

Reputation-based decisions for logic-based cognitive agents

Isaac Pinyol · Jordi Sabater-Mir ·
Pilar Dellunde · Mario Paolucci

Published online: 17 September 2010
© The Author(s) 2010

Abstract Computational trust and reputation models have been recognized as one of the key technologies required to design and implement agent systems. These models manage and aggregate the information needed by agents to efficiently perform partner selection in uncertain situations. For simple applications, a game theoretical approach similar to that used in most models can suffice. However, if we want to undertake problems found in socially complex virtual societies, we need more sophisticated trust and reputation systems. In this context, reputation-based decisions that agents make take on special relevance and can be as important as the reputation model itself. In this paper, we propose a possible integration of a cognitive reputation model, Repage, into a cognitive BDI agent. First, we specify a belief logic capable to capture the semantics of Repage information, which encodes probabilities. This logic is defined by means of a two first-order languages hierarchy, allowing the specification of *axioms* as first-order theories. The belief logic integrates the information coming from Repage in terms of image and reputation, and combines them, defining a typology of agents depending of such combination. We use this logic to build a complete graded BDI model specified as a multi-context system where beliefs, desires, intentions and plans interact among each other to perform a BDI reasoning. We conclude the paper with an example and a related work section that compares our approach with current state-of-the-art models.

I. Pinyol (✉) · J. Sabater-Mir · P. Dellunde
IIIA—Artificial Intelligence Research Institute, CSIC—Spanish Scientific Research Council,
Bellaterra, Barcelona, Spain
e-mail: ipinyol@iiia.csic.es

J. Sabater-Mir
e-mail: jsabater@iiia.csic.es

P. Dellunde
Universidad Autònoma de Barcelona, Bellaterra, Barcelona, Spain
e-mail: pilar.dellunde@uab.cat; pilar@iiia.csic.es

M. Paolucci
ISTC—Institute of Cognitive Sciences and Technologie, CNR—National Research Council, Rome, Italy
e-mail: mario.paolucci@istc.cnr.it

Keywords Reputation · Image · Belief logic · BDI agents · Trust · Multi-context systems

1 Introduction

Computational trust and reputation models have been recognized as key to design and implementation multi-agent systems [28]. These models manage and aggregate the information needed by agents to efficiently select partners in uncertain situations. In recent years, several models have been developed [43]. For simple applications, a game theoretical approach similar to that used in most models can be sufficient. However, if we want to undertake problems found in socially complex virtual societies, more sophisticated trust and reputation systems based on solid cognitive theories are needed. One such cognitive theory is defined in [9].

The theory [9] proposes that agents evaluate the performances of other agents according to certain criteria. These evaluations (social evaluations from now on) can be believed by the agents, communicated or both believed and communicated. According to [9] a social evaluation that is believed is an image, while a social evaluation that circulates in the society, is reputation. Therefore, an agent can have a good image of agent A as a seller, and at the same time acknowledges that A has a bad reputation as a seller. Furthermore, at this level, the theory describes a typology of possible decisions that autonomous agents can make involving social evaluations:

- *Epistemic decisions* cover the dynamics of beliefs regarding image and reputation, or in other words, decisions about updating and generating social evaluations.
- *Pragmatic–strategic decisions* are decisions of how to behave with potential partners using social evaluations information, and thus, how agents use them to reason.
- *Memetic decisions* refer to the decisions of how and when to spread social evaluations.

From a computational point of view, not all current state-of-the-art reputation models make a distinction between image and reputation, but all of them compute social evaluations. In fact, the field of reputation models has been mainly focused on epistemic decisions, while little attention has been paid to pragmatic–strategic and memetic decisions. Indeed, agents' decisions about how to use reputation information and how and when to spread them have been designed ad-hoc lacking any systematic or formal procedure. As mentioned before, this solution may suffice for simple environments, but in more complex societies pragmatic–strategic and memetic decision can be as important as epistemic decisions, and need more formal approaches as well.

This paper focuses on pragmatic–strategic decisions. Here the Repage reputation model is chosen as a paradigmatic example, because it is based on the cognitive theory defined in [9], to be integrated in a BDI (*Belief, Desire, Intention*) agent, providing then a formal integration of social evaluations in the agents' reasoning and decisions. To do so, in Sect. 2 we introduce the concepts of social evaluation in the framework of the cognitive theory of reputation that we use in the paper, and how Repage deals with these aspects. In Sect. 3 we define the belief logic of the agents (L_{BC}) as many-sorted first-order logic to capture the semantics of Repage information, which embraces probabilities. We use this logic in Sect. 4 to ground image and reputation Repage predicates in terms of beliefs. We highlight a typology of agents depending on the interaction between image and reputation. In Sect. 5 we finally specify the complete BDI agent architecture as a multi-context system whose logical reasoning process uses L_{BC} . In Sect. 6 we analyze some relevant reasoning points by presenting an example. In Sect. 7 we state the related work and compare some existing trust and reputation models with our

BDI + Repage system. Finally, in Sect. 8 we conclude our analysis and propose the future work.

2 Social evaluations: image and reputation

If we want to define an agent architecture capable of capturing the notions of reputation and use it in a classical practical reasoning, we need first to get in touch with the view that Conte and Paolucci introduced in [9]. This cognitive theory of reputation enhances what should be understood as *reputation* by contrasting it with a very related construct: *image*. According to this theory, both image and reputation are social evaluations. Image though, is a simple evaluative belief that tells how good or bad a given target results to be in a given context. Instead, reputation is a metabelief that acknowledges the existence of an evaluation that *circulates* in the society. Thus, when *A* accepts that agent *B* has a bad reputation as a *car driver*, *A* does not necessary believe such evaluation. It only believes that *B* is reputed to be bad as a *car driver*.

In the following two subsections we get in touch with these two constructs as defined in [9], providing a conceptual vision from a cognitive perspective.

2.1 Image

Image is an evaluative belief, a belief that describes an evaluation of a target, that can be a single agent or supra-agent (like groups or institutions), towards a specific context. In fact, from both [9] and [30] an image requires the context to be a goal that the agent wants to achieve. Hence, an agent *A* evaluates another agent *B* when *A* thinks that *B* is *good* or *bad* for achieving the goal. We will see how this constraint is relaxed when considering Repage [40], the computation model inspired is this cognitive theory and the base for our work. For instance, in the most simplified scenario, an agent can hold a *very good* image of *John* in the context of *obtaining 2 boxes of high-quality wine*.

The theory describes three sets of agents that participate in a given social evaluation as image [9]:

- *Evaluators*: A nonempty set of agents that share the evaluation. Hence, they must share the same goal.
- *Targets*: A nonempty set of agents or supra-agents that are evaluated by the evaluators.
- *Beneficiaries*: A nonempty set of agents that use the evaluations, and thus, also share the same goal.

Note that the sets of evaluators and beneficiaries do not necessary are the same. This is very clear in online reputation mechanism, like eBay [11], where buyers evaluate sellers and these evaluations are used by other buyers.

2.2 Reputation

The theory considers reputation as a belief about others' evaluations. From a broad sense, it can be considered a metabelief. The theory analyses the roles of agents participating in a given social evaluation as reputation [9]:

- *Evaluators*: A nonempty set of agents that share the evaluation. Hence, they must share the same goal.

- *Targets*: A nonempty set of agents or supra-agents that are evaluated by evaluators.
- *Beneficiaries*: A nonempty set of agents that use the evaluations, and thus, also share the same goal.
- *Third Parties*: A nonempty set of agents that acknowledge that some evaluators share the evaluation.

The first three sets are the same as for image. Here though, the theory introduces a third party agents group. This group shares the belief that a group of evaluators is endowed with the social evaluation. Third parties are the holders of the reputation, and often they completely include the set of evaluators. Third parties are those aware of the effects of reputation transmission and the ones that transmit reputation (so called *gossip*).

2.3 Towards the reputing agent

From the individual perspective, agents can be partially aware of such sets and can in fact act in all the roles.¹ We assume that our cognitive agent i is endowed with goals and beliefs and therefore, is able to generate evaluations about other agents. Thus, i acts as evaluator when performing epistemic decisions. As well, i can act as beneficiary when receives evaluations from a set of agents S . In this case, i knows that the agents in S act as evaluators. Also, when i decides to send its own evaluations to a set of agents D , i is aware that agents in D may act as beneficiaries, and that they will know that i is an evaluator. Curiously, i may not be aware that she is actually being targeted by others, but must be aware of such possibility and the consequences of achieving bad evaluations. Because of that, cognitive agents have the *motivation* to act accordingly to well-established social behaviors.

From the above conceptualization, the set of third-parties and the set of evaluators do not necessary coincide. In fact, only designers of virtual societies could completely discriminate both sets. In this paper, we focus on agents that can act both as evaluators and third-parties. We deal with an agent architecture that is capable of evaluating and managing other agents' behaviors (evaluator) and capable as well to acknowledge reputation (third-party agent). We tackle then situations in which the set of evaluators is included in the set of third-parties.

From this perspective, when an individual agent accepts or acknowledges a given reputation it indicates that the agent assumes that the nested evaluation circulates in the society, that most of the members of the society (third-parties in this case) would acknowledge the existence of a circulating voice about the target. In the same way than agents may share a given evaluation with others without being aware of others existence [9], third-parties may share a given reputation (a metabelief).

At the individual level, when a third party transmits reputation information about a given target to a set of agents, it does not necessary believe the corresponding image of the target. This is because reputation moves to a level above of image, the belief about the circulation of an evaluation.

In general, distributed reputation and trust models deal with agents only partially endowed with the capabilities above. Instead, the Repage system [40] addresses them in a quite complete way, since its theoretical framework comes from the cognitive theory of reputation defined in [9] that we have defined briefly in this section. Repage is designed to be part of agents architectures as a system that computes image and reputation information (epistemic

¹ This is not necessary always the case. We can think about agents societies whose members are buyers or sellers (not both), and in which only buyers perform evaluations of sellers. In this case, the set of evaluators is the set of buyers, and the set of targets, the set of sellers. In more generic situations though targets and evaluator can coincide.

decisions), but does not tell anything about how to use them in a practical reasoning process, which leads to pragmatic–strategic decisions. As said before, the latter is the main topic of the paper. In the next section we explain how Repage models social evaluations as image and reputation, detailing the information involved.

2.4 Social evaluations in Repage

Social evaluations in Repage incorporate three elements: the target, the context, and the value of the evaluation [35,40]. For instance, an evaluation may say that an agent A (target), as a car driver (context or role) is very good (value). The original implementation of Repage maintains the value associated to a social evaluation as a tuple of five positive values (summing to one), that we call *weights*: $[w_1, \dots, w_5]$, defining a probabilistic distribution. Each value has an associated label in a rating scale: *Very Bad* (VB), *Bad* (B), *Neutral* (N), *Good* (G) and *Very Good* (VG). The following predicates capture the information that Repage generates in terms of image and reputation. Let i be the evaluator agent that uses Repage, j a target agent, and r a context or role:

- $Img(j, r, [w_1, w_2, \dots])$: Represents the image predicate that the evaluator i has about j in the role r (we assume that agent i is always the holder of the predicate, the evaluator, and thus we omit it from the predicate). For instance,

$$Img(j, seller, [0.8, 0.2, 0, 0, 0])$$

indicates that the image that the evaluator has about j is VB (very bad) with a weight of 0.8 and B (bad) with a weight of 0.2. Furthermore, the linguistic labels must be contextualized in each role. For instance, in the role seller, VB may indicate that the quality of the obtained product after interacting with j is below 20, and B that the quality goes from 20 to 40. Then, agent i may know that after interacting with j in the role seller, that i will obtain a product of quality below 20 with a probability of 0.8, and a product of quality between 20 and 40 with a probability of 0.2. To compute such distribution Repage uses the history of interactions of i with j in the role seller, and third-party image communications from other agents. Because of that, image predicates tend to indicate what the agent *believes*, even though this is not necessary true, because some agents may combine such information with reputation predicates (see Sect. 3).

- $Rep(j, r, [w_1, w_2, \dots])$: Represents a reputation predicate. The conceptualization of the predicate is the same as for image. However, in this case Repage only uses communicated reputations to compute the distribution. For instance,

$$Rep(j, seller, [0.8, 0.2, 0, 0, 0])$$

indicates that the evaluator knows that the reputation of j as a seller is VB (very bad) with a weight of 0.8 and B (bad) with a weight of 0.2. When the labels are contextualized and since Repage only uses communicated reputations to compute them, this predicate indicates that i is aware that most of the agents in the society *say* (in the sense of gossiping) that after interacting with j as a seller one obtains a product of quality below 20 with a probability of 0.8, and a quality between 20 and 40 with a probability of 0.2.

As said before, Repage follows the idea that image and reputation are distinct objects, and offers a complete specification on how such predicates are computed from communications and direct experiences. Appendix A shows briefly the internals of Repage, although we refer to [40] for a detailed explanation.

2.5 Repage in logic-based agents

In this section we have explained what we understand for social evaluation and how the Repage system captures them in terms of image and reputation predicates. Aforesaid, both concepts are evaluative beliefs and therefore the information is part of the belief base of the agent. If we want logic-based agents to use this information these higher-order predicates must be grounded to combine them with other beliefs (not coming from the Repage system) and with the desires and intentions of the agent. In the next section we define the belief logic that our BDI agent architecture uses to perform logical reasoning over beliefs that model social evaluations coming from Repage, and beliefs that model the general knowledge that the agent has gathered. The main characteristic of the logic is that it can deal with several independent probabilistic distributions.

3 Defining the belief logic

In this section we describe the language L_{BC} to express agents’ beliefs and to reason about them. The language must be able to capture the semantics that Repage predicates bring over formulas. Since a social evaluation in Repage describes a behavior of a target agent in a role as a probability distribution, L_{BC} must capture probabilities over some underlying language of the agents’ ontology.

Agents also need to perform basic epistemic inferences. In general, agents observe and interact with the environment, incorporating knowledge to their respective bases. Obviously, we focus on the knowledge that comes from Repage system, which provides evaluations in terms of probabilities and that can be combined with other knowledge of the agent through logical inferences. This allows the agents to combine such knowledge with their desires to finally generate intentions and act in consequence to fulfill them. The idea is that L_{BC} must capture all the knowledge that agents believe at a given instant of time.

To define L_{BC} we use the approach described in [19] where languages are structured as a hierarchy. A different approach that also uses hierarchies of languages is the one taken by [17], that could be alternatively used for our purposes. Both works suggest that first-order logic is enough to define consistent theories of propositional attitudes for rational agents. In these papers, formulas φ from a certain propositional language A can be *embedded* into another language B as constants for the language, usually written as $[\varphi]$. For instance, we can have a language that describes possible weather events in cities: $Rain(Barcelona)$, $Sunny(Rome) \wedge Sunny(Berlin)$, and another language can talk about these events in terms of date/time: $Forecast(10/11/2010, [Rain(Barcelona)])$.

3.1 Preliminaries: an intuitive idea

We want to illustrate with an example the kind of reasoning we are expecting from the logic of belief. First, we recall that the Repage system provides probability distributions over the different roles that an agent plays. For example, in a scenario with buyers and sellers, a buyer can decide to evaluate sellers in two *roles*: the quality of the products they sell and the delivery time of the products.

