A Model for Generating Tourist Itineraries

Pierpaolo Di Bitonto, Francesco Di Tria, Maria Laterza, Teresa Roselli, Veronica Rossano, Filippo Tangorra
Department of Computer Science
University of Bari "Aldo Moro"
Via Orabona 4, 70125 - Bari – Italy
{dibitonto, francescoditria, marialaterza, roselli, rossano, tangorra}@di.uniba.it

Abstract— **A challenge in the recommender systems currently available for the tourism domain is how to suggest tourist itineraries in a specific geographical area (city or region). The proposed theoretical model allows items of intangible cultural heritage (events) such as processions, festivals, special markets, etc. to be characterized and correlated. The model features both a set of functions characterizing the events and a space-time relation that defines whether two events are correlated. The model allows itineraries to be constructed by computing the transitive closure of the space-time relation on the set of events. It can be used to construct itineraries at different grain sizes. This capacity makes the model scalable and easily applicable in the development of several applications. It has been implemented in a first order logic knowledge base in order to make an empirical evaluation of the model.**

Transitive closure; filtering; logical programming; cultural events

I. INTRODUCTION

The main challenge of recommender system research is to define methods that are ever more efficient in terms of both suggestion reliability and computational complexity. One of the most popular and interesting experimental contexts used to test the research results is tourism, where there are numerous details to be considered during the recommendation process. In this context, the paper illustrates in depth the theoretical model defined to describe and correlate intangible cultural heritage events. The model is the basis of the method for generating tourist itineraries implemented in the T-Path prototype [19]. Items of intangible cultural heritage are events that are strictly inherent to the culture and the history of a Country. During a trip, a tourist may be interested in visiting the main important events of a specific territory. Therefore, a recommender system for tourism should be able to select and present all the events occurring on a specific date in a given location. Moreover, it should be able to use the relationships among these events to suggest a trip that best fits the user requests. But, the question arises: *when are two events related to each other*? We can assume that two events are related to each other when they show both a space and time proximity, that is, the locations and the dates when they occur are near. Given such an informal definition, we can find all the sets of events that have a space-time relationship and use this relationship to generate a set. The retrieved items are represented in an ordered list of events; for each event, the date and the location are specified. This ordered list will be used to build a path that can be suggested to the tourist who would like to attend such events. The research process aims at answering the following recursive query: "*given an event x, which are all the possible tourist itineraries starting from x?*". The tourist itineraries are the result of the computation of the transitive closure of the space-time relationship on the considered events. The method proposed, if implemented in a knowledge-based recommender system which uses an ontological representation of the events and a logical program that computes the transitive closure [1], will be able to suggest and display several tourist itineraries in a 2-dimensional map. The existing recommender systems for tourism support users planning a trip by suggesting locations to visit, hotels to book, activities to do, and so on, on the basis of the user requests and profile, but they do not use space-time dimension to correlate items to be suggested.

The paper is organised as follows: section 2 presents some related works about methods used to suggest travel products and generate itineraries in recommender systems for tourism; section 3 presents the theoretical model; section 4 presents the transitive closure computation; section 5 illustrates a possible application of the theoretical model. Finally, in section 6 some conclusions and future research directions are outlined.

II. RELATED WORKS

Up to a few years ago the systems offered in the tourism domain were only able to suggest to the user single travel products or services such as destinations, accommodation, attractions, and so on. The methods used to make suggestions were above all content-based. They consisted of describing by their features the products to be suggested, inside a virtual catalog, and then matching these feature with others derived from the user inputs that can include information such as travel dates, number of people, budget, preferred activities, etc. Applications of these methods can be found in [2], where Triplehop's TripMatcher recommender system included in the Ski Europe website (www.ski-europe.com) and VacationCoach's Me-Print recommender included in the Travelocity website (www.travelocity.com), are described. Both the systems suggest travel destinations. Subsequently, several methods proposing whole travel plans and guiding

