Inteligencia Artificial 40(2008), 23-38

INTELIGENCIA ARTIFICIAL

http://erevista.aepia.org/

A Tourism Recommender Agent: from theory to practice

Ana Casali[1], Lluís Godo[2] and Carles Sierra[2]
[1]Depto. de Sistemas e Informática - Depto. de Cs. de la Computación
Facultad de Cs. Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario
Av Pellegrini 250, 2000 Rosario, Argentina.
acasali@fceia.unr.edu.ar
[2]Institut d'Investigació en Intel·ligència Artificial (IIIA), CSIC
Campus de la Universidad Autónoma de Barcelona s/n
08193 Bellaterra, España
{godo, sierra}@iiia.csic.es

Abstract In this paper a multiagent Tourism Recommender System is presented. This system has a multiagent architecture and one of its main agents, the Travel Assistant Agent (T-Agent), is modelled as a graded BDI agent. The graded BDI agent model allows us to specify an agent architecture able to deal with the environment uncertainty and with graded mental attitudes. We focus on the implementational aspects of the multiagent system and specially on the T-Agent development, going from the theoretical agent model to a concrete agent implementation.

Keywords: Graded BDI Agents, Recommender Systems, Tourism, Prolog.

1 Introduction

In the last years the Artificial Intelligence (AI) community has carried out a great deal of work on recommender systems [12, 17]. This kind of systems can help people to find out what they want, especially on the Internet. expect these systems to take personal preferences into account, and to infer and intelligently aggregate opinions and relationships from heterogeneous sources and data. Furthermore, we want such systems to be scalable, open, privacy-protecting and we want to get the recommendations with the least possible work on the users' behalf [13]. Thus, agent technology makes it possible to specify distributed, complex and autonomous recommender systems.

Among recommender systems we particularly concentrate on the tourism domain. The travel and tourism industry is one of the most important and dynamic sectors in Business-to-Consumer (B2C) e-commerce [18]. In this context, recommender applications can be valuable tools supporting for example, information search, decision making, and touristic package assembly. Moreover this is an interesting domain, where user's preferences and restrictions need to be considered. Because of the variety of possible needs, recommender systems can be modelled at different levels of complexity and knowledge-based approaches appear to be very suitable [2].

Also, several architectures have been proposed to provide agents with a formal support. Among them, a well-known intentional formal approach is the BDI architecture proposed by Rao and Georgeff [15]. This model is based on the explicit representation of the agent's beliefs (B), desires (D), and intentions

ISSN: 1137-3601(print); 1988-3064(on-line)

(I). Indeed, this architecture has evolved over time and it has been applied, to some extent, in several of the most significant multiagent applications developed up to now.

We consider that making the BDI architecture more flexible will allow us to design and develop agents potentially capable of having a better performance in uncertain and dynamic environments. Along this research line we have proposed a general model for *graded* BDI Agents (see [3, 4]), specifying an architecture able to deal with the environment uncertainty and with graded mental attitudes. In this agent model, belief degrees represent to what extent the agent believes a formula is true. Degrees of positive or negative desires enable the agent to set different levels of preference or rejection respectively. Intention degrees give also a preference measure but, in this case, modelling the cost/benefit trade off of reaching an agent's goal. Consequently, agents having different kinds of behavior can be modelled on the basis of the representation and interaction of these three attitudes.

In this work we present the development of a tourist recommender as a case study. The system goal is to recommend the best tourist packages on Argentinian destination according to the user's preferences and restrictions. The packages are provided by different tourist operators. This system has been designed using a multiagent architecture and we particularly use the g-BDI model to specify one of its agents, the Travel Assistant Agent (T-Agent). The purpose of this prototype implementation is to show that the g-BDI agent model is useful to develop concrete agents on a real domain.

In previous works we have presented the modelling process of a Travel Recommender Agent using the g-BDI architecture [5] and a general methodology for engineering g-BDI agents [6]. In this paper we describe the most relevant aspects of the tourism recommender system implementation and particularly we focus on the T-Agent implementation. This work is structured as follows. In Section 2 we briefly recall the g-BDI agent model. Then, in Section 3 the multiagent Tourism Recommender System is presented, and in Section 4 the main aspects of the Travel Recommender Agent (T-Agent) implementation are described. From Section 5 to 9 the different contexts, components of the T-Agent, are described. Finally, Section 10 contains some conclusions.

2 Graded BDI agent model

The graded BDI model of agent (g-BDI) allows to specify agent architectures able to deal with the environment uncertainty and with graded mental attitudes. In this sense, belief degrees on formulas are used to represent agent's uncertainty about the available environment information. Degrees of positive or negative desire allow the agent to set different levels of preference or rejection respectively. Intention degrees give also a preference measure but, in this case, modelling the cost/benefit trade-off of reaching an agent's goal. Thus, a higher intention degree toward a goal means that the benefit of reaching it is high, or the cost is low. Then, Agents having different kinds of behavior can be modeled on the basis of the representation and interaction of these three attitudes.

Since modelling different intentional notions by means of several modalities (B, D, I) can be very complex within a single logical formalism, the specification of the g-BDI agent model is done by using the framework of multi-context systems (MCS). Multi-context systems were introduced by Giunchiglia et.al. [8] as a framework allowing different formal (logical) components to be defined and interrelated. A particular MCS specification contains two basic components: contexts and bridge rules, which channel the propagation of consequences among context theories. Thus, a MCS is defined as a group of interconnected units or contexts $\langle \{C_i\}_{i \in I}, \Delta_{br} \rangle$. Each context C_i is defined by a tuple $C_i = \langle L_i, A_i, \Delta_i \rangle$ where L_i, A_i and Δ_i are respectively the language, axioms, and inference rules of the context. Δ_{br} is a set of bridge (inference) rules, that is, rules of inference with premises and conclusions in possibly different contexts, for instance a bridge rule like:

$$\frac{C_1 : \psi, C_2 : \varphi}{C_3 : \theta}$$

means that if formula ψ is deduced in context C_1 and formula φ is deduced in context C_2 then formula θ is added to context C_3. When a theory $T_i \subseteq L_i$ is associated with each unit, the specification of a particular MCS is complete.

The deduction mechanism of these systems is based on two kinds of inference rules, internal rules Δ_i, and bridge rules Δ_{br}, which allow to embed formulae into a context whenever the conditions of the bridge rule are satisfied.

