A Tourism Recommender Agent: From theory to practice.

Ana Casali and Armando Von Furth
Depto. de Sistemas e Informática FCEIA - UNR
Av Pellegrini 250, 2000 Rosario, Argentina.
acasali@fceia.unr.edu.ar

and

Lluís Godo and Carles Sierra
Institut d'Investigació en Intel·ligència Artificial (IIIA) - CSIC
Campus UAB, 08193 Bellaterra, Catalunya, 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 to specify an agent's 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 theoric agent model to the concrete agent implementation.

Keywords: Recommender Systems, Graded BDI Agents, 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]. This kind of systems can help people to find out what they want, especially on the Internet. Agent technology becomes invaluable by appreciating the facts that we 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 the systems to be scalable, open, privacy-protecting and we want to get the recommendations with the least possible work on users' behalf [7]. From the application of this technology results a community of distributed, complex and autonomous recommender agents.

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. In this context, recommender applications can be valuable tools supporting, for example, information search, decision making, and package assembly. Moreover this is an interesting domain, where diverse user's preferences and restrictions can be considered. Because of this variety, the recommendation systems can be treated in different levels of complexity and the knowledge-based approaches are very suitable [1].

Also, several architectures have been proposed to give agents a formal support. Among them, a well-known intentional formal approach is the BDI architecture proposed by Rao and Georgeff [9]. This model is based on the explicit representation of the agent's beliefs (B), its desires (D), and its intentions (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 [2]), 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. Then, 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 user's preferences and restrictions. The packages are provided by different tourist operators. This system is 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 real domain.

In previous works we have presented the modelling process of a Travel Recommender Agent using the g-BDI architecture [3] and a general methodology for engineering g-BDI agents [**?**]. 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 paper is structured as follows, in Section 2 we briefly introduce the g-BDI agent model. Then, in Section 3 the multiagent Tourism Recommender System is presented and in the next Section 4, the principal aspects of the T-Agent implementation are described. Finally, in Section 5 some conclusions are exposed.

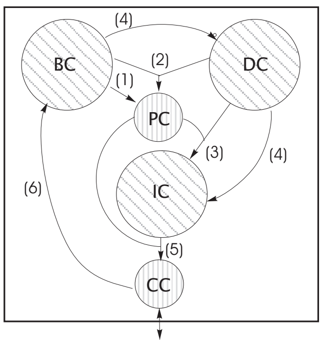
2. GRADED BDI AGENT MODEL

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 represent to what extent the agent believes a formula is true. 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 towards 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.

The specification of the g-BDI agent model is based on Multi-context systems (MCS) [**?**] to allow different formal (logic) components to be defined and interrelated. The MCS specification contains two basic components: units or contexts and bridge rules, which channel the propagation of consequences among theories. Thus, a MCS is defined as a group of interconnected units: $\left\langle \{C_i\}_{i\in I}, \Delta_{br} \right\rangle$, where each context $C_i \in \{C_i\}_{i\in I}$ is the tuple $C_i = \langle L_i, A_i, \Delta_i \rangle$ where L_i, A_i and Δ_i are the language, axioms, and inference rules respectively. When a theory $T_i \subseteq L_i$ is associated with each unit, the specification of a particular MCS is complete. Δ_{br} can be understood as rules of inference with premises and conclusions in different contexts.

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



Bridge rule (3)

$$\frac{DC : (D^{+}\varphi, d), PC : plan(\varphi, \alpha, P, A, c, r)}{IC : (I\varphi, f(d, c, r))}$$

Figura 1: Multi-context model of a graded BDI agent and a bridge rule example.

intentions (IC). We also consider two *functional* contexts: for Planning (PC) and Communication (CC). Thus, the g-BDI agent model is defined as the MCS: $A_g = (\{BC, DC, IC, PC, CC\}, \Delta_{br})$.

The overall behavior of the system will depend of the logic representation of each intentional notion in the different contexts and the bridge rules. The 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 can be seen in [2]. The left side of Figure 1 illustrates the g-BDI agent model proposed with the different contexts and the bridge rules relating them.