Role	Possible outcomes				
<i>Seller(Quality)</i>	<i>VeryGood_Quality</i>	<i>Good_Quality</i>	<i>Neutral_Quality</i>	<i>Bad_Quality</i>	<i>VeryBad_Quality</i>
<i>Seller(dTime)</i>	$dTime \leq 5$	$5 < dTime \leq 10$	$10 < dTime$		

Note that the possible outcomes for each role cover all the possibilities. How such information is finally codified as beliefs is one of the contributions of this work and it is explained in detail later. For the example, it is enough to realize that part of the information that the agent manages comes from an evaluation process that the Repage provides, while other comes from the general knowledge of the agent. This is what justifies such integration.

In our model, the desires of our agent i lead the practical reasoning process. The main idea is that for each desire, the belief logic should determine which *actions* allow the agent to achieve the desire and with which probability. For instance, agent i can desire the following with a strength of 0.9

$$(D^+(VeryGood_Quality \vee Good_Quality) \wedge dTime \leq 5 \wedge payLess(500), 0.9)$$

indicating that i desires to obtain a very good or good quality product delivered in less than 5 days and paying less than 500. Then, the logic of beliefs should provide which actions are capable of producing it. In concrete we would like the system to provide beliefs like

$$\begin{aligned} B(buy(Bob), (VeryGood_Quality \vee Good_Quality) \\ \wedge dTime \leq 5 \wedge paidLess(500), 0.45) \end{aligned} \quad (1)$$

$$\begin{aligned} B(buy(Alice), (VeryGood_Quality \vee Good_Quality) \\ \wedge dTime \leq 5 \wedge paidLess(500), 0.8) \end{aligned} \quad (2)$$

$$\begin{aligned} B(buy(Charlie), (VeryGood_Quality \vee Good_Quality) \\ \wedge dTime \leq 5 \wedge paidLess(500), 0.4) \end{aligned} \quad (3)$$

For instance, (1) indicates that after executing the action $buy(Bob)$, agent i will obtain

$$(VeryGood_Quality \vee Good_Quality) \wedge dTime \leq 5 \wedge paidLess(500)$$

with a probability of 0.45. The belief logic should deduce such information from more simple beliefs. For example, to deduce (1) agent i can hold the following predicates:

$$B(buy(Bob), (VeryGood_Quality \vee Good_Quality), 0.9) \quad (4)$$

$$B(buy(Bob), dTime \leq 5, 0.5) \quad (5)$$

$$B(buy(Bob), paidLess(500), 1) \quad (6)$$

Our approach suggests that formulas like (4) and (5) are *generated* from Repage. Note that (4) comes from the evaluation that agent i has about Bob in the role $Seller(Quality)$, while (5) from the evaluation of the same agent Bob in the role $Seller(dTime)$. The key idea is that Repage gives a probability distribution for each agent and role, and such probabilities can be combined under the assumption that distributions are stochastically independent. Then, the system should be able to infer from (4) and (5) the following:

$$B(buy(Bob), (VeryGood_Quality \vee Good_Quality) \wedge dTime \leq 5, 0.45) \quad (7)$$

where the probability of $0.45 = 0.9 \times 0.5$ is calculated following the standard probability computation for independent events. Also, the system should know that if a formula is always true (probability 1), like the case of (6) and it does not belong to any particular distribution, it can be combined using conjunction, to finally generate (1). Also, the formula (6) should

be calculated from the knowledge that i has about how much it cost to buy at Bob . In this sense, it is feasible and reasonable to assume that i should deduce (6) from:

$$B(\iota, \text{buy}(Bob), \text{paid}(350), 1) \quad (8)$$

$$B(\iota, \text{paid}(350) \rightarrow \text{paidLess}(500), 1) \quad (9)$$

where ι stands for an empty action. The beliefs are able then to codify knowledge that always holds after an action is executed (like formula (8)) and knowledge that always holds independently from the action (like formula (9)). If we want to keep an uniform notation, both kinds of formulas can be codified with probability 1.

The previous example illustrates the kind of reasoning we are looking for and the properties of the belief logic, which we enumerate in the following lines:

- (i) Evaluations from Repage codify the knowledge about the probabilities. This includes not only the assignment of probabilities for each agent and role, but the correct construction of the probability spaces. For instance, regarding Bob and role $Seller(dTime)$ the following beliefs could be generated:

$$\begin{array}{l} B(\text{buy}(Bob), dTime \leq 5, 0.5) \\ B(\text{buy}(Bob), 5 < dTime \leq 10, 0.3) \\ B(\text{buy}(Bob), 10 < dTime, 0.2) \\ \hline B(\text{buy}(Bob), dTime \leq 10, 0.8) \\ B(\text{buy}(Bob), dTime \leq 5 \vee 10 < dTime, 0.7) \\ B(\text{buy}(Bob), 5 < dTime, 0.5) \\ \hline B(\text{buy}(Bob), dTime \leq 10 \vee 10 < dTime, 1) \end{array}$$

- (ii) Repage provides evaluations for each agent and role in terms of image and reputation, which define two probabilistic distributions over the same agent and role that must be combined to finally generate beliefs. To avoid inconsistencies, we introduce besides the belief predicate B two more predicates, E (image) and S (reputation). Through the appropriate axioms we combine them to finally generate beliefs that do not fall into inconsistencies.
- (iii) When combining two formulas, in order to preserve a correct semantics and accuracy of the probabilities, we only can ensure that the resulting probability is correct when such formulas refer to the same action (so, the same agent) and talk about different roles, which we assume are stochastically independent. For this we need to codify into the belief predicates also the roles that are involved in the formula, and permit the combination of beliefs only when the intersection of such set of roles is empty. For instance, following the above example, the beliefs should be codified in the following way:

$$\begin{array}{l} B(\text{buy}(Bob), dTime \leq 5, 0.5, \{Seller(dTime)\}) \\ B(\text{buy}(Bob), 5 < dTime \leq 10, 0.3, \{Seller(dTime)\}) \\ B(\text{buy}(Bob), 10 < dTime, 0.2, \{Seller(dTime)\}) \\ \hline B(\text{buy}(Bob), dTime \leq 10, 0.8, \{Seller(dTime)\}) \\ B(\text{buy}(Bob), dTime \leq 5 \vee 10 < dTime, 0.7, \{Seller(dTime)\}) \\ B(\text{buy}(Bob), 5 < dTime, 0.5, \{Seller(dTime)\}) \\ \hline B(\text{buy}(Bob), dTime \leq 10 \vee 10 < dTime, 1, \{Seller(dTime)\}) \end{array}$$

Then the belief

$$B(\text{buy}(Bob), dTime \leq 5, 0.5, \{Seller(dTime)\})$$

can be combined with

$$B(\text{buy}(\text{Bob}), (\text{VeryGood_Quality} \vee \text{Good_Quality}), 0.9, \{\text{Seller}(\text{Quality})\})$$

because the intersection of the respective set of roles is empty and the action is the same. The resulting conjunction could be:

$$\begin{aligned} & B(\text{buy}(\text{Bob}), (\text{VeryGood_Quality} \vee \text{Good_Quality}) \wedge d\text{Time} \\ & \leq 5, 0.45, \{\text{Seller}(\text{Quality}), \text{Seller}(d\text{Time})\}) \end{aligned}$$

Note that we could also combine them with a disjunction:

$$\begin{aligned} & B(\text{buy}(\text{Bob}), (\text{VeryGood_Quality} \vee \text{Good_Quality}) \vee d\text{Time} \\ & \leq 5, 0.95, \{\text{Seller}(\text{Quality}), \text{Seller}(d\text{Time})\}) \end{aligned}$$

where $0.95 = 0.9 + 0.5 - 0.45$ is calculated following standard probabilistic computations. In both cases the set of roles is the same, since it is an indication of the roles that effect the formula. This mechanism prevents the logic to combine formulas which are not independent, so that the intersection of their respective set of roles is not empty.

The following subsection formalizes the syntax and semantics of the belief logic L_{BC} .

3.2 L_{BC} syntax and semantics

Following [19] we define two languages. The first one, denoted by L_{basic} , is the object language. L_{basic} is a classical propositional language that contains the symbols needed by the agents for writing statements about the application domain. The second language, denoted by L_{BC} , is the language the agents use to reason about beliefs, image and reputation. L_{BC} is a first-order many-sorted language that contains constant symbols for the formulas of the language L_{basic} .

For instance, in the example stated above, L_{basic} could be composed of the set of elementary propositions that we use to describe the possible outcomes of each role: VeryGood_Quality , Good_Quality , \dots , $d\text{Time} \leq 5$, $5 < d\text{Time} \leq 10$, \dots and the propositions $\text{Paid}(X)$ and $\text{PaidLess}(X)$ for each rational number X . Then, the language is constructed with the standard syntax of propositional logic that includes the symbols \neg , \wedge , \vee and \rightarrow necessary to express the base domain, as shown in the example.

L_{BC} is a first-order many-sorted language and contains four sorts:

- S_A : the sort representing actions.
- S_F : the sort representing formulas of the language L_{basic} .
- S_R : the sort representing the power set of roles.
- S_P : the sort representing probability values.

We use different letters for variables of different sorts of L_{BC} :

- a, a_1, a_2, \dots for variables of sort S_A
- x, x_1, x_2, \dots for variables of sort S_F
- r, r_1, r_2, \dots for variables of sort S_R
- p, p_1, p_2, \dots for variables of sort S_P .

Constants and predicate symbols of L_{BC} are identified by their sorts. The sort S_A includes a finite set of constant symbols C_A to denote actions. It also contains the constant ι to denote the special empty action. The sort S_F includes a set of constant symbols C_F to denote all formulas of the language L_{basic} . The set C_F contains constants of the form $[\sigma]$, where σ is

a formula of L_{basic} . The sort S_P includes a set of constant symbols C_P to denote rational numbers in the unit interval $[0, 1] \cap \mathcal{Q}$. For each $p \in [0, 1] \cap \mathcal{Q}$ we introduce the constant \overline{p} in the sort. However, in general, for the sake of clarity, we omit the overline notation for rational constants. Finally, the sort S_R includes a finite set of constant symbols C_R to denote finite sets of roles.

Before we proceed with the introduction of the L_{BC} syntax, it is important to remark two questions with respect to our notation. On the one hand, note that the symbols x, x_1, x_2, \dots are for variables of sort S_F in general, while the symbols $[\varphi]$ are constants of sort S_F that denote only formulas of the language L_{basic} . On the other hand, given a finite set of roles $\delta = \{R_1, \dots, R_l\}$, we have in C_R a constant, say c , denoting this set of roles. When we introduce the axiomatization of the logic, we use the notation $\mathcal{E}(c)$ to refer to the set δ denoted by constant c . For the sake of clarity we use sometimes the set of roles instead of the constant in some axioms. For instance, if the constant c denotes the set of roles

$$\{seller(quality), seller(dTime)\}$$

we will write the latter instead of c .

Now we specify the predicate symbols corresponding to various sorts. In the notation introduced below, the predicate symbol B , for instance, is written $B(S_A, S_F, S_P, S_R)$. This means that B is a predicate symbol of arity 4, with first argument in S_A , second argument in S_F , third argument in S_R and fourth argument in S_P . The language L_{BC} contains the following predicate symbols:

- Belief Predicate: $B(S_A, S_F, S_P, S_R)$.
- Image Predicate: $E(S_A, S_F, S_P, S_R)$.
- Reputation Predicate: $S(S_A, S_F, S_P, S_R)$.

L_{BC} contains various function symbols, that allow us to deal with parts of the agents' formulas and to express the reasoning of the agents. The functions applied to the sort S_F are one unary function $neg : S_F \rightarrow S_F$ for the negation of formulas, and the binary functions $con : S_F \times S_F \rightarrow S_F$ for conjunctions and $imp : S_F \times S_F \rightarrow S_F$ for implications.

For instance, if $[\varphi], [\phi] \in C_F$ then $imp([\varphi], [\phi])$ is interpreted as $[\varphi \rightarrow \phi]$, $con([\varphi], [\phi])$ as $[\varphi \wedge \phi]$, and $neg([\phi])$ as $[\neg\phi]$. The expression $or(x, y)$ stands for $\neg(con(\neg(x), \neg(y)))$. At first sight, all these functions can be regarded as purely syntactic transformations, but they are important in our construction because they allow us to write sentences that talk about parts of the formulas of L_{basic} .

The semantics of L_{BC} is the usual for a first-order many-sorted language. In this section we have presented only a few definitions and notation. A detailed introduction to the syntax and semantics of first-order many-sorted logics can be found in [12].

3.3 The basic axioms

In this section we define a theory Γ over L_{BC} , i.e. the axioms that agents use to reason. The theory contains the minimal formulas to describe the behavior of the predicates introduced above. We assume that the function product, sum and subtraction are defined for rational constants: $* : S_P \times S_P \rightarrow S_P$ where $* \in \{., +, -\}$. Remark that we are not giving an axiomatization of the logic, but only a set of axioms for a theory (a set of sentences in this first-order language closed under the logical consequence relation). For that reason we do not need to introduce inference rules. We assume that the deductive system is given (for instance, as defined in [12]).

RA: Conjunction

$$\forall ax_1x_2p_1p_2r_1r_2(B(a, x_1, p_1, r_1) \wedge B(a, x_2, p_2, r_2) \rightarrow B(a, con(x_1, x_2), p_1 \cdot p_2, r_3))$$

when $\mathcal{E}(r_1) \cap \mathcal{E}(r_2) = \emptyset$ and r_3 denotes the union of the two sets of roles $\mathcal{E}(r_1) \cup \mathcal{E}(r_2)$. Intuitively speaking, the axiom indicates that when two formulas talk about independent distributions (so, disjoint set of roles) we can ensure that the joint probability is the product.

MP: Modus ponens

$$\forall ax_1x_2r(B(a, x_1, 1, r) \wedge B_i(a, imp(x_1, x_2), 1, r) \rightarrow B_i(a, x_2, 1, r))$$

Note that this axiom (formulated as in [19]) indicates that agents use modus ponens when reasoning with formulas of the object language L_{basic} . It is an axiom of a theory, not of the logic. We assume an standard axiomatization of many-sorted first-order logic (cf. [12]) in which the modus ponens rule holds for every formula of the language.

NE: Necessity axiom for actions

$$\forall axr(B(t, x, 1, r) \rightarrow B(a, x, 1, r))$$

The axiom ensures that when the agent believes that a formula is true with a probability 1, after whichever action is performed, the formula will be also true.

CO: Completeness of probability

$$\forall ax(B(a, x, p, r) \rightarrow B(a, neg(x), 1 - p, r))$$

This ensures that when an agent knows the probability of a formula, also knows its complementary. The axiom is interesting because it states that a formula and its complementary cover all the probabilistic space.

Moreover, note that given $e \in C_A$ a constant denoting an action, $[\varphi_1], [\varphi_2] \in C_F, d_1, d_2 \in C_P$ denoting rational numbers and $c_1, c_2 \in C_R$ denoting sets of roles, when $B(e, [\varphi_1], d_1, c_1)$ and $B(e, [\varphi_2], d_2, c_2)$ hold, $\mathcal{E}(c_1) \cap \mathcal{E}(c_2) = \emptyset$, and $c_3 \in C_R$ is a constant denoting the union of sets of roles $\mathcal{E}(c_1) \cup \mathcal{E}(c_2)$, the previous axiomatization accomplishes the additive property of probabilistic spaces²:

$$B(e, con([\varphi_1], [\varphi_2]), d_1, c_3) \wedge B(e, con([\varphi_1], neg([\varphi_2])), d_2, c_3) \rightarrow B(e, [\varphi_1], d_1 + d_2, c_1)$$

Also, under the same condition the disjunction of independent formulas ensures the standard calculus of probabilities:

$$B(e, [\varphi_1], d_1, c_1) \wedge B(e, [\varphi_2], d_2, c_2) \wedge B(e, con([\varphi_1], [\varphi_2]), d_3, c_3) \rightarrow B(e, or([\varphi_1], [\varphi_2]), d_1 + d_2 - d_3, c_3)$$

GBEL: Ground Beliefs

$$B(e_1, [\varphi_1], 1, c_\emptyset) \\ \vdots \\ B(e_n, [\varphi_n], 1, c_\emptyset)$$

Those are the beliefs that describe the general knowledge of the agent. Each a_k is an action (possibly also the empty action ι), and each $[\varphi_k]$ denotes a proposition, a conjunction of propositions or a rule of the form $(\varphi_1 \wedge \varphi_m) \rightarrow \varphi$ from L_{basic} . The probabilistic distributions

² The standard formulation in probability theory is $Pr(A \cap B) + Pr(A \cap \bar{B}) = Pr(A)$.

are given by the predicates E and S interpreted by the Repage system, which ensures that the distributions are correct. To avoid inconsistencies we require that all the propositions are positive. c_{\emptyset} denotes the empty set of roles.