In the g-BDI agent model, we have *mental* contexts to represent beliefs (BC), desires (DC) and intentions (IC). We also consider two *functional* contexts: for Planning (PC) and Communication (CC). Thus, the g-BDI agent model is defined as a MCS of the form

$$A_g = (\{BC, DC, IC, PC, CC\}, \Delta_{br})$$

The overall behavior of the system will depend of the logic representation of each intentional notion in the different contexts and the bridge rules. For a complete specification of the g-BDI agent model, with the logic schema for each context (i.e. the language, axioms and inference rules) and a set of basic bridge rules, the reader is referred to [3].

Figure 1 illustrates the basic g-BDI agent model proposed with the different contexts and the bridge rules relating them. For instance, let us consider bridge rule (4) in Figure 1. As we have already mentioned, intention degrees are meant to represent a kind of trade-off between the benefit and the cost of reaching a goal by a plan execution. After the Intention context has inferred all the intention degrees and finds the best plan α_b that allows to get the maximum degree of intention i_{max} for φ, we also need a rule to establish the agent's interaction with the user as follows:

$$\frac{IC : (I\varphi, i_{max}), PC : bestplan(\varphi, \alpha_b, P, A, c)}{CC : C(recommends(\alpha_b))}$$

Namely, this bridge rule means that if the T-Agent intends φ at degree i_{max}, the maximum degree of all the intentions, then the T-Agent will recommend the plan α_b –*bestplan*– that will allow the agent to reach the most intended goal φ.

In order to represent and reason about graded notions of beliefs, desires and intentions, we use a modal many-valued approach. In particular, we follow the approach developed by Hájek et al. [11, 10] where uncertainty reasoning is dealt with by defining suitable modal theories over suitable many-valued logics. For instance, let us consider a Belief context BC where belief degrees are to be modeled as probabilities. Then, for each classical formula φ, we consider a modal formula $B\varphi$ which is interpreted as "φ is probable". This modal formula $B\varphi$ is then a *fuzzy* formula which may be more or less true, depending on the probability of φ. In particular, we can take as truth-value of $B\varphi$ precisely the probability of φ. Moreover, using a suitable many-valued logic, we can express the governing axioms of probability theory as logical axioms involving modal formulae. Then, the many-valued logic machinery can be used to reason about the modal formulae $B\varphi$, which faithfully respects the uncertainty model chosen to represent the degrees of belief. In this proposal, for all the mental contexts we have chosen the infinitely-valued Łukasiewicz logic expanded with rational truth-constants, but another selection of many-valued logics could be done for each unit as long as they provide the necessary connectives to express the intensional graded properties of the different modalities in each case.

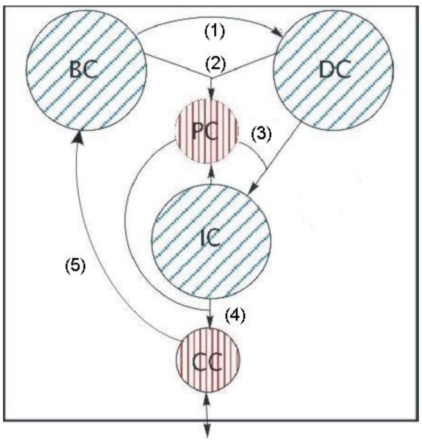

Figure 1: Multi-context model of a graded BDI agent

To set up an adequate axiomatization, for instance, for our probabilistic belief context logic, we need to combine axioms for the classical (two-valued) non-modal formulae, axioms of Łukasiewicz logic for modal formulae, and additional axioms for B-modal formulae according to the probabilistic semantics of the B operator. A similar approach is used to axiomatize, represent and reason under graded attitudes in the other mental contexts. A more detailed formalization of the adequate logics –language, semantics, axioms and rules– for the different contexts can be found in [3].

3 Tourism Recommender System

In this section we present the general architecture of the Tourism Recommender System. For the methodological aspects of the analysis and design stages of this case study the reader is referred to [6].

Inspired in the different components of a tourism chain, in the analysis phase we have identified the following roles: the Provider role (tourist package providers), the Travel Assistant role and Service roles (hotel chains, airlines, etc.). However, in this case study we don't deal with the service roles, we only mention them as necessary collaborators of the Provider role. Other functional roles have been identified as well, like for instance the Interface role, to manage the user interface, and the repository Maintenance role (R-Maintenance), to update and code into the system format the packages sent by the provider roles. In this simplified version of Recommender System, we define two agent's types: the Provider agent and the Travel Assistant Agent. We assign the interface role, the repository maintenance role and the travel assistant role to the Travel Assistant Agent (T-Agent).

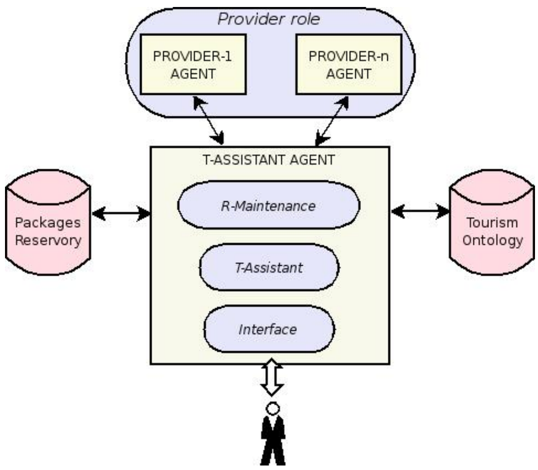

Figure 2: Multiagent architecture of the Tourism Recommender System

As it is natural in the Tourism Chain, different Tourist Operators may collaborate in the Provider role. To represent these different sources of tourist packages, we use two different provider agents (P-Agents). The internal architecture of the Provider agents is not considered in our implementation and for our purposes they are considered only tourist packages suppliers. The multi-agent architecture of the prototype version of the tourism recommender system, composed by the T-Agent and two Provider Agents, together with the main source of information they interact with (the destination ontology and the package repository) is illustrated in Figure 2. This multiagent system is easily scalable to include other providers.

The implementation of the Recommender system was developed using SWI-Prolog [1]. We decided to use prolog because is a suitable language to deal with logical deduction, which is the nature of the inference processes in our agent model. Also, SWI-Prolog is a multi-threaded version of prolog allowing an independent execution of the different contexts (i.e. in different threads). Furthermore, this prolog version is open source, it is well documented and includes a graphic interface tool in native language. A previous implementation of multi-context agents using this software [9] was a starting point for our development.

