Scalable Computing: Practice and Experience
Volume 12, Number 1, pp. 105–120. http://www.scpe.org
ISSN 1895-1767

TRUSTED BELIEFS FOR HELPFUL BEHAVIOR WHEN BUILDING WEB SERVICES *

IOAN ALFRED LEŢIA, RADU RĂZVAN SLĂVESCU †

Abstract. Composite software services often present uncertainty over their non-functional properties. To tackle this, one could model them as shared goals of an agent team which aims at maximizing the likelihood of success of the joint task. An architect is in charge with picking services and providers, while a consultant helps him, when possible, by suggesting alternative approaches. The multinomial version of the "Belief Recipe Tree" structure relies on beliefs built based upon prior mutual experiences of the consultant with various providers and/or abstract plans and revised after each interaction, exhibiting a higher flexibility in several web service building scenarios.

Key words: collaborative behavior, multi-agent systems, web services, trust, beliefs, subjective probability, agent teams

AMS subject classifications. 68T37, 68T42

1. Introduction. In the last decade, the development of web applications which employ e-services as basic blocks for deploying a complex functionality gained much momentum. Within this context, the paradigm of web services (WS) makes generous promises regarding the development of flexible and highly customized applications. Some particular tasks do find their solution in an existing web service, but this is the exception rather than the rule, as it is highly unlikely that one can find an already built WS able to solve her specific problem in each and every detail. In the vast majority of cases, matching the user's specific requirements demands designing a composite web service. Most of the time, this is made of simpler web services offered by a bunch of different providers. Building such a WS (i.e. a composite web service), demands solving at least two issues. On one hand, the developer must decompose the complex goal task into simpler ones, up to the basic level, where each of them can be solved by an existing WS. On the other hand, every basic task must be assigned to a concrete WS provider which must be selected among more providers of this type. Both issues requires considering functional as well as non-functional criteria. Hence, the system requires producing the intended output, but also guaranteeing some quality parameters like response time, reliability or cost.

A possible approach for this problem has been proposed, for example, in [3]. The cited paper describes a workflow based solution which takes into account the Quality of Service (QoS) of each component. A composite web service is shown to be reducible to a sequential flow whose QoS parameters are easy to analyze and predict. To achieve this, the system designer will need accurate data about the non-functional aspects of web services we mentioned above. The intrinsic volatility of such information makes it potentially unreliable though, as the values might change rapidly and alter the subsequent decisions. In the same time, the available information is inherently incomplete. This, combined with the possibility of having inconsistencies in the knowledge about the decomposition recipes lead to the conclusion that uncertainty must be accommodated when assembling web services.

Withing this context, developing a composite WS with appropriate reliability becomes a challenging task. To deal with it, one might regard this problem as the process of a joint goal being solved by a set of cooperative agents, and among them, WS architects and consultants. Typically, a consultant is employed by a WS architect in order to provide him/her with guidelines about designing a composite WS. This involves delivering both an abstract plan and a concrete list of WS providers for each plan step in order to maximize the likelihood of the whole WS to become up and running. Both agents above are interested in having this goal achieved, but the type of decisions they can make are quite different. The architect will be focused on choosing each service and its provider. The consultant will have to decide whether to adopt a helpful behavior towards her peer or to refrain from it. This behavior consists of suggesting his partner an alternative. The consultant must do this such that the global likelihood of success is maximized, whilst not providing its partner with more information than he/she paid for.

We argue the aforementioned issues can be solved by taking trust into consideration when an agent starts building composite web services. In order to prove this, we presented in [15] a possible extension of the

*Part of this work was supported by CNCSIS-UEFISCSU, National Research Council of the Romanian Ministry for Education, Research, Youth and Sport, under project ID_170/2009.

†Department of Computer Science, Technical University of Cluj-Napoca, Bariţiu 28, RO-400027 Cluj-Napoca, Romania ({letia,srazvan}@cs-gw.utcluj.ro)

Probabilistic Recipe Tree (PRT) to a structure we called Belief Recipe Tree (BRT) able to accommodate trust concerning web services and to allow making decisions based on this. Trust integrates a historical sequence of experiences into a single value and measures the confidence one should have in a service to meet its advertised non-functional parameters (e.g. the likelihood of success). We show how to develop and employ the BRT structure, which is actually a belief-based extension of the "Probabilistic Recipe Tree" in [14], for making the above type of decisions. Then, we suggest a way for building the beliefs an agent needs for cooperation over the WS design, based on the set of experiences the consultant has had with different providers and/or design plans. Since the beliefs are revised after each experience, it follows that the flexibility offered is higher due to the fact the providers which are not always truthful can be punished and the system can adjust its beliefs, as well as its subsequent decisions, accordingly.

The present paper elaborates on the idea by extending the BRT to multinomial opinions. This aims allowing agent to deal with a finer grained set of mutually exclusive sentences describing an agent's performance, like "The level of success is *poor/average/good* as opposite to a mere $failure/success$, in order to allow a more realistic model of the domain. The main contribution made is to extend the operations on BRT to the multinomial case. First, we revisit the structure of PRT and BRT and present some examples in more details. Then, we introduce the multinomial versions of the BRT operations. Finally, we present an example to illustrate the type of decisions which are allowed by the enriched BRT.

The rest of this paper is organized in the following way. Section 2 introduces the basic blocks of the Subjective Logic which are further needed for building and manipulating the BRT. Section 3 details the structure, components and operations on BRT, starting from a brief presentation of its purely probabilistic counterpart PRT. Section 4 illustrates in detail the ideas in a scenario of building a composite web service which generates car routes. The resulting web service ought to take into consideration both map details as well as traffic information and weather forcast when generating an autoroute. The ideas are then extended to multinomial opinions. Section 5 positions the approach among some other works in the field, while Section 6 concludes and sketches future lines of investigation.

2. Subjective Logic based Trust. Since uncertainty of sentences describing the world is inherent, several formalisms were developed in order to deal wit it. One such formalism is the Subjective Logic (SL) theory, introduced in [9] and extensively updated in [10]. This logic could be seen as a superset of the classical probability theory because, as its parameters approach some limit values, the theory is reduced to the classical one. SL deals with beliefs and includes operations on sentences similar to those in the Dempster-Shafer theory of beliefs [5, 22]. In the same time, it is compatible with Nilsson's probabilistic logic [18]. The present section briefly presents the ideas of SL we used for trust estimation and web service selection. For further details on the topic of SL, we refer to [10].

2.1. Subjective Logic Opinions. SL argues that a perfectly accurate assessment of probability is beyond the scope of human nature. Therefore, uncertainty involved in estimating the values of probabilities must be considered and assessed. The theory's basic block is the concept of *opinion* over a sentence. Given a proposition x, the opinion ω_x regarding the truth value of x is defined as a quadruple $\omega_x = (b_x, d_x, u_x, a_x)$. Its components represent the degree of belief (evidence supporting x), disbelief (evidence supporting $\neg x$) and uncertainty about the truth of x. By definition, they must sum up to 1. The atomicity a_x is a measure of the *prior* probability of the truth value of x. Within this subsection, we will focus on a universe of discourse comprising only binomial, mutually exclusive sentences (x and $\neg x$ respectively), thus a_x defaults to 0.5. In the next subsection, we will briefly describe the multinomial case, where there are more than 2 possible, mutually exclusive values for a sentence.

