Case base querying for travel planning recommendation

Francesco Ricci and Hannes Werthner
eCommerce and Tourism Research Laboratory
ITC-irst
via Sommarive 18
38050 Povo, Italy
Tel: (+39)-0461-314521 - Fax :(+39)-0461-302040
email: ricci@itc.it

Abstract

This paper describes the general architecture and function of an intelligent recommendation system aimed at supporting a leisure traveller in the task of selecting a tourist destination, bundling a set of products and composing a plan for the travel. The system enables the user to identify his own destination and to personalize the travel by aggregating elementary items (additional locations to visit, services and activities). Case-Based Reasoning techniques enable the user to browse a repository of past travels and make possible the ranking of the elementary items included in a recommendation when these are selected from a catalogue. The system integrates data and information originating from external, already existent, tourist portals exploiting an XML-based mediator architecture, data mapping techniques, similarity-based retrieval and online analytical processing.

Keywords: Recommendation System, Case-Based Reasoning, Trip Planning, Mediator Architecture, XML.

1 Introduction

Planning a travel towards a tourism destination is a complex problem solving activity. The term "destination" itself, i.e. the final goal of the travel plan, refers to a concept that is unclear and lacks a commonly agreed definition. First of all, the spatial extension of the destination is known to be a function of the traveller distance from the destination. For instance, Italy is a destination for a Japanese, but not for a European traveler who may focus on a specific region, such as Tuscany, rather than on a historical city or something else. Moreover, a destination is not a mere geographical entity; it may be a wish or a collection of activities or experiences. In that sense a destination cannot be conceptually separated from a travel plan to the destination, i.e., a destination is reshaped in the tourist perspective in many ways according to the activities that he will perform in (towards) that destination. Modeling this vague concept and the decision process that leads different users to their preferred destination is still an open and challenging research problem.

The majority of current web-based systems tackle this problem in a very simple way. They usually present, possibly with rich multimedia content, an archive of "pre-defined" tourist destinations or a catalogue of products (tourism services). Moreover, these systems start from rather simple assumptions on the decision process that will drive the traveler during the selection activities. In fact, these systems leave the man/machine interaction completely open to the user "navigation" choices on the information space they provide. In other words, current systems are not able to drive or carry out a focused dialogue aimed at the achievement of the final information goal, i.e., the product identification and selection (destination and services selection). For instance, they can't help the user to express some possibly vague interests and finally delivering a personalized recommendation that lists a target location and a set of additional services like lodging, car rental or cultural events.

This functional gap is being filled by a new promising category of intelligent applications called recommendation systems. Recommendation systems provide advice to users about products they might be interested in. Burke distinguishes three types of recommendation systems: collaborative- or social-filtering; content-based and knowledge-based (Burke, 2000).

Amazon.com is a very popular example of an eCommerce site that exploits a collaborative-filtering approach. At Amazon, data about a customer purchasing history are stored and book recommendations are compiled picking up books in the purchasing history of other customers with similar purchasing patterns.

News Dude (Billsus and Pazzani, 1999), which is an example of content-based recommendation systems, observes what online news stories the user has read and not read, and learns to present the user with articles he may be interested to read. Content-based systems are usually implemented as classifier systems based on machine learning research (Witten and Frank, 2000).

The third type of recommendation systems uses knowledge about users and the products to build up a recommendation. Knowledge may be expressed in the form of a detailed user model, a model of the selection and suggestion process, and a rich description of the items to be suggested. These systems often integrate both collaborative- and content-based filtering techniques (Pazzani, 1999) and may be "conversational" (Aha and Breslow, 1997,Göker and Thomson, 2000). Conversational systems, in contrast to a classical recommendation systems, which reply to a user query with a ranked set of results, mimic a real dialogue between the ``inquirer" and the ``advisor" to solve a user need.

Furthermore, knowledge-based recommendation systems may be based on Case-Based Reasoning. Case-Based Reasoning (CBR) is a problem solving methodology that faces a new problem or situation by first retrieving a past, already solved similar case, and then reusing that case for solving the current problem. For instance, if a CBR system provides suggestions for an accommodation, it may first ask to the user to specify some personal data and preferences (room price, type of accommodation, position in the town, etc.). Then the system may retrieve from a case base of lodging solutions a subset of cases that best match the input description and asks to the user some questions in order to finally select one case in the list and to build an offer from that case. In general, a case in the memory may represent an item previously suggested or the union of such an item and the current user data (his profile data and the feedback given to the item suggested).

In a CBR recommendation system the effectiveness of the recommendation is based on: the ability to match user preferences with item description; the tools used to explain the rationale for the match and to enforce the validity of the suggestion; the function provided for navigating the information space.

CBR is largely based on the notion and implementation of case similarity, and similarity based retrieval from the case base. Case bases have been traditionally implemented using ad-hoc data structures loaded in memory at system start up. However, integration of CBR with relational databases is becoming more and more important for many reasons, e.g., scalability; enterprise application integration; reuse of previous data. An initial set of works have therefore addressed this issue (Shimazu and Shibata, 1993,Burke, 2000,Göker and Thomson, 2000,Schumaker and Bergman, 2000). The main technical problem consists in the compilation of a similarity-based request into SQL, which does not support directly this type of query. This problem has been studied in the data base community (Chaudhuri and Gravano, 1999,Seidl and Kriegel, 1998) but it is still practically open in high dimensional spaces.