GI: Ground Images

Let $\varphi_1 \dots \varphi_n$ be formulas of L_{basic} that completely define the space of a distribution corresponding to role R . Let also $e \in C_A$ be a constant denoting an action and $d_1, \dots, d_4 \in C_P$. Then the following formulas are in the theory:

$$\begin{aligned}
 & E(e, [\varphi_1], d_1, \{R\}) \\
 & E(e, [\varphi_2], d_2, \{R\}) \\
 & \vdots \\
 & E(e, [\varphi_n], d_n, \{R\}) \\
 & E(e, [\varphi_1 \vee \varphi_2], d_1 + d_2, \{R\}) \\
 & E(e, [\varphi_1 \vee \varphi_3], d_1 + d_3, \{R\}) \\
 & \vdots \\
 & E(e, [\varphi_2 \vee \varphi_3], d_2 + d_3, \{R\}) \\
 & E(e, [\varphi_2 \vee \varphi_4], d_1 + d_4, \{R\}) \\
 & \vdots \\
 & E(e, [\varphi_1 \vee \varphi_2 \vee \varphi_3], d_1 + d_2 + d_3, \{R\}) \\
 & E(e, [\varphi_1 \vee \varphi_2 \vee \varphi_4], d_1 + d_2 + d_4, \{R\}) \\
 & \vdots \\
 & E(e, [\varphi_1 \vee \varphi_2 \vee \dots \varphi_n], 1, \{R\})
 \end{aligned}$$

They describe the full probabilistic space with the constraint that the disjunction of all the propositions belonging to the distribution corresponding to role R covers the complete space. For the kind of reasoning we want to perform, this is enough.

GR: Ground Reputations

Let $\varphi_1 \dots \varphi_n$ be formulas of L_{basic} that completely define the space of a distribution corresponding to role R . Let also $e \in C_A$ be a constant denoting an action and $d_1, \dots, d_4 \in C_P$. Then the following formulas are in the theory:

$$\begin{aligned}
 & S(e, [\varphi_1], d_1, \{R\}) \\
 & S(e, [\varphi_2], d_2, \{R\}) \\
 & \vdots \\
 & S(e, [\varphi_n], d_n, \{R\}) \\
 & S(e, [\varphi_1 \vee \varphi_2], d_1 + d_2, \{R\}) \\
 & S(e, [\varphi_1 \vee \varphi_3], d_1 + d_3, \{R\}) \\
 & \vdots \\
 & S(e, [\varphi_2 \vee \varphi_3], d_2 + d_3, \{R\}) \\
 & S(e, [\varphi_2 \vee \varphi_4], d_1 + d_4, \{R\}) \\
 & \vdots \\
 & S(e, [\varphi_1 \vee \varphi_2 \vee \varphi_3], d_1 + d_2 + d_3, \{R\}) \\
 & S(e, [\varphi_1 \vee \varphi_2 \vee \varphi_4], d_1 + d_2 + d_4, \{R\}) \\
 & \vdots \\
 & S(e, [\varphi_1 \vee \varphi_2 \vee \dots \varphi_n], 1, \{R\})
 \end{aligned}$$

IRB: Image-Reputation-Belief

Finally, the following axiom scheme combines E and S predicates over the same action, formula and distribution to generate beliefs. Depending of how we define the axioms, we can model different kinds of agents. The most general case is:

$$\forall x p_1 p_2 r (E(a, x, p_1, r) \wedge S(a, x, p_2, r)) \rightarrow B(a, x, h(p_1, p_2), r)$$

where $h : [0, 1] \cap \mathcal{D} \times [0, 1] \cap \mathcal{D} \rightarrow [0, 1] \cap \mathcal{D}$ is a function that combines the probabilities and preserves the probability distribution properties. An example of such a function could be the average, or weighted average function in order to give more importance to image or reputation information. Next section discusses it in more detail.

Equality Predicate

For all formulas φ, ϕ of L_{basic} the theory contains the following:

$$\begin{aligned} neg(\lceil \varphi \rceil) &= \lceil \neg \varphi \rceil \\ imp(\lceil \varphi \rceil, \lceil \phi \rceil) &= \lceil \varphi \rightarrow \phi \rceil \\ con(\lceil \varphi \rceil, \lceil \phi \rceil) &= \lceil \varphi \wedge \phi \rceil \end{aligned}$$

We must include them to ensure the completeness with respect to our intended semantics.

3.4 The basic semantics

In this subsection we show that the set of axioms presented above defines a first-order theory (say Γ) that is consistent. We do it by showing that the theory has, at least, a model that contains a set of positive atoms that exist in the model. Such model represents the reasoning process that the agent follows to deduce belief predicates. Following a similar approach than [19], we consider only models that contain ground terms of the language, so Herbrand models.

Proposition 1 *The theory Γ has a minimal model \mathcal{M} for any underlying language L_{basic} .*

Proof To prove it, we construct \mathcal{M} by induction following a *stratification* construction of the model. The main idea is to add the minimal number of atoms that accomplish the axioms, starting from the atoms that must be present in all the models, i.e. GBEL (ground beliefs), GI (ground images), GR (ground reputation) and the equality predicates for terms and rational numbers, and continuing by induction. Like in the construction of models used in logical programming, the strata k ($k \geq 1$) of the model includes all the generated atoms that require the application of at least k axioms to be created. Thus, the ground atoms generated from the axioms GBEL, GI, GR and equality predicates are in the first strata, and belong to the model \mathcal{M} (note that they are all positive), becoming the starting point of the construction. In the induction step we assume that \mathcal{M} already contains the atoms until the strata k . The generation of the strata $k + 1$ is done by applying any relevant axiom to the atoms already in \mathcal{M} .

The *application* of axioms in the induction step implies to add the minimal number of atoms that satisfy each axiom. We do not show the details on how each axiom creates and add new atoms. However, we illustrate it with the axiom RA (conjunction). Let us assume that the following ground atoms are already in the model.

$$B(\text{buy}(\text{john}), \text{VeryGood_Quality}, 0.9, \{\text{seller}(\text{quality})\})$$

$$B(\text{buy}(\text{john}), d\text{Time} \leq 5, 0.5, \{\text{seller}(d\text{time})\})$$

Then, to preserve the consistency of the model, the axiom RA is applied, and then, the following atom must be included into the model:

$$B(\text{buy}(\text{john}), \text{VeryGood_Quality} \wedge d\text{Time}, 0.45, \{\text{seller}(\text{quality}), \text{seller}(d\text{Time})\})$$

Under the assumption that GI and GR are well-constructed, so, they define correct probabilistic distributions, and that all GBEL axioms contain positive propositions, the construction of the model can be done for any underlying L_{basic} without falling into inconsistencies. \square

Given ground beliefs (GBEL), ground images (GI) and ground reputations (GR), the construction of the model \mathcal{M} gives us the belief formulas that the agent holds. Note that one and only one model exists, because all the axioms are universally quantified and do not contain disjunctions.

Also, note that under the assumption that GI and GR define correct probabilistic distributions, the axiomatization models the behavior of probability spaces for each role, and the combination of them when they are independent (different roles are involved). This is what the axiom IRB (image-reputation-belief) ensures.

3.5 Related work

Some current state-of-the-art logics inspired us for defining the logic. The probabilistic and dynamic notions have been mostly treated in epistemic logic [13,24], and in a simpler way in belief logic [6]. Propositional probabilistic variants of dynamic logic have been studied with the goal of analyzing probabilistic programs (for instance [25]).

Furthermore, some formalizations of trust using belief logic have been done [26], where trust is related to information acquisition in multi-agent systems, but in a crisp way. Similar to this, in [27], modal logic is used to formalize trust in information sources, also with crisp predicates. Here, actions and communicated formulas are also used.

Regarding fuzzy reasoning on trust issues, in [14] a trust management system is defined in a many-valued logic framework where beliefs are graded. Also, in [10] it is proposed a logic that integrates reasoning about graded trust (on information sources) and belief fusion in multi-agent systems. Our logic does not use graded beliefs. Instead, we use the notion of beliefs on probability sentences, since as we stated in Sect. 2, Repute social evaluations describe probabilities on the outcomes of future direct experiences.

Finally, in [36] a probabilistic dynamic belief logic is defined for dealing also with image and reputation notions. In this logic, beliefs and actions are considered normal modalities while probability predicates are considered non-standard modalities. In [36] only the expressiveness of the logic is explored.

Notice that we could have extended any of the previous logics (or other formalisms such as Gabbay's labelled deductive systems) to fulfill our original necessities. However, we wanted a very flexible logical framework with a very clear orientation towards possible implementations. Even when first-order logic is semi-decidable and it is not possible to guarantee very low complexities, it is indisputable that restricting the logic to Horn clauses together with other minimal assumptions, would ensure an easy adaptation to logic programming platforms.

4 Grounding image and reputation to L_{BC}

In this section we show how L_{BC} is capable of capturing image and reputation predicates from Repage, and how such information is transformed into the beliefs of the agent.

4.1 Image and reputation predicates

As explained in Sect. 2, image and reputation predicates computed from Repage are captured by the following expressions

- $Img(j, r, [V_{w_1}, \dots, V_{w_m}])$
- $Rep(j, r, [V_{w_1}, \dots, V_{w_m}])$

corresponding to the image and reputation of agent j playing the role r , from the point of view of the evaluator. We mention that the original implementation of Repage considers a tuple of five elements to represent the value of the evaluations. However, we generalize it, considering m ($m \geq 2$) elements. When in Repage the role and its labeled weights are defined, the role uniquely identifies an interaction model with two participants, and each w_k identifies a predicate, a formula from L_{basic} . To simplify, we can assume that the interaction model identified by a role is summarized in a single action.³ Thus, we presuppose the definition of a mapping $\mathcal{R}_{r,j}$ between a given role r and agent j to an action. In a similar way, we assume a mapping \mathcal{I}_{r,w_k} between each role r and label w_k to a formula written in L_{basic} .

We illustrate this with an example: In a typical market, the transaction of buying a certain product involves two agents, one playing the role of buyer (the evaluator) and the other playing the role of seller (j). From the point of view of the buyer, if she wants to evaluate other agents that play the role of seller, she knows that the associated action is *buy* at agent j . So, $\mathcal{R}_{seller,j}$ maps to *buy*(j). In the same way, the agent must know the meaning of each label w_k of Repage. Then, we can define that \mathcal{I}_{seller,w_1} is *veryBadProduct*, \mathcal{I}_{seller,w_2} is *okProduct*, etc.

In this mapping, the Repage predicate $Img(j, seller, [0.2, 0.3, \dots])$ indicates that the buyer believes that there is a probability of 0.2 that after executing the action $\mathcal{R}_{seller,j}$ (corresponding to the action *buy*(j)), she will obtain a \mathcal{I}_{seller,w_1} (*veryBadProduct*); with 0.3 that she will obtain \mathcal{I}_{seller,w_2} (*OKproduct*), etc. With reputation predicates the structure is similar, but the concept is different. In this case it indicates that the buyer believes that the corresponding evaluation is said by the agents in the group.

Following these indications, the representation of both predicates in L_{BC} is quite simple. Let j be an agent identifiers and r a role, then

$$\frac{Img(j, r, [V_{w_1}, V_{w_2}, \dots])}{\begin{matrix} E(\mathcal{R}_{rj}, \mathcal{I}_{r,w_1}, V_{w_1}, \{r\}) \\ E(\mathcal{R}_{rj}, \mathcal{I}_{r,w_2}, V_{w_2}, \{r\}) \\ \dots \end{matrix}} \quad \frac{Rep(j, r, [V_{w_1}, V_{w_2}, \dots])}{\begin{matrix} S(\mathcal{R}_{rj}, \mathcal{I}_{r,w_1}, V_{w_1}, \{r\}) \\ S(\mathcal{R}_{rj}, \mathcal{I}_{r,w_2}, V_{w_2}, \{r\}) \\ \dots \end{matrix}}$$

Repage ensures a correct probabilistic information in terms of a probabilistic distribution, and from these assignments it is easy to calculate the remaining disjunction probabilities necessary for the logical theory.

³ An interaction model can be seen as a set of actions to be performed by the agents.

As a matter of example and following the scenario above, let j_1, j_2 be agents, if Repage has generated the following predicates:

$$\begin{aligned} & \text{Img}(j_1, \text{seller}, [.1, .1, .1, .2, .5]) \\ & \text{Rep}(j_2, \text{seller}, [.6, .1, .1, .1, .1]) \end{aligned}$$

The logical theory should include regarding j_1

$$\begin{aligned} & E(\text{buy}(j_1), V \text{Bad Product}, 0.1, \{\text{seller}\}) \\ & E(\text{buy}(j_1), \text{Bad Product}, 0.1, \{\text{seller}\}) \\ & E(\text{buy}(j_1), OK \text{Product}, 0.1, \{\text{seller}\}) \\ & E(\text{buy}(j_1), \text{Good Product}, 0.2, \{\text{seller}\}) \\ & E(\text{buy}(j_1), \text{Bad Product}, 0.5, \{\text{seller}\}) \end{aligned}$$

And regarding j_2 :

$$\begin{aligned} & S(\text{buy}(j_2), V \text{Bad Product}, 0.6, \{\text{seller}\}) \\ & S(\text{buy}(j_2), \text{Bad Product}, 0.1, \{\text{seller}\}) \\ & S(\text{buy}(j_2), OK \text{Product}, 0.1, \{\text{seller}\}) \\ & S(\text{buy}(j_2), \text{Good Product}, 0.1, \{\text{seller}\}) \\ & S(\text{buy}(j_2), \text{Bad Product}, 0.1, \{\text{seller}\}) \end{aligned}$$

4.2 Relationship between image and reputation

One of the key points of Repage and the cognitive theory of reputation that underlies it [9] is the relationship between image and reputation. The theory states that both are social evaluations but distinct objects. With the representation we give for image and reputation in the L_{BC} and the axiomatization (the theory Γ), the difference depends on the relationship between the predicate E and the predicate S .