978-1-4244-8136-1/10/$26.00

the use in choosing their components (accommodation, flights, car, attractions to visit, etc.) were defined. The most commonly used method was based on case-based reasoning [5]: a problem-solving methodology that faces a new problem or situation by firstly retrieving a similar case, already solved in the past, and then reusing that case in order to solve the current problem [6]. This method was used in the DieToRecs [3] recommenders included in the Expedia website (www.expedia.com) and in Trip@dvice [4]. These systems work like a virtual travel agency: they suggest a pre-packaged offer or guide the user in the selection of the travel components to be assembled (flight, accommodation, car, etc.) that meet the user's needs and preferences. Trip@dvice, for example, stores completed travel plans in a Case Base, as good examples. During the recommendation session, the system retrieves travel characteristics and the cases most similar to the current one, through a similarity function including current and historic users. Another method used to suggest travel plan is knowledge-based. In this case the retrieval of the packages to be suggested was done using rules and relationships among concepts. An application is presented in [7], which illustrates a recommender that suggests the best travel packages for Argentinian destinations according to the user's preferences and restrictions. The packages to be recommended are retrieved using rules and two relationships: "to be similar to" and "to be better than", that allow the set of possible values that can satisfy the tourist's preferences to be expanded. Therefore, all these recommender systems plan the trip by a simple selection of each single component that the user should assemble according to her/his needs.

In the last year, advances in ubiquitous computing and wireless networking technologies have allowed applications for mobile devices to be created, that can offer the user tourist itineraries. Thus, the present challenge in this research is to define more and more efficient methods to construct such itineraries. In literature several methods for constructing itineraries of tourist spots such as churches, museums, entertainment, and so on. Among these methods, there are some based on fuzzy logic [15] and on the Travelling Salesperson Problem [16] methods based on shortest path algorithms, and so on. For example, the solution presented in [17] selects a set of spots to visit (according to the user preferences) using a genetic algorithm inspired by natural concepts such as mutation, selection and crossover. The number of selected spots depends on the time available. On the basis of the tourist's interests a visit time is associated to each spot. For example, more time will be spent by a tourist who is particularly interested in visiting the spot, less if she/he is moderately interested in visiting the spot. Then, the sequence of tourist spots is generated using the standard shortest path A* algorithm [16] and taking into account the visit time and the distance between a couple of spots measured as travel time. This method was used in the Macau recommender, a web-based tourist map guide system that suggests itineraries in Macau city (China).

In the same way, Lee et al. in [18] also use fuzzy logic and solve a Travelling Salesperson Problem. The fuzzy logic allows the selection of spots, while solving the Travelling Salesperson Problem allows construction of the paths. This method was used in a recommender system that plans travel itineraries in Tainan city (China). In this case the method selects three historic sites and five local food stores, after it has solved the Travelling Salesperson Problem on these eight locations.

Another way to construct spot itineraries can be found in [8]. In this work the problem of building a tour in a preferred geographical area was solved using a configuration problem approach: several components which satisfy the constraints (that are the user requests) are assembled in a tour. This method of constructing itineraries was implemented in the Smart Tourist Agenda Recommender (STAR) prototype, a system that supports tourists organizing a personalized agenda for a tour in the city of Turin (Italy). The information supplied by the user is the starting point of the tour, the kind of preferred restaurants, the attractions or events to attend and so forth. The result provided by the system is a list of things to do during the stay. These methods are the starting point of our research. They are able to construct spot itineraries but they are not able to consider the space-time dimension to correlate items.

The main contribution of this paper is to describe in a detailed manner the defined theoretical model to characterize and correlate the intangible heritage events. This model can be used to construct itineraries at any grain size level. Thus, any recommender system in which it is implemented will be able to propose an ordered list of events (intangible cultural heritage) the tourist can attend during her/ his visit.

III. THEORETICAL MODEL

The model defined herein describes events according to the space-time point of view by using a dedicated set of functions. Starting from the events and their characterization, the space-time relation is adopted to correlate couples of events.

The presented model can describe different situations, in which the events considered are placed in an specific space-time context (such as free entrance to the museum on Sunday morning, flea market on Saturday evening, and so on). The aim of the model is to allow possible tourist itineraries among the interesting events to be calculated and suggested to users.