This paper presents a knowledge-based CBR recommendation system that addresses the following objectives:

- **Destination shaping.** The user is supported in the process of selecting a personalized destination, i.e., a physical location and a set of activities and services to be consumed in that location or in nearby locations.
- **Cases drive personalization.** A set of cases, that store examples of previously suggested travel plans, constitute the basic knowledge source that the system leverages for driving the personalization and the recommendation of new travel plans.
- **Mediator architecture.** Cases are represented as semi-structured documents and are collected in an XML data store. The system accesses data using a set of views exported by a mediator. The mediator uses wrappers over existing data sources to retrieve all the needed information.
- **Support for query refining.** Query management is highly interactive, that is, a query result may be refined in a mixed initiative approach. For instance, when the user wants to relax or tighten a constraint, the system suggests the most reasonable relaxation or the minimal changes in query constraints that will yield a result with a manageable size.
- **Scoring with similarity metrics.** The relative ranking of alternative items is accomplished using case-based reasoning technologies and in particular with similarity based retrieval. The basic idea is that an item (activity, service, location) that is more similar to something that is contained in a user's past travel plan must be scored higher.

In this paper, first the user needs and the basic layout of the proposed approach are presented. Section 2 describes the working hypothesis for the destination model. Section 3 discusses the system architecture and briefly illustrates the data integration techniques used in the system. Section 4 then illustrates with additional details one of the system modules: the CBR query management component. Finally, Section 6 comments the current state of work.

2 Destination Modelling and Selection

A tourist destination may be modelled from different actors' perspectives (Werthner and Klein, 1999). DMOs (Destination Management Organizations), for instance, promote and reshape a geographical destination managing a network of actors (service providers) that contribute to the information content that is presented in a tourist web site.

From the opposite side, potential travellers (users of a web based information system) try to match this offered representation with their needs: geographical location, activities to perform in the location, services or events to consume, budget and timing constraints. The exact definition of a user-perceived destination is not clear and it is claimed that the elicitation of this rather vague concept calls for the adoption of an information tool based on the acquisition of real recommendation cases.

The tool must be initialised with a possibly simple model of the destination, but must be open to the user action, i.e., the user should be enabled to reshape a travel to his destination by aggregating simple elementary components. In a second step, having at disposal a historical repository of recorded human/computer interactions, it will be possible to build a more sophisticated model of the destination as it is perceived by the user, and build this new model as an evolution of the first one.

Figure 1 shows the basic model as a UML class diagram (Booch et al., 1999). In that diagram a rectangular box identifies a concept (class) and a link between two concepts denotes an association. An association terminating with an open diamond is an aggregation, i.e., a ``whole/part'' relationship. The central building block is the TravelAsset class (TA in short). This is described by (associated to) a LocationArea, and optionally an Activity performed, and a ServiceEvent which is consumed during a certain Timing. The numbers at the end of an association denote multiplicity, i.e., how many objects may be connected across an instance of an association. For instance, the ``0..1'' label at the end of the association between TravelAsset and Timing means that a TravelAsset may have 0 or 1 Timing, i.e., Timing is optional. A LocationArea may have relationships of spatial inclusion/containment into other LocationAreas. A ServiceEvent can be further specialized in a number of different sub-types, for instance: a lodging service, a climbing school, a recital, an exhibition. Analogously, Activity is the root of a hierarchy of traveler activities as: climbing, visiting, wandering or attending.

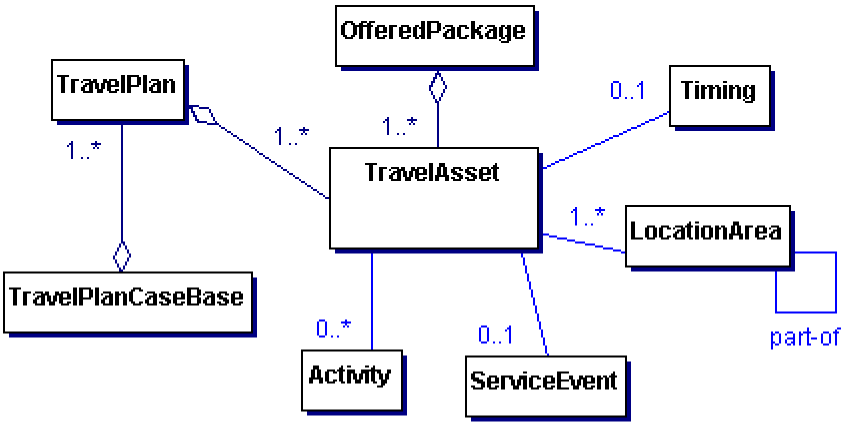

Figure 1: Destination main model.

The second major concept is TravelPlan, i.e., a user defined aggregation of TravelAssets building blocks. As an example of TravelPlan: ``skiing in Trento using the M.Bondone ski-pass the first week of January AND lodging in Vanezze in the Breeze Palace Hotel''. In this example, ``skiing in Trento using the M.Bondone ski-pass'' is the first TravelAsset and ``lodging in Vanezze in the Breeze Palace Hotel'' is the second one.

In the same way, but following the supplier view, the OfferedPackage is another aggregation of the same building blocks. The elicitation of the potential mismatch between the two destination types is one of the goals of our project. To achieve this, we enable the user to browse offered packages, and support him to build his own package.

