

# Automated Generation of Itineraries in Recommender Systems for Tourism

Pierpaolo Di Bitonto, Francesco Di Tria, Maria Laterza, Teresa Roselli,  
Veronica Rossano, and Filippo Tangorra

Dipartimento di Informatica, Università degli studi di Bari "Aldo Moro",  
Via Orabona 4, 70126 Bari, Italy  
{dibitonto, francescoditria, marialaterza, roselli,  
rossano, tangorra}@di.uniba.it

**Abstract.** Current recommender systems can support tourists in choosing travel products (accommodation, activities, means of transport, etc.), in planning long trips, and in profitably spending time in a specific geographical area such as a region (or a city). In the last case, the system should be able to construct itineraries suited to the tourist's interests. In this paper, a method for generating tourist itineraries in knowledge-based recommender systems is proposed. The method is based on a theoretical model that defines space-time relations among items of intangible cultural heritage (called events) and on transitive closure computation (of the relations), that is able to construct chains of events. The proposed method has been implemented in the T-Path recommender system, that suggests itineraries of cultural events occurring in the Apulia region.

**Keywords:** recommender systems, transitive closure, logical programming.

## 1 Introduction

Recommender systems are very useful tools for the tourism industry because they can support users in organizing long, medium and short trips, suggesting locations, accommodation, transport, and so on. For example, a tourist staying in the Apulia region could be interested in spending time visiting the main cities, landscapes or cultural events (festivals, processions, special markets, etc.). But planning itineraries is a complex problem and very difficult to face in an unknown geographical area. Thus, the tourist usually relies on maps, or guide books that suggest the main locations or attractions worth visiting. The improvements of technology are now offering the tourist more and more effective tools that are able not only to propose locations and/or attractions to visit, but also allow the tourist to select the most suitable itineraries for her/his interests. The research efforts in this domain have yielded numerous recommender systems aiming to support the tourist in making the best choices.

Most of the recommender systems presented in literature are able to suggest single items (flights, buildings, cities, etc.) on the basis of the tourist's requests. At this point, the tourist should create her/his own itinerary, choosing the items that fit her/his requirements. The newest recommender systems build itineraries automatically using

the Constraints Satisfaction Problem (CSP) approach [1]. This means that they generate a sequence of attractions or spots to be visited, filtering data according to the user's constraints (day of visit, cost, and so on) specified in the request.

Unlike the above approach, this paper proposes a method that builds tourist itineraries using transitive closure computation [2]. In particular, these itineraries are focused on intangible cultural heritage (called events) occurring in a geographical area. In other words, the main idea of our method is to detect a set of events (processions, festivals, special markets, etc.) with space and time relationships, in the sense that they are near both in space and in time.

Transitive closure computation aims at answering the following recursive query: "given an event  $e$ , what are all the possible paths starting from  $e$ ?". This computation is not done via Relational Calculus but via Predicate Calculus, as Aho and Ullman state in [3]: Relational Calculus does not suffice to express this kind of recursive query. Therefore, the itinerary planning process is generated by a logical program that uses a knowledge base. The proposed method has been implemented in the T-Path prototype, a knowledge-based recommender system that suggests tourist itineraries in the Apulia region.

The main contributions of our research work can be summarised as follows. Firstly, a novel method for generating itineraries using transitive closure computation is proposed. Secondly, the method is scalable. It can be used to generate itineraries at different grain sizes: the events can occur in regions, districts or cities (space relation), or in weeks, days, or hours (time relation). Because of this, the method could be implemented in any knowledge-based recommender system that uses an ontological representation of events and a logical program that computes transitive closure. Finally, the application domain is also a novelty in tourism recommender systems. In fact, these systems usually suggest items such as accommodation, flights, museums, etc. In our work the items suggested are itineraries composed of an ordered list of correlated events.

The paper is organised as follows: section 2 presents some related works about tourism recommender systems; section 3 presents the method for constructing tourist itineraries; section 4 gives an overview of the T-Path system architecture; section 5 shows a specific case study. Finally, some conclusions and future research directions are outlined.