Regarding the key question: *How does reputation influence image?*, Conte and Paolucci in [9] state that the relation is established basically at the pragmatic–strategic level of the agent. At this level, agents must decide which source of information to use. Typically, reputation information is used only if image information is not present, but from this perspective, reputation cannot influence the inner beliefs of the agent. However, from our logical perspective, this relationship seems closer and is defined by the axiom IRB (Image-Reputation-Belief):

$$\forall x p_1 p_2 r (E(a, x, p_1, r) \wedge S(a, x, p_2, r)) \rightarrow B(a, x, h(p_1, p_2), r)$$

Different functions $h : [0, 1] \cap \mathcal{Q} \times [0, 1] \cap \mathcal{Q} \rightarrow [0, 1] \cap \mathcal{Q}$ model different behaviors. We only require that h preserves the probability distribution properties. Some elaborated aggregation functions can be found in [39], but basically, they are based on weighted averages. Thus, a family of functions is determined by the expression:

$$h(p_E, p_S) = \frac{\delta_E \cdot p_E + \delta_S \cdot p_S}{\delta_E + \delta_S}$$

where $\delta_E, \delta_S \in \mathcal{Q}_{\geq}$. Table 1 summarizes the behavior of a family of agents depending on the values of δ_E and δ_S . Note that h can be defined globally, as it is in the axiomatization, but

Table 1 Different h function classes when it is based on a weighted average: $h(p_E, p_S) = \frac{\delta_E \cdot p_E + \delta_D \cdot p_S}{\delta_E + \delta_S}$

Class	Condition	Description
\mathcal{H}_1	$\delta_E \neq 0, \delta_S = 0$	Only image—The agent does not trust in reputation information
\mathcal{H}_2	$\delta_E \neq 0, \delta_S = 0$	Only reputation—The agent does not trust in image information
\mathcal{H}_3	$\delta_E = \delta_S \neq 0$	The agent considers that both sources of information have the same importance
\mathcal{H}_4	$\delta_E > \delta_S$	Image is more important than reputation
\mathcal{H}_5	$\delta_E < \delta_S$	Reputation is more important than image

we can have different functions for different distributions (roles). For instance, following the example above,

$$\begin{aligned} \forall x p_1 p_2 (E(a, x, p_1, \{Seller(Quality)\}) \wedge \\ S(a, x, p_2, \{Seller(Quality)\})) &\rightarrow B(a, x, h_q(p_1, p_2), \{Seller(Quality)\}) \\ \forall x p_1 p_2 (E(a, x, p_2, \{Seller(dTime)\}) \wedge \\ S(a, x, p_1, \{Seller(dTime)\})) &\rightarrow B(a, x, h_t(p_1, p_2), \{Seller(dTime)\}) \end{aligned}$$

where $h_q \in \mathcal{H}_2$ and $h_t \in \mathcal{H}_4$ (see Table 1 for a description of \mathcal{H}_2 and \mathcal{H}_4). This indicates that the evaluator does not trust its own experiences regarding the quality of the product and relies on reputation. Instead, regarding the delivery time the agent gives more importance to its own direct experiences. This configuration may look strange, but let us consider for instance an agent that is aware of its limitations regarding certain skills, or a robot agent that is aware that its sensors do not work well. In general, to establish this function on design time is quite difficult, because it requires precise knowledge of the society. Ideally, one can design metareasoning processes to establish the *best* function when the system is running in a real scenario. In fact, simple q-learning techniques suffices to some extend for this purpose [34].

5 Integrating Repage in a multi-context BDI agent

In the previous sections we have defined the language L_{BC} and a theory written in that language that expresses the reasoning process of the agent. We have also shown how the theory captures the semantics of image and reputation predicates coming from Repage, and how such information is combined to finally generate beliefs.

In this section, we propose a possible integration of Repage in a BDI agent.⁴ The underlying idea is to define a BDI agent, specified as a multi-context system, that uses the logic presented in Sect. 3 to describe the belief base of the agent. Then, such information would be combined with the desires of the agent and other functional components to generate intentions, which in turn would end up generating proper actions. In the first part of the section, we briefly introduce the notion of multi-context system and some of the related work regarding existent multi-context BDI specifications. The second part relies on the explanation of each element that compounds our BDI + Repage architecture.

⁴ A preliminary version of the model described in this section was originally published at [33].

5.1 Multi-context systems

Multi-context systems (MCS) provide a framework to allow several distinct theoretical components to be specified together, with a mechanism to relate these components [17]. These systems are composed of a set of contexts (or units), and a set of bridge rules. Each context can be seen as a logic and a set of formulas written in that logic. Bridge rules are the mechanisms to infer information from one context to another.

Giunchiglia and Serafini [17] proposed the following formalization of MCS: Let I be the set of context names, a MCS is formalized as $\langle \{C_i\}_{i \in I}, \Delta_{br} \rangle$:

- $C_i = \langle L_i, A_i, \Delta_i \rangle$, where L_i is a formal language with its syntax and semantics, A_i is a set of axioms and Δ_i the set of inference rules. Thus, L_i and A_i define an axiomatic formal system, a logic for the context C_i . Beside axioms, it is possible to include a theory T_i as predefined knowledge. All A_i , Δ_i and T_i are written in the language L_i .
- Δ_{br} is a set of bridge rules.

Bridge rules can be seen as inference rules among contexts. Each one has a set of antecedents (or preconditions) and a consequent (or postcondition). Then, when each formula in the antecedent is true in its respective context, the consequent becomes true as well (also in its context). A bridge rule is represented as follows:

$$\frac{C_{i_1} : \varphi_1, \dots, C_{i_n} : \varphi_n}{C_{i_x} : \varphi_x}$$

where $C_{i_k} : \varphi_k$ indicates that formula φ_k belongs to the context C_{i_k} , formulas $\varphi_1 \dots \varphi_n$ are the antecedents and φ_x is the consequent. Each φ_i is a formula that belongs to its respective context, and written in its own language. So, when the formulas $\varphi_1, \dots \varphi_n$ hold in their contexts, the formula φ_x is generated in the context C_{i_x} . However, we extend this approach by allowing in preconditions, comparisons between rational numbers. For this, the antecedent may include a set Q_1, \dots, Q_n (where $n \geq 0$) of extra conditions that must be evaluated as true to make the bridge rule applicable. Each Q_i has the form $r_1 \leq r_2$ where $r_1, r_2 \in \mathcal{Q}$ and \leq corresponds to the standard boolean comparison on rational numbers.

5.2 MCS and BDI agents

The use of MCS offers several advantages when specifying and modeling agent architectures [44]. From a software engineering perspective, MCS supports modular architectures and encapsulation. From a logical modeling perspective, it allows the construction of agents with different and well-defined logics, keeping all formulas of the same logic in their corresponding context. This increases considerably the representation power of logical agents, and at the same time, simplifies their conceptualization.

Also, the use of MCS to specify BDI is not new. The BDI architecture defined in [32] uses one context for each attitude; there is the belief context (B), the desire context (D) and the intention context (I). Each of them is equipped with a logic that corresponds to the premises that Rao and Georgeff [37] stated. Bridge rules among contexts determine the relationship between the attitudes and the type of agent: strong realism, realism and weak realism [37]. A communication context (C) is also included.

In [16], this specification is extended by means of a new commitment context, equipped with a deontic logic, creating then a new attitude of obligation. In [6] a multi-context BDI agent is specified and its attitudes are graded. Therefore, beliefs, desires and intentions are

multi-valued with grades from 0 to 1. For our BDI model, we take the logic defined for desires and intentions described in [6,7].

5.3 The multi-context BDI model

The specification of our BDI agent as a multi-context system is formalized with the tuple $Ag = \langle \{BC, DC, IC, PC, CC, RC\}, \Delta_{br} \rangle$. These correspond to Belief, Desire, Intention, Planner, Communication and Repute contexts respectively. The set of bridge rules Δ_{br} incorporates the rules 1, 2, 3, 4, P , Q and B (shown in Fig. 3) and the bridge rules A_I and A_R (shown in Fig. 2). Figure 1 shows a graphical representation of this multi-context specification. In the next sections we briefly explain each context and bridge rule.

5.3.1 Belief context (BC)

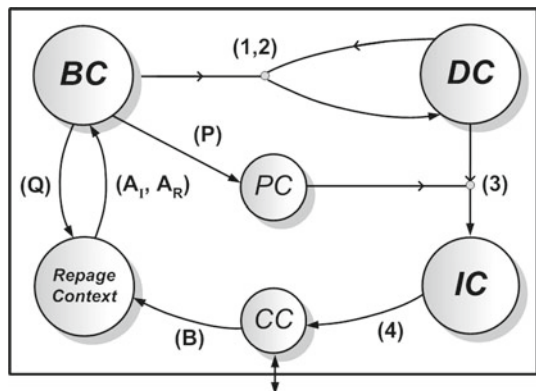
This context contains the beliefs of the agent. Hence, we use the logic introduced in Sect. 3, to integrate the knowledge coming from the reputation model Repute and other knowledge gathered by the agent. Since L_{BC} is a many-sorted first-order logic, the inference rules in this context are those from first-order logic. Thus, BC-context becomes an inference system that incorporates the theory defined in Sect. 3. Notice that this is how we have constructed the model for the theory.

5.3.2 Desire context (DC)

This context deals with the desires of the agent. Like the BDI model described by Rao and Georgeff in [37], they are attitudes that are explicitly represented and that reflect the general objectives of the agent. We consider that desires are graded, and for that, we use the multi-valued logic (DC -logic) based on the Lukasiewicz logic and described in [5]. The motivation for this decision arises when considering that reputation information has already a graded nature, in our case, represented as probabilities. Like in decision theory where agents manage expected utilities, we consider that from one side we obtain the probabilities, and from the other the strength of the desires. Combining them, we implement the idea of expected utility.

DC -language is built as an extension of a propositional language (in our case we use L_{basic} , the object language), by adding two fuzzy modal operators: D^+ and D^- . The intended meaning of $D^+\varphi$ is that the formula φ is desired by the agent holding it, and its truth degree, from

Fig. 1 The Repute context embedded in a multi-context BDI agent. Circles represent context and arrows represent bridge rules



0 (minimum) to 1 (maximum), represents the level of satisfaction if φ holds. The intended meaning of $D^-\varphi$ is that φ is negatively desired, and the truth degree represents the level of disgust if φ holds. Also, *DC*-logic includes truth constants \bar{r} where $r \in [0, 1] \cap \mathcal{Q}$, and the connectives $\&$ and \Rightarrow corresponding to the Lukasiewicz conjunction and implication respectively. In our architecture, agents' preferences are expressed by a set of desire expressions (both positive and negative) defining a theory.

We differentiate generic from concrete desires. Generic desires define the general preferences of the agent, and are formulas like $D^*\phi$, where $*$ stands for $+$ or $-$ and ϕ does not contain any action. Concrete desires are formulas like $D_\alpha^*\phi$ and define the desire to satisfy ϕ by executing action α . The original *DC*-logic from [5] does not consider subindex for the actions. However it uses this notation for the intentions (see next subsection). With this we indicate that a concrete desire takes into account the action to achieve the content. In this case, the grade represents the *expected* satisfaction level (or disgust if it is a negative desire) if the action is executed, implementing an equivalent *expected utility* from decision theory. Also, it serves to indicate that in the framework, actions do not behave as in dynamic logic. In our model, concrete desires are generated from generic desires and beliefs through bridge rules 1 and 2 (see Sect. 5.4.2).

Because in Lukasiewicz logic the formula $\phi \Rightarrow \varphi$ is 1-true iff the truth value of φ is greater or equal to that of ϕ , and the truth value of \bar{r} is exactly r , formulas like $\bar{r} \Rightarrow D^+\varphi$ in the theory of an agent i indicate that the level of *satisfaction* of agent i is at least r if φ holds. The same with negative desires and the level of *disgust*. From now on we will write these formulas as $(D^+\varphi, r)$ and $(D^-\varphi, r)$. The semantics is given in terms of a positive and negative preference distributions over the possible worlds. The axiomatization includes the classical logic axiom of propositional logic for non-modal formulas, plus the axioms of Lukasiewicz [20]. It is important to remark that the author defines the semantic condition that a world that is negatively desired to some extent cannot be positively desired. In terms of the axiomatization, this implies that the same formula cannot be both negatively and positively desired.

Note though that the inclusion of $D^-\varphi$ and $D^+\neg\varphi$ is completely valid. $D^-\varphi$ points out to the worlds that the agent does not want to reach, but this does not mean that he will try actively to avoid it. Instead, when we include $D^+\neg\varphi$ in the theory the agent will try to reach worlds where $\neg\varphi$ holds. We refer to [5] for technical details and proof of completeness of the logic.

5.3.3 Intention context (IC)

This context describes the intentions of the agent. Like in the Rao and Georgeff's BDI model [37], intentions are explicitly represented, but in our case generated from beliefs and desires. Also, we consider that intentions are graded, and for this we use the *IC*-logic defined in [6].

Similar to *DC*-logic, *IC*-logic is built on the top of a propositional language (in our case, the L_{basic}) defining a fuzzy modal operator to express formulas like $I_\alpha\varphi$. It indicates that the agent has the intention to achieve φ through the action α , and its truth degree (from 0 to 1) represents a measure of the trade-off between the benefit and counter-effects of achieving φ through α . Moreover, *IC*-logic is defined in terms of a Lukasiewicz logic in the same way as *DC*-logic. Also, formulas like $\bar{r} \Rightarrow I\varphi$ will be written as $(I\varphi, r)$. For the technical details and the proof of completeness we refer to [5].

Our system generates intentions through the bridge rule 3, from a positive concrete desire and the set of negative desires that may be achieved through the same action.

5.3.4 Planner context (PC) and communication context (CC)

The logic in the Planner context is a first-order logic restricted to Horn clauses. In this first approach, this context only holds the special predicate *action*, which defines a primitive action together with its precondition. We look forward to introducing plans as a set of actions in the future. Communication context is a functional context as well, and its logic is also a first-order logic restricted to Horn clauses with the special predicates *does* to perform actions, and *rec_jφ* to indicate that the agent has received the communication φ from agent *j*. They are first order predicates, not modalities

5.3.5 Repage context (RC)

The Repage context contains the Repage model. It is a functional context and we capture the information that the model computes with the predicates *Img(j, r, [V_{w1}, V_{w2}, ...])* and *Rep(j, r, [V_{w1}, V_{w2}, ...])*, corresponding to the image and reputation of agent *j* playing the role *r*. See Sect. 4 for a detailed analysis.

5.4 Bridge rules

5.4.1 Bridge rules A_I and A_R

Bridge rules A_I and A_R (see Fig. 2) are in charge of generating the corresponding *E* and *S* predicates from images and reputations respectively, as explained in Sect. 4. The key idea in this interface is that if the image or reputation information changes in Repage, the previously generated *E* and *S* predicates will not have the *support* to be valid any more, and thus, they must be out withdrawn from the theory (together with all the inferences performed so far from these predicates), placing the new ones instead. In this way, the theory is always consistent with the information that Repage computes.

5.4.2 Bridge rules 1, 2, 3, 4

Bridge rules 1 and 2 (see Fig. 3) transform generic desires to more concrete and realistic desires. To do this, these bridge rules merge generic desires from DC (with absolute values of satisfaction or disgust) with the information contained in BC, which includes the probability to achieve the desire by executing certain action. The result is a desire whose gradation has changed, becoming more realistic. This is calculated by the function *g*. If we define it as the product of both values, we obtain an expected level of satisfaction/disgust.⁵

Fig. 2 The bridge rules A_I and A_R (see Fig. 1). They translate image and reputation predicates respectively into the belief context

$$\begin{array}{l}
 \mathbf{A}_I: \frac{RC : \text{Img}(j, r, [V_{w_1}, V_{w_2}, \dots])}{\begin{array}{l} BC : E(\mathcal{R}_{rj}, \mathcal{I}_{r, w_1}, V_{w_1}, \{r\}) \\ BC : E(\mathcal{R}_{rj}, \mathcal{I}_{r, w_2}, V_{w_2}, \{r\}) \\ \dots \end{array}} \\
 \mathbf{A}_R: \frac{RC : \text{Rep}(j, r, [V_{w_1}, V_{w_2}, \dots])}{\begin{array}{l} BC : S(\mathcal{R}_{rj}, \mathcal{I}_{r, w_1}, V_{w_1}, \{r\}) \\ BC : S(\mathcal{R}_{rj}, \mathcal{I}_{r, w_2}, V_{w_2}, \{r\}) \\ \dots \end{array}}
 \end{array}$$

⁵ When *g* is defined as the product, the outcome is very similar to the notion of expected utility used in decision theory.