The intention degree trades off the benefit and the cost of reaching a goal, by a plan execution. One of the bridge rules included in the agent model (see Figure 1 right side) infers the degree of intention towards a goal φ ($I\varphi$) for each plan α that allows to achieve the goal. This value is deduced from the degree of desire $D^{+}\varphi$ (d), the expected satisfaction of the desire through the plan execution (r) and the cost (c) of the plan. This degree is calculated by a suitable function f.

In order to represent and reason about graded notions of beliefs, desires and intentions, we use a modal many-valued approach where uncertainty reasoning is dealt with by defining suitable modal theories over suitable many-valued logics. For instance, let us consider a Belief context 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 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 respect the uncertainty model chosen to represent the degrees of belief. In this proposal, for the mental contexts we choose the infinite-valued Łukasiewicz logic but another selection of many-valued logics may be done for each unit, according to the measure modeled in each case.

To set up an adequate axiomatization for our belief context logic we need to combine axioms for the crisp formulae, axioms of Łukasiewicz logic for modal formulae, and additional axioms for B-modal formulae according to the probabilistic semantics of the B operator. The same many-valued logic approach is used to represent and reason under graded attitudes in the other mental contexts. The formalization of the adequate logics –language, semantics, axiomatization and rules – for the different contexts is described in [2].

3. TOURISM RECOMMENDER SYSTEM

In this section we present the general architecture of the Tourism Recommender System. The methodological aspects of the analysis and design stages of this case study can be seen in [4].

Inspired in the different members of a Tourism Chain, in the analysis phase we have detected the following roles: the Provider role (tourist package providers), the Travel Assistant role and Services role (hotel chains, airlines, etc.). In this case study we don't deal with the Service role, we only mention it as a necessary collaborator of the Provider role. Other functional roles were captured i.e., the Interface role, to manage the user interface and the Reservory-Maintenance role (R-Maintenance), to charge, translate to an adequate format and discharge the tourist packages that are sent by the Provider role. 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 Reservory Maintenance role and the Travel Assistant role to the Travel Assistant Agent (T-Agent). 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 agents (P-Agents). This multiagent system is easily scalable to include other providers.

The agents in the Recommender system with the principal source of information they interact with (i.e., the destination ontology and the package reservory), are illustrated in Figure 2.

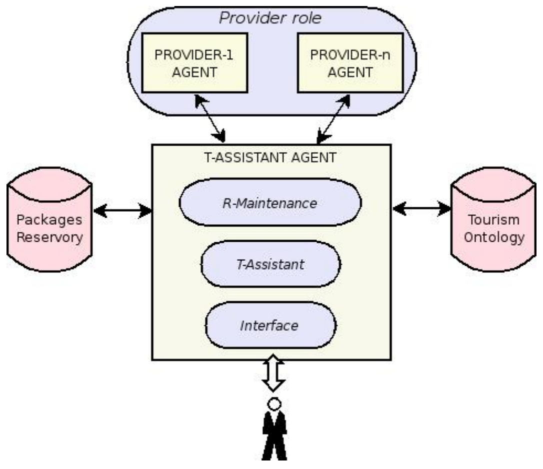


Figura 2: Multiagent architecture for the Tourism Recommender System

The implementation of the Recommender system was developed using SWI-Prolog [11]. This is a multi-threaded version of prolog allowing an independent execution of different contexts (i.e. in different threads). A prior implementation of multi-context agents using this software [6] was a starting point for our development. Furthermore, this prolog version is open source, it is well documented and includes a graphic interface tool in native language.

For our recommender system, each provider agent in the multi-agent systems may be executed in one thread and different threads correspond to the T-Agent components. The software has a set of instructions to deal efficiently with the message communication between threads.

3.1. Providers agents

In our multiagent recommender system two Tour Operator agents (*P-Agents*) are implemented, but the architecture enables to easily include other providers. These agents are only considered as packages suppliers and therefore, we do not get into the inner architecture of them. Each *P-Agent* runs in a different thread being in this way independent from each other and from the *T-Agent*. When the *T-Agent* requests for information, the *P-Agents* send all the current packages they can offer. The communication between agents is by message interchange.