Finally, TravelPlanCaseBase is a collection of user TravelPlans that are stored by the system. Many instances of this TravelPlanCaseBase class are possible. For instance, a ``Personal TravelPlanCaseBase'' can contain all the travel plans built by a specific user, or a ``Biker TravelPlanCaseBase'' can collect those that were built by the members of the bikers community. Partitioning TravelPlan objects in such case bases enables the implementation of collaboration filtering techniques in the CBR framework.

We note that the idea of representing a tourism service as an aggregation or elementary building blocks is shared by an IFITT initiative, the Reference Model Special Interest Group (RMSIG) (Höpken, 2000). Our model can be mapped to that of RMSIG and will reuse the basic components of that schema.

This paper concentrates on one part of the whole recommendation scenario, i.e., how the user is supported in the selection of the TravelAsset blocks from a catalogue, and how this process leads to the building and the storage of a collection of personalised TravelPlan cases. A TravelPlan bundles TAs, it is the unique entry point for all the information linked to the travel and a user history of TravelPlans provide the "fuel" for scoring the personalized recommendations (see Section 4).

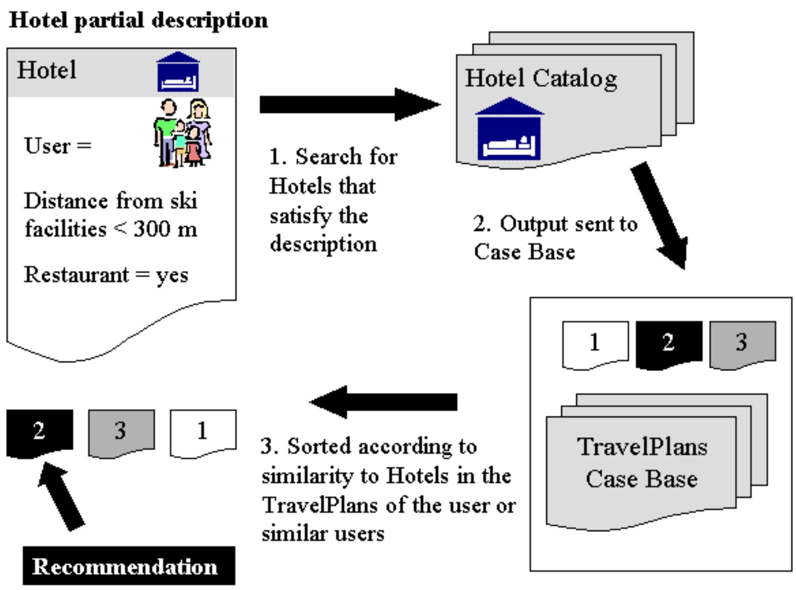

Figure 2: The recommendation process.

Figure 2 gives a snapshot of the recommendation process that is supported by the Case Base. Let us assume that the user wants to select a hotel in a given location. He starts by entering some partial description of the hotel (``close to the ski facilities and with a restaurant") and he also specifies some description of himself and his family (the group). In a second step, the system searches the catalogue and retrieves those hotels (in this example three are found) that satisfy those conditions. This search can be a traditional one or be based on similarity, i.e., the match can be only partial. In the third step, the case base of previously recorded TravelPlans is used to rank these three hotels. The hotel most similar to those that the same user or similar users have used in some past travel plan is ranked first.

Moreover, there is not a unique way to get recommendations exploiting the case base of TravelPlans. For instance, the user can perform a query directly on the case base of TravelPlans, looking for a travel made by other users and partially satisfying some constraints or preferences. Or the system itself can suggest some additional TravelAssets either selected by other similar users or because it fits well in the travel plan that the user is building. Due to lack of space not all the potential functions can be described, but it must be stressed that a case base of travel plans is a rich source of information and enables multiple strategies for tourism product recommendation.

3 System Architecture

In this Section, first the general architecture of the system is described and then the query component is further illustrated. Figure 3 shows the main components:

- **Graphical User Interface (GUI).** This module, together with the Dialogue Management and the Tourist Decision Management, is the principal responsible for user input analysis and content presentation. Focusing on the innovative aspects, the GUI relies on a model of the decision process extracted from an initial set of cases, that are based on the analysis of real human/human interactions. The model determines the dialogue flow, i.e. the set of pages delivered to the user's browser, the logical actions that the user can perform on a page, and the rules that, given a page and an action, outputs the next page. The initial model will be successively refined using the logs of the interactions.
- **CBR and Query Management.** This component processes queries posed by the user to the data tier. It is used both to interact with the case base of TravelPlans and to select and filter TAs, whose descriptions originate from external tourist portals. This component is described in the next Section.
- **Presentation Personalization.** Information content is contained in the data sources and must be presented to the user in accordance with the user profile information. Profile data are collected during interaction, e.g. at the user registration or when constraints are specified in a query. The derived model (XML-based) enables the online personalization of the presentation using Natural Language automatic generation techniques (Not and Zancanaro, 2000).
- **Travel Plan Management.** These functions enable the user to store, retrieve, update and share his history of travels. Moreover, a travel plan can be exploited, during the travel, to further suggest new activities or comment on a TA that the user has in his plan and will probably consume. The user has the possibility to control this process, exposing or shading information contained in his travel plan.

