Validation and Experimentation of a Tourism Recommender Agent based on a Graded BDI Model[1]

Ana CASALI [a,2], Lluís GODO [b] and Carles SIERRA [b]
[a] *Depto. de Sistemas e Informática, FCEIA - UNR , Argentina*
[b] *IIIA - CSIC, Barcelona, Spain*

Abstract. In this paper, a validation and an experimentation of the use of graded BDI agents is reported. This agent model has been proposed to specify agents capable to deal with the environment uncertainty and with graded attitudes in an efficient way. As a case study we focus on a Tourism Recommender Agent specified using this agent model. The experimentation on the case study aims at proving that this agent model is useful to develop concrete agents showing different and rich behaviours. We also show that the results obtained by these particular recommender agents using graded attitudes improve those achieved by agents using non-graded attitudes.

Keywords.

1. Introduction

In the last years, an increasing number of theories and architectures have been proposed to provide multiagent systems a formal support, among them the so-called BDI architecture [7]. This model has evolved over time and has been applied, to some extent, in several of the most significant multiagent applications developed up to now. With the aim of making the BDI architecture more expressive and flexible, in [2] a general model for Graded BDI Agents (g-BDI) has been proposed , specifying an architecture able to deal with the environment uncertainty and with graded mental attitudes. As a case study, a Tourism Recommender multiagent system has been designed and implemented where its main agent, the Travel Assistant Agent (*T-Agent*), has been modelled using the graded BDI model [3,4]. Actually, recommender systems [9,1] is an increasing area of interest within the Artificial Intelligence community where Agent technology becomes very valuable as it eases the expression of those different characteristics we expect from these systems (e.g. user profile oriented, able to aggregate relationships from heterogeneous sources and data, open and scalable) [6]. The agent-based tourism recommender

---

[1] L. Godo and C. Sierra acknowledge partial support of the Spanish CONSOLIDER CSD2007-0022 project *Agreement Technologies*.

[2] Corresponding Author: Ana Casali, Depto. de Sistemas e Informática, Facultad de Cs. Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, Av Pellegrini 250, 2000 Rosario, Argentina; E-mail: acasali@fceia.unr.edu.ar.

system that has been developed has the goal of recommending the best tourist packages on Argentinian destinations, provided by different tourist operators, according to user's preferences and restrictions.

In this work we report on the validation process of the recommender system as well as on the experimentation we have performed with the aim of proving different properties of the g-BDI model of agents. Namely, we have performed a sensitivity analysis to show how the g-BDI agent model can be tuned to have different behaviours by modifying some of its component elements. Also, we have done some experiments in order to compare the performance of recommender agents using the g-BDI model with respect to agents without graded attitudes. This paper is structured as follows. In Section 2 the g-BDI model of agent is succinctly described. Then, in Section 3 we present the relevant characteristics of the Tourism Recommender implementation. In Section 4, we describe the validation process of the *T-Agent* designed using the g-BDI model and implemented in a multithreaded version of Prolog. Finally, the results of the above mentioned experiments are reported in Section 5. We conclude in Section 6 with some final remarks.

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, agents showing different kinds of behaviour 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) [5] allowing different formal (logic) 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 theories. Thus, a MCS is defined as a group of interconnected units or contexts $\left\langle \{C_i\}_{i\in I}, \Delta_{br} \right\rangle$. 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 the language, axioms, and inference rules of the context respectively. Δ_{br} is a set of bridge (inference) rules, that is, rules of inference with premises and conclusions in possibly different contexts. When a theory $T_i \subseteq L_i$ is associated with each unit, the specification of a particular MCS is complete. 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) and a set of bridge rules (Δ_{br}). 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 behaviour of the system will depend on the logic representation of each intentional notion in the different contexts and the bridge rules. The left hand side of Figure 1 illustrates the g-BDI agent model proposed with the different contexts and some of the bridge rules relating them. As for example, we describe the bridge rule (see (3) in Figure 1) that infers the degree of intention towards a goal φ for each plan α that allows to achieve the goal ($I_\alpha \varphi$), in the next section 3.