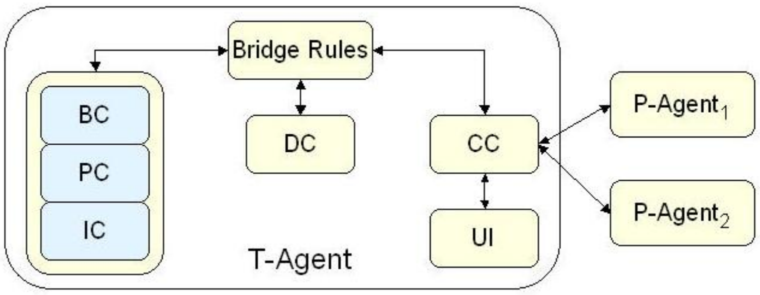


Figura 3: Multithread system scheme

In the real world each tourism 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* use. 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 principal role of the T-Agent is to give tourists recommendation 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).

Then, for developing the T-Agent the implementation of these interconnected components is needed. Each context has its own inference rules and facts, and they should not interfere. Choosing to use a thread for each context allows the desired separation but could slows considerably the application. The solution adopted for our implementation was to place some of these components in different threads. That is the case for the Communication context (CC), the Desire context (DC) and some bridge rules. 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.

For this multithreaded implementation the policy adopted following [6], 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 (has finished the internal deductions). Each Unit has got a message queue that retains the messages until they are been processed. A communication meta-interpreter, is devoted to synchronise the ongoing inference process and the arrival of new incoming messages.

In our prototype the principal interchange of messages is during the initial stage. In this phase the *T-Agent* asks the *P-Agents* for the current tourist packages. Answering this request, the *P-Agents* send many messages, each one containing a package offered. The software tool successfully support this intensive messages interchange.

The Communication context (CC) in the *T-Agent* is in charge of receiving these messages, translating them and immediately it sends them to the Belief context (BC). In this way the agent knowledge is increased with the package information. In the next subsections we described how the principal 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.

4.1. Communication Context

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

In an extended version of this system, all the interactions could be improved in many different aspects (e.g. being more dynamic).

4.1.1. Interaction with the P-Agents

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

Require the packages offered - The CC sends a message to each P-Agent asking them for the current touristic packages they offer.

Receive packages and translate them - The CC behaves as a wrapper, translating the incoming packages into the T-Agent format.

Send packages - Once the packages are put into the correct format, they are sent to the Planner context. The recommendation will derive from this package information and domain knowledge.

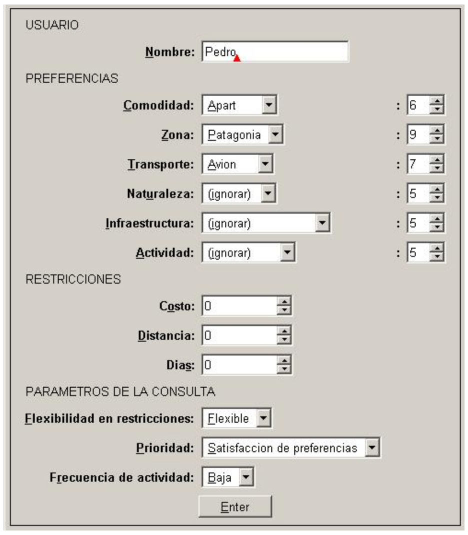
4.1.2. User interface

The user interface is in charge of explicitly acquiring the tourist's profile, giving him the resulting recommendation and receiving the user's feedback. In a first approach this interface was developed using the native graphic library of the software tool. As a tool requirement, this graphic interface runs in an independent thread in closed interaction with the CC one. This interface may be divided into the next sequential stages:

User's preferences acquisition: These preferences are explicitly acquired asking him to fill in a form. The tourist can select his preferences (positive desires) and restrictions (negative desires) assigning them a natural number from 1 to 10 to represent the level of preference (resp. restriction) in the selected item. Furthermore, he can choose different parameters as: the flexibility of restrictions (flexible or strict), the espected frequency of the selected activity (high or low) and the priority criterion to order the recommended packages (preference satisfaction, minimum cost or trust). An example of a tourist's selection using this interface is shown in Figure 4.

Once the user finishes his selection, the CC sends all the acquired information to the Desire context (DC).