A. Event characterization

The first step in the model definition is to characterize the event in terms of space and time. For this reason, the *Start*, *End*, *Location*, and *Distance* functions have been defined.

Let e be an event (if necessary provided with an index), let E be a finite set of events, let $\aleph$ be the set of natural numbers, and let L be the set of all possible names of existing geographical locations on Earth. The following functions are defined:

$Start: E \rightarrow \aleph$.

Start is a monotonically strictly increasing function that associates to each event e a natural number that expresses the date when the event starts: $\forall e \in E \; \exists \; s \in \aleph \mid Start(e) = s$. Since *Start* is a monotonically strictly increasing function, given a couple of events (e_1, e_2), if e_1 starts before e_2, then $Start(e_1) < Start(e_2)$.

End: $E \rightarrow \aleph$.

End is a monotonically strictly increasing function that associates to each event e a natural number that expresses the date when the event ends: $\forall e \in E \; \exists \; s \in \aleph \mid End(e) = s$. Since *End* is a monotonically strictly increasing function, given a couple of events (e_1, e_2), if e_1 ends before e_2, then $End(e_1) < End(e_2)$. We assume that $\forall e \in E$, $Start(e) \leq End(e)$.

$Location$: $E \rightarrow L$.

Location is the function that associates to each event e the name of the place where the event is located: $\forall e \in E$, $\exists \; n \in L \mid Location(e) = n$.

$Distance$: $L \text{x} L \rightarrow \aleph$.

Distance is the function that associates to each couple of names of the geographical places the distance between the elements of the couple: $\forall (l_1, l_2) \in L \times L$, $Distance(l_1, l_2) = d$, where $d \in \aleph$. If the distance between the two places of the couple is equal to 0, then the two places are coincident: $\forall (l_1, l_2) \in L \times L$, $Distance(l_1, l_2) = 0 \Leftrightarrow l_1 = l_2$.

The defined *Start*, *End*, and *Location* functions allow each event e to be characterised regardless of the granularity to be assigned to the considered events. For instance, by changing the interpretation of the *Start* and *End* functions, it is possible to define two events that occur on the same day, or in the same hour, or in the same week as contemporaneous. In the same way, it is possible to change the interpretation of the function *Location*, so as to identify as the place where the event is located a city, a district, or a wider region.

The choice of the model grain size will change the way the events are considered without changing the working logic of the model. This allows a high scalability level. This property is very important in a system intended to build cultural routes. The reasons are obvious if we compare the needs of a tourist who is planning a bicycle trip with those of a tourist who is planning to visit a wider region travelling by train or by car. In fact, a tourist travelling by train is able to move through cities very quickly.

Moreover, we can also take into account the needs of a Roman citizen (who is interested in cultural routes in her/his city) or the needs of a foreign tourist (interested in visiting Rome and other surrounding places), or those of other tourists who are planning to visit Rome, Florence, Venice, and Naples. For each tourist, the concept of distance among places can be very different.

B. Equivalent events

Independently of the space-time grain size level, it is possible that two or more different events may occur in the same place at the same time. In this case, the model should be able to consider them as a whole. In order to solve this problem, the Σ relation (coincident events) has been defined.

Let us define two events as coincident if they start at the same time, end at the same time, and occur in the same location. The Σ relation is applied as an equivalence relation on the set of the events. So, the quotient set E_Σ has been considered. Each element of E_Σ is a set of one or more coincident events.

The model considers equivalent events as a single set of elements with the same space-time properties. In this way, the model presents to the tourist a set of equivalent events and allows her/him to choose the events s/he prefers.

Definition 1. Let E be the set of events e_i, where $i \in \aleph$, that is $E := \{e_i, i \in \aleph\}$. The Σ relation is defined on E as follows: $\forall \; e_i, e_j \in E \; e_i \Sigma e_j \Leftrightarrow Location(e_i) = Location(e_j) \wedge Start(e_i) = Start(e_j) \wedge End(e_i) = End(e_j)$,

where:

- $i, j \in \aleph$,
- *Location*(e_i) is the smallest geographical place that contains the event e_i,
- *Start*(e_i) is the time when the event e_i starts,
- *End*(e_i) is the time when the event e_i ends.