For example, one can assign a value of 0.7 to the belief corresponding to the sentence "WS will succeed", 0.2 to the disbelief corresponding to the sentence "WS will succeed" (i.e. the belief corresponding to the sentence "WS will fail") and the remaining 0.1 to the uncertainty about the behavior of the WS. This one is due to the lack of perfect knowledge over the behavior of the given WS and can be seen as second order probability (modeling uncertainty over a first-order probabilities).

Given $\omega_x = (b_x, d_x, u_x, a_x)$, the corresponding probability expectation value (a generalization of a classical probability expectation), is defined by the formula below:

$$E(\omega_x) = b_x + a_x u_x \tag{2.1}$$

This definition corresponds to that of pignistic probability in [23] and is consistent with the principle

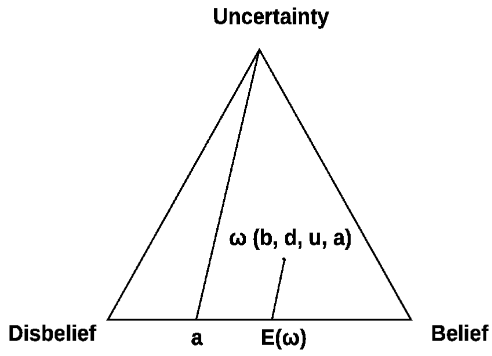

FIG. 2.1. *A Subjective Logic opinion inside the opinion triangle*

of equally dividing frame belief among its singletons. The interpretation of a_x is the relative proportion of singletons of x.

As shown in [10], an opinion coresponds to a point inside a triangle. Figure 2.1 illustrates this, displaying an opinion $\omega(b, d, u, a)$, the base rate a, the vertexes of maximum belief, disbelief and uncertainty and the probability expectation value $E(\omega)$.

We regard such an opinion over a sentence as the subjective trust hold by the issuing agent towards the object of that sentence. As an example, we may consider x to be the sentence above, assumed to be believed by the architect agent: "Web service WS_1 offered by provider P_1 will perform according to its parameters". In this case, $\omega_x = (0.7, 0.2, 0.1, 0.5)$ means the agent issuing sentence x (i.e. the system architect) beliefs the sentence to a degree of 0.7; its negation $\neg x$ to a degree of 0.2 and has an uncertainty degree of 0.1 about it, possibly because of the lack of complete evidence. The value of a is 0.5 as, so far, we have considered only a binary universe of discourse for each sentence. In this example, ω_x models the trust of the entity issuing sentence x towards WS_1. Trust could also be concerned with sentences describing generic solutions like "In order to obtain a WS of type T, one should combine a WS of type T_a and one of type T_b", or "Solution S_1 for the problem P is preferred by 55% of the developers".

2.2. Mapping Experiences into Opinions. Let us suppose one has already had a number of experiences with a binary event, e.g. a WS behaving with the advertised latency or not. Let us assume there were r positive (i.e. expected behavior from the WS side) and s negative (i.e. poor) experiences. Posterior probabilities of binary events can be represented by a family of probability density functions, namely the Beta distribution:

$$
\begin{aligned}
Beta(p|\alpha, \beta) &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \\
\alpha &= r + 2a \\
\beta &= s + 2(1-a) \\
\Gamma(z) &= \int_0^\infty t^{z-1} e^{-t} dt
\end{aligned}
\tag{2.2}
$$

with the restrictions $0 < a < 1$, $p \neq 0$ if $\alpha < 1$ and $p \neq 0$ if $\beta < 1$.

One possible interpretation for this is how likely the outcome of a future experience will have a given value (e.g. positive) and how much uncertainty exists within this prediction, given that r positive and s negative experiences have been recorded so far.

As showed in [10], there exists a function mapping the evidence space for a sentence (i.e. prior observations over its truth) into opinion space. Based on this correspondence, sentence opinions can be synthesized out of sequences of observations over those sentences. Opinions are further combined according to specific operators to be defined later. The resulting algebra of opinions is equivalent, both from the point of view of semantics and expressiveness, with the distributions, but it has the advantage of being much more efficient from the computational perspective.

If for the sentence x, we have r experiences supporting x and s supporting $\neg x$, then the opinion's compents b, d and u are computed in the following way:

$$b = \frac{r}{r+s+2} \tag{2.3}$$

$$d = \frac{s}{r+s+2} \tag{2.4}$$

$$u = \frac{2}{r+s+2} \tag{2.5}$$

Some limit cases can be considered: $b = 1$ means logical $TRUE$ (probability 1), $d = 1$ means logical $FALSE$ (probability 0), $u = 1$ means vacuous opinion (total uncertainty, absolute lack of experiences) and $b + d = 1$ gives the classical probability (no uncertainty). One should notice the latter case happens if the number of experiences is infinite. This property will be uses further in this paper, when our extended formalism reduces to the original one when the number of experiences approaches ∞.

One should notice that, from this perspective, a set of 3 positive and 1 negative outcomes for an experience is different from a set of 30 positive and 10 negative experiences. Though in both cases the probability of a positive experience is 75%, the opinions are $(0.500, 0.166, 0.334, 0.5)$ and $(0.714, 0.238, 0.05, 0.5)$, so the uncertainty over the expected experiment value decreases.

2.3. Subjective Logic Operators. In order to combine opinions on propositions, SL introduces a set of operators over opinions, whose semantics relies on probability distributions as well. The following two SL operators are used in this work:

1. conjunction $\wedge$: given $\omega_x = (b_x, d_x, u_x, a_x)$ and $\omega_y = (b_y, d_y, u_y, a_y)$, their conjunction $\omega_{x\wedge y}$, written $\omega_x \sqcap \omega_y$, has the following components:

$$b_{x\wedge y} = b_x b_y + \frac{(1-a_x)a_y b_x u_y + a_x(1-a_y)u_x b_y}{1-a_x a_y} \tag{2.6}$$

$$d_{x\wedge y} = d_x + d_y - d_x d_y \tag{2.7}$$

$$u_{x\wedge y} = u_x u_y + \frac{(1-a_y)b_x u_y + (1-a_x)u_x b_y}{1-a_x a_y} \tag{2.8}$$

$$a_{x\wedge y} = a_x a_y \tag{2.9}$$

E.g. if the solution for designing a composite WS is expressed in BPEL4WS as $sequence(RG, WF)$, and we have opinions concerning reliability $\omega_{RG} = (0.7, 0.2, 0.1, 0.5)$ and $\omega_{WF} = (0.6, 0.3, 0.1, 0.5)$ respectively, then the sentence describing the success likelihood of the composite WS would be $\omega_{RGWF} = (0.4, 0.51, 0.09, 0.25)$.