Bring the resulting recomendation: 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


USUARIO

Nombre:	Pedro

PREFERENCIAS

Comodidad:	Apart	: 6
Zona:	Patagonia	: 9
Transporte:	Avion	: 7
Naturaleza:	(ignorar)	: 5
Infraestructura:	(ignorar)	: 5
Actividad:	(ignorar)	: 5

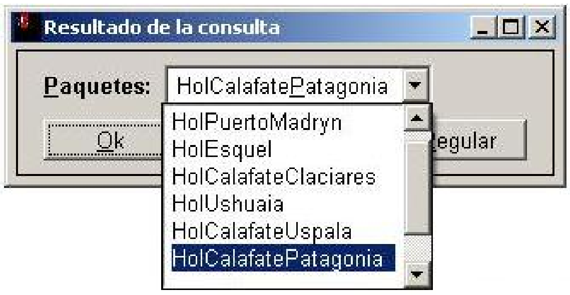
RESTRICCIONES

Costo:	0
Distancia:	0
Dias:	0

PARAMETROS DE LA CONSULTA

Flexibilidad en restricciones:	Flexible
Prioridad:	Satisfaccion de preferencias
Frecuencia de actividad:	Baja

Enter


Resultado de la consulta

Paquetes: HolCalafatePatagonia

HolPuertoMadryn
HolEsquel
HolCalafateClaciares
HolUshuaia
HolCalafateUspala
HolCalafatePatagonia

Ok
egular

Figura 4: User interface: tourist's preferences (left) and package recommendation (right).

(e.g. preference satisfaction). The first packages of this ranking are showed to the tourist and the user can visualize the information about them opening *pdf* files (may be used other multimedia files).

Receive Tourist'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: The user is satisfied with the ranking obtained.
- Different order: The given recommended packages are well considered by the user, but are in a different order than the user's own ranking. Then, he is able to introduce the three best packages in its right order.
- Fair: The user is not satisfied with the given recommendation. Then, the interface enables him to introduce a comment about his opinion.

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

4.2. 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 look for packages that will not make true any of 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. The message is first pre-processed by the meta-interpreter. In the following items we describe how the positive desires are built, in a similar way the negative ones are treated:

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 (e.g. 0.8). For instance, elementary desires in DC for a tourist's consultation (see selection in figure 4) are:

   ```
   desU(yLst([(zona, patagonia)]), 0.9)
   desU(yLst([(transporte, avion)]), 0.7)
   desU(yLst([(comodidad, hotel3)]), 0.5)
   ```

2. **Combined Desires:** After the elementary desires are added to the context, all the possible conjunctions are built. The degrees for the conjunctions are calculated following the guaranteed possibility model (see [2]) where the resulting degree is greater than the maximum of elementary degrees. Having in the DC formulae like $desU(y(D_1), G_1)$ and $desU(y(D_2), G_2)$ it is added the combined desire $desU(yLst([D_1, D_2]), G)$. In this prototype G is particularly computed by the following function:

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

   As for example, it is showed the code of one of the conjunctive combinations built from the elementary desires given above:

   ```
   desU(yLst([(zona, patagonia), (transporte, avion)]), 0.92)
   ```

We notice, that in the special case of some desire types (related to accommodation and particular resources) we consider that the tourist will also be satisfied (in some degree) if he receives from the package a similar value than the desired one. In these cases we are considering implicit rules like "If the tourist have a positive desires D in degree d and D is similar to D' in a degree s, then he also desires D' in a degree $d' = f(d, s)$". Even this rules are not explicitly coded in DC, they are used for the expected satisfaction computation (see next subsection 4.3.4).

The positive and negative desires are both passed by a bridge rule to the Planner context where the feasible packages to satisfy the tourist are selected.

4.3. Belief Context

In this context the *T-Agent* represents all the necessary knowledge about tourism and the argentinian domain: tourist packages, information about destinations and rules to infer how the different user's preferences may be satisfied (in some degree) by the feasible tourist packages.