## 2 Related Works

Current recommender systems for the tourism domain can be classified in two main groups: systems that suggest single items (or travel products) and systems that suggest a set of items. Examples of the first group are Triplehop's TripMatcher ([www.ski-europe.com](http://www.ski-europe.com)) and VacationCoach ([www.travelocity.com](http://www.travelocity.com)) [4]. They select travel destinations by filtering the items collected in a virtual catalogue according to the user's needs and preferences. The systems that suggest a set of items (the second group) support tourists in planning their trips, arranging transportation, accommodation, and everything else needed. Examples of these kind of systems are Expedia ([www.expedia.com](http://www.expedia.com)), DieToRecs [5], and Trip@dvice [6]. In these cases the recommendation process works like a virtual travel agency: it suggests a pre-packaged offer,

or else it guides the user in the selection of the travel components (flight, accommodation, car, etc.) according to the user's needs and preferences.

Such recommender systems support tourists in building their own itineraries; systems that can automatically build itineraries starting from the tourist's requests are still few. Some authors state that this is "due to the complexity of the task and doubts concerning scalability to handle large travel databases" [7]. The construction of itineraries needs to generate a sequence of items to be visited. In literature, the problem of generating itineraries is dealt with by filtering data according to the user's constraints (day of visit, cost, and so on) specified in the request. The Electronic Travel Planner (ETP) prototype [7], for example, stores information (such as duration, cost, and availability, dates and times) about travel products in a relational database. The problem of building itineraries is expressed in terms of constraints (day of visit, arrival/departure time and location, preferred time of visit, cost, etc.) and an objective function. The itinerary suggested to the tourist satisfies all the constraints and maximizes the value of the objective function. The value of the objective function is the difference between the "enjoyment value" of the itinerary and its cost.

Other examples of systems that use the Constraints Satisfaction Problem (CSP) approach are: *INTRIGUE* [8] and Smart Tourist Agenda Recommender (STAR) [9], recommender systems that suggest itineraries around the city of Turin (in north Italy), and the tourist guide in [10] that support users visiting the city of Salamanca (Spain). Conversely, in [11] the problem of generating itineraries in the Macau system (that extends the previous MacauMap system [12]) is solved by using the A\* graph search algorithm [13], using travel time as the distance value between tourist spots.

The method proposed in this paper and its application in a recommender system allows the tourist to receive a group of itineraries made up of a set of correlated events (procession, festival, special market, etc.). The chains of events are generated by means of transitive closure computation of the space-time relationships.

### 3 A Method for Generating Tourist Itineraries

In order to characterize the events considered (procession, festival, special market, etc.) and to determine whether two events have a space-time relationship, a set of functions and a space-time relation were defined as follows. Let  $E$  be a finite set of events  $e$  occurring in a geographical area (for example in the Apulia region). Details of defined functions are given below.

Let  $\mathcal{N}$  be the set of natural numbers, and  $L$  the finite set of all names of existing locations (at a specific grain size: district, city, region) in the geographical area considered. The functions *Start*, *End*, *Location*, and *Distance* were defined as follows:

- *Start*:  $E \rightarrow \mathcal{N}$  is the function that associates to each event  $e \in E$  a natural number that expresses the date when the event starts (for example the 1<sup>st</sup> February 2010 is the 32<sup>nd</sup> day of the year).
- *End*:  $E \rightarrow \mathcal{N}$  is the function that associates to each event  $e \in E$  a natural number that expresses the date when the event ends.
- *Location*:  $E \rightarrow L$  is the function that associates to each event  $e \in E$  the name of the location where the event occurs.

- *Distance*:  $L \times L \rightarrow \mathcal{N}$  associates to each couple of locations the distance between them. If the distance between the two places of the couple is equal to 0, then the two places are coincident. If the distance between two places is lower than a threshold value  $\varepsilon$  (conveniently fixed), then the places are near in space, otherwise the places are distant.