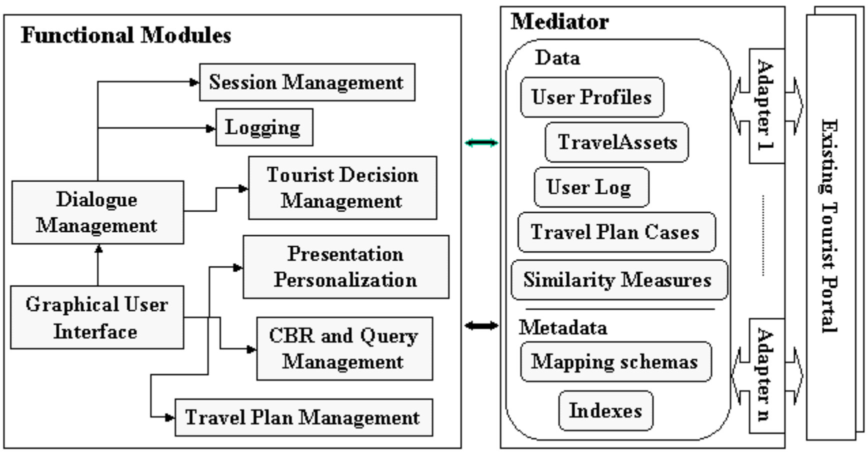

Figure 3: System architecture.

As already stated, the system is designed to integrate data from external sources. In the Mediator module the TA objects are mapped to the external repositories using adapters. Moreover, all the TravelPlans are stored locally along with User Profiles and the similarity measures to score the searched objects (TAs or TravelPlans) in accordance with user's preferences.

The data tier exploits a simple mediator architecture (Wiederhold, 1992,Florescu et al., 1998) that is described in more detail in another paper (Ricci et al., 2001). The mediator views are defined as XML Schema documents (Fallside, 2001) and are implemented either as SQL views over a relational database or as collection of XML documents, as it is depicted in Figure 4.

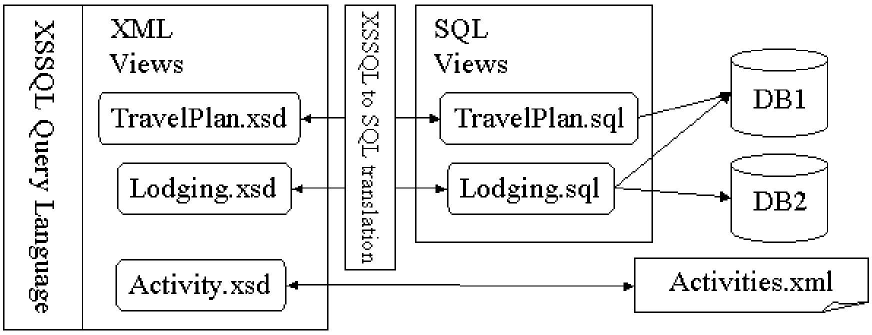

Figure 4: Integration of relational and XML data.

In that way, the XML Schemas (the .xsd files in Figure 4) are the user's exported views over the data contained in the information server and in the external sources. They represent the bridge between the user's requests and the target data, shadowing the details of data distribution and data management (e.g. XML versus relational tables or local versus external).

In order to integrate information contained in relational databases, our approach exploits the SQL views mapping functionality plus a canonical mapping between the SQL views layer and the XML Schemas layer as is already available in many DB to XML mapping products (e.g. (Oracle, 2000)). This approach is simpler than those that rely on XML query languages (Baru et al., 1999) (Fernandez et al., 2000) to build the view definitions but it suits properly our requirements (see the discussion in Section 5.4). A custom query language (XSSQL in Figure 4) has been developed to query the XML views. Queries in XSSQL over those views that are implemented as SQL views are translated runtime on the corresponding SQL queries to retrieve data from the physical storage system.

In this context the case base of TravelPlans is modelled as a custom XML view over the data and is implemented as an SQL view over a relational database. TravelPlans are modeled as XML documents but are stored in relational tables (see the discussion in Section5.3 for comparison with other approaches). Other types of information can be stored as XML documents (for instance the Activity descriptions in Figure 4). In this case no translation of XSSQL to SQL is needed and our custom query language is implemented directly on the XML representation.

4 CBR Query Management

The ``CBR query management" component collects input data from the user, builds a query on the mediated schema and then compiles a query plan that may involve both local and remote data. The results collected are then ranked using a similarity metric that measures the "distance" between a selected item and the “ideal” solution.

The queries processed by the system are Range Queries, i.e., the query constraints specify ranges of allowed values for features of the TAs. For instance: *Select all TA where Activity="lodging" and Service.type = "hotel" and hotel.cost < 60 and Location="Rome"*. Or, *Select all TA where Activity="canoeing" and Location.type="high-mountain or deep-canyon"*. To process such queries, the system first retrieves all the TAs that satisfy the given constraints and then measures the distance between a these TA and an “ideal” one (a TA either contained in a past TravelPlan or “synthesized” from the User Model).

4.1 Interactive Query Management

The query process is not linear, the user interacts with the system by looking at an initial result set and by refining the query conditions according to the result obtained. The proposed approach is illustrated in Figure 5.