4.3.1. Tourist packages

One of the most significant data structure in our system is the package structure. After analyzing nearly fourty argentinian packages selected from the Internet, a general structure capable of representing the information available in most of them, was proposed. Each package is represented as a list containing an identifier, a tour provider, the package cost and a travel-stay sequence represented by $Trip$ as 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)

As 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), carcelDeReincidentes],
      [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.

4.3.2. Destination ontology

The *T-Agent* needs to have information about the country and the different possibilities its diverse places bring. 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 vital to infer how 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 focussed on destinations (see e.g. [8]) including the resources they have, the activities they offer, etc. Inspired in some of them, the following features were extracted for the destination ontology in our prototype.

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

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

We use as *coordinates* the geographic coordinates provided by tInstituto Geográfico Militar de la República Argentina (http://www.geoargentina.com.ar). The *zone* assigned to each destination corresponds to the partition of argentinian provinces into zones, proposed by Secretaría de Turismo de la República Argentina (http://www.turismo.gov.ar).

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 was directly code in a prolog file, but it is possible for the *T-Agent* to receive an ontology built using an ontology editor (via XML code).

4.3.3. Special Relations in the domain

To increase the domain knowledge of the *T-Agent* other relations were included in the BC.This knowledge about related concepts 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 considered important to include the following relations:

1. **Similarity dictionary:** The BC includes a set of similarity relations between synonymous or similar concepts, according to the tourism domain. As the *T-Agent* has a Belief context that deals with graded information, this similarity relation may include a degree $g \in [0, 1]$ expressing the 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 silimarity 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(montaña, serro), 0.8)
   ```

2. ***"Better than"* relation:** For the accommodation concepts was added a "better than" relation expressing 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.

4.3.4. Preference Satisfaction Estimation

The Belief context is in charge of evaluating the estimation of how a tourist's desire D is satisfied after executing certain package α_P. From the belief degree r of $B([\alpha_P]D)$ and the degree d of desire D, this estimation is computed as the product $E = r.d$. Following the model presented in [2], the truth degree of $B([\alpha_P]D)$ is considered the probability of having D after following plan α_P. Since the degree of desire d is provided in the DC formulae, to complete the satisfaction estimation we need rules in the BC to compute the belief degree r of the formula $B([\alpha_P]D)$ according to the different desire types.

Basically a tourist plan may be considered as a time sequence of subplans (see [3] for details), and the satisfaction estimation depends on how much a preference is expected to be satisfied in each stage of a trip. As it was presented above, the packages are structured as: $Package ::= (Id, Provider, Cost, Trip)$ where $Trip$ is a travel-stay sequence $[(Travel_i, Stay_i)]$ $i = 1, n$. In our approach each $Travel_i$ and $Stay_i$ parts of the $Trip$ are considered as atomic package stages (sub-plans), amenable to satisfy desires. Packages α_P are therefore modelled as composed plans, $\alpha_P = \alpha_1; ..., ; \alpha_n$, alternating travel and stay sub-plans.

Then, the belief degree r of $B([\alpha_P]D)$ will depend on the probabilities r_i of having D after the execution of the each sub-plan α_i. For computing the degree r the T-Agent needs to estimate the components r_i and then to aggregate them using a suitable operator, i.e.

$(B([\alpha_1]D), r_1) \wedge ... \wedge (B([\alpha_n]D), r_n) \rightarrow (B([\alpha_P]D), \oplus_{i=1,n} r_i)$, where $\oplus$ is an appropriate aggregation operator.

A set of rules which play this aggregation role, depend on the kind of desire and on the user's priority criterion. In the following items we give the insights of this estimation for positive desires.

- *Elementary desires*

For evaluating the expected satisfaction of a desire D by executing a package α_P depending on the kind of desire D, the travel or stay stages of α_P are considered. As for example, if the desire is about transport, the travel stages are used, and if it is related to a natural resource, the stay parts are considered.

The underlying idea to compute the expected satisfaction is to take the proportion of the package where the user's desire is expected to be satisfied (in some degree) respect to the total trip. Furthermore, the estimation of how the different stages of a trip may satisfy a preference, may be also graded. In our approach we consider for this estimation, the similarity degree between the tourist's desire and the respective proposal in the package.

On the one hand if the offer is exactly or "better than" the user's preference, the expected satisfaction of achieving the chosen preference in this package stage, is consider the desire degree (i.e. $E = d$). On the other hand if the offer is similar, our approach is to take the similarity degree s (between asked and offered preferences) to compute the expected satisfaction of the user's desire by a package stage (i.e. $E = s.d$).

Then, if the package α_P is composed by different stages, i.e. $\alpha_P = \alpha_1; ...; \alpha_n$ the general way of computing the preference satisfaction of D by the execution of package α_P, i.e the degree r in the formula $(B[\alpha_P]D, r)$, is the next:

$$(B([\alpha_1]D), r_1) \wedge ... \wedge (B([\alpha_n]D), r_n) \rightarrow (B([\alpha_P]D), \frac{\sum_i r_i \times Time_{\alpha_i}}{TotalTime})$$

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

For instance, if D is about acommodation, $Time_{\alpha_i}$ computes the duration (in days) of the stay α_i and $TotalTime$ is the total duration of the trip. In the case of being D an activity and considering that the user has chosen the high activity frecuency, $Time_{\alpha_i}$ is the hours that the activity is programmed in the stage α_i of the package and $TotalTime$ is an estimation of the total number of hours that the activity could take along the trip.

Example: Let us assume a tourist has an accommodation preference of Apart Hotel represented by the desire D: $desU(yLst([(comodidad, apart)]), 0{,}7)$. Then, the *T-Agent* wants to evaluate the expected satisfaction of D through the package *holCalafatePatagonia* (see subsection 4.3.1). This package has two stay stages: in Calafate destination, with accommodation in Apart Hotel and in Ushuaia, providing a Hotel 3*.

Then, the degrees r_i are computed respectively as: $r_1 = 1$ and using the similarity relation $belU(similar(apart, hotel3), 0{,}75)$, $r_2 = 0{,}75$. Finally, the degree r corresponding to the belief degree of $B([holCalafatePatagonia]D)$ is computed as:

$$r = \frac{(1 \times 3d) + (0{,}75 \times 4d)}{7d} = 0{,}857$$

and the expected satisfaction of the selected preference through this package is $E = r.d = 0{,}599$.

When the tourist's desire is related to a destination resource (e.g. natural resources, activity) the belief degree of achieving the chosen preference by a plan execution, has another interesting characteristic. We have noticed that usually the packages have limited information about destinations and their resources. Thus, for the estimation of some preference satisfacion the *T-Agent* needs to use domain knowledge. In our prototype this information is structured in the destination ontology.

Using the same schema for evaluating the degree r presented above, the computation of r_i degrees are refined. They are computed after a *package-destinations* cross inference, to assess the presence of the tourist's selected preference in the package and in the destination information.

The strategy followed is to evaluate for each package stage α_i, the probability that it has of providing certain resource either it is explicitly offered in a package (r_{Pi}) or it is inferred by the ontology information (r_{Oi}). In this approach the T-Agent takes as the degree r_i the maximum of both estimations, i.e. $r_i = max\{r_{Pi}, r_{Oi}\}$.

- *Combined desires*

The DC theory includes conjunction of positive desires. To evaluate the probability of reaching the conjunction of elementary desires (D_1 and D_2) by the execution of a package α, we assumed that the desires are stochastically independent for each package P. Then, from the degrees r_1 and r_2 corresponding to the elementary desires, we can compute the belief degree in achieving their conjunction by executing the plan α, using the following rule:
$((B[\alpha]D_1, r_1), (B[\alpha]D_2, r_2)) \rightarrow (B[\alpha](D_1 \wedge D_2), r_1 \cdot r_2)$

We remark that the many-valued model of information representation and reasoning in the BC has many advantages. First, this model ables an expressive representation of the domain knowledge. Secondly, this approach allows the agent to evaluate in a more real way the expected satisfaction of preferences by the execution of diverse packages by a *package-ontology* cross inference. Finally, the treatment of this many-valued information makes it possible to compute in a graded way each expected satisfaction, giving the agent more information than a bi-valued approach.

4.4. Planner Context

The PC unit is vital for the *T-Agent* implementation, its theory is composed by $planner$ formulae. This context is responsible for looking for *feasible package*. A *feasible package* satisfies one of the positive desires (elementary or combined) and its execution cannot satisfy any restriction. These feasible plans are computed within this context using an appropriate searching method, that takes into account beliefs and desires injected by bridge rules from the BC and DC units, respectively.

Then, from the positive and negative desires, the package information, the beliefs of the agent about package destinations, the estimation of desire satisfaction by plan execution and the package cost; the Planner can find feasible packages (coded as $paqSi$ formulae). These touristic packages may fulfill the tourist's positive desires, but avoiding negative ones (do not satisfy them in degree greater than a threshold: $UmbralN$).

The following forward rule code 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)
```