Fig. 3 The bridge rules 1, 2, 3, 4, P, Q and B (see Fig. 1)

$$\begin{array}{l}
 \mathbf{1:} \quad \frac{DC : (D^+ \varphi, d_\varphi) \quad BC : B(\alpha, \varphi, p_\psi, Q)}{DC : (D_\alpha^+ \varphi, g(d_\varphi, p_\psi))} \\
 \\
 \mathbf{2:} \quad \frac{DC : (D^- \varphi, d_\varphi) \quad BC : B(\alpha, \varphi, p_\psi, Q)}{DC : (D_\alpha^- \varphi, g(d_\varphi, p_\psi))} \\
 \\
 \mathbf{3:} \quad \frac{DC : (D_\alpha^+ \varphi, \delta), PC : action(\alpha, P), PC : P \quad DC : (D_\alpha^- \psi_1, \delta_{\psi_1}), \dots, (D_\alpha^- \psi_n, \delta_{\psi_n})}{\delta - \sum_{k=1}^n \delta_{\psi_k} \geq 0} \\
 \quad \quad \quad \frac{}{IC : (I_\alpha \varphi, f(\delta, \sum_{k=1}^n \delta_{\psi_k}))} \\
 \\
 \mathbf{4:} \quad \frac{IC : (I_\alpha \varphi, \varepsilon_{max})}{CC : does(\alpha)} \\
 \\
 P, Q, B: \quad \frac{BC : B\varphi}{PC : \varphi}, \quad \frac{BC : B\varphi}{RC : \varphi}, \quad \frac{CC : rec_j \varphi}{RC : rec_j \varphi}
 \end{array}$$

Bridge rule 3 generates intentions. It takes into account both the expected level of satisfaction and the cost of the action. At the same time, executing an action to achieve certain formula can generate undesirable counter-effects. Thus, bridge rule 3 also takes into account the possible negative desires that can be reached by executing this action. In this bridge rule, for each positive realistic desire (D^+), we must include all negative desires (D^-) that can result from the same action. In this way we have the value of the positive desire (δ^+) and the sum of all negative desires (δ^-) that can be achieved by executing the same action. The strength of the intention that is created is defined by a function f . Different f functions would model different behaviors. In our examples we use the following definition: $f(\delta^+, \delta^-) = \max(0, \delta^+ - \delta^-)$.

Finally, bridge rule 4 instantiates a unique intention (the one with maximum degree) and generates the corresponding action in the communication context.

5.4.3 Bridge rules P, Q and B

Bridge rules P and Q allow the planner and Repage context respectively to be aware of the beliefs of the agent. The planner context uses this information to build plans, actions and their preconditions. Repage uses the information to configure the mappings \mathcal{R} and \mathcal{T} .

Rule B reflects the reaction of the communication context once it receives communicated images, communicated reputation, third party images from other agents and fulfillment predicates. The content of these communications is directly introduced in Repage, which will update its information.

6 Putting the model to work

In this section we analyze the reasoning processes performed by an executable version of the model presenting an example.

The base scenario we use involves a BDI agent that, as a manager of a small restaurant, needs to periodically order wine to refill the stock. In this scenario, several providers are available. The information our agent wants to capture about them includes reliable

information, for instance the price she will have to pay, but also uncertain information such as the delivery time of the orders and the quality of the wine. While reliable information is introduced as beliefs of probability 1, uncertain information will result in beliefs of lower probability values.

This situation can be formalized in multiple ways. We can define four possible pairwise disjoint predicates for the quality of the wine: *poorWine*, *averageWine*, *goodWine*, *excellentWine* (pW , aW , gW and eW from now on) and five pairwise disjoint predicates for the delivery time: *days(0, 1)*, *days(2, 3)*, *days(4, 5)*, *days(6, 10)*, *days(11, ∞)* indicating respectively a delivery time up to 1 day, between 2 and 3 days etc. Also we define the predicates *paid(X)*, *paidLess(X)*, *paidMore(X)* to indicate that the agent has paid X , less than X and more than X respectively, and the implication relation $\text{paid}(X) \rightarrow \text{paidLess}(Y)$ when $X < Y$, and $\text{paid}(X) \rightarrow \text{paidMore}(Y)$ when $X > Y$. The predicate *budget(X)* indicates that the money she has in the budget is X . This knowledge and the implication among predicates must be introduced also as beliefs.

The interaction model defining the purchase of wine indicates that providers act as *wine-Sellers*, but agent i wants to evaluate them in the two independent dimensions: the quality of the wine and the delivery time. Thus, Repage uses the roles *wineSeller(quality)* and *wineSeller(dTime)*. The mapping \mathcal{R} (see Sect. 4.1) of these two roles points to the same action *buyWine* (*buy* from now on), which then summarizes the entire interaction model. The mapping \mathcal{T} of the role *wineSeller(quality)* relates w_1 to *poorWine*, w_2 to *averageWine* etc, and the mapping \mathcal{T} of the role *wineSeller(dTime)* relates w_1 with *days(0, 1)*, w_2 with *days(2, 3)*, etc.

6.1 The initial knowledge

In this world, our agent knows the existence of four providers represented by *alice*, *bob*, *charlie* and *debra* respectively. Our agent is aware of their prices, and so this knowledge is introduced as beliefs:

$$\begin{aligned} B(\text{buy}(\text{alice}), \text{hasWine} \wedge \text{paid}(1000), 1, e_{\emptyset}) \\ B(\text{buy}(\text{bob}), \text{hasWine} \wedge \text{paid}(900), 1, e_{\emptyset}) \\ B(\text{buy}(\text{charlie}), \text{hasWine} \wedge \text{paid}(400), 1, e_{\emptyset}) \\ B(\text{buy}(\text{debra}), \text{hasWine} \wedge \text{paid}(1300), 1, e_{\emptyset}) \end{aligned}$$

Bridge rule P introduces the information above into the planner context in order to generate the corresponding plans (simple actions in this case). It follows then, that in PC we find

$$\text{action}(\text{buy}(\text{alice}), \text{hasMoreMoney}(1000))$$

indicating that the action of buying wine from *alice* is preconditioned on the budget having more than 1000.

6.2 Study cases

6.2.1 Exploring the space: case 1

Our agent is new to the business and only *trusts* her own direct experiences. It means that axiom IRB uses a function h of the class \mathcal{H}_1 . Since she is just starting the business, she is

mostly concerned about the quality of the wine rather than the delivery time. She has a budget of 1350 ($budget(1350)$) for the purchase. Regarding her desires, she would be satisfied with paying up to 1350 for an excellent wine. With the same strength she would be satisfied paying up to 800 for a good wine. In any case, she needs the wine. What she does not want is a poor or average wine. Lower on her priority list is obtaining the wine quickly, but still a long delivery time is not desired. These preferences can be formalized as desires in the DC as follows:

$$\begin{aligned}
 &(D^+(hasWine \wedge paidLess(1350) \wedge eW), .9) \\
 &(D^+(hasWine \wedge paidLess(800) \wedge gW), .9) \\
 &(D^+hasWine, .7) \\
 &(D^-pW, 1) \\
 &(D^-aW, .8) \\
 &(D^-days(11, \infty), .5) \\
 &(D^-days(6, 10), .4)
 \end{aligned}$$

Since she does not have any information about the providers, Repage predicates contain the maximum possible uncertainty. For instance, the corresponding image predicates for *charlie* are:

$$\begin{aligned}
 &Img(charlie, wineSeller(quality), [.25, .25, .25, .25]) \\
 &Img(charlie, wineSeller(time), [.2, .2, .2, .2, .2])
 \end{aligned}$$

Under these conditions the reasoning process leads to a random choice between three agents (*charlie*, *bob* and *alice*) to achieve the desire *hasWine*. In the following lines we briefly explain the most relevant steps.

Bridge rule A_I generates beliefs in the BC from images. As said before, the epistemic decision is not done at this rule but inside Repage, which computes image and reputation. In the case of *charlie* this rule is activated regarding the role *wineSeller(quality)* as:

$$\begin{aligned}
 &\underline{RC : Img(charlie, wineSeller(quality), [.25, .25, .25, .25])} \\
 &BC : E(buy(charlie), pW, .25, \{wineSeller(quality)\}) \\
 &BC : E(buy(charlie), aW, .25, \{wineSeller(quality)\}) \\
 &BC : E(buy(charlie), gW, .25, \{wineSeller(quality)\}) \\
 &BC : E(buy(charlie), eW, .25, \{wineSeller(quality)\}) \\
 &\dots
 \end{aligned}$$

All possible outcomes after buying from *charlie* have the same probability. This rule also generates the probabilities of disjoint formulas:

$$\begin{aligned}
 &BC : E(buy(charlie), pW \vee aW, .50, \{wineSeller(quality)\}) \\
 &BC : E(buy(charlie), pW \vee gW, .50, \{wineSeller(quality)\}) \\
 &BC : E(buy(charlie), pW \vee eW, .50, \{wineSeller(quality)\}) \\
 &BC : E(buy(charlie), aW \vee gW, .50, \{wineSeller(quality)\}) \\
 &BC : E(buy(charlie), aW \vee eW, .50, \{wineSeller(quality)\}) \\
 &BC : E(buy(charlie), gW \vee eW, .50, \{wineSeller(quality)\}) \\
 &BC : E(buy(charlie), pW \vee aW \vee gW, .75, \{wineSeller(quality)\})
 \end{aligned}$$

$$\begin{aligned}
 BC &: E(\text{buy}(\text{charlie}), pW \vee gW \vee eW, .75, \{\text{wineSeller}(\text{quality})\}) \\
 BC &: E(\text{buy}(\text{charlie}), aW \vee gW \vee eW, .75, \{\text{wineSeller}(\text{quality})\}) \\
 BC &: E(\text{buy}(\text{charlie}), pW \vee aW \vee gW \vee eW, 1, \{\text{wineSeller}(\text{quality})\})
 \end{aligned}$$

The previous E predicates are directly transformed to B predicates through the axiom IRB, which uses a h function belonging to \mathcal{H}_1 (only images are taken into account). In BC, because of the assumption that the quality and delivery time dimensions are stochastically independent, probabilistic inference rules of the L_{BC} theory are applied. For example, from $B(\text{buy}(\text{charlie}), eW, .25, \{\text{wineSeller}(\text{quality})\})$ and $B(\text{buy}(\text{charlie}), \text{days}(0, 1), .2, \{\text{wineSeller}(\text{time})\})$ can be deduced

$$B(\text{buy}(\text{charlie}), eW \wedge \text{days}(0, 1), .05, \{\text{wineSeller}(\text{quality}), \text{wineSeller}(\text{time})\})$$

where .05 is the product of .25 and .2. In particular, and for the interest of our example, the following belief is also generated:

$$\begin{aligned}
 B(\text{buy}(\text{charlie}), \text{hasWine} \wedge \text{paid}(400) \wedge eW, .25, \\
 \{\text{wineSeller}(\text{quality}), \text{wineSeller}(\text{time})\})
 \end{aligned}$$

Bridge rules 1 and 2 are executed for each generic positive and negative desire respectively. For instance, rule 1 is fired for the first desire as follows:

$$\begin{aligned}
 DC &: (D^+(\text{hasWine} \wedge \text{paidLess}(1350) \wedge eW), .9) \\
 BC &: B(\text{buy}(\text{charlie}), \text{hasWine} \wedge \text{paidLess}(1350) \wedge eW, .25, Q) \\
 \hline
 (D_{\text{buy}(\text{charlie})}^+(\text{hasWine} \wedge \text{paidLess}(1350) \wedge eW), g(.9, .25))
 \end{aligned}$$

If we consider that $g(p, q) = p \cdot q$, the resulting grade of the positive concrete desire is .225. It indicates that performing the action of buying from *charlie* to obtain an excellent wine and paying less than 1350 has an expected level of satisfaction of .225. Of course, for the same desire bridge rule 1 can be executed several times because different actions can lead to the same desire. Negative desires fire bridge rule 2 generating concrete negative desires. They indicate the expected level of disgust if the action is executed.

These negative desires are used in bridge rule 3 to take into account possible counter-effects of satisfying certain desire. Rule 3 is executed only one time for each positive concrete desire. For example, considering the desire above:

$$\begin{aligned}
 DC &: (D_{\text{buy}(\text{charlie})}^+(\text{hasWine} \wedge \text{paidLess}(1350) \wedge eW), .225) \\
 DC &: (D_{\text{buy}(\text{charlie})}^-(\text{days}(11, \infty)), .08) \\
 DC &: (D_{\text{buy}(\text{charlie})}^-(\text{days}(6, 10)), .08) \\
 DC &: (D_{\text{buy}(\text{charlie})}^-(aW), .2) \\
 DC &: (D_{\text{buy}(\text{charlie})}^-(pW), .25) \\
 PC &: \text{action}(\text{buy}(\text{charlie}), \text{budgetMore}(400)) \\
 PC &: \text{budget}(1100) \rightarrow \text{budgetMore}(400) \\
 \hline
 IC &: (I_{\text{buy}(\text{charlie})}(\text{hasWine} \wedge \text{paidLess}(1350) \wedge eW), f(.225, .61))
 \end{aligned}$$

In this case, notice that the expected level of satisfaction of achieving the desire by buying from *charlie* is .225 but its counter-effects bring an expected level of disgust of .61. Taking $f(\delta^+, \delta^-) = \max(0, \delta^+ - \delta^-)$, this intention has a grade of 0. Why would we perform an action if we expected from it to obtain more disgust than benefit?

If the intention had the maximum degree, bridge rule 4 would generate the corresponding action. In our example, after calculation, the intentions with a grade higher than 0 result to be:

$$\begin{aligned} &(I_{buy(charlie)}hasWine, .14) \\ &(I_{buy(bob)}hasWine, .14) \\ &(I_{buy(alice)}hasWine.14) \end{aligned}$$

As expected, since Repage does not have any information and our agent needs to buy wine, a random choice can be made among these possibilities. Buying from *debra* is not considered because in rule 3 the precondition of having a budget greater than 1300 does not hold (see the action definition in the planner context). Assuming that she picks $(I_{buy(charlie)}, hasWine, .14)$, bridge rule 4 is fired executing the action $buy(charlie)$.

The result of this transaction fulfills the agent's desires in terms of delivery time and quality. This information is inserted into Repage by means of the bridge rule B. Repage evaluates the outcomes and updates the values of image and reputation. In the next reasoning process, this information will be introduced as beliefs by bridge rule A_I and A_R , as we have shown at the beginning of this case.

Continuing with our example, we suppose that *charlie* delivers the wine quite fast, in less than one day, but the quality of the wine is not very good. This makes Repage update image predicates as

$$\begin{aligned} &Img(charlie, wineSeller(quality), [.4, .4, .1, .1]) \\ &Img(charlie, wineSeller(time), [.45, .25, .1, .1, .1]) \end{aligned}$$

We recall here that w_1, w_2, \dots in the role $wineSeller(quality)$ correspond to pW, aW, \dots meanwhile in the role $wineSeller(time)$ they correspond to $days(0, 1), days(2, 3), \dots$ respectively.