In order to represent and reason about graded mental attitudes we use a modal many-valued approach where reasoning about graded uncertainty, preferences and intentions

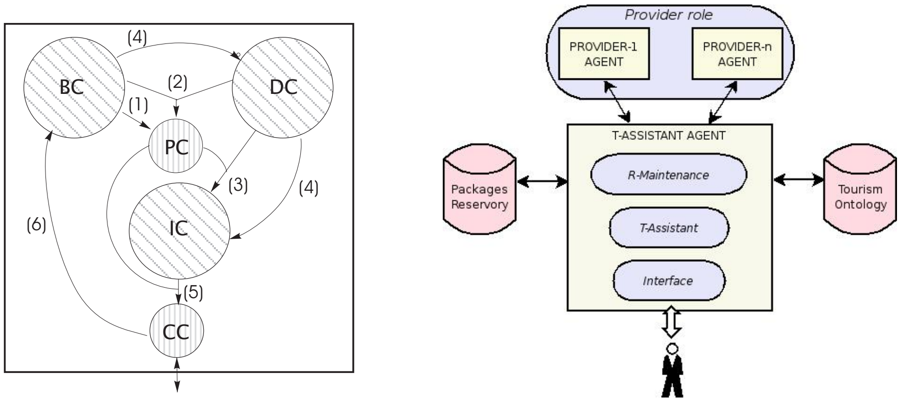

Figure 1. Multi-context model of a graded BDI agent (left) and the multiagent architecture of the Tourism Recommender system (right)

is dealt with by defining suitable modal theories over suitable many-valued logics. The formalization of the adequate logics (i.e. language, semantics, axiomatization and rules) for the different contexts and the basic bridge rules can be seen in [2].

3. A Tourism Recommender System

For a prototype version of the Tourism Recommender System, we define two agent's types: provider agents and a Travel Assistant agent. As it is natural in the Tourism Chain, different Tourist Operators may collaborate/compite for the provider role. To represent these different sources of tourist packages, we use for this implementation two different agents (*P-Agents*), but the multiagent architecture is easily scalable to include other providers. These agents are only considered as packages suppliers and therefore, we do not get into their internal architecture. The agents in the Recommender system with the main source of information they interact with (i.e., the destination ontology and the package reservory) are illustrated in Figure 1 (right). The implementation of the Recommender system was developed using a multi-threaded version of prolog [3] allowing an independent execution of different contexts (i.e. in different threads). The principal role of the Travel assistentt agent (*T-Agent*) is to provide tourists with recommendations about Argentinian packages and it can be suitably modelled as an intentional agent and particularly, by a g-BDI model. This agent model is specified by a multicontext architecture having mental and functional contexts Next, we briefly describe how the contexts have been implemented in order to obtain the desired behaviour of the *T-agent* (for a detailed description see [4]).

Communication Context (CC): The CC is the agent's interface and is in charge of interacting with the tourism operators (*P-Agents*) and with the tourist that is looking for recommendation. The *T-Agent*, before beginning its recommendation task, updates its information about current packages (carrying out its reservory maintenance role). It behaves as a wrapper translating the incoming packages into the *T-Agent* format and sends them to the Planner context. The user's interface has been developed as a Web service application and it is responsible for:

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

- *Acquiring user's preferences:* they are explicitly obtained from the user by filling in a form. The tourist can set her preferences (positive desires) and restrictions (negative desires) and assign them a natural number from 1 to 10 to represent the level of preference (resp. restriction) for the selected item. Preferences are given about the following issues: geographic zone, natural resources, infrastructure, accommodation, transport or activities. The constraints are related to the maximum cost she is able to afford, the days available for traveling and the maximum total distance she is willing to travel. Once the user finishes his selection, the CC sends all the acquired information to the Desire context DC.