Given the above-described functions, it is possible to define the space-time relation  $\Delta$  between two events as follows:  $\forall e_i, e_j \in E, e_i \Delta e_j \text{ iff } e_i \Delta_t e_j \wedge e_i \Delta_s e_j$ , where:

- $i, j \in \mathcal{N}$
- $e_i \Delta_t e_j \Leftrightarrow \text{Start}(e_i) \leq \text{Start}(e_j) \leq \text{End}(e_i)$
- $e_i \Delta_s e_j \Leftrightarrow \text{Distance}(n_i, n_j) \leq \varepsilon$ , whereas  $\varepsilon \in \mathcal{N}$  is a fixed threshold,  $\text{Location}(e_i) = n_i$ ,  $\text{Location}(e_j) = n_j$ .

Given a set of events, on the basis of this relation it is possible to build paths like  $e_1, e_2, \dots, e_n$  where  $e_i$  is correlated in space and time to  $e_{i+1}$ , for  $i = 1, \dots, n-1$ . For example, starting from event  $e_i$ , we need to know whether there is an event  $e_{i+1}$  such that  $e_i$  and  $e_{i+1}$  have a space-time relationship (represented by the symbol “ $\rightarrow$ ”). If such an  $e_{i+1}$  exists, in a recursive way, we look for another event  $e_{i+2}$ , and so forth. When no other event satisfies the  $\Delta$  relation, the transitive closure computation ends. The result is a path composed of a chain of events that have a pairwise space-time relationship ( $e_i \rightarrow e_{i+1} \rightarrow e_{i+2} \rightarrow \dots \rightarrow e_n$ ). A deeper and more detailed discussion of the proposed method is reported in [14].

### 4 T-Path Architecture

The described method was implemented in the T-Path prototype, a knowledge-based recommender system that proposes tourist itineraries to see events occurring in different cities in the Apulia region. In the T-Path architecture it is possible to distinguish three logical layers (Fig. 1): (a) the data layer, that manages the data sources and builds an integrated knowledge base; (b) the application layer, that generates itineraries by transitive closure computation; and (c) the presentation layer, that displays the results of this computation, in other words the itineraries.

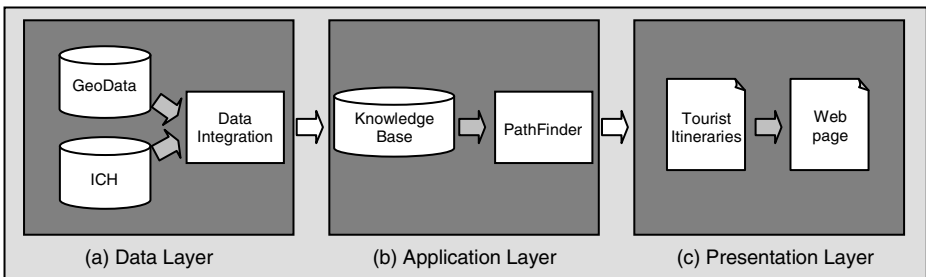


Fig. 1. T-Path architecture

## 4.1 Data Layer

The T-Path Data Layer is made up of two relational databases (Fig. 1 - a), namely GeoData and ICH (Intangible Cultural Heritage). The GeoData database stores information about the geographical structure of the considered territory. This database contains, for example, information such as “Bari is a city”, “Bari is located in Apulia”, and “Apulia is an Italian region”. Moreover, for each city the geographical coordinates (such as latitude and longitude) are also stored. In this way, it is possible to know the distance in kilometers between two cities. The ICH database stores data about cultural events (such as fairs, religious events, and musical concerts). Each event is classified in a predefined category and is described by a set of attributes. The attributes are used to define the space-time collocation of the events. Therefore, for each event, the name, the location, and the date are provided. For example, this database contains information such as: name of the event “Santa Lucia fairs”; location “Lecce (LE)”, and date: “from December 13 to December 24”. The geographical data and the ICH data are extracted from the source databases and loaded into a knowledge base through an integration process. The Data Integration module is a software component that dynamically generates the facts of the knowledge base, by mapping relational schemas to predicates. According to our mapping strategy, the integration process generates a fact in the knowledge base for each tuple in a relation.