6.2.2 Receiving reputation information: case 2

After a while, our agent needs to buy more wine. She has exactly the same desires as before and the same budget, so she is mainly interested in the quality of the wine rather than delivery time. But this time, her image information about *charlie* has changed. Furthermore, we assume that she has received several reputation communications, about both *charlie* and *alice*. This information makes Repage generate the following reputation predicates:

$$\begin{aligned} &Rep(charlie, wineSeller(quality), [.5, .3, .1, .1]) \\ &Rep(alice, wineSeller(quality), [.1, .2, .2, .4]) \end{aligned}$$

The reputation information regarding *charlie* coincides more or less with the image our agent has about him. This is not the case with *alice*. Through bridge rule A_R these predicates generate beliefs into BC. For *charlie*:

$$\begin{aligned} &RC : Rep(charlie, wineSeller(quality), [.5, .3, .1, .1]) \\ &BC : S(buy(charlie), pW, .5, \{wineSeller(quality)\}) \\ &BC : S(buy(charlie), aW, .3, \{wineSeller(quality)\}) \\ &BC : S(buy(charlie), gW, .1, \{wineSeller(quality)\}) \\ &BC : S(buy(charlie), eW, .1, \{wineSeller(quality)\}) \end{aligned}$$

Note that these beliefs refer to what others say, not what our agent really believes. Since our agent only *trusts* herself, she does not take into account these predicates. In terms of the *BC*-logic it indicates that there is no relationship between operator *S* and operator B_i so far. This situation is also common: we can accept that a given person has a bad reputation, that most people *say* this, even when we believe the opposite [9].

Under these conditions, the reasoning process is similar to the previous case. This time though, *charlie* is no longer a possible choice, since the last experience with him was bad regarding the quality of the wine. Bridge rule 3 generates the intention to buy from *charlie* with a very low grade, in fact zero, since it is likely a poor or average wine would be delivered. In this case, the generated intentions are

$$(I_{buy(bob)}hasWine, .14)$$

$$(I_{buy(alice)}hasWine, .14)$$

Our agent chooses *alice*. This time we suppose the result is in tune with the expectations of our agent; she obtains a good wine, even though the delivery time is not very fast. Repage updates image predicates regarding *alice* as follows:

$$Img(alice, wineSeller(quality), [0, 0, .15, .85])$$

$$Img(alice, wineSeller(time), [0, 0, 0, .1, .9])$$

6.2.3 Keeping the same desires: case 3

Maintaining the exact same desires as case 1 and 2, the next time that our agent wants to buy wine, she has the following intentions whose grade is higher than 0:

$$(I_{buyWine(bob)}hasWine, .14),$$

$$(I_{buyWine(alice)}hasWine, .35)$$

$$(I_{buyWine(alice)}(hasWine \wedge paidLess(1350) \wedge eW), .365)$$

Since *alice* provided wine that was mostly excellent, and this is the main concern of our agent, she chooses again to buy from *alice*, but to satisfy the desire $hasWine \wedge paidLess(1350) \wedge eW$. The option to buy from *bob* appears due to the uncertainty around his performance. We suppose that the resulting transaction confirms the same results as the previous case: an excellent wine but a long delivery time.

6.2.4 Changing desires: case 4

This time our agent accepts the suddenly request to host a big birthday banquet that will take place in less than 12 days. Her cellar is not prepared for this event, so, she needs to order more wine. In this situation, her desires are different, since delivery time is now a key issue while the quality of the wine drops in importance:

$$(D^+hasWine \wedge paidLess(1350) \wedge days(0, 1), .9)$$

$$(D^+hasWine \wedge paidLess(800) \wedge days(2, 3), .7)$$

$$(D^-pW, .2)$$

$$(D^-aW, .2)$$

$$(D^- \text{days}(11, \infty), .8)$$

$$(D^- \text{days}(6, 10), .7)$$

Thanks to her previous interactions with the providers our agent already has some information about their performance. In this case, the only intention with a degree higher than 0 is

$$(I_{\text{buyWine(charlie)}}(\text{hasWine} \wedge \text{paidLess}(1350) \wedge \text{days}(0, 1)), .095)$$

She picks *charlie*, and the results are like the first time she bought from him in case 1: a short delivery time but a low quality.

6.2.5 Using reputation information: case 5

Several weeks after the successful banquet, our agent recuperates her initial desires and needs to order wine again. During this time she has heard about both *bob* and *debra*'s reputations which indicates that both offer excellent wines and that furthermore *debra* is capable to deliver the order in a day. This is not the case with *bob*:

$$\text{Rep}(\text{bob}, \text{wineSeller}(\text{quality}), [0, 0, .05, .95])$$

$$\text{Rep}(\text{bob}, \text{wineSeller}(\text{time}), [.1, .2, .3, .3, .1])$$

$$\text{Rep}(\text{debra}, \text{wineSeller}(\text{quality}), [0, 0, 0, 1])$$

$$\text{Rep}(\text{debra}, \text{wineSeller}(\text{time}), [1, 0, 0, 0, 0])$$

This information is introduced through rule A_R as S predicates. Unfortunately for our agent, *alice* notifies that she will not be available this time because she will be on holidays. Because of that, and because the reputation information she received in case 2 was in concordance with what she really believed, our agent starts *trusting* what others gossip. In this new scenario, the IRB axiom is set to use a h function belonging to \mathcal{H}_2 (only reputation is taken into account). Thus, the axiom IRB states the following:

$$\forall x p_1 p_2 r (I(a, x, p_1, r) \wedge S(a, x, p_2, r)) \rightarrow B(a, x, p_2, r)$$

It means that reputation predicates from *Repage*, once they have been inserted into the BC-context as S predicates, they become belief predicates. For instance, regarding *bob* in the role of *wineSeller(quality)*, rule A_R generates, among others, the following predicate: $S(\text{buy}(\text{bob}), eW, .95, \{\text{wineSeller}(\text{quality})\})$, meaning that people is gossiping that with a probability of .95, the wine will be excellent when buying from *bob*. Since our agent *believes* what it gossiped due to axiom IRB, it can be deduced that $B(\text{buy}(\text{bob}), eW, .95, \{\text{wineSeller}(\text{quality})\})$. In this case, the only non-zero graded intention generated is

$$(I_{\text{buy}(\text{bob})}(\text{hasWine} \wedge \text{paidLess}(1350) \wedge eW), .565)$$

From the activation of bridge rule 3 as follows:

$$\begin{array}{l}
DC : (D_{buy(bob)}^+(hasWine \wedge paidLess(1350) \wedge eW), .0855) \\
DC : (D_{buy(bob)}^-(days(11, \infty), .08) \\
DC : (D_{buy(charlie)}^-(days(6, 10), .21) \\
PC : action(buy(bob), budgetMore(900)) \\
PC : budget(1100) \rightarrow budgetMore(900) \\
\hline
IC : (I_{buy(bob)}(hasWine \wedge paidLess(1350) \wedge eW), f(.855, .29))
\end{array}$$

We suppose in this situation that the results are not as the agent expects, obtaining an average wine. Thus, Repage image predicates are updated as:

$$\begin{array}{l}
Img(bob, wineSeller(quality), [.3, .4, .2, .1]) \\
Img(bob, wineSeller(time), [.1, .2, .3, .3, .1])
\end{array}$$

6.2.6 Image and reputation interference: case 6

Note that in the previous situation, the image about *bob* in the role *wineSeller(quality)* contradicts *bob*'s reputation in the same role. This has already happened in case 2 with *alice*, but axiom IRB was only taking into account image information. In this new case, we assume that the IRB uses a h function from the class \mathcal{H}_4 , where both image and reputation are taken into account but image is more important. As a matter of example, we set function h as

$$h(p_E, p_S) = \frac{7 \cdot p_E + 3 \cdot p_S}{10}$$

We show how the reasoning process proceeds. Regarding the role *wineSeller(quality)*, through bridge rule A_I the following E predicates are generated into the belief context:

$$\begin{array}{l}
E(buy(bob), pW, 0.3, \{wineSeller(quality)\}) \\
E(buy(bob), aW, 0.4, \{wineSeller(quality)\}) \\
E(buy(bob), gW, 0.2, \{wineSeller(quality)\}) \\
E(buy(bob), eW, 0.1, \{wineSeller(quality)\}) \\
\dots
\end{array}$$

and through bridge rule A_R the following:

$$\begin{array}{l}
S(buy(bob), pW, 0, \{wineSeller(quality)\}) \\
S(buy(bob), aW, 0, \{wineSeller(quality)\}) \\
S(buy(bob), gW, 0.05, \{wineSeller(quality)\}) \\
S(buy(bob), eW, 0.95, \{wineSeller(quality)\}) \\
\dots
\end{array}$$

Then, the presence of axiom IRB with the h function defined above combines both predicates generating a new probability distribution. In this case:

$$\begin{array}{l}
B(buy(bob), pW, 0.21, \{wineSeller(quality)\}) \\
B(buy(bob), aW, 0.28, \{wineSeller(quality)\}) \\
B(buy(bob), gW, 0.155, \{wineSeller(quality)\}) \\
B(buy(bob), eW, 0.355, \{wineSeller(quality)\}) \\
\dots
\end{array}$$

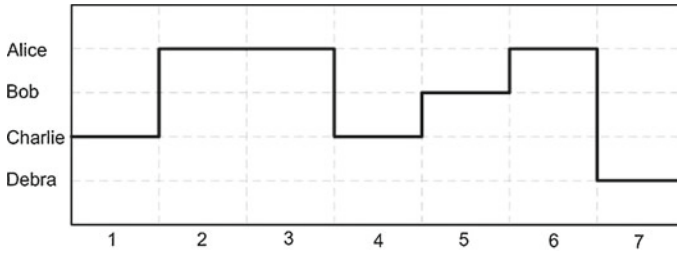


Fig. 4 The choices of the agent throughout the situations explained in this section

In this way we preserve the properties of probability distributions, reflecting in the resulting beliefs a combination of the both source of information: image and reputation from Repage.

Turning again to the example above, note that the resulting beliefs for *bob* presents a distribution that model an almost uncertain distribution, here values are close to 0.25. This make sense since image and reputation information regarding *bob* where quite contradictory. In this situation, our agent picks *alice* (Fig. 4).

6.2.7 Increasing the budget: case 7

To conclude, we want to show the effect of a simple environment change. In this case, our agent decides to increase the wine budget to 2000. With exactly the same desires and the same reputation and image information as before, the reasoning process generates the maximum intention to buy from *debra*. This provider was always filtered out at bridge rule 3 because the precondition of buying from *debra* (to have more than 1300) was never fulfilled. Thus, the intention to buy from *debra* is only slightly higher than buying from *alice*.

6.3 Implementation details

The scenario and each one of the situations have been implemented in Prolog.⁶ An implementation of logical systems usually entails the simplification or limitation of some aspects of the logic. In our case, we assume that each logical formula is expressed as a Horn clause and that modal operators are first-order predicates. Also, we do not accept logically omniscient agents that use a forward-reasoning engine, even when some implementations of multi-context systems use this approach [44]. Instead, we take advantage of the backward-reasoning engine of Prolog.

Note that the multi-context system specification of our BDI agent models an agent whose purpose is to execute a single action. This action is generated through rule 4 by choosing the intention of maximum grade. For this choice the agent must generate all possible intentions, which are created through rule 3 from desires, and so on. This schema follows a backward-reasoning algorithm that can be implemented in Prolog.

Thus, considering predefined knowledge as Prolog predicates, and inference rules and bridge rules as Prolog rules, the agent's reasoning can be started by asking Prolog to satisfy the predicate *does(A)*. While this is an oversimplification of what should be understood as multi-context systems, for simple examples the results are coherent and useful. We plan to

⁶ The source code can be download at <http://www.iia.csic.es/~ipinyol/sourceJAAMAS09.zip>.

study implementation issues in the future, and the effects of the simplifications in the desirable properties of the system.

7 Related work

To finish the paper we want to put our model in contrast with some of the current state-of-the-art reputation and trust models. Hence, in this section we compare our BDI + Repute model with other existing models by defining four dimensions of analysis that have not been explicitly tackled in the most popular reviews. In the review, we only explore distributed models, those that consider trust or reputation as a subjective information managed by each individual agent.

7.1 Trust dimension

Even when we have not explicitly defined a trust model, from a cognitive point of view, our BDI + Repute architecture can be considered a trust model, in the sense that there is a reasoning path that leads to a decision to rely in someone. We do not want to differentiate between models classified as *trust* and others as *reputation*. We strongly believe that the distinction between both *kinds* of models does not rely on a clear consensus in the community. For instance, the *type* dimension that Sabater provides in his classification [43] is not based on any objective fact, but on what the authors of the models claim.

On the contrary, when facing these concepts from a more cognitive perspective, the distinction becomes clearer. From the concept of social trust [8], occurrent and dispositional trust [21, 15] and pragmatic–strategic decisions pointed out by Conte and Paolucci in [9], *trust* implies a decision. Trust can be seen as a process of practical reasoning that leads to the decision to interact with somebody. Regarding this aspect, some models provide evaluations, rates, scores etc. for each agent to help the decision maker with a final decision. Instead, others specify how the actual decision should be made. From our point of view, only the latter cases can be considered trust models. We recall here that in this case, the decisions are also pragmatic–strategic, in the sense described in [9].

Table 2 summarizes the models that from our definition should be considered trust models. We mark them with ‘✓’. For instance, the model defined by Marsh [29] is a trust model because it indicates exactly to whom to interact with. The final decision is made through a well-defined threshold. Another example is the model defined by Sen and Sajja [46]. Even when this model is usually considered a reputation model, the fact is that it defines a decision making process that identifies to whom to interact with, and then, fits in our definition of trust.

Models marked with ‘–’ are those that we do not consider trust models. They calculate measures or evaluations to help a decision making process. For instance, the AFRAS model [3, 4] gives evaluations in terms of fuzzy sets, and the shape of these fuzzy numbers also determines a reliability measure. However, there is no mechanism that tells the agent how to use such evaluations. This situation is similar as in the Repute model. As explained before the model only gives support to the creation of image or reputation predicates, not how such information is combined.

Finally, we use the mark \sim to indicate that the model does not give an explicit decision mechanism, but that it is rather dependent on the current desires of the agent. For instance, the Regret model [41] provides for each agent and context a *trust* value, together with a reliability

Table 2 Computational models against our classification dimensions

Model	Trust	Cognitive	Procedural	Generality
Abdul-Rahman et al.	–	–	~	–
AFRAS	–	–	✓	✓
Castelfranchi et al.	✓	✓	–	✓
FIRE	~	–	✓	✓
ForTrust	✓	✓	–	✓
Marsh	✓	–	~	✓
LIAR	✓	–	✓	–
Regret	~	–	✓	✓
Repage	–	~	✓	✓
Schillo et al.	–	–	✓	✓
Sen and Sajja	✓	–	✓	–
Yu and Singh	✓	–	✓	–
BDI + Repage	✓	✓	✓	✓

measure. The trust value is calculated through aggregation of the information from several sources. One of the sources is defined by an ontology, which already determines which information is considered more important.⁷ Hence, the goals of the agent are somehow codified in this ontology, and the final trust value obtained is an indicator of which possible target agent matches better with the desires of the agent. However, since it offers a reliability measure the decision is not yet possible. For instance, let us assume that agent *a* has a trust value of 0.6 with a reliability of 1. On the other hand, another agent *b* has a trust value of 0.8 with a reliability of 0.4. Which is the best option? It still requires a decision making process. However, it is clear that with similar reliability measures, the agent with highest trust value is the chosen one. FIRE model [22] shows a similar situation.