- *Showing the resulting recommendation:* as a result of the *T-Agent* deliberation process, the CC receives from the Intention context a ranking of feasible packages that satisfies some or all of the tourist preferences. Then, he can visualize the information about them (i.e. the description of the transport, destination, accommodation, activities) opening suitable files.

- *Receiving Tourist's feedback:* After analyzing the ranking of the recommended packages, the user can express through the CC interface her opinion about the recommendation. Namely, the user can select one of the following three possible evaluations:

1. *Correct*: when the user is satisfied with the ranking obtained.
2. *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.
3. *Incorrect*: The user is not satisfied with the given recommendation. Then, the interface enables him to introduce a (textual) comment with his opinion.

All the information resulting from the user data entry is stored to evaluate the system behaviour.

Desire Context (DC): As the *T-Agent* is a *personal agent*, its overall desire is to maximize the satisfaction of the tourist's preferences. Thus, in this context the different tourist's graded preferences and restrictions are respectively represented as positive and negative desires. For instance, the preferences of a tourist that would like to go to a mountain place and to travel by plane but not more than 2000 kms could be represented by the following theory:

$$\mathcal{T}_{DC} = \{(D^+(resources, mountain), 0.9), (D^+(transport, air), 0.7),$$
$$(D^+[(resources, mountain), (transport, air)], 0.92), (D^-(distance, 2000), 0.5)\}$$

The *T-Agent* uses the desires as pro-active elements, and are passed by a bridge rule to the Planner context that looks for feasible packages.

Belief Context (BC): In this context the *T-Agent* represents all the necessary knowledge about tourism and the Argentinian domain: tourist packages (each package is represented as a list containing an identifier, a tour provider, the package cost and a travel-stay sequence), information about destinations (represented by a destination ontology) and rules to infer how much preferences can be satisfied (to some degree) by the feasible tourist packages. This context also contains knowledge about *similarity relations* between concepts to extend the possibility of satisfying a tourist with similar preferences than the actually selected ones. Besides, the BC is in charge of estimating the extent (the belief) $B([\alpha_P]\varphi)$ to which a desire (preference) φ will be achieved when selecting a given package α_P.

Planner Context (PC): This context it is responsible for looking for *feasible packages*. A package is *feasible* when it satisfies at least one of the positive desires and its execution does not violate any restriction. These plans are computed within this context using an appropriate search method, that takes into account beliefs and desires injected by bridge rules from the BC and DC units, respectively. This set of packages is passed to the Intention context which is in charge of ranking them.

Intention Context (IC) and a Bridge rule example: In the IC the *T-Agent* finds the intention degree for each feasible package that is expected to satisfy a desire φ. There is a bridge rule that infers the degree of $I_\alpha\varphi$ for each package α that allows to achieve φ. This value is computed by a function f that suitably combines factors like the degree d of desire about φ, the belief degree r in achieving φ by executing the plan α, and the (normalized) cost c of the plan α.

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

Different functions can model different individual agent behaviours. In the *T-Agent* this function is defined as a weighted average: $f(d, r, c) = (w_1 * d + w_2 * r + w_3 * (1 - c))/(w_1 + w_2 + w_3)$, where the different weights w_i are set by the *T-Agent* according to the priority criterion selected by the user (minimum cost or preference satisfaction). Once the rule has been applied to all the feasible plans, the IC has a set of graded intention formulae. Using the intention degrees the *T-Agent* makes a package ranking that communicates to the CC and then, through the user interface, it is provided back to the user as recommendation.

4. Validation of the Recommender Agent

It has been shown in [3,4] that the g-BDI architecture described above is useful to model recommender systems. In this section we try to answer whether this system provides satisfactory recommendations. This recommender system is accessible via Internet[4] allowing an online and a multiuser access. To analyze its behaviour the user's opinion is crucial. This opinion is given after he receives the ranking of the tourist packages the system recommends. We want to know whether the *T-Agent* is a personal agent satisfying, to some degree, a set of different users. As the process of information classification is generally a complex and personal task, and may differ among persons, we want to measure the average system behaviour over a population.

