The benefits of surprise in dynamic environments: from theory to practice

Emiliano Lorini, Michele Piunti

The Benefits of Surprise in Dynamic Environments: From Theory to Practice*

Emiliano Lorini[1,2] and Michele Piunti[1,3]

[1] ISTC - CNR, Rome, Italy
[2] IRIT, Toulouse, France
[3] Università degli studi di Bologna - DEIS, Bologna, Italy
{emiliano.lorini,michele.piunti}@istc.cnr.it

Abstract. Artificial agents engaged in real world applications require accurate resource allocation strategies. For instance, open systems may require artificial agents with the capability to filter out all information which are irrelevant with respect to the actual intentions and goals. In this work we develop a model of surprise-driven belief update. We formally define a strategy for epistemic reasoning of a BDI-inspired agent, where surprise is the causal precursor of a belief update process. According to this strategy, an agent should update his beliefs only with inputs which are surprising and relevant with respect to his current intentions. We also compare in practice the performances of agents using a surprise-driven strategy of belief update and agents using traditional reasoning processes.

"*A wealth of information creates a poverty of attention, and a need to allocate that attention efficiently*"
[H. A. Simon talks at Johns Hopkins & CIOS Conf. in Tokyo, Fall 1969].

1 Introduction

Realistic cognitive agents are by definition resource-bounded [1], hence they should not waste time and energy in reasoning out and reconsider their knowledge on the basis of every piece of information they get. They need some filter mechanism which is responsible: 1) for signaling the inconsistency between beliefs and an incoming input which is relevant with respect to the current task; 2) for the revision of beliefs and expectations on the basis of the incoming relevant information. Our claim is that one of the main functions of surprise in cognitive agents is exactly this. In this work we will develop a computational model of a cognitive agent where a surprise-based filter of belief change is implemented. The computational model we will present consists in the operationalization of two general hypothesis. On one hand, we suppose that at each moment an agent is focused and allocates his attention on a particular task that he is trying to solve and on a certain number of intentions which represent the pragmatic solutions that the agent has selected in order to accomplish the task [2]. The agent ignores all incoming inputs which are not relevant with respect to the current task on which he is focused

* This research is supported by the European Project MindRACES (IST-511931).

and only considers those information which are relevant. On the other hand, we suppose that if a relevant input turns out to be incompatible with respect to the pre-existent beliefs of the agent, surprise arises. The surprise reaction is a causal precursor of a belief update process. In fact, a surprise with a certain intensity relative to the incoming relevant input "signals" to the agent that things are not going as expected and that beliefs must be reconsidered. Other authors [3,4] have attributed to surprise a precise and crucial functional role in mind by stressing that it is perhaps the most important causal precursor of a process of belief change.[1] The main objective of this paper is to clarify such a functional role of surprise in mind by integrating a surprise-based mechanism of belief update into a $belief - desire - intention$ (BDI) computational model [7,8]. The BDI is a well-established framework which is aimed at describing an agent's mental process of deciding moment by moment on the basis of current beliefs, which action to perform in order to achieve some goals.[2] The computational model of surprise-based belief change presented in this paper has also the ambition to bridge the existing gap between formal and computational models of belief change and psychological models of belief dynamics. Indeed, most of the authors in the tradition of belief change theory have been mainly interested in finding rationality principles and postulates driving belief change (this is for instance the main purpose of the classical AGM theory [10]) without investigating the causal precursors of this kind of process (they have implicitly assumed that when an agent perceives some fact such a perception is always a precursor of a belief change).