7.2 Cognitive dimension

Although this dimension has already appeared in other surveys, the provided definitions are quite vague. In this dimension we differentiate models that have clear representations of trust, reputation, image etc. in terms of cognitive elements such as beliefs, goals, desires, intentions, etc. From our perspective, models that achieve such representation explicitly describe the epistemic and motivational attitudes that are necessary for the agents to have *trust* or to hold social evaluations. In this sense, in models that achieve a cognitive representation, final values of trust and reputation are as important as the structure that supports them. These models are usually very clear at the conceptual level, but lack in computational aspects.

Often, models that are not endowed with this property consider the model as a black box that receives inputs and issues trust and reputation *values*. Because of that, the internal calculation process cannot be considered by the agent, only the final values. Moreover, the integration with the other elements of the agent remains unclear because motivational attitudes are assumed or mix with the calculus. However, their computational aspects are usually quite well defined.

⁷ For instance, to calculate the trust of agents as sellers, the ontology can define that this is evaluated through the price in an 80% and through the delivery time in a 20%.

In Table 2 we show the summary of the reviewed models against this dimension. We marked with ‘✓’ the ones with such property, and ‘–’ the lack of it. We mark the Repage model with ‘~’ because the internal structure is based on predicates that have associated cognitive notions, but it does not have an explicit representation of them. In fact, Repage integrates into first-order like predicates, mixing also epistemic and motivational attitudes. The model presented in this paper, the BDI + Repage model makes explicit these missing cognitive components.

7.3 Procedural dimension

Often, models offer a nice way to represent and deal with trust and reputation, but there is no explanations on how they are archive. This is quite common in cognitive models, which focus on the internal components of trust and reputation, on a descriptive dimension, but not how such components are built. However, some non-cognitive models do not give explicit details on the calculus of their evaluations. We must recall here that we focus on the epistemic decisions, not on the creation and combination of motivational attitudes (goal-based).

The model introduced by Castelfranchi and Falcone [8] regarding social trust does not give details on how the beliefs are created. ForTrust model [15,21] redefines the notion of social trust and introduces cognitive reputation but still epistemic decisions remain unclear. On the contrary, models like AFRAS [3,4] and Regret [38,42] describe until the last detail how evaluations are created and how they are aggregated.

We mark Marsh [29] and Abdul-Rahman et al. [1] models with ‘~’ to indicate that in general they provide all the calculations, but left some initial values. For instance, in the former, the model does not indicate how direct interactions are evaluated. The author indicates that this is left open and dependent of the context (and we totally agree with it). The same happens with the latter model.

7.4 Generality dimension

The last dimension we want to analyze refers to the generality of the model. In this dimension we want to classify the models that have a general purpose ‘✓’ versus the ones that focus on very particular scenarios ‘–’. For instance, the model by Abdul-Rahman et al. [1] is a non-general model that focuses on the trust on the information provided by witness agents. The same happens with the model by Yu and Singh [48], which is designed for agents participating in a very structured peer-to-peer network, where evaluations are only done in terms of quality of services. Obviously, the models that have such specification obtain good results and very acceptable computational complexities.

On the contrary, models built for general purposes can be adapted to multiple scenarios and are perfect then for general agents architectures. Regret [38,41] and BDI + Repage model [33] are good examples of such models. Again, Table 2 summarizes in the last column this property against the surveyed models.

7.5 Brief explanation of the models

7.5.1 A-Rahman and Hailes

This model [1] uses the term *trust*, and its main characteristic relays on that evaluations are represented with a discrete set of four elements. The model is fed by two sources: direct

experiences and third party communications of direct experiences. The representation of the evaluations is done in terms of the discrete set $\{vt$ (*very trustworthy*), t (*trustworthy*), u (*untrustworthy*), vu (*very untrustworthy*) $\}$. Then, for each agent and context the system keeps a tuple with the number of past own experiences or communicated experiences in each category. For instance, agent A may have a tuple of agent B as a seller like $(0, 0, 2, 3)$, meaning that agent A has received or experienced 2 results as untrustworthiness and 3 as very untrustworthiness. Finally the *trust* value is computed taking the maximum of the tuple values. In our example for agent A , agent B as a seller would be very untrustworthy. In case of tie between vt and t and between u and vu the system gives the values U^+ (mostly trustworthy) and U^- (mostly untrustworthy) respectively. In any other tie case the system returns U^0 (neutral).

7.5.2 AFRAS

The model presented by Carbo et al. [4] uses fuzzy sets to represent reputation values. The idea is that the latest interaction that an agent has with a partner, that is also valued as a fuzzy set, updates the old fuzzy set reputation value through a weighted aggregation. To calculate the weights, they introduce the *remembrance for memory*, a factor that allows the agent to give more weight to the latest interaction or to the old reputation value. The novelty of this approach relies on the reliability of the reputation value, since it is intrinsically represented in the fuzzy set. So, a wide fuzzy set for a reputation value indicates a high level of uncertainty, meanwhile narrow ones, implies a more reliability.

The model also deals with the recommendations sent by other members of the society. The recommendations are aggregated together with the direct interactions. The level of reliability of this witness information will depend on the good or bad reputation of the senders. In this case then, recommendations from a very well reputed sender could have the same weight than direct interactions.

7.5.3 Castelfranchi and Falcone

Castelfranchi and Falcone in [8] define trust as a mental state composed of a set of goals and beliefs and strongly related to the notion of delegation. In a more formal way, let i, j be two agents, the cognitive components that make i trusts j regarding the goal g are the following [8]⁸:

- *Goal Seeking*: i has the goal g .
- *Competence Belief*: i believes that j is capable of obtaining g from a set of actions (summarized in the action α)
- *Disposition Belief*: i believes that j will actually perform α to obtain g . This belief makes agents predictable.
- *Dependence Belief*: i believes that she needs/depends on j to perform the task.

Competence and disposition beliefs, together with the goal are the *core trust*. They model the ability and willingness of the agent j to achieve g . They are evaluative beliefs and are constituents of the image and reputation of j in the sense described at the beginning in the previous chapter. This property was already mention in [30], where the authors exemplify

⁸ The authors describe other beliefs and goals that are part of the trust mental state, like fulfillment belief or wishes. However, for the sake of clarity we obviate them because they are a direct cause of the beliefs shown in the list.

which kinds of beliefs compose evaluations, and the capabilities that cognitive agents must achieve in order to be evaluators.

7.5.4 ForTrust

The ForTrust model presented in [21] refine the notion of social trust by differentiating *occurrent* from *dispositional* trust. The former is understood as the trust on other agents to act here and now, and coincide with the core trust definition given by Castelfranchi and Falcone. In contrast, dispositional trust denotes the disposition of the trustee to perform an action in order to obtain a potential goal when some conditions hold [21].

From a more technical perspective the authors define *occurrent* trust with the predicate $OccTrust(i, j, \alpha, \varphi)$, indicating that i trusts j here and now to perform action α to obtain goal φ . As in the definition of core trust from Castelfranchi and Falcone [8], the components embrace an *occurrent* goal, an *occurrent* capability belief, an *occurrent* power belief and an *occurrent* intention belief. More formally:

$$\begin{aligned}
 OccTrust(i, j, \alpha, \varphi) =_{def} & OccGoal_i(\varphi) \wedge \\
 & Belief_i(OccCap(j, \alpha)) \wedge \\
 & Belief_i(OccPower(j, \alpha, \varphi)) \wedge \\
 & Belief_i(OccIntends(j, \alpha))
 \end{aligned}$$

The beliefs on the *occurrent* capability and *occurrent* power correspond to the competence beliefs, while *occurrent* intention to the disposition belief. Regarding *dispositional* trust, the background components are the same but we move from *occurrent* goals to *potential* goals, and from *occurrent* beliefs to *potential* beliefs. Following [21], *dispositional* trust is defined as follows:

$$\begin{aligned}
 DispTrust(i, j, \alpha, \varphi) =_{def} & PotGoal_i(\varphi) \wedge \\
 & Belief_i(CondCap(j, \alpha)) \wedge \\
 & Belief_i(CondPower(j, \alpha, \varphi)) \wedge \\
 & Belief_i(CondIntends(j, \alpha))
 \end{aligned}$$

7.5.5 ReGreT

The ReGreT system presented by Sabater [41] is maybe one of the most complete reputation and trust models, since it takes into account several advantages of all the models presented so far.

ReGreT uses direct experiences, third party information and social structures to calculate trust, reputation and levels of credibility. In this model, trust is a function of direct trust, only calculated through direct experiences, and reputation. The incorporated reputation model uses transmitted information, social networks analysis, system reputation and prejudices (to infer reputation values of unknown agents from their belonging group). It also incorporates a credibility module to evaluate the truthfulness of witness information, that of course, takes into account the reputation and trust of the information provider. It provides reliability measures for trust, reputation and credibility values.

Finally, an important aspect of this model is the consideration for an ontological dimension. They defined the trust of agent a on b towards certain context φ as $T_{a \rightarrow b}\varphi$. The situation φ is totally contextualized, and may depend on other elements. To describe the relationships of contextualized environment, it is assumed an ontology that describes this knowledge, that could be seen as the current *preferred desires* or *goals* of the agent.

7.5.6 FIRE

The FIRE model introduced by Huynh et al. [22] incorporates similar elements than Regret. It computes as well a *trust* value for each agent and a reliability measure. It uses direct trust computed though direct experiences (extracted from Regret as the same authors claim), witness information (similar to Regret) and certified reputation. The last one is a completely new component. Certified reputations *are ratings presented by the rated agent about itself which have been obtained from its partners in past interactions* [22]. The authors argue that this could be seen as the recommendation letters or references when applying for a job position.

The model uses role-based trust to determine the elements that contribute to the calculation of trust. This component is similar to the ontology dimension of Regret. Therefore, they can be seen as the desires (or goals) of the agent.

7.5.7 Marsh

This model [29], one of the first that appeared in literature, talks explicitly about trust, and only takes into account direct experiences. It defines three kinds of trust.

- *Basic Trust*: T_x^t represents the trust disposition of agent x at time t .
- *General Trust*: $T_x(y)^t$ represents the general trust that agent x has on y at time t without specifying any situation.
- *Situational Trust*: $T_x(y, \alpha)^t$ represents the trust of agent x on the target agent y in the situation α . Marsh defines a basic formula to calculate it:

$$T_x(y, \alpha)^t = U_x(\alpha)^t \cdot I_x(\alpha)^t \cdot \overline{T_x(y)^t} \quad (1)$$

where $U_x(\alpha)^t$ is the utility that agent x gains from situation α , $I_x(\alpha)^t$ is the importance for agent x in the situation α , and $\overline{T_x(y)^t}$ is the estimation of general trust after taking into account all information related to $T_x(y)^t$. The author proposes three ways to calculate this estimation: the mean, the maximum and the minimum of all past experiences.

7.5.8 Yu and Singh

In this model [47], the result of direct interactions is stored as what the authors call quality of service (-QoS-). Agents only keep the most recent interactions, and each agent defines a threshold for each partner over which she is classified as a trustworthy agent.

Also, the model incorporates for each agent a *TrustNet* structure, in a similar way as Schillo et al. [45] and Histos [49]. The difference is that agents being queried can refer to other agents. The initial agent will take into account the information only if the refereed agents are not too far in the social tree. The model uses Dempster Shafer evidence theory to aggregate the information from different source agents.

7.5.9 LIAR

The LIAR model presented by Muller and Vercouter [31] focuses on the detection of fraud and reputation management in the communications. The authors use a normative language to formalize prohibited situations in terms of the information sent by the agents and the commitments that they set. Through this, the model defines a procedure capable to detect lies.

The model mainly uses two different *kinds* of reputation: Direct Experience-Based Reputation and Observation-Based Reputation. With this information agents can decide whether to *trust* or *distrust* the information sent by a given source agent. The authors detail the decision making process for the trust decision, and thus, from our perspective, it becomes a trust model. The model is framed in peer-to-peer networks.

7.5.10 Schillo et al.

The model presented by Schillo et al. [45] was designed for societies or environments where the evaluation of interactions between agents has a boolean nature, for instance, *good* or *bad*. For this reason it works perfectly in scenarios like the prisoners dilemma. The idea is that the result of an interaction computes the honesty of the partner by checking what she claimed and what she finally did. Taking into account all the results in the interactions, the model calculates the probability on the honesty in the next interaction, by simply dividing the number of interactions where the agent was honest by the total number of interactions. Then, let A, B be agents, where A has observed B being honest h times on a total of n interactions, the probability for A that B will be honest the next interaction is calculated by $T(A, B) = \frac{h}{n}$.

This naive idea is complemented with a very interesting source of information. They incorporate a social network, a *TrustNet* data structure, for each agent. The idea is that agents can query other agents that have met before. This witness information will be a set of interaction results, not a summary of them, that agents can incorporate to their probability calculus.

7.5.11 Sen and Sajja

In the model presented by Sen and Sajja [46] the authors explicitly talk about reputation. The model considers two kinds of direct experience: direct interaction and direct observation. The idea is that only direct interactions give an exact perception of the performance of the agents. The authors suppose that observations are noisy, and that may differ from reality. Due to this difference, the impact than direct interactions have on the updating rule of reputation values is much higher than direct observations. They represent the reputation values as real numbers in the interval $[0, 1]$ where 0 represents the worst reputation and 1 the best one, in a linear function.

In addition, in their model agents can query other agents about the performance of other partners, being the answer always a boolean, good or bad. From this witness information, agents calculate the number of positive and negative answers received about the same partner.

8 Conclusions and future work

In this paper, we have presented a possible integration of a cognitive reputation model, Repage, in a BDI agent architecture. The agent has been specified using multi-context systems, where each attitude has been integrated in a different a context. We used L_{BC} , a

hierarchical first-order language to represent the beliefs of the agent, allowing probabilistic reasoning. In particular we show how Repage social evaluations, image and reputation, are translated into probabilistic formulas written in L_{BC} , and under which conditions image and reputation information influence each other. The complete BDI reasoning process is done by also allowing graded desires and graded intentions, and stating appropriate bridge rules to relate them.

From the example it should be clear that on one hand epistemic decisions play a crucial role in the pragmatic–strategic decisions of the agent, and that a formal model for its integration improves the conceptualization of the reasoning process. On the other hand, the consequences of pragmatic–strategic decisions may effect the epistemic decisions. Even when in this work we have used Repage model, we took it as a paradigmatic example of a cognitive reputation model. Other models with similar (or simplified) notions could be also used in the BDI framework.

The classification dimensions that we provide in the relate work section enhance the contribution of this paper. Nevertheless, we want to point out several considerations:

1. Even when we are placing the BDI + Repage model as a trust model, we want to clarify that the architecture is more general. We do not explicitly define trust, but it emerges from a set of beliefs, desires and intentions when a decision is made and such decision involves an action to interact with another agent.
2. Also, when trust emerges from the reasoning, it can be completely defined in terms of a mental state composed of beliefs, desires and intentions. Hence, we classify it as a cognitive model. When a decision is made and such decision involves an interaction with another agent, the mental state can be seen as trust, in the sense described by Castelfranchi and Falcone [8] and Herzig et al. [21].
3. Moreover, bootstrapping is possible, becoming, as far as we know, the only trust model that has a cognitive representation and at the same time, an analytical formulation to update and calculate the cognitive components of trust.
4. Finally, the model has a general purpose. It is not attached to any underlying network typology nor ontology, and thus, it could and should be adapted to the peculiarities of the environments, although we believe that this knowledge could be codified as beliefs.