Theorem 1. The Σ relation is an equivalence relation on E.

It is very simple to show the reflexive, the symmetric, and the transitive properties of Σ on E. For the sake of brevity it has therefore been omitted. From Theorem 1, E_Σ is the quotient set, that is the set of equivalence classes generated by the Σ relation, whereas the elements of E_Σ are defined as follows.

Definition 2. Let $j \in \aleph$ be the number of events. Then, $\forall [e_j]_\Sigma \in E_\Sigma$, $[e_j]_\Sigma = \{e_i^{'} \in E, \; i \in \aleph \mid Location(e_1^{'}) = Location(e_2^{'}) = \ldots = Location(e_n^{'}) \wedge Start(e_1^{'}) = Start(e_2^{'}) = \ldots = Start(e_n^{'}) \wedge End(e_1^{'}) = End(e_2^{'}) = \ldots End(e_n^{'})\}$.

For the sake of simplicity, in the rest of the paper the elements of $[e_j]_\Sigma$ are represented by e_j when no ambiguity arises. In other words, if events start at the same time, end at the same time, and occur in the same place, then they are equivalent and we take them as a whole.

C. Space-time relation

The Δ relation has been defined on E_Σ. In particular, Δ is satisfied if the considered events are related in space and time. We have also defined the *C* relation in order to generate chains of events. The *C* relation correlates events if a chain of arbitrary length composed of Δ-correlated events exists. It has been proved that *C* is an ordering relation on E_Σ, so each chain of events built by means of *C* is an ordered set of events. Such ordered lists of events are the itineraries to be suggested to the tourists.

Definition 3. Let E_Σ be the set of equivalence classes generated by Σ relation on *E*. The Δ relation is defined as follows.

$\forall e_i, e_j \in E_\Sigma$, $e_i \Delta e_j$ iff $e_i \Delta_t e_j \wedge e_i \Delta_s e_j$, where:

- $i, j \in \aleph$,

- $e_i \Delta_t e_j \Leftrightarrow Start(e_i) \leq Start(e_j) \leq End(e_i)$,
- $e_i \Delta_s e_j \Leftrightarrow Distance(Location(e_i), Location(e_j)) \leq \varepsilon$, whereas $\varepsilon \in \aleph$ is a quantity small as desired and fixed.

According to Definition 2, $e_i \Delta e_j$ can be also written as $\Delta(e_i) = e_j$. Moreover, the Δ_s relation allows adaptation of the intended interpretation (semantics) of the *Distance* function by suitably varying the value of ε. Moreover, note that if $e_i \Delta_t e_j$ holds, then e_i precedes e_j.

IV. Transitive Closure Computation

Definition 4. Let us consider E_Σ and Δ as defined previously. The powers Δ^m of Δ are: $\Delta^0 = \Im E_\Sigma$ and $\Delta^{k+1} = \Delta^\circ \Delta^k$, where $m \in \aleph$, $\Im E_\Sigma$ is the identity relation on E_Σ such that $\forall x \in E_\Sigma$ it results that $x \Delta x$, and ° is the composition operator. Given Δ and Δ^p as any power of Δ (with $p \in \aleph$) both relations defined on E_Σ, $\Delta^\circ \Delta^p(x) = y \Leftrightarrow \exists\, e_i$, for $i \in \aleph$ | $\Delta(x) = e_i \wedge \Delta^p(e_i) = y$.

Let us consider E_Σ and Δ. The transitive closure t of Δ is the smallest transitive relation Δ' on E_Σ that contains Δ:

$$t(\Delta) = \bigcup_{k \geq 0} \Delta^k$$

where $t(\Delta)$ contains the union of all the couples of Δ-correlated events. Therefore, by computation of the transitive closure of Δ on E_Σ, we obtain a set of pairwise-related events.