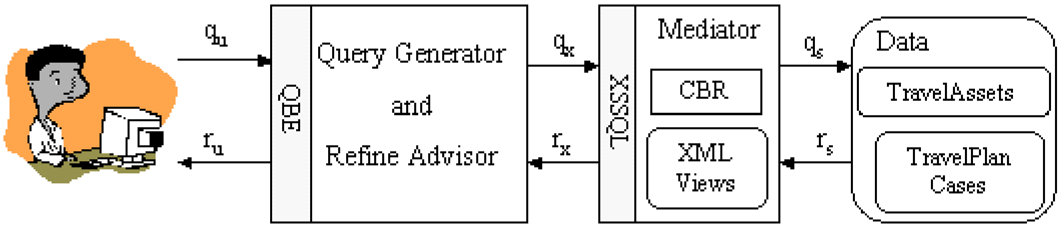

Figure 5: Query processing.

The typical interaction will proceed as follows:

1. The user enters a query q_u in a simplified query by example language (QBE). The user does not have to know the query language used for the real data access. q_u is a range query, a set of features are constrained to be in given ranges of values.
2. The ``Query Generator and Refine Advisor" translates the user query $q_u$ to the language (XSSQL) and over one of the views exported by the Mediator, and it sends the translated query $q_x$ to the ``Mediator" for the execution.
3. The ``Mediator", according to data mapping information, translates q_x into a new query that is written in the query language of the target data source. For instance, TravelAssets are stored in a relational database, hence SQL is used. The engine executes the query on the specified data and returns the result set r_s to the ``Mediator", that passes it back to the ``Query Generator and Result Analyser".
4. The ``Query Generator and Result Analyser" examines the result. If it is ``reasonable" (for instance the result size is less than a given threshold) then results are ranked, asking the Mediator for the "Similarity Based Scoring", and sent it back to the user (r_u). Otherwise, if the result set has not a manageable size or is too small, then the Query Generator and Refine Advisor module suggests changes to the user query. The module proposes to relax or tighten a query constraint using some meta information (indexing and data summary) provided by the Mediator. So a new query q'_u is built and proposed to the user. If the user agrees then the query process is re-initiated.
5. The new query q'_x is executed and a new result r'_x is produced. The interaction is now again at the stage described above and could loop until a stop condition is satisfied.

Note that the similarity-based scoring, performed by the CBR module, is not executed directly on the whole target data set, but it is performed on the result set of an intermediary range query. This is a common approach used to deal with the computational complexity inherent to nearest neighbour queries to databases and XML data sets as well (Chaudhuri and Gravano, 1999,Seidl and Kriegel, 1998). We also believe that from the user point of view (especially for TravelAsset selection) it is preferable to clearly separate what is "logically" satisfied in the result set from what is only partially matched in the result. We therefore assume that the user will constrain a set of "required" features, which are more import for him, and the system will score the results using these ``required" features and some “optional" features. For instance, if he is looking for a hotel, the user may select those having a certain maximum price and certain required facilities (e.g. no-smoking room and AC). The system will score the selected candidates by measuring the distance from an “ideal" hotel (e.g. the cheapest) and therefore taking into consideration all the features, possibly with different grades (see also Section 5.2 for a comparison with other approaches).

The known limitation of this approach (as is illustrated in the typical interaction quoted above) is that a range selection may end up with no result at all. This could happen in our system but if this is the case then the system is able to identify the cause of such a failure and to suggest a repair action (relaxation or tighten). For instance, if the user issues the following query: *Select all TA where Activity="lodging" and Service.type = "hotel" and hotel.cost < 60 and Location="Rome"*. Should the result set be void, the system "looks" in a neighbor of the data selected by the query and determines that the failure is due to both the cost and the location constraints. It then suggests two repair options: increase the price bound to 100; or expand the location to an area surrounding Rome.

This function is enabled by: (a) segmentation of the information space in buckets; (b) indexing of these buckets. The system uses an indexing technique called bitmap indexes (O'Neil and Quass, 1997,Chaudhuri and Dayal, 1997) that have been introduced in OLAP systems (online analytical processing). Bitmap indexes allow to compute the size of a result set without actually executing the query and they can speedup multi-dimensional access to the data as needed by the example query shown above.

4.2 Ranking Using the Case Base

As a final issue we discuss how a set of results may be ranked using the case base, i.e., the library of TravelPlans. Let us assume that a TA can be described as an element of $X = \prod_{i=1}^{n} X_i$, i.e. a n-dimensional vector. For the sake of simplicity we are limiting the discussion to linear data structures. What follows can be generalized to semi-structured data types as XML documents. Each X_i is the space of possible values for the i-th feature. For instance, $X_i = R$ for real features, like the price of a room or the latitude of a location. There might be also Boolean features ($X_i = \{true, false\}$) or other discrete symbolic features like the type of the "cuisine" (e.g., $X_i = \{italian, french, arab, indian\}$).

A simple Range Scored Query may be expressed as a list of features and for each feature a set of allowed values (range): $x_1 \in R_1 \wedge \ldots \wedge x_k \in R_k$, where $k \leq n$ and $R_i \subseteq X_i$. If $x = (x_1, \ldots, x_n)$ is a TA that satisfies the query range constraints then the score for x is computed as follow:

$$d_{\min}(x) = \min_{y \in PTA} \sum_{i=1}^{n} w_i d_i(x_i, y_i),$$

$$Score(x) = 1/(1 + d_{\min}(x)).$$

where *PTA* is the set of all TAs that appear in some past Travel Plan of the user, $0 \leq w_i \leq 1$ is a weight used to balance features relative relevance, and d_i is the metric used for distance computation on each feature dimension. The above formulas imply that, the closer is x to some TA in a past travel, the better it scores. Therefore if $\bar{x}$ is the TA with maximum score, i.e., with minimal distance from the set PTA, the system will suggest $\bar{x}$ to the user. Moreover, if $\bar{y}$ is the TA in *PTA* such that $d_{\min}(\bar{x}) = \sum_{i=1}^{n} w_i d_i(\bar{x}_i, \bar{y}_i)$, i.e., that with minimal distance from $\bar{x}$, then $\bar{y}$ can be shown to the user to explain the rationale of the suggestion, i.e., $\bar{x}$ is suggested because it is very similar to $\bar{y}$, an item that the user used in the past.