In our multiagent recommender system the two Tour Operator agents (*P-Agents*) implemented runs in a different thread, so in this way being independent from each other and from the *T-Agent*. When the

[1]http://www.swi-prolog.org

T-Agent requests for information, the *P-Agents* send to *T-Agent* all the current packages they can offer. The communication between agents is by message exchange.

In the real world, each tourist operator may structure the tourist packages in a different way and using its own terminology. To experiment with heterogeneous providers, we use different field names in the plan structure used in each *P-Agent*. Then, these structures are translated into the format the *T-Agent* uses. Thus, a wrapper functionality is needed and it is carried out by the Communication context of the *T-Agent*. In a more complete multiagent recommender architecture a wrapper agent may be included.

4 T-Agent Implementation

The main role of the T-Agent is to provide tourists with recommendations about Argentinian packages. This agent may be suitable modelled as an intentional agent and particularly, by a *g-BDI agent model*. This agent model is specified by a multicontext architecture having mental and functional contexts (i.e, *BC, DC, IC, PC and CC*) and a set of bridge rules (*BR*s).

Thus, the implementation of these interconnected components is needed. Each context has its own inference rules and theories, and they should not interfere. Using a thread for each context allows the desired separation but could considerably slow down the execution. The solution adopted for our implementation was to place only some of these components in different threads. That is the case for the Communication context (CC), the Desire context (DC) and some bridge rules. However, since the Belief (BC), Planner (PC) and Intention (IC) contexts interchange quite a lot of information, for efficiency reasons they run in the same thread. The multithread scheme for the T-Agent in the multiagent system is illustrated in Figure 3, where the yellow boxes represent different threads and the arrows their interactions.

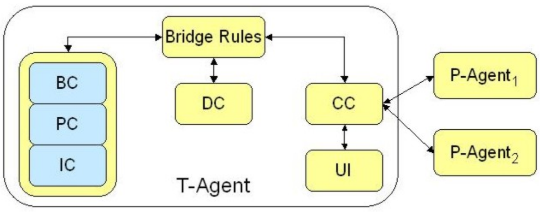

Figure 3: Multithread system scheme

For this multithreaded implementation, following [9], the policy adopted is to have asynchronous threads and asynchronous communication. It means that the messages are sent and received at any time, but they are processed only when the unit is inactive (i.e. when it has finished the internal deductions). Each unit has a message queue that retains the messages until they have been processed. A communication meta-interpreter is devoted to synchronize the ongoing inference process and the arrival of new incoming messages. In our prototype, the exchange of most part of the messages is made during the initial phase. In this phase the *T-Agent* asks the *P-Agents* for the current tourist packages. To answer this request, the *P-Agents* send back a number of messages, each one containing an offered package. The software tool successfully supports this intensive message exchange.

The Communication context (CC) in the *T-Agent* is in charge of receiving these messages, it translates them into a suitable format and it immediately sends them to the Belief context (BC). In this way the agent's knowledge is increased with the package information. In the next sections we described how the main multi-context components of the *T-Agent* are implemented in order to obtain the desired behavior. We begin with the Communication context that provides the agent with a unique and well-defined interface with the environment.

5 Communication Context

The Communication context (CC) constitutes the *T-Agent* interface and makes it possible to encapsulate the agent's internal structure. This context takes care of the sending and receiving of messages to and from other agents in the multiagent society where our graded BDI agents lives. The CC in the *T-Agent* is in charge of interacting with the tourist operators (*P-Agents*) and with the tourist user that is looking for a recommendation.

5.1 Interaction with the P-Agents

Before beginning its recommendation task, the *T-Agent* updates its information about current packages (carrying out its repository maintenance role). This is achieved by the CC through the following steps:

- **Requiring the packages offered:** The CC sends a message to each P-Agent asking them for the current touristic packages they offer.
- **Receiving packages and formatting them:** As the information coming from each *P-Agent* has different format the CC behaves as a wrapper, translating the incoming packages into the T-Agent format.
- **Sending packages:** Once the packages are put under the correct format, they are sent to the Planner context. The recommendation will be based on the information about packages and on domain knowledge.

5.2 User interface

The user interface is in charge of explicitly acquiring the tourist's profile, providing him with the resulting recommendation and receiving the user's feedback. In a first approach this interface was developed using the native language. Later on, to facilitate the access to the recommender system it has been implemented as a Web service[2]. This interface process goes through the following sequential steps:

- **Acquiring user's preferences:** User's preferences are explicitly acquired asking him to fill in a form. The tourist can specify his preferences (positive desires) and restrictions (negative desires), assigning them a natural number from 1 (minimum) to 10 (maximum) to represent the level of preference or rejection in the selected item. Furthermore, he can choose different parameters: the flexibility of restrictions (by specifying them as flexible or hard), the expected frequency of the selected activity (high or low) and the priority criterion to rank-order the recommended packages (preference satisfaction or minimum cost). An example of a tourist's preferences specification using this interface is shown in Figure 4.

  Once the user finishes his specification, the CC sends all the acquired information to the Desire context (DC).
- **Showing the resulting recommendation:** As result of the *T-Agent* deliberation process, the CC receives from the Intention context (IC) a ranking of feasible packages that satisfies some of the tourist preferences. The ranking is ordered also taking into account the priority criterion he has selected (e.g. preference satisfaction). The first nine packages of this ranking are shown to the tourist and the user can visualize the information about them by opening suitable files (e.g *pdf* files).
- **Receiving user's feedback:** After analyzing the ranking of the recommended packages the user can express through the interface his opinion about the recommendation. For this task, the options considered are the following:

  - *Correct:* when the user is satisfied with the ranking obtained.

  - *Different order:* when the recommended packages are fine for the user, but they are ranked in a different order than the user's own order. In such a case, the user is able to introduce the three best packages in the right order.

  - *Incorrect:* when the user is not satisfied with the given recommendation. Then, the interface enables him to introduce a (textual) comment about his opinion.

[2]http://musje.iiia.csic/eric/

All the information resulting from the previous steps (i.e., the tourist's preferences, the recommendation given and the user's feedback) is stored to evaluate the system performance.