## 4.2 Application Layer

The T-Path Application Layer is made up of the knowledge base and the PathFinder program (Fig. 1 - b). The knowledge base (see Table 1) contains the facts expressed using the Predicate Calculus. The predicates, derived from the functions of the method defined in section 3, are divided into two groups: the first group (first five predicates) describes the geographical knowledge; the second group (last two predicates) defines the space and time features of an event. The two groups of predicates are related to each other because the location of an event is a city.

The next examples aim to clarify the predicate instantiation.

**Table 1.** Knowledge base predicates

Predicate	Semantics
city(X, Y)	X is a city in province of Y.
province(X, Y)	X is a province of Y, where Y is a region.
region(X, Y)	X is a region of Y, where Y is a nation.
nation(X)	X is a nation, where X = 'Italy'.
near(X, Y)	The city X is near to the city Y.
location(e, Y)	The location where the event e happens is Y, where Y is a city.
day(e, G, A, Z)	The event e happens on the day G of the year A. Z is the duration of the event, in terms of days.

**Example 1.** The following facts state that Italy is a nation, Apulia is an Italian region, Bari and Lecce are Apulian provinces, and Bari is close to Lecce.

```
nation(Italia).
region(Apulia, Italia).
province(Bari, Apulia).
province(Lecce, Apulia).
city(Bari, Bari).
city(Lecce, Lecce).
near(Bari, Lecce).
```

**Example 2.** The following facts state that the festivity of San Nicola (Saint Nicholas) occurs recurrently in Bari on the 8<sup>th</sup> of May (that is the 128<sup>th</sup> day of 2009), and it lasts 3 days. The *location* predicate recalls the *Location* function, whereas the grain size is the city level of the geographical hierarchy. The *day* predicate recalls the *Start* and *End* functions. Thanks to these functions, both the starting date and the duration of an event are defined.

```
location('festivity of Saint Nicholas', Bari)
day('festivity of Saint Nicholas', 128, 2009, 3).
```

The PathFinder program builds a path (if it exists) via transitive closure computation using event  $e \in E$  as an initial event. Then the PathFinder writes the result of the computation in an XML file that will be displayed to the tourist by the presentation layer. The pseudo-code of the program is shown below.

```
createXML(T)
for each  $e \in E$ 
     $P_e = tc(e)$ 
    PathXML = generatePath( $P_e$ )
    add(PathXML, T)
end for
storeXML(T)
```

The core of the PathFinder is a logical program able to execute recursive queries and to compute the transitive closure. The logical program has been created using the Prolog programming language [15] and the Swi-Prolog development environment [16]. The logical program provides an answer to the following recursive query: “*what are all the possible paths starting from a specific event  $e$ ?*”. In the logical program (shown below)  $tc(e)$  is the goal and  $e$  is the starting event. The inferring rules recall the space-time relation of the method.

```
tc(e):-          location(A, X), rel(A, X).
rel(A, X):-     location(B, Y), spatial(X, Y),
                temporal(A, B), print(A, B),
                retract(location(A, X)), rel(B, Y).
spatial(X, Y):- near(X, Y); near(Y, X).
temporal(A, B):- day(A, Sa, Aa, Da),
                 day(B, Sb, Ab, Db), A \= B, Aa == Ab,
                 Dura is Sa + Da, Sb >= Sa, Sb <= Dura.
print(X, J):-   write(X), write('•'), writeln(J).
```

The output produced (the chain of events) by the program is stored in an XML file, defined by the following Document Type Definition (DTD).

```

<!ELEMENT output (path*)>
<!ELEMENT path (stage+)>
<!ELEMENT event (name, location, date, brochure)>
<!ELEMENT location (city, province, region)>
<!ELEMENT event (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT province (#PCDATA)>
<!ELEMENT region (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT brochure (#PCDATA)>
<!ATTLIST stage id CDATA>
    
```