Regarding the distance metrics d_i, that determines the similarity between cases, there are two possibilities: for numeric features d_i is the (normalised) absolute value of the difference of feature values, for symbolic features it is basically the Hamming distance (Wettschereck et al., 1997). More in general the system offers a collection of heterogeneous metrics that are adapted and tuned as more cases are acquired (Wilson and Martinez, 1997,Blanzieri and Ricci, 1999).

It must be noted that the PTA set is empty before an history of interaction is collected and a case base is built. Therefore, at the beginning PTA will be identical for all users and will contain a set of ideal prototypes that represent generally optimal cases (for instance the hotel with best "quality/cost" ratio).

The features' relative importance in the scoring are determined by the weights w_i. Initially, i.e. when *PTA* is empty, we will use $w_i = 1$ if the i-th feature was specified in the query, and 0 otherwise. This incorporates a simple assumption, that is, the features specified as constraint matter to the user. After a case base is collected the weights can be more finely tuned to reflect user preferences using a methodology described elsewhere (Wilson and Martinez, 1997,Ricci and Avesani, 1999). This introduces a learning step that, using the case base which reflects a history of user usage of the system, will build a model of the user preferences.

5 Related Work

5.1 Trip Planning and Destination Selection

Trip planning is already supported by some commercial sites, for instance expedia and biztravel. Expedia (www.expedia.com) enables the user to define a new trip by selecting a flight or a hotel or a car (rental). Then the user can select one additional services, hence adding a map of the zone or a travel guide, revise the services status (reserved or not) and print a summary of the travel. In that site multimedia content (audio, images, video) represent an important part of the user interface that is used for the final plan aggregation. Similar functions are supported by biztravel.

Our system is different from these examples in many aspects. First of all these systems do not score alternative options with respect to user preferences, or when user preferences are used these must be explicitly inserted by the user. In other words these systems are not able to reason on the user's past behaviour and to infer data or preferences from previous choices. Moreover, these systems do not take advantage of previous trip plans of the user (or of similar users) to customize or personalise the current recommendation.

5.2 Case Based Recommendation and Tourism

The application of CBR to Travel and Tourism is in a very early stage. Lenz has been the first to apply CBR technologies to this domain in the CABATA system(Lenz, 1996,Lenz, 1999). In CABATA, CBR is mainly exploited as a tool to issue similarity-based queries to a catalogue. In this setting the user enters the partial specification of an item and the system retrieves the most similar from the catalogue. So, for instance, the user enters a partial description like ``cost=100 and location=Rome", and the system retrieves all the hotels that satisfy those conditions (if any) plus those that do not match all these requirements but are similar, e.g., an hotel that costs 110 and whose location is Rome.

Our framework extends this model in many directions. Firstly, the scoring and ranking mechanism is based not only on the user input conditions but also on the travel plan cases stored in the personal case base of the user (or similar users). Therefore, referring to the example quoted above, we can rank and differentiate even the hotels that satisfy the query conditions (``cost=100 and location=Rome") by measuring the distance from similar hotels used in the past. In CABATA all the hotels that logically satisfy the query conditions have zero distance from the probe and therefore are not differentiated in the recommendation. This points out another major difference, i.e., the capability of our system to pipe a similarity query after a traditional range query, i.e., to compute similarity matches only on those cases that satisfy some strict logical conditions. This is, in our view, closer to the user needs, i.e., to enable the user to express both strict and ``soft" constraints.

The second major aspect that differentiates our system from CABATA, refers to the case structure itself. Our case base is made of TravelPlans, i.e. a complex structure made of more elementary components (TravelAsset). This case base is used both when the elementary components retrieved from a catalogue must be scored, and when similarity based retrieval is performed on the TravelPlan case base itself. CABATA only performs this last function, i.e., similarity based retrieval on the case base.

5.3 Case bases and XML

The intersection of CBR and XML is becoming an interesting area of research. Shimazu (Shimazu, 1998) has been among the first to advocate the exploitation of XML in CBR application. XML was introduced first as a case representation language, i.e, for structuring the knowledge in a specific real case, and secondly, because of the special support of XML for textual data. In the CARET/XML system each case is modeled with a case profile that contains: a) a link to an XML file with the full case description; b) a set of fields (and field values) extracted from the corresponding XML file. Case profiles are stored in a RDBMS. This is an interesting and early implementation of an XML to RDBMS (partial) mapping. The similarity computation is performed only on that part of the case that is stored in the RDBMS.

Sengupta et al. (Sengupta et al., 1999) propose a classification of CBR implementation into three types: Web-Based; Enterprise and Task-Based. The distinctions between these three models are mainly due to the case representation language and physical storage system: Web-Based systems use XML, Enterprise systems uses RDBMS and Task-Based systems use ad hoc representation languages and various storage systems. Our proposal merges the web-based and the enterprise models using a Mediator architecture.