The correlations between events are now used to generate sequences of events by applying C, the events chaining operator. If x and y are two elements of E_Σ, and a chain of events $e_1,\ldots, e_n$ exists, such that $\Delta(x) = e_1 \wedge \Delta(e_1) = e_2 \wedge \ldots \wedge \Delta(e_{n-1}) = e_n \wedge \Delta(e_n) = y$, then x and y are linked by a path of Δ-correlated events. Obviously, starting from an event e_i if there are two events e_j and e_z correlated in space-time dimension with e_i, we can construct two chains of events. The formalization of the events chaining is provided by the following definition of the *C* operator.

The necessity to introduce the C operator is due to the consideration that the events correlated using the powers of Δ are not ordered, because this relation is not transitive. On the other hand, the informal meaning of C is that two events are correlated whether there is a path between them. Therefore, the C relation can be easily proven that it is an ordering relation. The formal definition can be found below.

Definition 5. $\forall x, y \in E_\Sigma$, we have that $C(x) = y$ iff $\Delta(x) = y \vee \exists e_1, \ldots, e_n \in E_\Sigma$, such that $\Delta(x) = e_1 \wedge \Delta(e_1) = e_2 \wedge \ldots \wedge \Delta(e_n) = y$, where $n \in \aleph$. Thus, for $n = 0$, we have that $\Delta(x) = x$. For $n = 1$, $\Delta(x) = y$. For $n = 2$, $\Delta(x) = e_1 \wedge \Delta(e_1) = e_2 \wedge \Delta(e_2) = y$, and so on.

Corollary 1. $\forall x, y \in E_\Sigma$, $C(x) = y$ iff $\exists e_1, \ldots, e_n \in E_\Sigma$, with $n \in \aleph$, such that $\Delta(x) = e_1 \wedge \Delta(e_1) = e_2 \wedge \ldots \wedge \Delta(e_n) = y \Leftrightarrow \Delta\underbrace{\Delta \ldots \Delta}_{n-1}(x) = y \Leftrightarrow \Delta^n(x) = y$.

Theorem 2. $\forall x, y \in E_\Sigma$, $C(C(y)) = x \Rightarrow C(y) = x$.

Proof. $\forall x, y \in E_\Sigma$, $C(y) = x \Leftrightarrow \exists e_1, \ldots, e_n \in E_\Sigma$, with $n \in \aleph$, such that $\Delta(y) = e_1 \wedge \Delta(e_1) = e_2 \wedge \ldots \wedge \Delta(e_n) = x \Leftrightarrow \Delta\underbrace{\Delta \ldots \Delta}_{n-1}(y) = x \Rightarrow C(\underbrace{\Delta \ldots \Delta}_{n}(y)) = x \Leftrightarrow \exists e_1', \ldots, e_m' \in E_\Sigma$, with $m \in \aleph$, such that $\Delta(\Delta^n(y)) = e_1' \wedge \ldots \wedge \Delta(\Delta^n(e_m')) = x \Rightarrow \underbrace{\Delta \ldots \Delta}_{m}(\Delta^n(y)) = x \Rightarrow \Delta^{m+n}(y) = x$,

where, for $m + n = w \in \aleph$, $\Delta^w(y) = x \Rightarrow C(y) = x$. □

Theorem 3. C is an order relation on E_Σ.

Proof. C is an order relation if it satisfies the reflexive, the anti-symmetric, and the transitive properties.

For the reflexivity, it is sufficient that (on the basis of the Definition 3, for $n = 0$), $\forall x \in E_\Sigma$, $\Delta(x) = x \Rightarrow x \Delta_t x \wedge x \Delta_s x \Rightarrow Start(x) \leq Start(x) \leq End(x) \wedge Distance(Location(x), Location(x)) = 0 \leq \varepsilon \in \aleph \Rightarrow C(x) = x$. In fact, $Start(x) = a \in \aleph$ and $a \leq a$. Moreover, $Start(x) \leq End(x)$ for definition of the *End* function. Finally, $Distance(Location(x), Location(x)) = 0$ for definition of the *Distance* function.