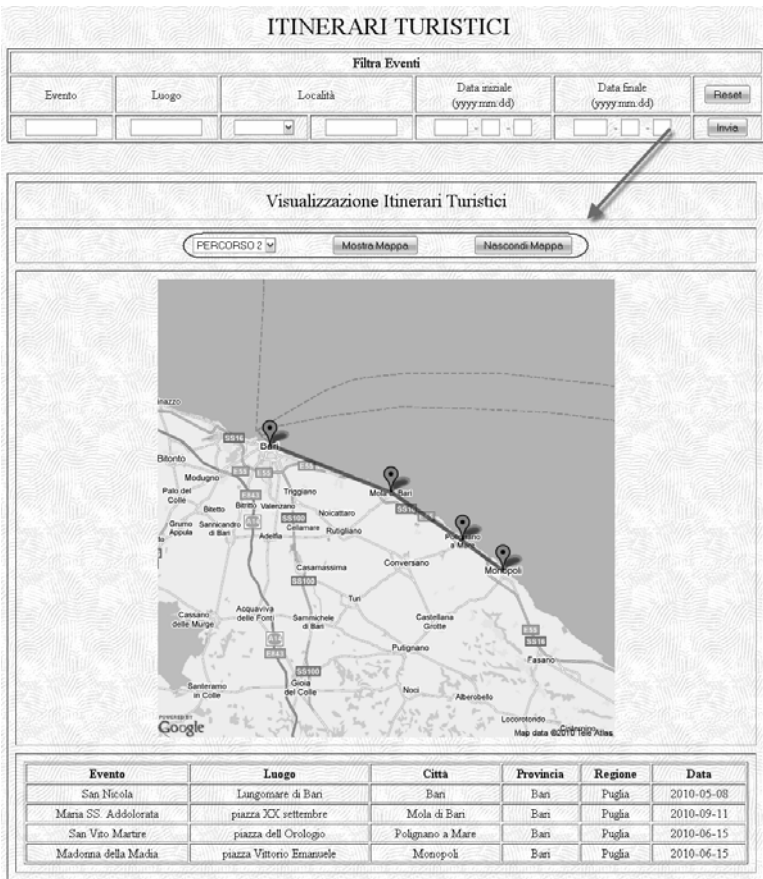


Fig. 2. A tourist itinerary computed by the Pathfinder

### 4.3 Presentation Layer

The T-Path Presentation Layer (Fig. 1 - c) aims at displaying the content of the XML file produced by the PathFinder. The information on the itineraries is visualized using both a geographical map, thanks to the integration of the Google Map [17] in the system, and a table that reports all the details of the events (Fig. 2). Moreover, it is possible to navigate the XML file in order to filter the contents on the basis of the tourist search criteria.

## 5 T-Path at Work

In T-Path, the most important religious events in the Apulia region occurring between the end of 2009 and the first months of 2010 have been collected. Apulia is a region in the South of Italy, being the easternmost region in our Nation. This region is located along the Adriatic Sea and down to the Ionian Sea. Its provinces are Bari, which is the chief town, Lecce, Brindisi, Foggia, and Taranto.

As examples, some of the events stored in the T-Path prototype are reported below (in Italian):

1. Fiera di Santa Lucia, from December 13 2009 to December 24 2009 in Lecce (LE).
2. Presepe vivente di Sanarica, from December 24 2009 to January 6 2010 in Sanarica (LE).
3. Presepe artistico, from December 24 2009 to January 11 2010 in Muro Leccese (LE).
4. Grande presepe artistico, from December 1 2009 to February 2 2010 in Salice Salentino (LE).
5. Focura per Sant'Antonio abate, January 16 2010 in Novoli (LE).
6. Presepe vivente di Bethlehem, from December 25 2009 to January 6 2010 in Grumo Appula (BA).
7. La notte dei Magi, January 5 2010 in Valenzano (BA).
8. Presepe Palazzo Marchesale, from December 8 2009 to January 6 2010 in Adelfia (BA).
9. Calata dei re Magi, January 6 2010 in Lizzano (TA).
10. Festa di San Ciro, from January 1 2010 to January 31 2010 in Grottaglie (TA).