We use the implementation of the *T-Agent* modelled as a g-BDI agent and a set of 40 tourism packages offered to the *T-Agent* by the provider agents. We have collected a set of 52 queries made to at least 30 different users, most of them students of our Department. The preferences and restrictions introduced by them as input to the system, together with the system results and the user's feedbacks, constitute our *N-cases* set. Each case in the dataset is composed by:

- **User's Input**: a user ID and his graded preferences and restrictions.

[4]http://musje.iiia.csic.es/eric/

- **Agent's Result**: the system returns a ranking of at most nine packages.
- **User's Feedback**: as explained in the previous section, after analyzing the information of the recommended plans the user provides a feedback by evaluating the result as: (1) Correct, (2) Different order or (3)Incorrect. In this validation process we consider feedback types (1) and (2) correspond indeed to *satisfactory results* since the user can find what he wants among the recommended options.

Actually, for the validation purposes, we have only taken into account those cases which included the user's feedback.

Results: From the selected 52 cases (*N-cases*) we have separated the ones having a satisfactory feedback (including Correct or Different order) from the non satisfactory ones. The cases where the user provides his own ranking (Option 2: Different order) are indeed very valuable because it means that the user analyzed the offers proposed by the system, while cases with the Option 1 (Correct order) sometimes they correspond to a "quick answer". The results obtained for the *N-cases* classified by the different feedback categories are shown in the following Table. From these results, the global behaviour of the T-Agent may be considered useful in most cases (73% of *N-cases*).

Queries (*N-cases*)	Correct order	Different order	Incorrect	Satisfactory (*S-cases*)
52	21	17	14	38
100%	40.4%	32.7%	26.9%	73.1%

In order to give a general measure of the *T-Agent* results over the satisfactory cases (*S-cases*), we have evaluated how close is the *T-Agent* ranking with regard to the user's own ranking. For this, we choose the Block (Manhathan) distance between the position of the first three packages selected by the user and their position in the system ranking. This distance was adopted because it is appropriate for capturing positional differences. Namely, assume the user's feedback is $U_i = (P_{i1}, P_{i2}, P_{i3})$ and the *T-Agent* ranking for this consult is $R_i = (R_1, R_2, ..., R_9)$. Then, if $P_{i1} = R_j$, $P_{i2} = R_k$, $P_{i3} = R_n$, the distance between the user's the system rankings is defined by:

$$Dist\,(U_i, R_i) = |1 - j| + |2 - k| + |3 - n|$$

The frequencies of the block distance corresponding to the *T-Agent* results for all the *S-cases* can be seen in Figure 2. We analyzed the incorrect cases and the comments attached (if any) about the user dissatisfaction with respect to the system recommendation and they were somewhat scattered. Apart from that, in some of these incorrect cases we detected a system shortcoming related to the tourism knowledge base, the destination ontology used for this experimentation was incomplete with respect to the popular knowledge. Therefore, we believe the *T-Agent* behaviour may be improved by completing these ontologies. Finally, the *S-cases* set of satisfactory results (see table in Figure 2) yields an average distance of 2.95 in the scale $[0, 18]$, and hence giving a good global measure result. Summarizing, we have obtained satisfactory results of the Recommender System in this validation process that allows us to claim that "the *T-Agent* recommended rankings over Tourism packages are in most cases near to the user's own rankings".

5. Experimentation

In this section we present the experimentation we have made following two directions. The first one, we call it Sensitivity Experimentation, has the purpose of analyzing how much the general g-BDI agent architecture can model concrete agents having different behaviours by modifying some of its components. The second one aims at checking whether the distinctive feature of the g-BDI agent model, which is the gradual nature of mental attitudes, actually makes a difference (in terms of better results) with simulated BDI non-graded models.