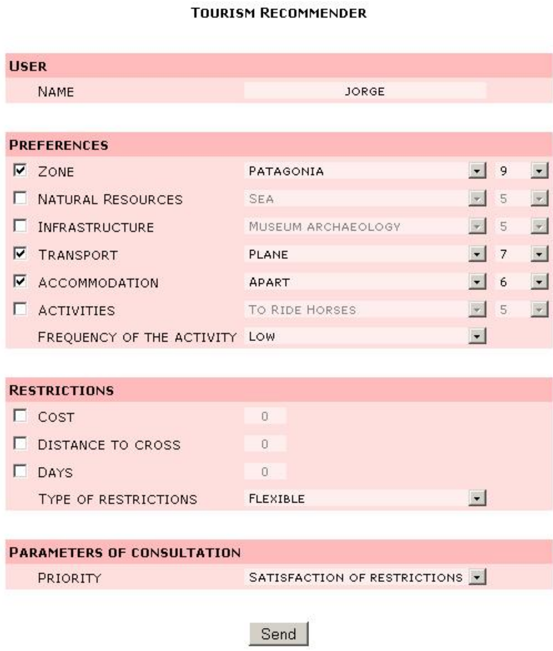

Figure 4: User interface - tourist's preferences

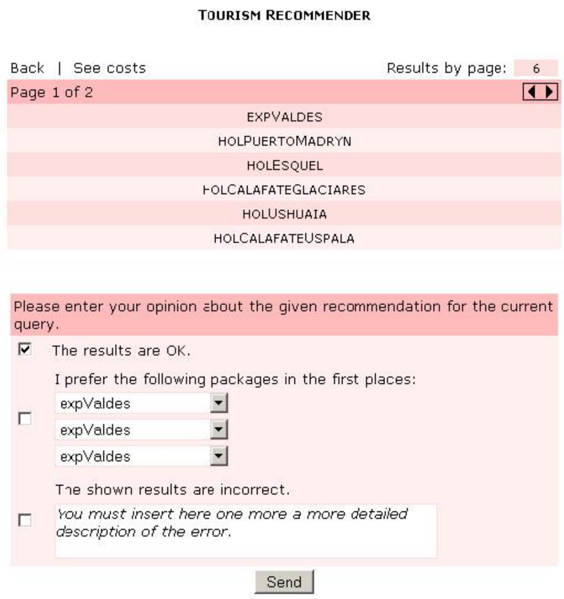

Figure 5: User interface - package recommendation and user feedback

6 Desire Context

As the *T-Agent* is a kind of *personal agent*, its overall desire is to maximize the satisfaction of tourist's preferences. Then, in this context the different tourist's graded preferences and restrictions are respectively represented as positive and negative desires.

On the one hand, the negative desires are used as strong constraints, namely, the T-Agent will discard those packages not satisfying them. On the other hand, from the elementary positive desires all their conjunctions are built as combined desires. The *T-Agent* will use all these desires as pro-active elements, looking for different packages that will allow tourists to satisfy any of them. Then, the theory in this context is constituted by positive and negative desires (represented by *desU* formulae).

The user's preferences are acquired in the CC by the user interface and are introduced in a list to the DC. In the following items we describe how the positive desires are built (negative desires are treated in a similar way):

1. Elementary desires: The DC takes each desire from the list received from the CC, normalizes its degree (i.e. mapping it from $\{1, ..., 10\}$ into $(0, 1]$) and adds it to the context formulae. The structure of these formulae is: $desU(y(Desire, Value), NormalizedDegree)$

The relation $y(Desire, Value)$ represents a positive desire where the first argument is the class of desire (e.g. transport "transporte") and the second is the value the tourist has chosen (e.g. plane "avion"), followed by the normalized degree. For instance, the formal expression in the DC of the elementary desires corresponding to the tourist's preferences specified in Figure 4 are:

```
desU(y(zona, patagonia), 0.9)
desU(y(transporte, avion), 0.7)
desU(y(comodidad, apart), 0.6)
```

2. Similar Desires: In the special case of some types of desires (e.g. those about accommodation, transportation, natural resources), we consider that a tourist can also be satisfied (to some lower degree) with a package that offers similar facilities to the ones originally specified. Also, in the particular case of accommodation, we assume tourists will be also satisfied if they receive a better accommodation than the selected one.

Therefore, the Belief context contains instances of the domain dependent relations "*to be similar to*" and "*to be better than*" which are used to expand the set of possible values that would satisfy the user's preferences. Indeed, these instances are used to generate new desires into the DC by means of rules like "If the T-Agent has a positive desire X at least to a degree d and he believes that X is similar to Y at least to a degree s, then he also desires Y at least to a degree $d' = d \cdot s$" and "If the T-Agent has a preference about an accommodation X at least to the degree d and Z is an accommodation *bether than* X, then he also desires Z at least to degree d". These rules are formalized by suitable bridge rules.

3. Combined Desires: After the elementary desires are added to the context, all possible conjunctions are built. The conjunctions are attached a degree greater or equal than the maximum of elementary degrees, and hence is computed in accordance with the guaranteed possibility model (see [1]). Namely, if the DC contains formulae like $desU(y(D_1), G_1)$ and $desU(y(D_2), G_2)$, a combined desire $desU(yLst([D_1, D_2]), G)$ is also added. In our prototype G is computed by the following function:

```
calcularGraduacion(G1, G2, G) :-  G is G1 + ((1 - G1) * G2)
```

As way of example, we show the code of one of the conjunctive combinations built from the elementary desires given above:

```
desU(yLst([(zona, patagonia), (transporte, avion)]), 0.97)
desU(yLst([(zona, patagonia), (comodidad, apart)]), 0.96)
desU(yLst([(transporte, avion), (comodidad, apart)]), 0.88)
desU(yLst([(zona, patagonia), (transporte, avion), (comodidad, apart)]), 0.988)
```

Both, the positive and negative desires are passed by means of a bridge rule to the Planner context where the feasible packages that satisfy the tourist's preferences are selected.

7 Belief Context

In the Belief context the *T-Agent* represents all the necessary domain knowledge about tourism and in particular about tourism in Argentina: tourist packages, information about destinations and some special domain-dependent relations. Also, in this context the belief degrees of achieving the different desires by executing alternative plans, are computed. We describe next the representation of these kinds of information.