For each city the latitude and longitude were also stored to know the distance (expressed in kilometres) between the cities. We fixed the  $\varepsilon$  threshold distance at 50 km. Let  $n_1$  and  $n_2$  be two cities, the criterion for the space relationship is

$$Distance(n_1, n_2) < 50, \text{ that is, } n_1 \text{ is near } n_2 \text{ iff } Distance(n_1, n_2) < 50.$$

As an example:

- $Distance(\text{Lecce}, \text{Sanarica}) = 32.78 < 50 \Rightarrow \text{Lecce is near to Sanarica}$
- $Distance(\text{Lecce}, \text{Adelfia}) = 131.34 > 50 \Rightarrow \text{Lecce is distant from Adelfia.}$



Of course, the *Distance* function is symmetric, i.e.  $Distance(n_i, n_j) = Distance(n_j, n_i)$ , for  $i \neq j$ , and reflexive, as  $Distance(n_i, n_i) = 0$ ,  $\forall i \in \mathfrak{N}$ . Starting from these data the transitive closure computation will be computed. For the sake of simplicity, in this case study we refer to a specific event using the *code* field. Supposing that 8 is the start event, T-Path searches whether another event  $e$  exists such that  $8 \rightarrow e$ . Since:

- $Start(8) = \text{December 8, 2009}$ ,
- $End(8) = \text{January 6, 2010}$ ,
- $Start(6) = \text{December 25, 2009}$ ,
- $End(6) = \text{January 6, 2010}$ ,
- $Location(8) = \text{Adelfia}$ ,
- $Location(6) = \text{Grumo Appula}$ ,
- $Start(8) \leq Start(6) \leq End(8)$ ,
- $Distance(8, 6) = 13.56 \Rightarrow \text{Adelfia is near Grumo Appula}$ ,

the T-Path system concludes that event 8 has a space-time relationship with 6, so it is possible to build the path  $8 \rightarrow 6$ .

In a recursive way, the T-Path system searches for another event  $f$  such that  $6 \rightarrow f$ :

- $Start(6) = \text{December 25, 2009}$ ,
- $End(6) = \text{January 6, 2010}$ ,
- $Start(7) = \text{January 5, 2010}$ ,
- $End(7) = \text{January 5, 2010}$ ,
- $Location(6) = \text{Grumo Appula}$ ,
- $Location(7) = \text{Valenzano}$ ,
- $Start(6) \leq Start(7) \leq End(6)$ ,
- $Distance(6, 7) = 15.03 \Rightarrow \text{Grumo Appula is near Valenzano}$ ,

The T-Path system concludes that events 6 and 7 have a space-time relationship and builds the path  $6 \rightarrow 7$ . Since no other event has a space-time relationship with 7, the computation ends. The final result of this recursive query is path  $A = \{8 \rightarrow 6 \rightarrow 7\}$ . In the same way, T-Path finds the following further paths:

- $B = \{4 \rightarrow 5\}$ ,
- $C = \{4 \rightarrow 10 \rightarrow 9\}$ ,
- $D = \{4 \rightarrow 1 \rightarrow 2 \rightarrow 3\}$ ,
- $E = \{4 \rightarrow 1 \rightarrow 3 \rightarrow 2\}$ .

where A, B, C, D and E are independent paths and each path is formed by an ordered set of events. The XML file of the path B is reported below.

```
<output>
  ...
  <path>
    <event id = 4>
      <name>Grande presepe artistico</name>
      <location>
        <city>Salice Salentino</city>
        <province>Lecce</province>
```

```

        <region>Apulia</region>
    </location>
    <date> December 1, 2009</date>
</event>
</path>
<path>
    <event id = 5>
        <name> Focura per Sant'Antonio abate
    </name>
        <location>
            <city>Novoli</city>
            <province>Lecce</province>
            <region>Apulia</region>
        </location>
        <date> January 16, 2010</date>
    </event>
</path>
    ...
</output>

```

The XML tree built is then displayed to the tourist using both a map where the path is depicted and a table showing all the details of the chain of events (Fig. 2). Moreover, the tourist can view the details of all the itineraries computed using a select box.

## 6 Conclusions

Up to now, the problem of automatically building itineraries to be proposed by recommender systems has been faced using the Constraints Satisfaction Problem (CSP) approach. The output is a sequence of attractions or spots to be visited, filtered according to the tourist's constraints (day of visit, cost, and so on) specified in the request.

