag additional destinations to visit, the accommodation to stay and the activities to practise.

3.2 Seeking for Inspiration

If the tourist is more recommendation-driven, the selection process will be supported from the system in the form of pictorial representations of former trips (third door). In this case, the user does not have to specify general travel or travel item preferences to obtain recommendations, but rather immediately receives six complete trips, represented by images about the destination and the accommodation and the two most important and characterizing features (Fig. 14.4). The user can then access detailed information about each trip. After examining the recommendations, or, simply, inspired by the shown images, the user can get six new different recommendations simply by expressing interest about one of them; the system, exploiting the fact that the user is interested in that particular alternative, proposes six other alternatives more focused on the user wishes. In this way, the user can browse the catalogue space without explicitly specifying constraints and wishes. When the user finds the proposal that suits the needs, he or she can add the suggested products to the personal travel bag.

Fig. 14.4. Seeking for Inspiration.

The system exploits a case base of travel bags that is built by the community of DieToRecs users as well as catalogues provided by TISCover AG and APT Trentino (DMO). The case structure is hierarchical (Ricci et al., 2002b) and implemented as an XML view over a relational database. DieToRecs integrates case-based reasoning with interactive query management. When asked to retrieve a travel product, DieToRecs tries to cope strictly with user needs and, if this is not possible, it suggests query changes that will produce acceptable results. Similarity-based retrieval is exploited: (i) when a complete travel is searched (second and third door); and (ii) when the single products (as the result set of the user’s query) must be ranked (first door).

The system is structured into two main components: (i) the GUI component, which manages the interaction with the user, gets the inputs, handles customization and shows the results; and (ii) the recommendation component, which is responsible for the work behind the interface – the database access, user case management and the recommendation process itself. Sections 4.1 and 4.2 describe these two components in more detail.

4.1 DieToRecs GUI structure

The DieToRecs system was developed as component-based architecture and consists of several components that interact with each other. The units encapsulate functionalities essential for managing the DieToRecs system and are described below. Figure 14.5 shows the cooperation among these engines.

Fig. 14.5. Cooperation among engines in the DieToRecs system.
The Dialog Engine serves as a central management component to support the interaction between user and system, to read the input, to validate it, to manage the workflow and to return the response back to the user. This demonstrates that the Dialog Engine is responsible not only for the representation but also for the control of the user navigation. The engine consists of four main components to handle the user input validation and session handling, the workflow steering, the recommendation process and the integration of data from the recommendation component into the visualization fragments.

In close cooperation with the Dialog Engine two other engines work in the background to administrate information on session level and, if activated, on permanent level. The Session Engine is responsible for authenticating and authorizing users according to their identification and permission rights as well as guaranteeing that a user’s session will not be lost during interaction. It overcomes stateless hypertext transfer protocol (HTTP) connections by keeping sessions alive. The Tracking Engine manages tracking of interesting data such as user input or user behaviour. Tracked data is logged in the local database and provides methods to retrieve once tracked and logged data to support the recommendation process.

The Screen Flow Engine informs state-dependent engines about workflow sequences. It is responsible for delivering information about what is the next step that should be delivered to the user. The request comes from the Dialog Engine. To be able to respond the Screen Flow Engine needs to know the current status from this request. Consequently, depending on this current status and certain other conditions, the next step will be explored and delivered back to the Dialog Engine.

The GUI Adaptation Engine handles all relevant actions for supporting the Dialog Engine with essential layout information for the next required step that has to be visualized and delivered to the user. It assembles a web page through the adaptation of single fragments described in HTML, XML or JavaServer Pages (JSP). Following the engine model it only performs work in case the Dialog Engine requests some information; so, it is not active by itself. The key part of the GUI Adaptation Engine asks its subworkers for information, combines this information, prepares the necessary fragments and sends them to the requester, all based on the state information received with the request.