7.1 Tourist packages

One of the most significant data structures in our system is the package structure. After analyzing nearly forty Argentinian packages selected from the Internet, a general structure which is capable of representing the information available in most of them has been adopted. Each package is represented as a list containing an identifier, a tour provider, the cost and a travel-stay sequence as it can be seen in the following structure:

Package	::=	(Id, Provider, Cost, Trip)
Trip	::=	[(Travel, Stay)]
Travel	::=	(Transport, Road)
Stay	::=	(Destination, Days, Accommodation,[Activity])
Activity	::=	activity(Sport, Hours)| excursion(Resource, Hours, Name)

For example, the prolog representation of the package named *holCalafatePatagonia* is presented below:

```
paq(id(holCalafatePatagonia), costo(1900),
  [(viaje(avion, aire), estadia(calafate, dias(3), comodidad(apart),
    actividades([
      [act(cityTour), horas(4)],
      [exc(parqueNacional), horas(8), peritoMoreno]]))),
  (viaje(avion, aire), estadia(ushuaia, dias(4), comodidad(hotel3),
    actividades([
      [act(cityTour), horas(1.5)],
      [exc(museo), horas(1), finDelMundo],
      [exc(historia), horas(1), carcelReincidentes],
      [exc(parqueNacional),horas(2),tierraDelFuego],
      [exc(lago), horas(1), escondido],
      [exc(lago), horas(1), fagnano]]))),
   (viaje(avion, aire), null)])
```

Notice that in the last element of the travel list the stay is null, representing the return travel.

7.2 Destination ontology

The *T-Agent* needs to have information about the country and the different possibilities its places bring about. Usually the packages have little information about the destinations and the resources available in them. This domain knowledge is complementary to the package information and very important to infer whether a trip including certain destinations can satisfy some tourist preferences (e.g. natural resources). To structure the knowledge about Argentinian tourism, we analyzed different tourism ontologies and most of them were focused on destinations (see e.g. [14]) including the resources they have, the activities they offer, etc. Inspired in them, the following features were extracted for defining the destination ontology in our prototype:

Destination	::=	(Name, Coordinates, Zone, [NaturalResource], [ArtificialResource], [Activity])
Coordinates	::=	(X, Y)
NaturalResource	::=	Resource
ArtificialResource	::=	Resource
Resource	::=	(KindOfResource, Name)

The information of almost fifty Argentinian destinations (i.e. all the destinations related to the packages used) has been introduced to fill in this ontology. This information has been extracted from official web-sites.

We use as *coordinates* the geographic coordinates provided by the Instituto Geográfico Militar de la República Argentina[3]. The geographical *zone* assigned to each destination corresponds to the partition of Argentinian provinces into zones proposed by the Secretaría de Turismo de la República Argentina[4]. An example of the destination structure for the *Ushuaia* city is presented below:

```
localidad(nombre(ushuaia), provincia(tierraDelFuego),
          gps(54.80, 68.31), zona(patagonia),
          naturaleza([(parqueNacional,tierraDelFuego),
                      (canal,beagle), (bahia,lapatala), (lago,roca),
                      (lago,fagnano), (lago,elEscondido),
                      (laguna,negra), (rio,grande)]),
          infraestructura([(museo,finDelMundo),
                      (museo,regional),(museo,acatushun),
                      (historia,presidio),(ingenieria,trenFinDelMundo)],
          actividades([avistajeFauna,esqui,navegacion,pesca,trekking]))
```

The ontology used in this prototype has been directly coded in a prolog file, but it is possible for the *T-Agent* to receive an ontology built using an ontology editor (via XML code).

7.3 Special relations in the domain

As already mentioned in Section 6, to increase the domain knowledge of the *T-Agent*, some special relational predicates have been included in the BC language. This allows to encode knowledge about related concepts that makes it possible for the *T-Agent* to expand the search to other terms related to the ones expressed in the tourist's preferences and are used in the selection of the best packages for the tourist. In this implementation we have considered important to include two kinds of relations:

1. "to be similar to" relation: The BC includes a set of instances of the *similar* predicate on pairs of synonymous or similar concepts according to the tourism domain, composing a so-called similarity dictionary. As the *T-Agent* belief context deals with graded information, these instances may include a degree $g \in [0, 1]$ expressing a sort of semantical distance between terms. The formulae in this dictionary are structured as: $belU(similar(term_1, term_2), g)$.

For instance, we show a fragment of this similarity dictionary:

```
% accommodation category
belU(similar(apart, hotel3), 0.75)
belU(similar(camping, campamento), 1.0)

% nature category
belU(similar(lago, embalse), 0.7)
belU(similar(montaa, cerro), 0.8)

% transport category
belU(similar(bus, colectivo), 1.0)
belU(similar(bus, trafic), 0.9)
```

2. *"to be better than"* relation: For the accommodation concepts a "better than" relation has been added to express whether an accommodation is better than another one. This transitive relation allows the *T-Agent* to expand the search of the packages that satisfy the user's preferences, to those that include accommodations better than the selected one.

[3] http://www.geoargentina.com.ar
[4] http://www.turismo.gov.ar

7.4 Beliefs on desires fulfillment

The T-Agent needs to compute in which degree a particular desire is believed to be fulfilled after a plan execution. This means to compute the degree r of the formula $B([\alpha_P]D)$, where α_P is a tourism package and D is a desire (elementary or combined). This belief degree r is necessary for the agent to estimate the *expected satisfaction* $E(D, \alpha)$ of a desire D by a plan α_P, as we will see later in Section 9.1 where this expectation is estimated by the value $E = r \cdot d$, where r is the degree of $B([\alpha_P]D)$ and d the degree of desire D. Notice that, following the model presented in [3], the truth degree of $B([\alpha_P]D)$ is considered as the probability of making D true after following plan α_P. In the following we describe how such a probability is estimated from according to the different types of desire types and plans.

Basically, a tourist plan may be considered as a temporal sequence of subplans and the global satisfaction depends on how user's preferences are expected to be satisfied through each stage of the plan trip. As it was presented above, the packages are structured as:

$$Package ::= (Id, Provider, Cost, Trip)$$

where $Trip$ is a travel-stay sequence $[(Travel_i, Stay_i)]$, $i = 1, \ldots, n$. In our approach each pair $(Travel_i, Stay_i)$ is considered as an atomic package stage (sub-plan), amenable to satisfy some desires. Packages α_P are therefore modelled as composed plans, $\alpha_P = \alpha_1; \ldots; \alpha_n$, alternating travel and stay sub-plans.

Then, the expected satisfaction $E(D, \alpha_P)$ of a desire $D = D_1 \wedge ... \wedge D_n$ through the execution of the plan α_P is computed in our model (see Section 9.1) from the expected satisfactions values $E_{ij} = E(D_j, \alpha_i)$ of the elementary desires D_j by the execution of the elementary sub-plans α_i. In turn, to compute each of the E_{ij}'s, the belief degree r_{ij} of achieving the desire D_j through the subplan α_i execution (corresponding to the degree of the formula $B([\alpha_i]D_j)$) is needed. This is described next.