Our work has been also influenced by (Hayes and Cunningham, 1999). Hayes and Cunningham argued that a case base document should be considered no differently from a document containing industry standard data. Only those information, required by CBR function, that cannot be derived from the original schema must be added. The reader is referred to (Ricci et al., 2001) for a more detailed description of our technical approach and the differences introduced with respect to (Hayes and Cunningham, 1999).

5.4 Information Integration in Tourism

Nowadays, information integration and mediator systems are one of the most prominent research issues for web applications. There is a number of research projects (see (Levy, 1998,Florescu et al., 1998,Levy and Weld, 2000) for an introduction and a list of projects) and software products (e.g. Tamino, Nimble, Excelon, etc.) that compete in this arena. It is not possible to report on all of them, but we would like to describe some applications of these technologies to the tourism domain.

Barish et al. (Barish et al., 2000) have developed an application (TheaterLoc) that allows users to retrieve information about theaters and restaurants for a variety of cities in the United States. TheatherLoc is based on the Ariadne system that make possible to query web sources (html) as if they were databases. At the heart of the Ariadne approach there is a domain model that provides a unifying ontology for describing the sources' content. TheaterLoc integrates heterogeneous sources consuming their data at the highest representation level, i.e., that of the user interface (html). The advantage, compared to our approach that integrates data at the lowest representation level (relational tables), is that applications can be developed with no impact on the data providers. However, it requires that a constant update of the wrappers to the external sources. In fact the html pages of the integrated systems can easily be changed by the information providers. Moreover, it must be noted that this approach cannot support a complex query process as that proposed here, i.e., interactive query refinement based on indexing and similarity based query.

(Wös and Dunzendorfer, 2001) advocates the necessity to set up adequate data interchange facilities between heterogeneous systems to cope with the tourists' requests for an extensive data collection. They propose an architecture (TIS-QL) that requires that both clients and servers interact with each other through adapters. The client adapter behaves for the client as a proxy towards the server, whereas the server adapter is a translator from (to) a general datamodel to (from) the server specific datamodel. They also show, in case the adapters are not implemented, how to integrate applications using XML and a metadata description of the mappings between different standards. This approach is powerful and very general, but would represent an overkill in our application scenario. We do not need to implement a mechanism that will support any peer to peer data interchange, we only need to map information from a set of providers to a single client, the recommendation component. With respect to (Wös and Dunzendorfer, 2001) our approach is obviously more limited but more suited to our requirements and goals.

6 Conclusion

This paper describes a work in progress at a newly established research centre on eCommerce and Tourism (http://ectrl.itc.it). The proposed approach is being implemented for the ``Azienda di Promozione Turistica" of Trentino (Trentino DMO), and will be shortly validated within a use group. A more comprehensive version will be developed as a main result of a European IST Project (DIETORECS, in collaboration with: TIScover AG - Travel Information Systems (A), Institute for Tourism and Leisure Studies - Vienna University of Economics and Business Administration (A), National Laboratory for Tourism and eCommerce - University of Illinois at Urbana-Champaign (USA) and Azienda per la Promozione Turistica del Trentino (I)).

The main result of our approach is a comprehensive middleware for issuing personalized recommendation to a leisure traveller in identifying and aggregating elementary services or activities to be consumed. It is based on the principle that suggestion effectiveness relies on a combination of factors like: appropriate destination modelling; data retrieval and filtering with both sharp and approximate matching; scoring using personal preferences that can be derived from a base of previous cases.

Moreover, the recommendation system, which is based on that middleware, should help to better understand the user needs and behaviour, and will clarify the possible mismatch between offered destination packages and user's wishes. Besides, the case base of TravelPlans, which is an output of the user interaction with the system, will enable additional advanced functions like: TravelPlan composition with TravelAssets contained in other similar TravelPlans or tourist support during the travel (m-commerce).

Acknowledgements

This project is partially funded by CARITRO foundation. We would like to thank B. Arslan, E. Not and D. Fesenmaier for their feedback and comments.

References

Aha, D. and Breslow, L. (1997). Refining conversational case libraries. In *Case-Based Reasoning Research and Development, Proceedings of the 2nd International Conference on Case-Based Reasoning (ICCBR-97)*, pages 267-278. Springer.

Barish, G., Knoblock, C. A., Chen, Y.-S., Minton, S., Philpot, A., and Shahabi, C. (2000). The theaterloc virtual application. In *Proceedings of the Twelfth Annual Conference on Innovative Applications of Artificial Intelligence (IAAI-2000) Austin, Texas*, pages 980-987.

Baru, C., Gupta, A., Lundäscher, B., Marciano, R., Papakonstantinou, Y., and Velikhov, P. (1999). Xml-based information mediation with MIX. In Exhibitions Program of ACM SIGMOD 99.

Billsus, D. and Pazzani, M. (1999). A hybrid user model for news story classification. In *Proceedings of the Seventh International Conference on User Modeling, UM '99*, Banff, Canada.

Blanzieri, E. and Ricci, F. (1999). Probability based metrics for nearest neighbor classification and case-based reasoning. In Althoff, K.-D., Bergmann, R., and Branting, L. K., editors, *Proceedings of the 3rd International Conference on Case-Based Reasoning Research and Development (ICCBR-99)*, volume 1650 of *LNAI*, pages 14-28, Berlin. Springer.