The GUI Customization Engine provides interaction services independent from content that have to be customized for an individual user. More generally, it supports an individualized interaction between a specific user and the system. An example of such services is a personalized welcome message for registered users after logging into the system. The GUI Customization Engine supports the Dialog Engine in the same manner as the GUI Adaptation Engine. It reacts in case the Dialog Engine requests some information. To minimize system traffic the engine has to know the next step of the GUI sequence from the Dialog Engine to retrieve only those services that are relevant for the specific user and also relevant for the concerned step.
4.2 Recommendation component structure

The main goal of the recommendation component is to enable the application to provide personalized recommendations about products stored in the product catalogues. This component has been designed to be easily integrated in existing tourist websites to enable recommendation functions. Figure 14.6 shows the overall structure of the recommendation component.

It is structured in the following main components:

- **The CaseManager.** Is the component devoted to the management of the current user’s case? It collects all the session data acquired from the user during the interaction, like the expressed travel wishes and the products added to the user’s travel bag, and makes them available to the Recommendation Engine component.

- **The Recommendation Engine.** Exploiting the current case and the case base implements the recommendation algorithms to find the products to be recommended to the user. In particular, it provides the Single-item Recommendation, the Complete Travel and the Seeking for Inspiration functions. It uses the XML Mediator and the Metric components.

- **XML Mediator.** It allows storing and retrieving data (XML documents) from existing data repositories. XML Mediator returns the XML documents matching queries that are expressed by means of a query language we have defined. In this integration architecture, the existing repositories contain data modelled according to their local data models. One role of

![Fig. 14.6. Recommendation component.](image-url)
the XML Mediator is to expose a unified integrated data model, which is mapped into the underlying local data models (Global as View approach) (Mananoescu et al., 2001). The XML Mediator has to translate the query expressed on the integrated data model and produce the XML documents having the structure defined by the integrated data model by collecting the pieces of information from the local data models of the integrated repositories. Furthermore, XML Mediator implements the Intelligent Query Management technology, identifying the query refinement suggestions to be proposed to the user.

- **The Metrics.** This component provides the tools for measuring similarity among XML documents and for sorting these documents, which can be products or recommendation sessions (cases). The recommendation algorithms implemented in the Recommendation Engine exploit similarities among products and recommendation sessions to identify the products to be recommended. The reader is referred to Chapter 8 for more details on this.

XML is widely used by all the components. All the data passed to, and provided by, the component to the application are XML documents, making certain that the functions are available to different web architectures. In addition, XML enables the document structures to be dynamic and allows the framework to work with different data models and case structures.

5. Conclusions

The DieToRecs system represents a new generation of travel recommender systems that can cope with individual differences in travel wishes and decision styles. We have empirically validated the system by A/B comparisons with more traditional approaches. The validation was conducted in Austria and has shown that the recommendation functions included in this system help users in travel planning by increasing ease of use and overall satisfaction. Perhaps, most importantly, users tend to accept the products that are recommended. A second evaluation has been done in Europe (Italy and Austria) and in the USA and has produced a large number of cases that will be used as training cases for the adaptive behaviour of DieToRecs. However, both research and software engineering issues are still open. From the research side, the most important aspects are how to dynamically adapt the recommendation algorithms to the current user, how to dynamically bundle travel proposals based on other users’ past cases and how to manage cases that contain incomplete information (i.e. unspecified travel preferences). From the software engineering point of view, integrating recommendation technologies in commercial tourism sites poses specific challenges: the algorithms should be optimized to be able to handle the workload in terms of concurrent users and number of recommendations per seconds that such systems require; the number of cases produced by the system becomes quickly very large and, thus, algorithms for case base management should be identified; tourism products change often
and, thus, recommendation algorithms should cope with a case base, which may contain products changed or not existing any more, by adapting the recommendations to the new context.

Chapter Summary

This chapter presented DieToRecs, a novel case-based travel advisory system. DieToRecs has been designed by incorporating a human decision model that stresses individual differences in decision styles. DieToRecs supports multiple decision styles and provides personalized recommendations exploiting a case base of recommendation sessions, which are stored by the system. Users can access the system through three main functional doors that fit to complementary groups of decision styles. Whichever the door used to enter the system, users can eventually switch the type of support required. The application relies on a component-based architecture, featuring a set of computational engines. One of these engines encapsulates the core functionality of the system – the methodologies used to rank products and manage queries. Other engines are dedicated to the personalization of the GUI and to track user activity. The system has been empirically validated and represents a value-added service for future destination management systems.