This paper presents a method for building itineraries incorporated in a knowledge-based recommender system. The proposed method aims to find a chain of events using transitive closure computation on the basis of the space-time relations. This is possible thanks to the use of the theoretical model, that consists of a set of functions that characterize each single event, and a space-time relation that allows the events to be correlated. The theoretical model can be easily modified so as to correlate events at different grain sizes, both in terms of time and space. Moreover, it can be incorporated in any knowledge-based recommender system that uses an ontological representation of events and has a logical program able to execute recursive queries. The paper illustrates an implementation of the proposed method in T-Path, a recommender system prototype that suggests itineraries of cultural events (festivals, processions, special markets, etc.) in the Apulia region. Future developments will allow the chain of events to be tailored to the tourist's profile. For example, if the tourist is a religious man, the system will attribute high priority to the presentation of the whole chain of religious events (such as processions, historical parades, etc.).

## References

1. Tsang, E.: *Foundations of Constraint Satisfaction*. Academic Press, London (1993)
2. Elmasri, R., Navathe, S.B.: *Fundamentals of Database Systems*, 5th edn. Addison-Wesley, Reading (2006)
3. Aho, V., Ullman, J.D.: Universality of data retrieval languages. In: 6th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages, pp. 110–119. ACM Press, New York (1979)
4. Ricci, F.: Travel recommender systems. *J. IEEE Intelligent Systems* 17, 55–57 (2002)
5. Ricci, F., Fesenmaier, D.R., Mirzadeh, N., Rumetshofer, H., Schaumlechner, E., Venturini, A., Wöber, K.W., Zins, A.H.: DieToRecs: a case-based travel advisory system. In: *Destination Recommendation Systems: Behavioral Foundations and Applications*. CABI Publisher International, Wallingford (2006)
6. Venturini, A., Ricci, F.: Applying trip@advice recommendation technology. In: 17th European Conference on Artificial Intelligence, pp. 607–611. IOS Press, Amsterdam (2006), <http://www.visiteurope.com>
7. Dunstall, S., Horn, M., Kilby, P., Krishnamoorthy, M., Owens, B., Sier, D., Thiebaut, S.: An automated itinerary planning system for holiday travel. *J. Information Technology & Tourism* 6, 1–33 (2004)
8. Petrone, G., Ardissono, L., Goy, A.: INTRIGUE: personalized recommendation of tourist attractions for desktop and handset devices. *J. Applied Artificial Intelligence* 17, 687–714 (2003)
9. Goy, A., Magro, D.: Dynamic Configuration of a Personalized Tourist Agenda. In: *IADIS International Conference WWW/Internet 2004*, pp. 619–626. IADIS, Madrid (2004)
10. Corchado, J.M., Pavón, J., Corchado, E.S., Castillo, L.F.: Development of CBR-BDI agents: A tourist guide application. In: Funk, P., González Calero, P.A. (eds.) *ECCBR 2004*. LNCS (LNAI), vol. 3155, pp. 547–559. Springer, Heidelberg (2004)
11. Biuk-Aghai, R.P., Fong, S., Si, Y.-W.: Design of a Recommender System for Mobile Tourism Multimedia Selection. In: *2nd International Conference on Internet Multimedia Services Architecture and Application*, pp. 1–6. IEEE Press, Los Alamitos (2008)
12. Biuk-Aghai, R.P.: MacauMap: Next generation mobile travelling assistant. In: *Proceedings of Map Asia* (2004)
13. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. *J. IEEE Transactions on Systems Science and Cybernetics* 4, 100–107 (1968)
14. Di Bitonto, P., Di Tria, F., Laterza, M., Roselli, T., Rossano, V., Tangorra, F.: A model for generating tourist itineraries. In: *4th International Workshop on Semantics in Data and Knowledge Bases* (2010) (submitted)
15. Sterling, L., Shapiro, E.: *The Art of Prolog: Advanced Programming Techniques*, 2nd edn. MIT Press, Cambridge (1994)
16. SWI-Prolog, <http://www.swi-prolog.org>
17. Google Map, <http://maps.google.it>