2. disjunction $\vee$: given $\omega_x = (b_x, d_x, u_x, a_x)$ and $\omega_y = (b_y, d_y, u_y, a_y)$, their disjunction $\omega_x \vee \omega_y$, written $\omega_x \sqcup \omega_y$, has the following components:

$$b_{x\vee y} = b_x + b_y - b_x b_y \tag{2.10}$$

$$d_{x\vee y} = d_x d_y + \frac{a_x(1-a_y)d_x u_y + (1-a_x)a_y u_x d_y}{a_x + a_y - a_x a_y} \tag{2.11}$$

$$u_{x\vee y} = u_x u_y + \frac{a_y d_x u_y + a_x u_x d_y}{a_x + a_y - a_x a_y} \tag{2.12}$$

$$a_{x\vee y} = a_x + a_y - a_x a_y \tag{2.13}$$

E.g. if the solution for designing a WS called RG requires choosing either service RG_1 or service RG_2, with reliability $\omega_{RG_1} = (0.7, 0.2, 0.1, 0.5)$ and $\omega_{RG_2} = (0.8, 0.1, 0.1, 0.5)$ respectively, then the sentence describing the success likelihood of WS would be $\omega_{RG} = (0.92, 0.043, 0.037, 0.75)$.

The definitions of the conjunction and disjunction operators we have just presented were first introduced in the paper [11]. Here, they are called normal binomial multiplication and comultiplication and denoted by the operators $\cdot$ and $\sqcup$ respectively. The two operations are shown to be very good approximations of the analytically correct operators applied to the Beta probability density functions. Whilst the exact operations quickly become unmanageable, their SL counterparts preserve the simplicity thus allowing analyzing complex models [10]. The outcome, albeit approximate, will approach the exact results as the uncertainty decreases.

2.4. Multinomial Opinions. If we consider k possible, mutually exclusive values for a sentence, e.g. the reliability level could be $X = poor, average, good$ ($k = 3$), we need to shift to multinomial opinions. Such an opinion would be $\omega_X = (\vec{b_X}, u_X, \vec{a_X})$, with $u + \sum_{i=1}^{k} \vec{b_{x_i}} = 1$ and $\sum_{i=1}^{k} \vec{a_{x_i}} = 1$. The vector contains k values for the belief levels corresponding to each possible value of the sentence, one value for the level of uncertainty and k values for atomicity. We will write $b(x_i)$ to denote $\vec{b_{x_i}}$.

The components of the corresponding probability expectation vector are defined by:

$$E(x_i) = b(x_i) + a(x_i)u \tag{2.14}$$

To illustrate, let us suppose we have the sentences "Performance of WS_1 is bad", "Performance of WS_1 is average", "Performance of WS_1 is good", the opinion might be, for example, $\omega_x = (0.2, 0.2, 0.5, 0.1, 0.3, 0.4.0.3)$ meaning the the system architect beliefs the performance to be *bad* to a degree of 0.2, *average* to a degree of 0.2, *good* to a degree 0.5 and has an uncertainty degree of 0.1 The probability expectation vector would then be $(0.23, 0.24, 0.53)$.

Posterior probabilities of this type of events can be represented by another family of probability density functions, namely the Dirichlet distribution instead of the Beta distribution, but the approach is similar. For the sentence x, if we have the experience vector $(\vec{r}, \vec{a})$, mapping to opinions can be made by:

$$b(x_i) = \frac{r(x_i)}{2 + \sum_{i=1}^{k} r(x_i)} \tag{2.15}$$

$$u = \frac{2}{2 + \sum_{i=1}^{k} r(x_i)} \tag{2.16}$$

Next, we introduce the definition of multinomial multiplication, which extends the conjunction operation to the multinomial case. This definition is introduced in [12] as the Assumed Uncertainty Mass method. Given two sets of multinomial opinions X and Y, the method for computing their conjunction relies on generating all the elements of the Cartesian product $X \times Y$ and redistributing some of the belief mass on the rows and columns to singleton elements and to the global set $X \times Y$. This is performed such that the expected value of each singleton in the Cartesian product will be equal to the product of the expected values of its components.

Let us assume we have two multinomial opinions $\omega_X(\vec{b_X}, u_X, \vec{a_X})$ having possible values $x_1, \ldots, x_k$ and $\omega_Y = (\vec{b_y}, u, \vec{a_y})$ with possible values $y_1, \ldots, y_l$. Computing the conjunction $\omega_x \sqcap \omega_y$ is based on the formulas below.

First, we compute the expected utilities of the singletons in $X \times Y$:

$$E((x_i, y_j)) = (b(x_i) + a(x_i)u_X)(b(y_j) + a(y_j)u_Y) \tag{2.17}$$

Then we estimate an intermediate uncertainty:

$$u^I_{X\times Y} = u^{Rows}_{X\times Y} + u^{Columns}_{X\times Y} + u^{Frame}_{X\times Y} \tag{2.18}$$

$$u^{Rows}_{X\times Y} = 1 - \sum b^{Rows}_{X\times Y} \tag{2.19}$$

$$b^{Rows}_{X\times Y} = \begin{pmatrix} u_X b(y_1) & \ldots & u_X b(y_l) \end{pmatrix} \tag{2.20}$$

TABLE 2.1
Multinomial opinion multiplication

	belief			atomicity			Expected value		
	poor	*avg*	*good*	*poor*	*avg*	*good*	*poor*	*avg*	*good*
success	0.026	0.038	0.099	0.060	0.150	0.090	0.059	0.119	0.147
failure	0.046	0.058	0.192	0.140	0.350	0.210	0.122	0.248	0.306

$$u_{X\times Y}^{Columns} = 1 - \sum b_{X\times Y}^{Columns} \tag{2.21}$$

$$b_{X\times Y}^{Columns} = \left(\begin{array}{ccc} b(x_1)u_Y & \dots & b(x_k)u_Y \end{array} \right) \tag{2.22}$$

$$u_{X\times Y}^{Frame} = u_X u_Y \tag{2.23}$$

Based on this, we find the product uncertainty:

$$u_{X\times Y} = min \left\{ u_{X\times Y}^{(i,j)}, (x_i, y_j) \in X \times Y \right\} \tag{2.24}$$

$$u_{X\times Y}^{(i,j)} = \frac{u_{X\times Y}^{I} E((x_i, y_j))}{b_{X\times Y}^{I}((x_i, y_j)) + a(x_i)a(y_j)u_{X\times Y}^{I}} \tag{2.25}$$

Beliefs and atomicity levels are then computed:

$$b_{X\times Y}((x_i, y_j)) = E((x_i, y_j)) - a_X(x_i)a_Y(y_j)u_{X\times Y} \tag{2.26}$$

$$a_{X\times Y}((x_i, y_j)) = \frac{E((x_i, y_j)) - b((x_i, y_j))}{u_{X\times Y}((x_i, y_j))} \tag{2.27}$$

Given the multinomial opinion multiplication, its disjunctive counterpart is computed in a similar way, taking into consideration that:

$$E((x_i, y_j)) = E(x_i) + E(y_j) - E(x_i)E(y_j) \tag{2.28}$$

As an example, we might consider the following situation. The consultant considers using a WS of type T for a complex WS. There were 4 prior experiences labeled as *success* and 2 labeled as $failure$ for this design solution. For the moment, the a priori probabilities are assumed known; section 6 discusses how they can be obtained. Thus, for this type of WS, we may have the opinion $\omega_X = (0.250, 0.500, 0.250, 0.300, 0.700)$. On the other hand, we have one WS provider which, in the past, has offered a latency considered *poor* in 5, *average* in 10 and *good* in 13 cases, hence the opinion $\omega_Y = (0.167, 0.333, 0.433, 0.067, 0.200, 0.500, 0.300)$. We would like to know the belief level for the sentence "The design solution based on a WS of type T is *success* and the latency is *good*". Table 2.1 summarizes the belief and atomicity levels for this situations. As we can see, the belief and atomicity for the sentence above are 0.192 and 0.210 respectively, for an uncertainty level of 0.541 and an expected value of 0.306.