The C relation is anti-symmetric iff $\forall x, y \in E_\Sigma$, $C(x) = y \wedge C(y) = x \Rightarrow x = y$. We note that:
$C(x) = y \wedge C(y) = x \Rightarrow C(C(y)) = y \wedge C(C(x)) = x$, for substitution. However, $C(C(y)) = C(y) = y \wedge C(C(x)) = C(x) = x$, for Theorem 3. So, we have that $C(y) = y = C(x) \Rightarrow x = y$. In other words, if $C(x) = y \wedge C(y) = x$ holds, then a chain of length 0 exists and it is reduced to one element $x = y$.

The C relation is transitive iff $\forall x, y, z \in E_\Sigma$, $C(x) = y \wedge C(y) = z \Rightarrow C(x) = z$. On the basis of the linkage of events, we have:
(4) $C(x) = y \Leftrightarrow \exists e_1, \ldots, e_n \in E_\Sigma, n \in \aleph$, such that $\Delta(x) = e_1 \wedge \ldots \wedge \Delta(e_n) = y \Leftrightarrow \Delta^n(x) = y$,
(5) $C(y) = z \Leftrightarrow \exists e_1', \ldots, e_m' \in E_\Sigma, m \in \aleph$, such that $\Delta(y) = e_1' \wedge \ldots \wedge \Delta(e_n') = z \Leftrightarrow \Delta^m(y) = z$, then, by substituting (4) in (5), it follows that $\Delta^m(\Delta^n(x)) = z \Rightarrow \Delta^{m+n}(x) = z \Rightarrow C(x) = z$. □

Each chain of ordered events is an automatically generated itinerary that can be presented to a tourist. On the basis of the $t(\Delta)$, it is possible to generate tourist itineraries of length k, where k is the fixed point of the recursive algorithm**.**

V. Implementation of the Theoretical Model

The section examines how it is possible to use the theoretical model to construct itineraries in knowledge-based systems using transitive closure computation on the basis of the space-time relationships among events. This computation is not possible via Relational Calculus but via Predicate Calculus, as Aho and Ullman [9] noted that Relational Calculus does not suffice to express such a recursive query. Therefore, the implementation depends on a logical program that relies on a knowledge base.

Firstly, in order to define the knowledge base, we have to provide an ontological representation of the domain.

A. Ontological representation

An ontology is a formal representation of concepts of the domain of interest. In this case, we need to describe items of intangible cultural heritage that are events related to the history of a territory (religious events, gastronomic festivals, wine festivals, and so on). This representation allows us to model the main entities of the domain and the semantic relationships existing among these entities. An ontology is usually graphically depicted as a structure composed of a set of nodes (that represent the entities) and a set of oriented arcs (that represent the relationships among entities) [10, 11]. The ontology at the base of the model is shown in Fig. 1.

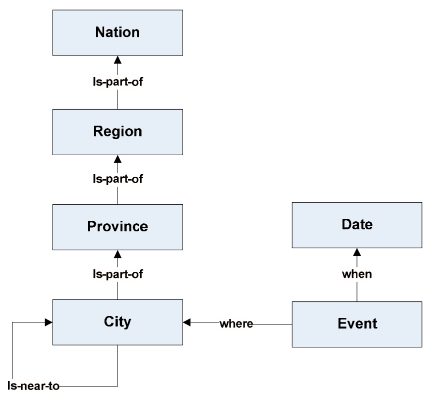

Figure. 1. Ontological representation of the domain.

There are several languages used to represent an ontology. The language we used is Predicate Calculus [12].

We have defined two predicate groups related each other. The first four predicates (see Table I) describe the knowledge about the geography of a territory. For example, they allow the hierarchical structure of a Country to be defined : a nation is composed of regions, a region is composed of provinces, and a city can be a province or not and so on. The last predicate among these (that is, *near*) recalls the Δ_s relation and allows us to define which cities are close each other. Thus, given two cities X and Y, if $X \Delta_s Y$, then *near*(X, Y), that is, X is near Y if the distance between X and Y is lower than a preestablished threshold value. We note that this predicate is not symmetric as *near*(X, Y) ≠ *near*(Y, X).