7.4.1 The case of elementary desires

For evaluating the belief degree r in which a package α_P will fulfill an elementary desire D, the agent focuses on either the travel stages or the stay stages in the α_P depending on the kind of desire D specifies. For example, if D is about transport then, only the travel stages in α_P are considered, while if D is related to a natural resource then only the stay stages of α_P are considered. In any case, the belief degree is computed using a set of rules that depend on the kind of desire and on the user's priority criterion.

For example, the BC has a rule setting that "*if the desire D is about accommodation of category c and $stay_i$ of package α_P (i.e. the subplan α_i) offers an accommodation better or equal than c, then the belief degree of fulfilling the desire D by subplan α_i is $r_i = 1$*". In other words, in case D and α_i satisfy these conditions, such a rule would create the formula $(B([\alpha_i]D, 1)$ in the agent's BC theory.

When the tourist's desire D is related to a destination resource (e.g. natural resources, activity) the belief degree of fulfilling it by a plan execution has another interesting characteristic. We have noticed that packages have usually limited information about destinations and their resources. Thus, for belief estimation purposes, besides the package information, the *T-Agent* may need further knowledge about destinations. In our prototype this information is structured in a destination ontology. This amounts to extend the computation of the degree r_i of $B([\alpha_i]D)$ to a *package-destinations* cross inference to assess the fulfillment of the tourist's selected preference in a certain destination, using not only the package supplied information but also the available information about the destination. Therefore, the strategy which is followed is, for each package stage α_i, to evaluate the probability of α_i providing a certain resource D both from explicit information offered in the package (r_{Pi}) and from information inferred from the destination ontology (r_{Oi}). Finally, the T-Agent takes as degree r_i the maximum of both estimations, i.e. $r_i = \max\{r_{Pi}, r_{Oi}\}$.

7.4.2 Combined desires

The DC theory includes conjunction of positive desires. To evaluate the probability of fulfilling the conjunction of elementary desires (e.g. $(D_1 \wedge D_2)$) by the execution of a package α, we assumed that, as random variables, the elementary desires are stochastically independent. Then, from the degrees r_1 and r_2 corresponding to the elementary desires D_1 and D_2 respectively, we can compute the belief degree in achieving their conjunction by executing the plan α using the following rule:

$$\frac{(B[\alpha]D_1, r_1), (B[\alpha]D_2, r_2)}{(B[\alpha](D_1 \wedge D_2), r_1 \cdot r_2)}$$

For example, consider the T-Agent has the following combined desire D specified in the DC:

```
desU(yLst([(zone,patagonia),(activity,rafting)]), 0.8)
```

the agent has also in her belief context BC the belief degrees of obtaining the elementary desires by a package α, which are respectively $(B[\alpha]\mathit{patagonia}, 1.0)$ and $(B[\alpha]rafting, 0.7)$. Following the rule given above, the T-Agent computes that the belief degree for the combined desire is: $(B[\alpha](\mathit{patagonia} \wedge \mathit{rafting}), 0.7)$.

8 Planner Context

The Planner Conext (PC) is fundamental for the *T-Agent* implementation. The PC unit is assumed to contain a set of available plans, coded as instances of the predicate *planner* with *paq* formulae (see below). The Planner context is responsible for looking among them for *feasible packages*. By *feasible package* we mean a package that fulfills, to some degree, one of the positive desires (elementary or combined) and avoids, as post-condition, the satisfaction of the agent's negative desires above to a given threshold $UmbralN$. The set of feasible plans is determined within this context using an appropriate searching method that takes into account information injected by bridge rules from the BC and DC units, including positive and negative desires, information about packages (including their cost), the agent's beliefs about package destinations and the estimation of the agent's desires fulfillment by the different plan executions. The following forward rule encodes this in the Planner context.

```
des(yLst(DeseosP), _), des(nLst(DeseosN), UmbralN),
planner(paq(IdPaq, Proveedor, Costo, _Recorrido)),
bel(contiene(IdPaq, DeseosP), R),
bel(not(contiene(IdPaq, DeseosN)), UmbralN),
bel(costoNormalizado(Costo, CN), 1)
--:
planner(paqSi(IdPaq, Proveedor, CN, DeseosP), R)
```

For each feasible package, with identifier $IdPaq$, this rule cretaes into the PC theory an instance of the *planner* predicate with a *paqSi* formula with identifier $IdPaq$. Note that in each instance of a feasible package, its normalized cost ($CN \in [0, 1]$) is used instead of the actual cost.

After the PC has identified the set of feasible packages, they are passed to the Intention context, which is in charge of ranking of these packages according to the user's preferences.

9 Intention Context

In order to rank the feasible packages to be offered to the user, the Intention context IC of the *T-Agent* is in charge of estimating the intention degree for each feasible package as a trade off between the benefit (expected satisfaction) and the cost of reaching the user's desires through that package. Thus, first, this context estimates the expected satisfaction $E(D, \alpha)$ of a tourist's desire D assuming she selects a package α. Second, using a suitable bridge rule, it computes the intention degree (the truth degree of the formula $I_\alpha D$) towards the desire D by executing a tourist package α using a function that combines the expected satisfaction $E(D, \alpha)$ and the normalized package cost CN. In the following Subsections we give some insights of how this estimations are implemented in the *T-Agent*.

9.1 Estimating the expected satisfaction of desires

For estimating the expected satisfaction of a tourist's desire D assuming she selects a package α_P, the underlying idea is to consider that each plan α makes D a binary random variable, with a probability distribution

$$Prob_\alpha(D = true) = r,\ Prob_\alpha(D = false) = 1 - r.$$

Now, if d is the user's positive desire degree for making D true, and assuming the positive desire of making D false is 0, then the user's expected satisfaction degree by achieving D through the plan α is clearly $E(D, \alpha) = r \cdot d + (1 - r) \cdot 0 = r \cdot d$.