Notice that this rule uses formula coming from DC (des formulae) and from BC (bel formulae). For each feasible package named $IdPaq$ the normalized cost ($CN \in [0, 1]$) is computed and used instead of its actual cost.

After the PC has found the set of feasible packages, they are passed to the Intention context in charge of making a ranking of these packages.

4.5. Intention Context

The *T-Agent* in this context (IC) finds the intention degree for each feasible package. As we have previously mentioned, the intention degree trades off the benefit and the cost of reaching a goal or desire, through a plan execution.

There is a bridge rule that infers the degree of $I_\alpha(D)$ for each package α that allows to achieve D. This value is deduced from the degree of desire D (GD), the belief degree in achieving D by executing the plan α (GR), the cost of the plan α (CN) and the trust in the package provider (GT).

The intention degree for $I_\alpha(D)$ is calculated by a function f that suitably combines all these factors. Different functions can model different individual agent behaviors. In the *T-Agent* this function is defined as a weighted average:

$$f = (w_d * GD + w_r * GR + w_c * (1 - CN) + w_t * GT)$$

where the different weights w_i are set by the *T-Agent* according to the priority criterion selected by the user (minimum cost, preference satisfaction or trust).

The following bridge rule infers the intention formulae related to the package Id with the corresponding intention degree (G):

```
planner(paqSi(Id, Proveedor, CostoN, DeseosP), GR),
  bel(trust(Proveedor), GT),
  des(yLst(D), GD),
  bel(prioridad(PU), 1),
  f(GD, GR, GT, CN, PU, G)
  --:  int(paqRecomendado(Id), G)
```

Once the rule has been applied to all the feasible plans the IC has a set of graded intention formulae. Using the intention degree the *T-Agent* makes a package ranking that communicates to the CC. We opted to select the first N [1] packages to recommend the tourist.

Finally, the selected packages are passed to the CC and then, through the user interface the *T-Agent* gives the tourist this ranking as recommendation. For instance, in Figure 4 it is showed a tourist's preference selection (on the left) and the resulting recommended ranking (on the right). After analyzing the results, the user is asked to give the system his feedback.

5. 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 many-valued 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. Secondly, 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 diverse packages. Finally, the intention degree of a plan towards a desire satisfaction may be computed as a function of diverse factors (e.g. satisfaction, cost, trust). As we can obtain diverse agent behaviors defining different functions for intention computation, these become a crucial point in the agent model.

The first experimentation of this prototype was carried out with good results. Considering 150 recommendations, the 75 % of the user's opinions were aceptable (correct or different order) and among them, the 70 % was correct. Now we are working in the adjustment of the T-Agent behavior using this user's feedback. As for future work we plan to simulate a crisp BDI version of the T-Agent as to experiment and compare with the graded BDI model of this agent.

[1] N is an agent parameter currently set as $N = 6$

REFERENCIAS

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

[2] Casali A., Godo Ll. and Sierra C. Graded BDI Models For Agent Architectures. Leite J. and Torroni P. (Eds.) *CLIMA V, LNAI* 3487, pp126-143, Springer-Verlag, Berling Heidelberg, 2005.

[3] Casali A., Godo Ll. 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*, Ed. Max Bramer (ISBN 0-387-34654-6). Boston: Springer, 415-424, 2006.

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

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

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

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

[8] Prantner Kathrin, E-Tourism, DERI Innsbruck,5 October 2004 (www.deri.org)

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

[10] Ricci F., Travel recommender Systems, in IEEE Intelligent Systems, November/December 2002, pages 55–57, 2002.

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

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