3. Belief Recipe Trees. This section presents the Belief Recipe Tree (BRT), an extension of the Probabilistic Recipe Trees (PRT) introduced in [14]. Most of this section, as well as the next one, are borrowed from [15]. Some examples have been added and the multinomial case has been investigated and compared with the previous solution.

The central idea we advocate is that building a complex web service could be regarded as building up a plan, so we may use this structure for web service architecture elaboration. As a general approach, an architect is in charge with picking services and providers, while a consultant helps him, when possible, by suggesting alternative approaches, provided such a decision is rational from the point of view of the total utility.

3.1. Probabilistic Recipe Trees. This subsection briefly describes the PRT for self-containment purposes. For details and formal definitions, we refer to [14].

A PRT for an action α is basically a tree which incorporates a probability distribution over the recipes for accomplishing α. A node in a PRT represents an action together with some associated properties: a leaf node represents a basic (i.e. atomic) action, while an intermediate node represents a complex one. Intermediate nodes are either AND or OR nodes. Each child of an AND node represents an action component of a recipe for completing the AND node action. Each child of an OR node represents a non-deterministic choice of a recipe for competing the OR node action. Each branch which descends from an OR node has associated a probability (which is assumed known a priori) for the corresponding child node to be selected as a recipe for completing the OR node action.

The context of an action, denoted C_α, is the complex of information an agent bases her decision on at a specific moment of time. The predicate $Context(C_\alpha, G_1, \alpha, T)$ is used to express the idea that C_α is the context in which agent G_1 believes, at time T, that action α is done. Function $cba.basic(G_1, \beta, C_\beta)$ returns the probability that agent G_1 can bring about the basic level action β within context C_β. Similarly, $cba.cost(G_1, G_2, \beta, C_\beta)$ returns the cost paid by agent G_1 when the basic level action β is done by agent G_2 in context C_β. Function $V(G_1, \alpha, C_\alpha)$ returns the utility for G_1 if action α is performed in context C_α and includes, for complex actions, both the gain for the action itself and for its sub-actions. All *cba*, *cost* and V are considered intrinsic properties of actions, known by all agents, although obtaining good estimations of them is not trivial for systems comprising more agents and little prior interactions.

Function $p\_CBA(PRT_\alpha, C_\alpha)$ returns the probability of action α to succeed, given context C_α and the tree PRT_α. For leaf nodes, the returned value is *cba.basic*, for AND nodes, a product of children probabilities, while for an OR node, weighted average of children probabilities. Finally, function $Cost(G_i, PRT_\alpha, C_\alpha)$ returns the expected cost to be paid by agent G_i when the group carries out the recipes in PRT_α in context C_α. The returned value are as follows: *cost.basic* in case of leaf nodes, the sum of children costs for AND-type nodes and a weighted average of children costs for OR nodes.

An agent is assumed to be able to perform helpfully, conveying and asking actions. A helpful action requires the agent to be committed (believes PRT_α is the best way and all other agents intend to carry out PRT_α) and to consider that performing γ would increase the group utility. Conveying information action requires the agent to be committed and to believe that the conveyed information would increase the group utility, provided that there is some agent which may perform an action based on the received information. The asking information action can be done if there is another agent committed and he believes that he possesses information which would increase the group utility.

3.2. Opinion based PRT: Belief Recipe Trees. A BRT deals with opinions instead of mere probabilities. The rationale behind extending the PRT structure into BRT is the necessity of endowing the user with a mechanism of generating and adjusting the probability values needed by the tree. In the same time, the BRT "converges" towards the PRT when the number of experiences increases.

The BRT structure basically serves for representing in a compact manner the alternative recipes which might be used for achieving a goal (e.g. for building a composite web service). The size of such a structure is showed to be $\mathcal{O}(nm)^d$, where n is the number of potential recipes for every action, m is the average number of steps in one such action and d is the number of levels of decomposition required to reach the atomic level of WS. This makes it exponentially smaller than the trivial $\mathcal{O}(n^{m^d})$ [14].

A BRT comprises terminal nodes, which model atomic tasks, and non-terminal nodes which represents composed tasks. Each node has attached the following information:

(i) an opinion describing the trust level the evaluator has in the success of the task represented by that node

(ii) a cost, which estimates how much the plan effector must pay in order to have the task corresponding to the node completed

(iii) an income modeling the benefit obtained if the task is accomplished.

The opinions attached to a terminal node estimate the success likelihood of that task, based on its own prior accomplishments; the opinions in the non-terminal nodes are computed based on those of their descendants. The costs for a non-terminal node are computed based on those of descendants and on the likelihood of a branch of being selected, while the income incorporates the benefits for achieving the tasks both in the descendants and in the node itself.

We consider a recipe as a sequence of steps to be performed in order to accomplish a goal. There exists a set of basic tasks which are the atomic constituents of every such recipe. They are described by the properties $cost_N$ and $income_N$, representing the cost and the reward respectively for completing the recipe step at the level of the node N. These are assumed common and known across the community. Unlike the original approach, the property "can bring about" (*cba*, the success likelihood for a specific service) is modeled as an SL *opinion* owned by the evaluator.

The BRT contains the following types of nodes:

1. leaf nodes: they are atomic tasks. For our case, such an task is selecting a specific atomic web service.
2. AND nodes: each child of an AND node must contain a constituent of the join task in the node. It is used to model the situation of a complex recipe consisting of two or more goals which must be achieved to fullfill the whole goal; for our case, it models a composed web service.
3. OR nodes represent possible alternatives for achieving a specific goal, e.g. making a nondeterministic choice of one alternative over the other in case one need a web service of type T and two concrete web services of that type, Tws_1 and Tws_2 are available but just one needs to be chosen.

Opinion-based BRT functions are introduced and serve for modeling helpful behavior in the very same way their PRT counterpart do; each of these functions works on the goal in the BRT's root and evaluates the most appropriate recipe for achieving it.