TABLE I. GEOGRAPHICAL 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.

Using these predicate we can define facts like “Bari and Lecce are cities” “Bari and Lecce are provinces of Apulia region”, “Apulia is a region of Italy”, “the Italian Country is a nation”, and also “Bari city is close to Lecce city”.

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

Other two predicates (see Table II) can define the space-time features of the events. The first predicate recalls the *Location* function, whereas the grain size level is the city level of the geographical hierarchy. The second predicate recalls the *Start* and *End* functions. Thanks to these functions, we can know the starting date of an event, as well as its duration.

TABLE II. SPACE-TIME PREDICATES.

location(A, X)	The location where the event A happens is X, where X is a city.
day(A, Sa, Aa, Da)	The event A happens on the day Sa of the year Aa. Da is the duration of the event, in terms of days.

Using these two predicates we can define facts like “The Saint Nicholas’ festivity is held in Bari” and “The Saint Nicholas’ festivity occurs recurrently the 128th day of the 2009 (that is the 8th of May)”

```
location(‘festivity of Saint Nicholas’,
Bari)

day(‘festivity of Saint Nicholas’,128,
2009, 3).
```

B. Logical program

The logical program represents the implementation of the recursive query devoted to the computation of the transitive closure. The logical program can be realized using Prolog [13] as programming language and Swi-Prolog [14] as software environment.

The program computes the transitive closure starting from a given initial event. For example, starting from the event e_1, we need to know whether an event e_2 exists such that e_1 and e_2 have a space-time relationship. If such a e_2 exists, in a recursive way, we look for another event e_3 such that e_2 and e_3 are related on the basis of the same relationship. When no other event satisfies the relation, the computation ends and the result is a cultural route, composed of a set of ordered events having a space-time relationship. In this case, the path is $e_1 \rightarrow e_2 \rightarrow e_3$ (that is, $e_1 \Delta e_2 \Delta e_3$).

The logical program is shown below.

```
start(A):-        location(A, X), rel(A, X).
rel(A, X):-       location(B, Y), space(X, Y),
                  time(A, B), print_couple(A, B),
                  retract(location(A, X)), rel(B,Y).
space (X, Y):-    near(X, Y); near(Y, X).
time(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_couple(X, J):- write(X), write('→'), writeln(J).
```

This program is able to find all couples of events that satisfy the Δ relation. If two events satisfy this relation, then they have a space-time relationship.

The *space* inferring rule allows retrieval all the events that have a space relationship, according to the *near* predicate. Then, given two events e_1 and e_2, a space relationship between e_1 and e_2 exists, iff e_1 is near e_2 or e_2 is near e_1.

The *time* inferring rule recalls the Δ_t relation. Then, given two events e_1 and e_2, there exists a time relationship between e_1 and e_2, iff e_1 precedes e_2 (that is, the date e_1 happens is lower than the date e_2 happens), e_2 occurs after the beginning of e_1 and before e_1 ends.

Finally, if e_1 and e_2 have space-time relationships (that is, $e_1 \Delta e_2$), where e_1 is the starting event, then, the program (partial) output is $e_1 \rightarrow e_2$. The computation continues recursively, considering e_2 as the starting event.

VI. CONCLUSIONS

The paper explores the theoretical model that underlies the method for generating tourist itineraries implemented in T-Path prototype [19]. The model, based on several functions and a space-time relation, allows items of intangible cultural heritage (festivals, processions, special markets, etc.) to be characterized and correlated.

In the context of cultural tourism, the model can be implemented in a knowledge-based recommender system in order to plan a complete cultural route that the user can follow during her/his travel. In this way we can consider space and time relationships while selecting the set of correlated events and so build cultural routes by computation of the transitive closure.

Moreover, the model allows itineraries to be generated at different grain size levels. This capacity makes the model scalable and applicable in the development of several applications. Finally, it is general. In fact, it could be used in other context domains in order to suggest chains of items. By modifying the Δ relation and the item description, the model could suggest friendship paths in a social network (friend of friends), religious paths, gastronomic paths, navigational paths on the web, and so on. Currently, the model does not take into account the use preferences and the events are ordered according to the starting date. Our future work will focus on detailing the model in order to include new relations and the event duration. This will contribute to generate more optimized tourist itineraries.