Therefore, to estimate the value $E(D, \alpha)$ one needs to estimate both the probability r of achieving D by α and the (positive) desire degree d for D. These values can be directly obtained from the BC and DC contexts respectively for atomic package components and elementary desires. Indeed, if the plan α consists of a sequence of travel-stay components $\alpha = \alpha_1; \ldots ; \alpha_k$ and the desire D is a conjunction of elementary desires $D = D_1 \wedge \ldots D_n$, then the agent contains in her contexts:

- **BC:** instances $(B[\alpha_i]D_j, r_{ij})$, for $i = 1, k;\ j = 1, n$ and
- **DC:** instances (D^+D_j, d_j), for $j = 1, n$, and (D^+D, d) where $d = 1 - \Pi_{j=1,n}(1 - d_j)$ (see DC context in Section 6)

and hence, for each i, j we can estimate $E(D_j, \alpha_i)$ with the values $r_{ij} \cdot d_j$. Then, in order to come up with a estimated value for $E(D, \alpha)$, we follow the following steps:

(i) $E(D_j, \alpha)$ is computed for each elementary desire D_j, $j = 1, n$ and then

(ii) $E(D, \alpha)$ is estimated for the combined desire D.

Next we give some insights of how the estimation is done first, for elementary and then, for combined positive desires.

(i) Elementary desires. The expected satisfaction $E(D_j, \alpha)$ of an elementary desire D_j through the execution of the plan α may be computed from the expected satisfactions $E_{ij} = E(D_j, \alpha_i)$ by the execution of its sub-plans α_i, using an appropriate aggregation operator $\oplus$, i.e.:

$$E(D_j, \alpha) = \oplus(E(D_j, \alpha_1), \ldots, E(D_j, \alpha_k))$$

For computing each E_{ij}, the agent uses the probabilities r_{ij} and desire degrees d_j injected from the BC and the DC respectively by a bridge rule. A set of rules play the aggregation role to obtain the expected satisfaction. These rules depend on the kind of desire.

The underlying idea to compute the expected satisfaction of a user's desire D_j (with degree d_j) is to consider the proportion (in terms of duration) of the package components where D_j is expected to be satisfied with respect to the whole trip proposed by the package. Furthermore, the estimation of how much a component of a package α_P (a sub-plan α_i) satisfies a preference may be also graded and is computed depending on what the offer of the sub-plan is, as it is explained next: In our approach we consider for this estimation the relationship of the tourist's desire D_j with the actually proposed in the package D'_j:

- if D'_j is exactly D_j or "better than" than D_j, the expected satisfaction of D_j by the package component α_i is taken as $E(D_j, \alpha_i) = r_{ij} \cdot d_j$, where r_{ij} is the belief degree of $B([\alpha_i]D_j)$.
- if D'_j is similar to D_j to the degree s (see previous subsection 6), the expected satisfaction of D_j by the package component α_i is taken as $E(D_j, \alpha_i) = r'_{ij} \cdot d \cdot s$, where r'_{ij} is the belief degree of $B([\alpha_i]D'_j)$.

Then, if the package α is composed by different stages, i.e. $\alpha = \alpha_1; \ldots ; \alpha_n$, the general way of computing the expected satisfaction $E(D_j, \alpha)$ of the desire D_j by the package α, is defined as

$$E(D_j, \alpha) = \frac{\sum_i E(D_j, \alpha_i) \cdot Time_{\alpha_i}}{TotalTime},$$

where $Time_{\alpha_i}$ and $TotalTime$ are computed according to the kind of desire D_j.

For instance, if D_j is about accommodation, $Time_{\alpha_i}$ denotes the duration (in days) of the stay α_i and $TotalTime$ is the total duration of the trip. On the other hand, in the case of D being an activity, if the user's preferences specify to do the activity with *high frequency* (see Section 5.2), $Time_{\alpha_i}$ is the duration (in hours) of the activity programmed by α_i and $TotalTime$ is an estimation of the total number of hours the activity could take during the whole trip.

Example: Let us assume a tourist has an accommodation preference of *Apart-Hotel* represented by the desire D:

$$desU(y(comodidad, apart), 0.7).$$

Using the similarity relation between this type of accommodation and a 3 star hotel represented by the instance

$$belU(similar(apart, hotel3), 0.75)$$

the T-Agent considers the tourist will also be satisfied to some degree if he is accommodated in a *3-star Hotel*, i.e. he is assumed to also have the desire D' represented by:

$$desU(y(comodidad, hotel3*),\ \ 0.7 \cdot 0.75)$$

The *T-Agent* evaluates the expected satisfaction of accommodation through the package *holCalafatePatagonia*. This package has two stay components: 3 days in Calafate with *Apart-Hotel* accommodation and 4 days in Ushuaia with a *3-star Hotel* accommodation (see for details subsection 7.1). Considering that in the BC were computed $r_1 = 1$ and $r'_2 = 1$ then, the expected satisfaction of each one of these two package components are respectively as follows:

- $E_1 \ = \ r_1 \cdot d \ = 1 \cdot 0.7 \ \ = 0.7$ and
- $E_2 \ = \ r'_2 \cdot d' = 1 \cdot 0.525 = 0.525$

Finally, the expected satisfaction of the accommodation desire by the package *holCalafatePatagonia* is computed as the average of the expected satisfactions E_1 and E_2, considering the duration in days of each stay, i.e.:

$$E \ = \frac{(E_1 \cdot 3) + (E_2 \cdot 4)}{7} = 0.6$$

(ii) Combined desires. If the agent has a combined desire $D = D_1 \wedge ... \wedge D_n$ with degree d (i.e. the formula $(D^+ D, d)$ is in the DC) and she has selected a plan α, for each desire D_j the agent can compute the expected satisfaction $E(D_j, \alpha)$ as it was shown in item (i). Then, the agent can estimate the probabilities:

- $Prob_\alpha(D_j = true) = E(D_j, \alpha)/d_j$, for each desire D_j and
- $Prob_\alpha(D = true) = \Pi_{i=1,n} Prob_\alpha(D_j)$.

Finally, she can estimate the expected satisfaction of the combined desire as follows:

- $E(D, \alpha) = Prob_\alpha(D) \cdot d$.

9.2 Computing the intention degrees

After the T-Agent has estimated the tourist's expected satisfaction $E(D, \alpha)$ for each desire D and feasible package α, the corresponding intention degrees are computed in the IC. Namely, for each desire D and a feasible package α to achieve D, the following bridge rule is used to infer a degree for the formula $I_\alpha(D)$:

$$\frac{PC : fplan(D, \alpha, P, A, c), IC : (E(D, \alpha), e)}{IC : (I_\alpha D, f(e, c))}$$

This degree is computed, by means of a suitable function f combining the expected satisfaction of the desire through the plan execution (e) and the cost of the plan (c). Different functions can model different individual agent behaviors. In the *T-Agent* this function is defined as a weighted average:

$$f(ES, CN) = \frac{w_{es} \cdot e + w_{cost} \cdot (1 - c)}{w_{es} + w_{cost}}$$

where w_{es} and w_{cost} are weights which are set by the *T-Agent* according to the priority criterion selected by the user (minimum cost, preference satisfaction).