Each node in the BRT has associated a specific opinion (as opposite to a mere probability), which represents its likelihood of being successful. For leaf nodes (corresponding to atomic WS) the *cba* property is estimated based on r and s. Each time an agent, either architect or consultant, employs a specific WS, it logs the performance of that WS, labeling it as either *positive* or *negative*, thus increasing r or s respectively. We assume the existence of an underpinning WS taxonomy, which classifies each WS given by its URI into a class according to its functionality. The performance of a specific WS will be recorded in conjunction with its corresponding functionality class. For example, the *http://www.webservicex.net/globalweather.asmx?wsdl* web service will have its performances logged as a $WeatherForcastWS$ class member. Later on, if a WS in the $WeatherForcastWS$ class is needed, *http://www.webservicex.net/ globalweather.asmx?wsdl* will be considered for selection. The opinion for a node at this tree level is then updated based on r and s:

$$\omega_N = (\frac{r}{r+s+2}, \frac{s}{r+s+2}, \frac{2}{r+s+2}, a) \tag{3.1}$$

For a non-terminal node N, the opinion ω_N is computed based on those of the node descendants:

1. AND node: if D_N is the subset of all direct descendants of an AND node N, ω_s is the opinion in a direct descendant s of N (where $s \in D_N$), then the opinion ω_N in the node N is:

$$\omega_N = \bigsqcap_{s \in D_N} \omega_s \tag{3.2}$$

It gives the likelihood of N being successful based on the success of every subtree of it.

2. OR node: if D_N is the subset of all direct descendants of an OR node N, ω_s is the opinion in a direct descendant s of N (where $s \in D_N$), $\omega_{b(s)}$ is the opinion associated to the tree branch going from N to s, then the opinion ω_N is:

$$\omega_N = \bigsqcup_{s \in D_N} \omega_{b(s)} \sqcap \omega_s \tag{3.3}$$

This opinion describes the likelihood of N being successful taking into consideration the potential of success of each subtree alone. One should note that, for an OR node, each descending branch (not just the nodes) is endowed with an opinion. These branch opinions are aimed to model information of the type "for the job X, 70% of the specialists would recommend solution Y", thus expressing knowledge about recipes rather than about particular executors. They are manipulated like any other BRT opinion.

Costs of each node are computed as below:

1. leaf node N: $cost_N$ is assumed known and given
2. AND node: if D_N is the subset of all direct descendants of an AND node N:

$$cost_N = \sum_{s \in D_N} cost_s \tag{3.4}$$

3. OR node: if D_N is the subset of all direct descendants of an OR node N, $\omega_{b(s)}$ is the opinion associated to the tree branch going from N to s and $EV(\omega_{b(s)})$ its expected value:

$$cost_N = \sum_{s \in D_N} cost_s * EV(\omega_{b(s)}) \tag{3.5}$$

Expected utilities of each node describe its profit ($income - cost$), weighted by the likelihood of this being achieved. They and are computed as follows:

1. leaf node N: if $income(N)$ is the price the customer agreed to pay for the delivery of N (this is assumed to be known by agents), then the profit will be:

$$eval(N) = income(N) - cost(N) \tag{3.6}$$

2. AND node: if D_N is the subset of all direct descendants of an AND node N, $income(N)$ is the price the customer agreed to pay for the delivery of N, ω_N is the opinion associated to node N and $EV(\omega_N)$ its expected value:

$$eval(N) = EV(N) * income(N) + \sum_{s \in D_N} eval(s) \tag{3.7}$$

3. OR node: if D_N is the subset of all direct descendants of an OR node N, $income(N)$ is the price the customer agreed to pay for the delivery of N, ω_N is the opinion associated to node N and $EV(\omega_N)$ its corresponding expected value, $\omega_{b(s)}$ is the opinion associated to the tree branch going from N to s and $EV(\omega_{b(s)})$ its expected value:

$$eval(N) = EV(N) * income(N) + \sum_{s \in D_N} eval(s) * EV(\omega_{b(s)}) \tag{3.8}$$

3.3. BRT for helpful behavior. We used the BRT structure for making decisions in case of a cooperative activity where only partial team member involving is needed, though without risking to compromise the global team goal.

The kind of decision to be made is as follows: if a team comprising A and C aim at building a composite WS and C has just gained some knowledge about new possible recipes for building a part of the system, is it rational, in team profit terms, for C to inform A about this or not? The following algorithm, similar to that in [14], is proposed for making such a decision.

The predicate $Committed(G_1, GR, \alpha)$ uses the belief recipe tree BRT_α, which has α in its root, in order to select the design solution α over any other similar solution β and commit to it. As an example, α might be a design goal like "Build a route generator WS". Formally, G_1 is committed to α iff G_1 believes that BRT_α maximizes the group's GR utility:

$$\begin{gathered}\exists BRT_\alpha \; BEL(G_1, \; \forall BRT_\beta \; BRT_\beta \neq BRT_\alpha \; \Rightarrow \\ Eval(BRT_\beta) \leq Eval(BRT_\alpha)) \; \wedge Int.Th(G_1, SelectedBRT(BRT_\alpha))\end{gathered} \tag{3.9}$$

If an agent is committed to α and believes that sending information o to his/her partner will increase the group utility, he/she will do so (see Algorithm 3.3.1).

The next section presents an trace excerpt of this algorithm in order to decide over helpful behavior.

Algorithm 3.3.1 Commitment decision

if $Committed(G_1, GR, \alpha)$ **then**
  $BRT_\alpha = PredictBRT(G_1, GR, \alpha, C_{GR})$
  $C_\beta = ContextUpdate(C_\beta, o)$
  $BRT_\beta = PredictBRT(G_1, G_2, \beta, C_\beta)$
  $BRT_\alpha^o = BRTReplace(BRT_\alpha, BRT_\beta)$
  $utility = Eval(BRT_\alpha^o) - Eval(BRT_\alpha)$
end if
if $utility \geq CommunicationCost(G_2)$ **then**
  $Int.To(G_1, Communicate(G_1, G_2, o))$
end if

4. Running model. The example in this section describes a simple scenario in which a team of agents make cooperative decisions about sharing knowledge using BRT for assessing the benefits of this.

Let us suppose we have two agents: a system architect A and a consultant C who intend to build a web service called *Navigator*. This web service should build routes between pairs of addresses for car drivers. When building such a route, the system takes into consideration information about the existing ways between start and destination points, as well as information concerning the traffic on various routes and about the weather forcast in that area. We will assume there exist simple web services offering each type of information above; they are respectively called Mws_1, Mws_2, Mws_3 (map web services), Tws_1, Tws_2 (traffic info web services) and $Wfws_1, Wfws_2$ (weather forcast services).

The whole system can be built as a combination of two web services: *Route Generator* and *Weather Forcast*. The former prepares routes taking into consideration both map and traffic information (thus having two service components: *Mapws* and *Trafficws*); the latter returns information concerning the weather in the region which will be visited. The customer agreed to pay money both for the partial components (e.g for the *Route Generator* part) as well as for the final delivery. Building the whole system will bring an income of \$40K; developing the *Route Generator* alone will bring \$30K, while the *Weather Forcast* part will get \$2K.

We assume both A and C know general recipes for building such a system, so their BRTs are similar; however, their knowledge about specific web services capable of carrying each atomic task might be different. We consider this assumption reasonable for software developers. A depiction of the common BRT is given in Figure 4.1. A square nodes represents an individual WS, while a circle corresponds to either an AND or an OR node.

Building the system can be seen as a collaborative task involving agents A and C. Generating the solution recipe is equivalent to building a shared plan describing how each step is performed. Agent A must therefore come up with a list specifying either a recipe or an atomic web service for each needed task (e.g. the *Route Generator* will be done by combining Mws_2 and Tws_1 into a *Simple Route Generator*, while $Wfws_1$ will do the job of *Weather Forcast*).