REFERENCES

[1] R. Elmasri and S.B. Navathe, Fundamentals of Database Systems, 5th ed., USA: Addison Wesley, 2006.

[2] F. Ricci, “Travel recommender systems,” IEEE Intelligent Systems, vol. 17, Nov./Dec. 2002, pp. 55-57.

[3] F. Ricci, D.R. Fesenmaier, N. Mirzadeh, H. Rumetshofer, E. Schaumlechner, A. Venturini, K.W. Wöber, and A.H. Zins, “DieToRecs: a case-based travel advisory system. In Destination Recommendation Systems: Behavioral Foundations and Applications,” CABI Publisher International, Wallingford, pp. 227-239, 2006.

[4] A. Venturini and F. Ricci, “Applying trip@advice recommendation technology to www.visiteurope.com,” Proc. 17th European Conference on Artificial Intelligence, IOS Press, Aug. 28-Sept. 1, 2006. pp. 607-611.

[5] A. Aamodt and E. Plaza, “Case-based reasoning: foundational issues, methodological variations and system approaches,” Artificial Intelligence Communications, vol. 7, Mar. 1994, pp. 39-59.

[6] F. Ricci and H. Werthner, “Case-based querying for travel planning recommendation,” Information Technology and Tourism, vol. 4, Nov. 2001, pp. 215-226.

[7] A. Casali, L. Godo, and C. Sierra, “A Tourism Recommender Agent: from theory to practice,” Inteligencia Artificial, vol. 40, 2008, pp. 23-38.

[8] A. Goy and D. Magro, “Dynamic Configuration of a Personalized Tourist Agenda,” Proc. IADIS International Conference WWW/Internet (IADIS 04), IADIS, Oct. 6-9 2004, pp. 619-626.

[9] V. Aho and J.D. Ullman, “Universality of data retrieval languages,” Proc. 6th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL '79), ACM Press, January 29-31, 1979, pp.110-119, doi: http://doi.acm.org/10.1145/567752.567763.

[10] K. Knight, E. Rich, and B. Nair, Artificial Intelligence, McGraw-Hill Education, 2008.

[11] N.J. Nilsson, Artificial Intelligence: A New Synthesis. San Francisco: Morgan Kaufmann, 1998.

[12] E.W. Dijkstra and C. S. Scholten, “Predicate Calculus and Program Semantics,” New York: Springer-Verlag, Inc. 1990.

[13] L. Sterling and E. Shapiro, “The Art of Prolog: Advanced Programming Techniques,” 2nd ed. USA: MIT Press. 1994.

[14] SWI-Prolog: www.swi-prolog.org.

[15] L. Zadeh, “Outline of a new approach to the analysis of complex system and decision process,” IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, Jan. 1973, pp. 28-44.

[16] P.E. Hart, N.J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths”, IEEE Transactions on Systems Science and Cybernetics, vol. 4, July 1968, pp. 100-107, 10.1109/TSSC.1968.300136

[17] P. Robert, S.F. Biuk-Aghai, and S. Yain-Wha, “Design of a Recommender System for Mobile Tourism Multimedia Selection,” Proc. 2nd International Conference on Internet Multimedia Services Architecture and Applications, IEEE Press, Dec. 2008, pp. 1-6, doi: 10.1109/IMSAA.2008.4753931.

[18] C.-S. Lee, Y.-C. Chang, and M.-H. Wang, “Ontological recommendation multi-agent for Tainan city travel,” Expert Systems with Applications, vol. 36, April 2009, pp. 6740-6753, doi: 10.1016/j.eswa.2008.08.016.

[19] P. Di Bitonto, F. Di Tria, M. Laterza, T. Roselli, V. Rossano, and F. Tangorra, “Automated generation of itineraries in recommender systems for tourism,” Proc. International Workshop on Web Engineering and Tourism (WEBTOUR), Springer Verlag, July 2010.