Booch, G., Rumbaugh, J., and Jacobson, I. (1999). *The Unified Modeling Language User Guide*. Object Technology Series. Addison Wesley Longman, Reading, Mass.

Burke, R. (2000). *Encyclopedia of Library and Information Science*, chapter Knowledge-based Recommender Systems. To appear.

Chaudhuri, S. and Dayal, U. (1997). An overview of data warehousing and olap technology. *SIGMOD Record*, 26(1):65-74.

Chaudhuri, S. and Gravano, L. (1999). Evaluating top-k selection queries. In *Proceedings of the 25th VLDB Conference*, pages 397-410, Edinburgh, Scotland.

Fallside, D. C. (2001). Xml schema part 0: Primer. http://www.w3.org/TR/xmlschema-0/. W3C Proposed Recommendation.

Fernandez, M., Tan, W., and Suciu, D. (2000). Silkroute: Trading between relations and xml. *Computer Networks*, 33(1-6):723-745.

Florescu, D., Levy, A., and Mendelzon, A. (1998). Database techniques for the world-wide web:a survey. *SIGMOD Record*, 27(3):59-74.

Göker, M. H. and Thomson, C. A. (2000). Personalized conversational case-based recommendation. In *Advances in case-based reasoning: 5th European workshop, EWCBR-2000, Trento, Italy, September 6-9, 2000: proceedings*, pages 99-111. Springer.

Hayes, C. and Cunningham, P. (1999). Shaping a cbr view with xml. In *Case Based Reasoning Research and Development, Proceedings of the International Conference on Case Based Reasoning (ICCBR)*, pages 468-481, Monastery Seeon, Germany. Springer.

Höpken, W. (2000). Reference model of an electronic tourims market. In *Information and Communication Technologies in Tourism 2000 - Proceedings of the International Conference (ENTER2000)*, pages 265-274. Springer.

Lenz, M. (1996). Imtas - intelligent multimedia travel agent system. In *Information and Communication Technologies in Tourism (Proceedings of ENTER-96)*, pages 11-17. Springer.

Lenz, M. (1999). Experiences from deploying cbr applications in electronic commerce. In *Proceedings of the German Workshop on CBR*.

Levy, A. Y. (1998). The information manifold approach to data integration. *IEEE Intelligent Systems*, 13:12-16.

Levy, A. Y. and Weld, D. S. (2000). Intelligent internet systems. *Artificial Intelligence*, 118:1-14.

Not, E. and Zancanaro, M. (2000). The macronode approach: mediating between adaptive and dynamic hypermedia. In *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, AH'2000*, pages 167-178, Trento, Italy.

O'Neil, P. and Quass, D. (1997). Encoded bitmap indexing for data warehouses. In *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 38-49, Tucson, Arizona, USA.

Oracle (2000). Oracle8i application developer's guide - xml.

Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13:393-408.

Ricci, F. and Avesani, P. (1999). Data compression and local metrics for nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):380-384.

Ricci, F., Mirzadeh, N., and Werthner, A. V. H. (2001). Case-based reasoning and legacy data reuse for web-based recommendation architectures. In *Proceedings of the Third International Conference on Information Integration and Web-based Applications & Services*, pages 229-241, Linz, Austria.

Schumaker, J. and Bergman, R. (2000). An efficient approach to similarity-based retrieval on top of a relational database. In *Advances in case-based reasoning: 5th European workshop, EWCBR-2000, Trento, Italy, September 6-9, 2000: proceedings*, pages 273-284. Springer.

Seidl, T. and Kriegel, H. (1998). Optimal multi-step k-nearest neighbor search. In *Proceedings of the ACM SIGMOD Conference*, pages 154-165, Seattle, WA.

Sengupta, A., Wilson, D. C., and Leake, D. B. (1999). On constructing the right sort of cbr implementation. In *CBR workshop at International Joint Conference on Artificial Intelligence IJCAI-99*.

Shimazu, H. (1998). A textual case-based reasoning system using xml on the worl-wide web. In Smyth, B. and Cunningham, P., editors, *Advances in Case-Based Reasoning*, volume 1488 of *Lecture Notes in Computer Science*, pages 274-285. Springer.

Shimazu, H. K. H. and Shibata, A. (1993). Retreiving cases from relational data-base: Another stride towards corporate-wide case-base systems. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence IJCAI-93*, pages 909-915, Chambéry, France, August 28 -September 3. Morgan Kaufmann.

Werthner, H. and Klein, S. (1999). *Information Technology and Tourism - A Challenging Relationship*. Springer.

Wettschereck, D., Mohri, T., and Aha, D. W. (1997). A review and empirical comparison of feature weighting methods for a class of lazy learning algorithms. *AI Review Journal*, 11:273-314.

Wiederhold, G. (1992). Mediators in the architecture of future information systems. *IEEE Computer*, 25(3):38-49.

Wilson, D. R. and Martinez, T. R. (1997). Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 11:1-34.

Witten, I. H. and Frank, E. (2000). *Data Mining*. Morgan Kaufmann Publisher.

Wös, W. and Dunzendorfer, A. (2001). Configurable EDI for heterogeneous tourism information systems based on XML. In *Information and Communication Technologies in Tourism 2001, ENTER2001*, pages 93-101. Springer.