Consultant C has just learned there might also exist a web service called *Complex Route Generator* which does the jobs of *Mapws* and *Trafficws* in one step; however, architect A is not aware of this yet.

The problem consultant C tries to solve is whether he/she should get involved into conveying this new piece of knowledge to A or not. In order to achieve this, C makes an assessment of the chances/costs for A building the system, based on his own knowledge.

The agent which builds the plan has information about each atomic service above; this information is kept in form of the number of positive (success) and negative (failure) experiences the agent has recorded when exploring a specific service. Please note that, for the time being, every experience can have only 2 outputs: $success$ or $failure$. E.g. for the service Mws_1, the number of prior successes is 1 and of prior failures is also 1. Thus, according to Section 3, the opinion concerning Mws_1 is $(0.25, 0.25, 0.50, 0.50)$.

4.1. Running sample. This subsection traces step-by-step the above scenario; it presents the opinions at each level and also the decision to be made by the consulting agent C.

Table 4.1 summarizes the opinions of C concerning different individual web services. For example, in the first line, one can see the information concerning Mws_1. The number of positive prior experiences r was 1, the number of prior negative experiences s was also 1, thus the opinion concerning Mws_1 is $(0.250, 0.250, 0.500, 0.500)$.

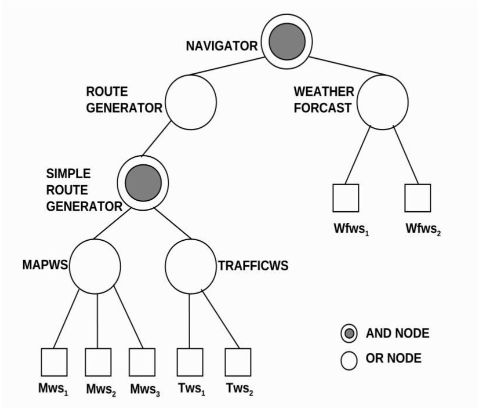

FIG. 4.1. *Web Service selection example*

TABLE 4.1
Atomic WS evidence and opinions

WS	**r**	**s**	**b**	**d**	**u**	**a**
Mws_1	1	1	0.250	0.250	0.500	0.500
Mws_2	3	1	0.500	0.167	0.333	0.500
Mws_3	1	2	0.200	0.400	0.400	0.500
Tws_1	5	1	0.625	0.125	0.250	0.500
Tws_2	3	2	0.428	0.286	0.286	0.500
$Wfws_1$	78	22	0.765	0.216	0.019	0.500
$Wfws_2$	67	33	0.657	0.222	0.111	0.500

C should also take into consideration the opinions for each branch. As presented in Section 3 the opinions concerning these options are used to model knowledge of the type "for the job X, 70% of the specialists would recommend solution Y", without specifying who should actually perform Y.

Considering the OR-type node *Mapws*, let us assume that the opinions for its descending branches Mws_1, Mws_2 and Mws_3 have the values $(0.200, 0.400, 0.400, 0.50)$, $(0.250, 0.250, 0.500, 0.50)$ and $(0.400, 0.200, 0.400, 0.50)$ respectively. Then, the opinion in the *Mapws* node is obtained by performing a logical OR among the logical ANDs from each branch, which leads to the result $(0.433, 0.288, 0.279, 0.578)$.

For the (AND-type) node *Simplerg*, the opinion in the node is computed as a simple logical AND between its descendants; the result would be $(0.345, 0.465, 0.190, 0.253)$.

Following the algorithm described above we get that the opinion for the *Navigator* task is $(0.227, 0.635, 0.138, 0.553)$.

Table 4.2 presents the opinions assumed for each branch and the opinions computed on this ground for every node in the BRT.

For the given costs and profits, we get *Eval*(*Navigator*) = \$5,202.40. Excerpts of this computation process are presented in Table 4.3, which contains the probability expectation E, cost, income and expected value *Eval* for the nodes in the example tree.

Now agent C might take into consideration the fact that he/she has just discovered another way of building the *Route Generator* part: there exists a new service called *Complex Route Generator* which does this job, having cost \$10,000. For the individual and branch opinions of $(0.666, 0.167, 0.167, 0.500)$ and $(0.333, 0.0, 0.667, 0.500)$ respectively, the new value for *Eval*(*Navigator*) will be \$7,919.38, thus it worth telling A about this, as long as the cost of communication is lower than the additional benefit. Figure 4.2 presents the new situation.

Let us suppose that the web service fails because the selected CRG_1 has failed twice. Then, the evidence and beliefs are adjusted to reflect this experience; their new values are those in Table 4.4.

In this case, the corresponding evaluations are \$5,202.40 (no composite web service disclosure) and \$4,180.70 respectively. In this case, the rational choice for C would be not to suggest an alternative for the route generator part. One should notice this is not feasible in the classical approach when the values of probabilities remain the

TABLE 4.2
Plan alternatives opinions

WS	**b**	**d**	**u**	**a**
Mapws to Mws_1	0.200	0.400	0.400	0.500
Mapws to Mws_2	0.250	0.250	0.500	0.500
Mapws to Mws_3	0.400	0.200	0.400	0.500
Node *Mapws*	0.433	0.288	0.279	0.578
TRAFFICWS to Tws_1	0.600	0.200	0.200	0.500
TRAFFICWS to Tws_2	0.375	0.375	0.250	0.500
Node TRAFFICWS	0.590	0.249	0.161	0.437
Node *Simplerg*	0.345	0.465	0.190	0.253
RG to *Simplerg*	0.666	0.167	0.167	0.500
Node RG	0.272	0.555	0.173	0.126
$Wfws_1$	0.714	0.143	0.143	0.500
$Wfws_2$	0.667	0.222	0.111	0.500
Node *Wf*	0.780	0.181	0.390	0.437
Node NAV	0.227	0.635	0.138	0.553

TABLE 4.3
Costs and profits

WS	**E**	**Cost**	**Income**	**Eval**
Mws_1	0.500	1,000.00	0.00	
Mws_2	0.666	2,000.00	0.00	
Mws_3	0.400	3,000.00	0.00	
Mapws	0.594	3,033	0.00	
Tws_1	0.500	4,000.00	0.00	
Tws_2	0.666	5,000.00	0.00	
Route	0.523	0.00	30,000.00	2,170.80
Weather	0.608	0.00	2,000.00	950.86
Navigator	0.545	0.00	40,000.00	5,202.40

same and the selection could change due to $Selected\_PRT$ behavior only.

This example illustrates the advantages of a BRT over a PRT. A BRT allows building trust from prior mutual experiences and also updating it when further evidence becomes available, rather than keeping it unchanged. It also provides more flexibility as it takes knowledge uncertainty, due to the lack of enough evidence, into account when making decisions.

4.2. PRT versus BRT and multinomial BRT. In a scenario similar with the above one, we take into consideration 2 situations: in the first one, the first weather forcast WS has a performance record of $r = 3000, s = 1000$; in the second situation, the same WS has a performance record of $r = 3, s = 1$. The values for all other WS are unchanged. Since the ratio r/s is the same, the corresponding PRT would be identical in the two situations. The BRT version would led to the results in Table 4.5.