5.1. Sensitivity model experimentation

We have performed two experiments to analyze how the overall recommender system behaviour can be modified by tuning some of the *T-agent* components. First, in *Experiment 1* we change the theory of one of the mental contexts, the desire context DC, using another way for computing the desire degree for each preference combination. Then, in *Experiment 2* we modify the bridge rule (1) definition by changing the function f to obtain the intention degree.

Experiment 1 For this experiment we follow the next steps:
(1) We use the tourism recommender agent *T2-Agent*: this agent was developed changing in the *T-Agent* the desire context. The modification in this context is related to the way the desire degrees are computed. The underlying idea was to weight not only the preference degree but also the number of preferences we are considering in each combined desire, as to give more relevance to the desires that combine a higher number of preferences. For this purpose in the Desire Context of the *T2-Agent* we use as degree for desire D the value

$$d' = 1/2 * (d + \frac{CardD}{CardPref})$$

where d is the degree used in the *T-Agent*, $CardD$ is the number of preferences considered in the desire D and $CardPref$ is the number of preferences selected by the user.
(2) We consider the *S-cases* (see Validation 1) where the results were satisfactory.
(3) The *user's inputs* of the *S-cases* are run in the *T2-Agent*
(4) We compare the *T2-Agent* results with the *S-cases user's feedbacks* we have for the *T-Agent* and compute distances.
Results: In the experiment we compare the ranking proposed by *T2-Agent* with the feedback of the *S-cases* consisting of the first three packages extracted from the *T-Agent* recommendation. Some of these packages may not be found in the *T2-Agent* answer. As in the validation process, we use the Block distance to have a global measure of the *T2-Agent* performance. For the missing packages, we take an optimistic approach assuming that the distance is 10 (supposing that the missing packages would be in the first place immediately after those appearing in the ranking). The distance frequency corresponding to the *T2-Agent* results for all the *S-cases* are shown on the table and graphic of Figure 2. For this experiment we use two global measures, the *average* of distances excluding the cases having missing packages and the *total average* that includes all the cases. The average of the distances between the *T2-Agent* ranking and the feedback of the *S-cases* is 2.85 and the total average is 4.23. Comparing the first global measure with the one ob-

tained with the *T-Agent* results, we notice that this measure is slightly better than the one obtain for the *T-Agent* results. We think that in a direct measure of the performance of the *T2-Agent* (comparing the *T2-Agent* ranking with the corresponding user's feedback) we would have had better results. Thus, we can conclude that *T-Agent* and *T2-Agent* share a similar behaviour, but the *T2-Agent*'s results are a little bit closer to the user's selections.

Experiment 2 We follow the same steps used in *Experiment 1* but in this case, for item (1) we use a tourism recommender agent called *T3-Agent*. This new agent has been defined from the *T-Agent* by changing one of its bridge rules. Namely, we have modified bridge rule (1) (see Section 3) that computes the intention degree of a package α in order to satisfy a set of user's preferences φ. We have used for *T3-Agent* a function that assigns an intention degree according to two different priorities (Preference Satisfaction or Minimum Cost) by defining two lexicographic orderings, namely:

- when the *Preference Satisfaction* criterion is selected, we consider the intention $I_\alpha\varphi$ described by the 3-tuple $(d', r, 1-c)$, where d' is the desire degree of φ, r is the belief degree in satisfying the user's preferences φ by the considered plan α, and c is the cost of the plan α. Then, we use the lexicographic order on the product space $[0,1]^3$ to rank the 3-tuples and hence the intentions.
- when the *Minimun Cost* criterion is selected, we consider $I_\alpha\varphi$ described by the 3-tuple $(d', 1-c, r)$ and then we rank intentions by lexicographically ordering these tuples.