In the future we are interested in studying the resolution of cognitive dissonances, situations in which the agent cannot decide which action to perform due to contradictory information. This research direction is somehow related to argumentation issues. Parsons et al. [32] use a multi-context BDI agent to build an argumentation framework that we could adapt in our model. Also, work related to coherence analysis can be used for such purpose. For instance, in the work [23], a deductive coherence framework is formalized in the basis of a BDI multi-context agent, and it could be used to solve these problems.

Another important part of this research line involves the empirical study of certain properties regarding image and reputation through simulations, and therefore implementation of the model using a logic-based multiagent platform, like JASON [2]. One point that we are specially interested is in the study on how graded trust conditions affect the overall performance of societies, and therefore, how the relation between image and reputation is relevant in determining the dynamics of the society. Regarding this issue, we are implementing a simple scenario using JASON where a simplified version of our BDI model is introduced. We plan to exhaustively study the performance of such simulation platform.

Acknowledgments This work was supported by the EC with the project LiquidPub (STREP FP7-213360), by the Spanish Education and Science Ministry with the projects AEI (TIN2006-15662-C02-01), AT (CONSOLIDER CSD2007-0022, INGENIO 2010), ARINF (TIN2009-14704-C03-03) and RepBDI (Intramural

200850I136), by the ESF Eurocores-LogICCC/MICINN with the project FFI2008-03126- E/FILO, and by the Generalitat de Catalunya under the grants 2009-SGR-1433 and 2009-SGR-1434. We also would like to thank the anonymous referees for their valuable comments and suggestions.

Appendix A: Internals of Repage

In the Repage architecture we find three main elements, a *memory*, a set of *detectors* and the *analyzer* (see Fig. 5). In the memory, predicates are conceptually organized in levels of abstraction and inter-connected. Each predicate that belongs to one of the main types (image, reputation, shared voice, shared evaluation, valued communication and outcome) contains an evaluation that refers to a certain agent in a specific role. We maintain the value associated to a predicate as a tuple of n positive values (summing to one), that we call *weights*, plus a strength value: $\{w_1, \dots, w_n, s\}$. Originally, only five values were considered, each one associated with a linguistic label: *Very Bad (VB)*, *Bad (B)*, *Neutral (N)*, *Good (G)* and *Very Good (VG)*.

The network of dependences specifies which predicates contribute to the values of others. Each predicate (except those at the bottom level) has a set of antecedents and at the same time contributes to the calculation of other predicates. The *detectors*, inference units specialized in each particular kind of predicate, receive notifications from predicates that have changed or that appear in the system, like new communications or new fulfillments, and use the dependences to recalculate the new values and to populate the memory with new predicates. The aggregation of evaluations is done with a weighted product (see [40] for a short discussion and [18,39] for details about the aggregation operation). Let m be the number of evaluations

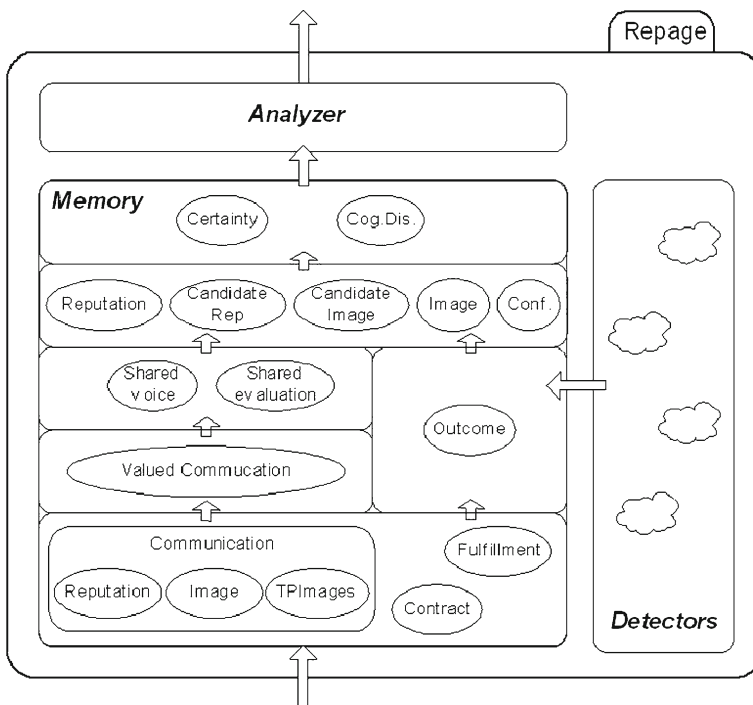


Fig. 5 The Repage architecture

w^j to aggregate, indicating their weight by w_i^j . The aggregated evaluation w is calculated as follows:

$$\forall i : 0 \leq i \leq n - 1 : w_i = \frac{\prod_{j=1}^m w_i^j}{\sum_{i=0}^{n-1} \prod_{j=1}^m w_i^j} \quad (2)$$

Furthermore, each predicate has associated a strength that is function of its antecedents and of the intrinsic properties of each kind of predicate. As a general rule, predicates that resume or aggregate a bigger number of predicates will hold a higher strength. However, strength is closely related to bias factors, rules that for instance, give more importance to direct experiences that indirect experiences, and that may come from sociology or psychology theories, or from simple common sense.

At the first level of the Repage memory we find a set of predicates not evaluated yet by the system.

- *Contracts*: agreements of the future interaction between two agents. For instance, in an e-Commerce environment, an agent may expect the maximum quality of a product that for sure the seller is saying will offer.
- *Fulfillment*: the result of the interaction. In the same e-Commerce example, the fulfillment would be the real quality of the product the agent got.
- *Communications*: Information that other agents may communicate about others' evaluations. These communications may be related to three different aspects: the image that the informer has about a target, the image that according to the informer a third party agent has, and the reputation that the informer has about the target.

In level two we have two kind of predicates:

- *Valued communication*: The subjective evaluation of the communication received that takes into account, for instance the image the agent may have of the informer as informant. Communications from agents whose credibility in terms of image or may be reputation are low, will not be considered as strongly as the ones coming from well reputed informers.
- *Outcome*: The agent's subjective evaluation of the direct interaction. From a fulfillment and a contract a *detector* builds up an outcome predicate that evaluates the particular transaction.

In the third level we find two predicates that are only fed by valued communications. On one hand, a shared voice will hold the information received about the same target and same role coming from communicated reputations. On the other hand, shared evaluation is the equivalent for communicated images and third party images.

Shared voice predicates will generate candidate reputation, and share evaluations together with outcomes, candidate image. In this fourth level candidate reputation and candidate images aren't strong enough to become a full reputation and image respectively. New communications and new direct interactions will contribute at this level to enrich these predicates and therefore "jump" to images and reputations. For details on the remaining elements we refer to [40].

References

1. Abdul-Rahman, A., & Hailes, S. (2000). Supporting trust in virtual communities. In *Proceedings of the Hawaii's international conference on systems sciences, Maui, Hawaii*.

2. Bordini, R. H., Hübner, J. F., & Wooldridge, M. (2007). *Programming multi-agent systems in AgentSpeak using Jason*. Chichester: Wiley.
3. Carbo, J., Molina, J., & Davila, J. (2002). Comparing predictions of sporas vs. a fuzzy reputation agent system. In *3rd International conference on fuzzy sets and fuzzy systems, Interlaken* (pp. 147–153).
4. Carbo, J., Molina, J., & Davila, J. (2002). Trust management through fuzzy reputation. *International Journal in Cooperative Information Systems*, 12(1), 135–155.
5. Casali, A. (2008). *On intentional and social agents with graded attitudes*. PhD thesis, Universitat de Girona.
6. Casali, A., Godo, L., & Sierra, C. (2004). Graded models for BDI agents. In J. Leite & P. Torroni (Eds.), *CLIMA V, Lisboa, Portugal* (pp. 18–33).
7. Casali, A., Godo, L., & Sierra, C. (2008). A logical framework to represent and reason about graded preferences and intentions. In *Proceedings of KR'08, Sydney, Australia*.
8. Castelfranchi, C., & Falcone, R. (1998). Social trust. In *Proceedings of the first workshop on deception, fraud and trust in agent societies, Minneapolis, USA* (pp. 35–49).
9. Conte, R., & Paolucci, M. (2002). *Reputation in artificial societies: Social beliefs for social order*. Dordrecht: Kluwer.
10. Demolombe, R., & Liau, C. (2001). A logic of graded trust and belief fusion. In *Proceedings of the 4th workshop on deception, fraud and trust in agent societies* (pp. 13–25).
11. eBay. (2002). *eBay*. <http://www.eBay.com>.
12. Enderton, H. B. (1972). *A mathematical introduction to logic*. New York: Academic Press.
13. Fagin, R., & Halpern, J. (1994). Reasoning about knowledge and probability. *Journal of the ACM*, 41(2), 340–367.
14. Flamínio, T., Pinna, G. M., & Tiezzi, E. B. P. (2008). A complete fuzzy logical system to deal with trust management systems. *Fuzzy Sets and Systems*, 159(10), 1191–1207.
15. ForTrust. (2009). *ForTrust: Social trust analysis and formalization*. <http://www.irit.fr/ForTrust/>.
16. Gaertner, D., Noriega, P., & Sierra, C. (2006). Extending the BDI architecture with commitments. In *Proceedings of the CCIA'06* (pp. 247–257).
17. Giunchiglia, F., & Serafini, L. (1994). Multilanguage hierarchical logic (or: How we can do without modal logics). *Journal of AI*, 65, 29–70.
18. Grabisch, M., Orlovski, S. A., & Yager, R. R. (1998). Fuzzy aggregation of numerical preferences. In *Fuzzy sets in decision analysis, operations research and statistics* (pp. 31–68).
19. Ganz, J., Kraus, S., & Perlis, D. (2000). A logic for characterizing multiple bounded agents. *Autonomous Agents and Multi-Agent Systems*, 3(4), 351–387.
20. Hájek, P., Godo, L., & Esteva, F. (1995). Fuzzy logic and probability. In *Uncertainty in artificial intelligence conference* (pp. 237–244).
21. Herzig, A., Lorini, E., Hubner, J. F., & Vercouter, L. (2010). A logic of trust and reputation. *Logic Journal of the IGPL*, 18(1), 214–244.
22. Huynh, T., Jennings, N., & Shadbolt, N. (2006). An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2), 119–154.
23. Joseph, S., Dellunde, P., Schorlemmer, M., & Sierra, C. (2008). Formalizing deductive coherence: An application to norm evaluation. In *NORMAS* (pp. 158–172).
24. Kooi, B. P. (2003). Probabilistic dynamic epistemic logic. *Journal of Logic, Language and Information*, 12(4), 381–408.
25. Kozen, D. (1983). A probabilistic PDL. In *STOC '83: Proceedings of the fifteenth annual ACM symposium on theory of computing, New York, NY, USA* (pp. 291–297). ACM.
26. Liau, C. J. (2003). Belief, information acquisition, and trust in multi-agent systems: A modal logic formulation. *Artificial Intelligence*, 149(1), 31–60.
27. Lorini, E., & Demolombe, R. (2008). From binary trust to graded trust in information sources: A logical perspective. In S. Barber, R. Falcone, J. Sabater-Mir, & M. Singh (Eds.), *Trust in agent societies*, volume 5396 of *LNAI* (pp. 205–225). New York: Springer.
28. Luck, M., McBurney, P., Shehory, O., & Willmott, S. (2005). *Agent technology: Computing as interaction (A roadmap for agent based computing)*. AgentLink.
29. Marsh, S. (1994). *Formalising trust as a computational concept*. PhD thesis, Department of Mathematics and Computer Science, University of Stirling.
30. Miceli, M., & Castelfranchi, C. (2000). The role of evaluation in cognition and social interaction. In *Human cognition and social agent technology* (pp. 225–259). Amsterdam: John Benjamins.
31. Muller, G., & Vercouter, L. (2005). Decentralized monitoring of agent communications with a reputation model. In R. Falcone, K. S. Barber, J. Sabater-Mir, & M. P. Singh (Eds.), *Trusting agents for trusting electronic societies, theory and applications in HCI and E-commerce*, volume 3577 of *Lecture notes in computer science*. New York: Springer.

32. Parsons, S., Sierra, C., & Jennings, N. (1998). Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3), 261–292.
33. Pinyol, I., and Sabater-Mir, J. (2009). Pragmatic–strategic reputation-based decisions in BDI agents. In *Proceedings of the AAMAS'09, Budapest, Hungary* (pp. 1001–1008).
34. Pinyol, I., & Sabater-Mir, J. (2010). Metareasoning and social evaluations in cognitive agents. In *Autonomic computing and communications systems*, volume 23 of *LNISCT* (pp. 220–235). Berlin: Springer.
35. Pinyol, I., Sabater-Mir, J., & Cuni, G. (2007). How to talk about reputation using a common ontology: From definition to implementation. In *Proceedings of the ninth workshop on trust in agent societies, Hawaii, USA* (pp. 90–101).
36. Pinyol, I., Sabater-Mir, J., & Dellunde, P. (2008). Probabilistic dynamic belief logic for image and reputation. In *Proceedings of the CCIA'08, Empuries, Spain*.
37. Rao, A. S., & Georgeff, M. P. (1991). Modeling rational agents within a BDI-architecture. In J. Allen, R. Fikes, & E. Sandewall (Eds.), *Proceedings of KR'91* (pp. 473–484). San Mateo, CA: Morgan Kaufmann Publishers.
38. Sabater-Mir, J. (2003). *Trust and reputation for agent societies*. PhD thesis, IIIA-CSIC, Barcelona, Spain.
39. Sabater-Mir, J., & Paolucci, M. (2007). On representation and aggregation of social evaluations in computational trust and reputation models. *International Journal of Approximate Reasoning*, 46(3), 458–483.
40. Sabater-Mir, J., Paolucci, M., & Conte, R. (2006). Repage: Reputation and image among limited autonomous partners. *JASSS*, 9(2).
41. Sabater, J., & Sierra, C. (2001). Regret: A reputation model for gregarious societies. In *Proceedings of the fourth workshop on deception, fraud and trust in agent societies, Montreal, Canada* (pp. 61–69).
42. Sabater, J., & Sierra, C. (2002). Reputation and social network analysis in multi-agent systems. In *Proceedings of AAMAS-02, Bologna, Italy* (pp. 475–482).
43. Sabater, J., & Sierra, C. (2005). Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1), 33–60.
44. Sabater-Mir, J., Sierra, C., Parsons, S., & Jennings, N. R. (2002). Engineering executable agents using multi-context systems. *Journal of Logic and Computation*, 12(3), 413–442.
45. Schillo, M., Funk, P., & Rovatsos, M. (2000). Using trust for detecting deceitful agents in artificial societies. *Applied Artificial Intelligence* (Special issue on trust, deception and fraud in agent societies).
46. Sen, S., & Sajja, N. (2002). Robustness of reputation-based trust: Boolean case. In *Proceedings of the first international joint conference on autonomous agents and multiagent systems (AAMAS-02), Bologna, Italy* (pp. 288–293).
47. Yu, B., & Singh, M. P. (2001). Towards a probabilistic model of distributed reputation management. In *Proceedings of the fourth workshop on deception, fraud and trust in agent societies, Montreal, Canada* (pp. 125–137).
48. Yu, B., & Singh, M. P. (2003). Detecting deception in reputation management. In *AAMAS '03: Proceedings of the second international joint conference on autonomous agents and multiagent systems* (pp. 73–80). New York: ACM Press.
49. Zacharia, G. (1999, September). *Collaborative reputation mechanisms for online communities*. Master's thesis, Massachusetts Institute of Technology.