If an uncertainty threshold is taken into consideration, e.g. 0.01, the BRT will lead to different decisions: in situation 1, the best is to commit to the weather forcast service, while in situation 2, this decision will be blocked and probably more evidence will be required. Considering uncertainty in a BRT offers the opportunity of making this type of decisions.

Table 4.6 shows the results in case more than 2 levels of quality are considered for the QoS. Here, we have a simple situation where only the *Route* and $Weather$, together with $Navigator$, are considered. The possible values for the $Weather$ WS performance are *good* and *bad*; *Route* can be *blue*, *silver* and *gold*. Let us consider the vector values: $(0.780, 0.181, 0.039, 0.500, 0.500)$ for $Weather$ and $(0.272, 0.025, 0.530, 0.173, 0.500, 0.400, 0.300)$ for *Route*. In this case, the expected value for the system to offer a *good* weather forcast AND a *gold* level route generator is 0.110. This offers the designer the opportunity to assess the system performance in a more meaningful manner, as opposite to a mere "does/does not work" in the previous version of the BRT.

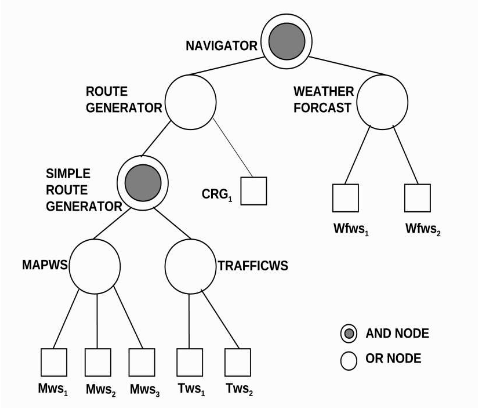

Fig. 4.2. *Adding an alternative plan*

Table 4.4
Alternative atomic WS evidence and opinions

WS	**r**	**s**	**b**	**d**	**u**	**a**
Mws_1	1	1	0.250	0.250	0.500	0.500
Mws_2	3	1	0.500	0.167	0.333	0.500
Mws_3	1	2	0.200	0.400	0.400	0.500
Tws_1	5	1	0.625	0.125	0.250	0.500
Tws_2	3	8	0.428	0.286	0.286	0.500
$Wfws_1$	78	22	0.765	0.216	0.019	0.500
$Wfws_2$	67	33	0.657	0.222	0.111	0.500
CRG_1	1	2	0.200	0.400	0.400	0.500

5. Related work. Within the context of knowledge potential uncertainty, combined with differences in goals and/or commitments, an agent can employ trust, seen as a "reduction of complexity" mechanism [17], inside the decision making process. Pressure for building realistic trust models comes from the area of delegation in multi-agent systems (selecting appropriate partners for tasks which require cooperation), as well as from the problem of security and quality of the knowledge available in large scale open environments (e.g. in the field of Semantic Web [1]). Thus, trust as an estimation of the likelihood of success of a specific task, was included among the ingredients involved in decision making and is present in many papers and research lines. Among them, we count the multi-agent systems and service-oriented architectures. Our work borrows ideas from both fields in order to tackle the problem of building web services.

In the area of multi-agent systems, the approach presented in [19, 20] aims to incorporate the well known Repage reputation model [21] into a BDI architecture and to model beliefs, desires and intentions as contexts connected by bridge rules, like in [6]. A Repage context is added, whose mission is to aggregate the information obtained from third party sources into sentence credibility level. A particular probabilistic logic is further employed in order to allow context manipulation; intentions are generated by an inference process conducted within this logic. However, the mechanism focuses on the perspective of a specific agent rather than committing to a shared goal as a part of a team.

The improved SharedPlan formalism in [8, 7] introduces a model for the dynamics of agent intentions in collaborative activity which integrates group decision making and group intentions updating. Agents can commit to joint activities and also can have only partial knowledge about how to perform an action, but the formalism does not address uncertainty aspects. In order to solve this issue, paper [14] introduced the Probabilistic Recipe Tree, which defined a probability distribution over the potential recipes which can solve the goal in its root node. The PRT structure is arguably more compact and works faster than the prior solutions (e.g. the mechanism described in [24]). However, the way the probabilities involved in reasoning are estimated and adjusted is not given. The solution our paper proposes for this is to use subjective probabilities. This will offer the opportunity of both estimating the probabilities those and modeling the uncertainty about these

Table 4.5
BRT versus PRT decisions

Sit.	**r**	**s**	**b**	**d**	**u**
1	3000	1000	0.749	0.250	0.001
2	3	1	0.5	0.167	0.333

Table 4.6
Multinomial Navigator WS expected values and opinions

Navigator	belief			Expected value		
	blue	*silver*	*gold*	*blue*	*silver*	*gold*
bad	0.263	0.057	0.433	0.287	0.075	0.438
good	0.049	0.001	0.105	0.072	0.019	0.110

estimations as no large number of mutual prior interactions is usually available. Extending the PRT into the Belief Recipe Tree structure and adjusting all operations accordingly represent the main contributions of this paper.

Trust has also gained attention to the researchers in the service-oriented field. For example, paper [25] employs a novel trust model in order to solve the service collaboration problem. The model enjoys a set of desirable properties, such as temporal dynamics, context-awareness and the possibility of exchanging opinions concerning trustees. The service providers make simple decisions like granting and revoking resource access based on the trust level computed this way. The approach we presented allows reasoning about complex actins, i.e. actions whose completion involves more than one actions and one goal. Trust is involved at least at two levels: at the atomic level, by measuring one entity's own capability, and at the composed level, by assessing the global likelihood of success for the combined goal. The model fullfills the requirements of context-awareness, rule-orientation, non-symmetry, and temporal dynamics.

The paper [16] proposes a method for finding a possible reconfiguration region containing replaceable services which allow the system to meet the original QoS specification if a service fails and affects a complex of processes. However, the need of reasoning in advance over the likelihood of such an event lead to a more general approach including trust as a measure of service reliability. We took this latter path in this paper.

Paper [26] deals with incremental trust evaluation of WS providers, based on the feedback from the WS clients side. A fuzzy logic based method for connecting service period length and reputation is introduces. Our solution is based on incorporating knowledge on particular WS provider and on generic solutions into opinions along tree branches and leaves. The operations defined on BRT allow one to accommodate both types of knowledge.

A similar, but slightly different approach is introduced in [4]. Experiences are recorded objectively, based on a shared ontology aimed to allow describing past interactions in detail, filter the unreliable ones and then integrate them into global opinions. This offers the trustors a standard common language for recording their experiences, but makes no steps for supporting decisions which involves complex goals. We advocate the idea of grounding trust for complex actions on trust levels for basic actions because we consider rather unlikely to have a reasonable amount of experiences of each type of complex tasks in order to make a similarity-based judgment, so trustors will have to rely to a large extent on their own experiences, but being able to decide when to ask for more information.