Results: The resulting distance frequencies of the results of the *T3-Agent*, for the *S-cases* are shown in Figure 2 (compared with the ones obtained by *T-Agent* and *T2-Agent*). The average of the distances in this case was 4.97, worst than the previous experiments, and the total average is 6.73. This means that the ranking obtained by this *T3-Agent* is farther from the user's ranking obtained in the validation process than the results of the previous versions *T-Agent* and *T2-Agent*. This fact does not mean that *T3-Agent* behaviour is necessarily worst, perhaps it finds other options different from the ones found by the *T-Agent*. This result may be interpreted that the way this agent computes the intention degree is different from the way the other recommender agents do it. Then, we can state that the g-BDI model allow us to engineer recommender agents having different behaviors.

Distance	Frequency		
	T-Agent	T2-Agent	T3-Agent
0	21	11	5
1	1	4	3
2	2	4	3
3	3	5	3
4	2	1	2
5	0	1	2
6	2	4	3
7	2	2	0
8	0	0	3
9	0	0	2
10	0	1	3
11	3	2	1
12	0	1	4
13	0	1	1
14	1	0	1
17	1	0	0
18	0	0	1
26	0	0	1
30	0	1	0
Average	2.95	2.85	4.97
Tot.Average	2.95	4.23	5.73

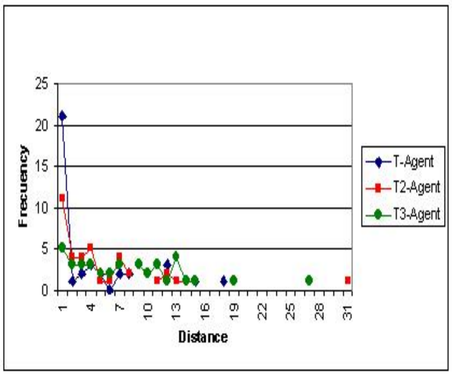

Figure 2. Distance frequencies Table (left) and Graphic (right)

5.2. *Graded vs. non-graded model comparison*

The aim of this experimentation is to compare the g-BDI model with non-graded (two-valued) BDI architectures. We want to show that the graded model of agent allows us to implement recommender agents having better results than the ones based on non-graded BDI models. We use the *T-Agent* and *T2-Agent* prototypes as g-BDI model implementations. Since the development of a Tourism Recommender using another traditional BDI architecture would be a highly time-demanding task and since also different factors would possibly interfere in the comparison of the results (e.g. how the agent builds plans, which decision process she uses), for simplifying and clarifying purposes we have decided to use simulated non-graded versions of the g-BDI architecture of the tourism agent. Starting from the recommender agents *T-Agent* and *T2-Agent* we keep their multicontext architecture and their logic schemes for contexts [5]. Then we introduce some thresholds to make the desire and belief attitudes two-valued (i.e., their degrees will be allowed to only take values in $\{0, 1\}$). The intention degrees have been left many-valued as to obtain a ranking of the selected packages.

Experiment 3 We have followed the same procedure as for the previous ones but, for this case, we use a family of Tourism Recommender agents called *Cij-Agents*. These agents derive from the recommender *T-Agent* or *T2-Agent* and simulate two-valued models of BDI agents. Each *Cij-Agent* has been developed by introducing thresholds in the context DC (U_d) and in the context BC (U_b) of the *T-Agent* and *T2-Agent*, to decide which formulae in these contexts are considered to hold (i.e. those with degree 1) and which do not (i.e. those with degree 0). Then, the following internal processes are introduced in these contexts:

- **DC:** before introducing formulae like $(D^+\phi, d)$ in the DC it is checked whether $d \geq U_d$; if so, the formula $(D^+\phi, 1)$ is added in the context, otherwise this desire is discarded (supposing $(D^+\phi, 0)$).
- **BC:** the same happens when the belief context evaluates the degree r of formulae like ($B[\alpha]\varphi$, r), if $r \geq U_b$ then the formula $(B[\alpha]\varphi, 1)$ is added to the BC, otherwise its degree is considered to be 0.