After the bridge rule has been applied to all the feasible plans, the IC has in its theory a set of graded intention formulae. The intention degrees are used by the *T-Agent* to rank the feasible packages that communicates to the CC. We opted to select the first nine packages to recommend the tourist.

Finally, the selected packages are passed to the CC unit and then, through the user interface the *T-Agent* outputs to the user the ranking as the system recommendation. For instance, Figure 4 shows a tourist's preference selection and the resulting recommended ranking is shown in Figure 5. After analyzing the recommended packages, the user is prompted by the system to provide his feedback.

10 Conclusions

A prototype of multiagent Tourism Recommender system has been implemented. A multiagent approach is suitable for this kind of systems dealing with heterogeneous and distributed information. Particularly we used a g-BDI architecture for modelling the T-Agent, showing in this way that this model is useful to develop concrete agents in real domains.

We remark that the graded model of information representation and reasoning in the g-BDI agent has many advantages for this implementation. First, this model enables an expressive representation of the domain knowledge (agent beliefs), the user's preferences (desires) and the resulting intentions. Second, the implemented approach allows the agent to expand the retrieval of feasible packages using similarity relations and domain knowledge, not explicitly included in the package information. Also, the treatment of many-valued information makes it possible to compute in a graded way the expected satisfaction of the different tourist's preferences by the execution of several packages. Finally, the intention degree of a plan toward a desire satisfaction may be computed as a function of different factors (e.g. expected satisfaction, cost). A valuable characterictic in the g-BDI agent model is than we can obtain different agent behaviors by defining different functions for intention computation.

A first experimentation of this prototype has been carried out with promising results (see [7] for a preliminary report). Considering 52 queries, 73% of the user's opinions were satisfactory (namely 40.4% with *correct order* and 32.7% with *different order* as user feedbacks). Furthermore, we have performed some experimentations using this recommender agent with the aim of proving different properties of the g-BDI model of agents. On the one hand, we have performed a sensitivity analysis to show how the g-BDI agent model can be tuned to have different behaviors by modifying some of its component elements. On the other hand, we have also done some experiments in order to compare the performance of recommender agents using the g-BDI model with agents without graded attitudes.

Acknowledgments

Ana Casali thanks her students for their help on the recommender system implementation. Lluís Godo and Carles Sierra acknowledge partial support of the Spanish project *Agreement Technologies* (CONSOLIDER CSD2007-0022, INGENIO 2010).

References

[1] Benferhat S., Dubois D., Kaci S. and Prade, H. Bipolar representation and fusion of preferences in the possilistic Logic framework. *In Proceedings of the 8th International Conference on Principle of Knowledge Representation and Reasoning (KR02)*, pp. 421-448, 2002.

[2] Burke R. Knowledge-based recommender systems. *Encyclopedia of Library and Information Systems*, 69, 2000.

[3] Casali A., Godo L. and Sierra C. Graded BDI Models For Agent Architectures. Leite J. and Torroni P. (Eds.) *CLIMA V, LNAI* 3487, pp. 126-143, Springer-Verlag, Berling Heidelberg, 2005.

[4] Casali A., Godo L., Sierra C. Modelos BDI graduados para arquitecturas de agente. *Inteligencia Artificial* 26, pp. 67-75, 2005.

[5] Casali A., Godo L. and Sierra C. Modelling Travel Assistant Agents: a graded BDI Approach. Proceedings of the IFIP-AI, WCC, Volume 217, *Artificial Intelligence in Theory and Practice*, Max Bramer (ed.), Springer, pp. 415-424, 2006.

[6] Casali A., Godo L. and Sierra C. A Methodology to Engineering Graded BDI Agents. Proccedings WASI, *CACIC 2006*, 12 pag . Potrero de Funes, Argentina, 2006.

[7] Casali A., Godo L., Sierra C. Validation and Experimentation of a Tourism Recommender Agent based on a Graded BDI Model. In: *ArtificiaL Intelligence Research and Developement*, T. Alsinet et al. (Eds.), Series: Frontiers in Artificial Inteligence and Applications 184, IOS Press, pp. 41-50, 2008.

[8] Ghidini C. and Giunchiglia F. Local Model Semantics, or Contextual Reasoning = Locality + Compatibility, *Artificial Intelligence*, 127(2):221-259, 2001.

[9] Giovannucci Andrea. Towards Multi-Context Based Agent Implementation, *Technical Report IIIA-CSIC*, 2004.

[10] Godo L., Esteva F. and Hájek P. Reasoning about probabilities using fuzzy logic. *Neural Network World*, 10:811-824, 2000.

[11] Hájek P., Godo L. and Esteva F. Fuzzy logic and probabitity, In Proc. of the *International Conference on Uncertainty in Artificial Intelligence (UAI 95)*, pp. 237-244, Montreal, Canada, 1995.

[12] Montaner M., López B., de la Rosa J.L. A Taxonomy of Recommender Agents on the Internet, *Artificial Intelligence Review*, Kluwer Academic Publishers. Volume 19, Issue 4, pp. 285-330. June, 2003.

[13] Niinivaara O. Agent-Based Recommender Systems. *Technical Report, University of Helsinki*, Dept. of CS, 2004.

[14] Prantner K. E-Tourism Working Draft, DERI Innsbruck, October 2004 (www.deri.org)

[15] Rao A. and Georgeff M. BDI Agents from Theory to Practice, *Technical Note 56, AAII*, April 1995.

[16] Ricci F. Travel recommender Systems, in *IEEE Intelligent Systems*, November/December 2002, pp. 55-57, 2002.

[17] Terveen L.G. and Hill W. Beyond Recommender Systems: Helping People Help Each Other. In Carroll, J. (ed.), *HCI in the New Millennium.* Addison Wesley, 2001.

[18] Werthner H. Intelligent Systems in Travel and Tourism, in Proceeding of the 18th *International Joint Conference on Artificial Intelligence, IJCAI-03*, pp. 1620-1625, Acapulco, Mexico, 2003.