The work presented in [27] addresses the problem of automatically composing web services within the context of uncertainty about their successfull invocations. Their solution is based on combining probabilistic situation calculus and hierarchical, symbolic planning and can deal with uncertainty and scale efficiently to large compositions. Our solution assumes a number of prior known recipes for hierarchically decomposing a task into subtasks up to the level of basic actions; this might be a weak point when compared to [27] as one cannot guarantee the completeness of such a collection of recipes. On the other hand, our approach offers flexibility on assessing the uncertainty involved in decisions and also on deciding whether the trustor does or does not know enough about a specific trustor in order to start reasoning over the interaction.

6. Conclusions and future work. Many times, the process of developing a complex WS implies collaborative design, hence the importance of deciding to which extent one should assist her peer. This paper has presented a solution to this problem based on the novel BRT structure. A BRT is an extention of a PRT, endowed with trust values in form of subjective opinions describing both service providers and design recipes. Operations on the tree nodes are extended accordingly. The key advantage of it is that it permits estimating the likelihood of success and make rational decisions even if a small number of prior interactions are at hand, adjusting beliefs when further evidence is available, and deciding whether to seek for more information versus to rely on the available one. Agents are offered the possibility to deal with multinomial instead of binomial opinions, for an even richer (i.e. finer grained) set of labels for performance description. Belief Recipe Trees, together with the corresponding versions of algorithms, offer a more realistic approach for the uncertainty present in designing complex SOA, as well as an improved flexibility in making sound decisions.

The first goal for future work is to conduct extensive experiments in scenarios involving an increased number of team members, tasks and resources needed for accomplishing the tasks, in order to investigate the performance and scalability of the approach. This should be combined with different methods for distributing uncertainty (here, we used only the Assumed Uncertainty Mass for the conjunction and a De Morgan based difference for disjunction, but some other solutions should also be explored).

Addressing the problem of accurate estimation of the base rates also deserves attention. Following the line in [13], the base rate can be estimated by the general frequency. The main shortcoming would be that a great amount of observations is still required for high quality estimation. Another line to pursue originates in [2]. Machine learning algorithms can be used for clustering, then learning stereotypes about classes of agents. The learned stereotypes can be further used to estimate the base rates, alone or after aggregating them into stereotypical reputations. This line is more appropriate in agent groups whose life-span is much shorter than that of the whole system and the agent pool is large enough to render the frequency estimation unfeasible. Hopefully these improvements will lead to a more realistic world representation and better agent decisions.

Acknowledgments. We are grateful to the anonymous reviewers for the very useful comments.

REFERENCES

[1] G. Antoniou and F. van Harmelen, *A Semantic Web Primer (Second Edition)*, MIT Press, Cambridge, MA, 2008.
[2] C. Burnett, T. J. Norman, and K. Sycara, *Bootstrapping trust evaluations through stereotypes*, in Proceedings of The 9th International Conference on Autonomous Agents and Multiagent Systems, Toronto, Canada, 2010, pp. 241–248.
[3] J. Cardoso, J. Miller, A. Sheth, and J. Arnold, *Quality of service for workflows and web service processes*, Journal of Web Semantics, 1 (2004), pp. 281–308.
[4] M. Şensoy, J. Zhang, P. Yolum, and R. Cohen, *Poyraz: Context-aware service selection under deception*, Computational Intelligence, 25 (2009), pp. 335–366.
[5] A. P. Dempster, *Upper and lower probabilities induced by a multivalued mapping*, The Annals of Mathematical Statistics, 38 (1967), pp. 325–339.
[6] F. Giunchiglia and C. Ghidini, *A local models semantics for propositional attitudes*, in Formal Aspects of Context, P. Bonzon, M. Cavalcanti, and R. Nossum, eds., Kluwer Academic Publishers, Dordrecht, 2000, pp. 161–174.
[7] B. J. Grosz and L. Hunsberger, *The dynamics of intention in collaborative activity*, Cognitive Systems Research, 7 (2006), pp. 259 – 272.
[8] B. J. Grosz and S. Kraus, *Collaborative plans for complex group action*, Artificial Intelligence, 86 (1996), pp. 269–357.
[9] A. Jøsang, *A logic for uncertain probabilities*, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 9 (2001), pp. 279–311.
[10] A. Jøsang, *Subjective Logic*, Draft book, 2010.
[11] A. Jøsang and D. McAnally, *Multiplication and comultiplication of beliefs*, International Journal of Approximate Reasoning, 38 (2004), pp. 19–51.
[12] A. Jøsang and S. O'Hara, *Multiplication of multinomial subjective opinions*, in Proceedings of the 13th International Conference on Information Processing and Management of Uncertainty (IPMU2010), Dortmund, Germany, 2010, pp. 248–257.
[13] A. Jøsang, S. O'Hara, and K. O'Grady, *Base rates for belief functions*, in Proceedings of the Workshop on the Theory of Belief Functions, 2010.
[14] E. Kamar, Y. Gal, and B. J. Grosz, *Incorporating helpful behavior into collaborative planning*, in Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, Budapest, Hungary, 2009, pp. 875–882.
[15] I. A. Letia and R. R. Slavescu, *Helpful behavior based on trust for web services*, in Proceedings of the 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania, 2010, pp. 526–533.
[16] K.-J. Lin, J. Zhang, Y. Zhai, and B. Xu, *The design and implementation of service process reconfiguration with end-to-end qos constraints in soa*, Service Oriented Computing and Applications, (2010), pp. 1–12.
[17] N. Luhmann, *Trust and Power*, John Wiley & Sons Ltd., 1979.
[18] T. J. Norman and C. Reed, *A model of delegation for multi-agent systems*, in Foundations and applications of Multi-Agent Systems, LNAI 2403, Springer-Verlag, 2002.
[19] I. Pinyol and J. Sabater-Mir, *Pragmatic-strategic reputation-based decisions in BDI agents*, in Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, Budapest, Hungary, 2009, pp. 1001–1008.

[20] I. Pinyol, J. Sabater-Mir, P. Dellunde, and M. Paolucci, *Reputation-based decisions for logic-based cognitive agents*, 2010, pp. 1–42.
[21] J. Sabater-Mir, M. Paolucci, and R. Conte, *Repage: Reputation and image among limited autonomous partners*, Journal of Artificial Societies and Social Simulation, 9 (2006), p. 3.
[22] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, 1976.
[23] P. Smets and R. Kennes, *The transferable belief model*, Artificial Intelligence, (1994), pp. 191–234.
[24] M. Tambe and W. Zhang, *Towards flexible teamwork in persistent teams: Extended report*, Autonomous Agents and Multi-Agent Systems, 3 (2000), pp. 159–183.
[25] M. Uddin, M. Zulkernine, and S. Ahamed, *Collaboration through computation: incorporating trust model into service-based software systems*, Service Oriented Computing and Applications, 3 (2009), pp. 47–63.
[26] Y. Wang, K.-J. Lin, D. S. Wong, and V. Varadharajan, *Trust management towards service-oriented applications*, Service Oriented Computing and Applications, 3 (2009), pp. 129–146.
[27] H. Zhao and P. Doshi, *A hierarchical framework for logical composition of web services*, Service Oriented Computing and Applications, 3 (2009), pp. 285–306.

Edited by: Dana Petcu and Alex Galis
Received: March 1, 2011
Accepted: March 31, 2011