As for the setting of the different thresholds, we have analyzed the desire and belief degrees distribution in the *T-Agent* and *T2-Agent* previous executions. We have experimented different thresholds and it turns out that with three different values (0.4, 0.5 and 0.6) we can obtain a good representation of the whole variations in the agents' results. Then, we have defined the following "two-valued" BDI agents using three thresholds in each case:

- Deriving from *T-Agent*:

1. *C14-Agent* uses $U_d = U_b = 0.4$
2. *C15-Agent* uses $U_d = U_b = 0.5$
3. *C16-Agent* uses $U_d = U_b = 0.6$

- Deriving from *T2-Agent*:

4. *C24-Agent* uses $U_d = U_b = 0.4$
5. *C25-Agent* uses $U_d = U_b = 0.5$
6. *C26-Agent* uses $U_d = U_b = 0.6$

Then, we have run the *S-cases* in each agent of this two-valued family and compared the results with the *S-cases* feedback computing the Block distances. As in the previ-

[5]This is possible as the many-valued frameworks used for the mental contexts are extensions of classical logic used in the two-valued models.

ous experimentation we have used the distance average and the total average as global measures.

Results: The resulting distance frequencies of the results of these two families of crisp agents deriving from *T-Agent* and *T2-Agent* are respectively shown in Figure 2. The average distance and total average resulting from this experiment are gathered in the following table.

	T-Agent family		*T2-Agent* family	
	Average	**Tot. Average**	**Average**	**Tot. Average**
g-BDI model	2.95	2.95	2.85	4.23
$U_d = U_b = 0.4$	6.43	14.06	4.04	8.36
$U_d = U_b = 0.5$	6.43	14.83	3.50	8.07
$U_d = U_b = 0.6$	4	17.41	3.55	14.43

Comparing the averages obtained with the two-valued models of recommenders (deriving from *T-Agent* and *T2-Agent*) we can see that those corresponding to the thesholds 0.4 and 0.5 are very similar. The average achieved with the threshold 0.6 is the best in the *T-Agent* family and is almost the best in the *T2-Agent* one, but the total average is greater, meaning that we have more packages of the *S-cases* feedback out of the system ranking. In both families we can see that the distance average of the recommenders using graded models are better than the simulated two-valued ones (using three different thresholds). These results give support to the claim that the recommender agents modelled using graded BDI architectures provide better results than the ones obtained using two-valued BDI models.

6. Conclusions

In this work we have focused on the validation and experimentation of g-BDI agents using as a case study a Tourism recommender agent. First, the results of the validation peformed allows us to conclude that g-BDI agents are useful to build recommender systems in a real domains such as tourism, providing satisfactory results. Second, we have also performed a sensitivity analysis showing that a g-BDI agent architecture can engineer concrete agents having different behaviours by suitably tuning some of its components. Finally, the results of a third experiment support our claim that the distinctive feature of recommender systems modelled using g-BDI agents, which is using graded mental attitudes, allows them to provide better results than those obtained by non-graded BDI models.

References

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

[2] 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.

[3] Casali A., Godo L. and Sierra C. Modeling Travel Assistant Agents: a graded BDI Approach. IFIP-AI, WCC. Artificial Intelligence in Theory and Practice., Max Bramer (Ed.) (ISBN 0-387-34654-6), pp. 415-424 Springer Verlag, 2006.

[4] Casali, A., Von Furth, A., Godo, Ll., Sierra, C., A Tourism Recommender Agent: From theory to prectice, WASI-CACIC 2007 (ISBN 978-950-656-109-3) pp 1548-1561, Corrientes, Argentina, 2007.

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

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

[7] Rao, A. and Georgeff M. BDI agents: From theory to practice. *In proceedings of the 1st International Conference on Multi-Agents Systems*, pp 312-319, 1995.

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

[9] 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.