The paper is organized as follows. In section 2 we provide the abstract model of a BDI cognitive agent by formalizing his informational attitudes (volatile beliefs and expectations which change over time and the stable knowledge about the dependencies of objects in the environment) and motivational attitudes (intentions and desires). In section 3 we apply the abstract model of a BDI cognitive agent to a specific foraging scenario. In section 4 the cognitive architectures of two general typologies of BDI agents are designed. The first typology corresponds to a standard BDI agent [7,8]. The second typology corresponds to a BDI agent endowed with a surprise-based filter of belief update (we call it $BDIS$ agent). In section 5 we report the results of some simulative studies in the scenario described in section 3. We compare the performances of the BDI agent and $BDIS$ agent in different conditions of environmental dynamism.

2 The Abstract Model of an Agent's Mental State

The abstract model of an agent's mental state is made of a set of $n \leq 1$ random variables **VAR** $= \{X_1, ..., X_n\}$. We suppose that each random variable $X_i \in$ **VAR** takes values from the set $Val_{X_i} = \{x_1, ..., x_r\}$, with $r > 1$. For each set Val_{X_i} we define the corresponding set $Inst_{X_i} = \{(X_i = x_1), ..., (X_i = x_r)\}$ of all possible instantiations

[1] Other functional roles have been attributed to surprise. For instance, some authors conceive surprise as a shortcut for attention [5]. The felt feedback of surprise is responsible for redirecting attention towards the unexpected and surprising stimuli, and for concentrating cognitive resources on them. According to other authors surprise is responsible for a shift from an automatic level of performance to a deliberate level [6].

[2] The idea to introduce emotions in a BDI system is not new. See for example [9].

of random variable X_i. Besides, we write $Inst = \bigcup_{X_i \in \textbf{VAR}} Inst_{X_i}$ to denote the set of all possible instantiations of all random variables.

We have a set $\Gamma \subseteq Inst$ of perceived data which fixes the value of certain variables that an agent perceives at a certain moment. For example, $\Gamma = \{(X_i = x_i)\}$ means "an agent sees that the observable variable X_i has value x_i". We denote with $\Gamma_{Var} = \{X_i \in \textbf{VAR} | \exists x_i \ s.t.(X_i = x_i) \in \Gamma\}$ the subset of **VAR** which includes the variables that an agent observes at a certain moment, that is, all those variables which have (at least) one instantiation in Γ. Here we suppose that for all $X_i \in \Gamma_{Var}, Inst_{X_i} \cap \Gamma$ is a singleton, that is, we suppose that an agent cannot perceive two different instantiations of the same variable. We use the notation $\Gamma(X_i)$ to denote that singleton, that is, for any $X_i \in \Gamma_{Var}, Inst_{X_i} \cap \Gamma = \Gamma(X_i)$.

We also use a simple bayesian network K which represents the joint probability distribution over the set of random variables **VAR**. A bayesian network is a directed acyclic graph (DAG) whose nodes are labeled by the random variables in **VAR** and the edges represents the causal influence between the random variables in **VAR**. Given an arbitrary random variable X (i.e. an arbitrary node) in the bayesian network K we denote with $anc(X)$ the ancestors of X. Formally, Z is an ancestor of X in the bayesian network K if there is a directed path from Z to X in K. Moreover, given an arbitrary random variable X in the bayesian network K, we denote with $par(X)$ the parents of X in the bayesian network. Formally, Z is a parent of X in the bayesian network K if Z is an ancestor of X in K which is directly connected to Z. Finally, we associate to each random variable X in K a conditional probability distribution $P(X|par(X))$, where each conditional probability distribution can be represented in terms of a conditional probability table. The bayesian network K encodes the agent's causal knowledge of the environment. Here we suppose that this part of the agent's knowledge is stable and can not be reconsidered.

In our general model, we also encodes the agent's beliefs and expectations that can change over time, i.e. the agent's volatile expectations and beliefs [11]. Given a random variable $X_i \in \textbf{VAR}$, we denote with $\sum_{X_i}$ the set of all possible probability distributions over the random variable X_i. Then, we denote with $\prod_{X_i \in \textbf{VAR}} \sum_{X_i}$ the set of all possible combinations of probability distributions over the random variables in **VAR**. Besides, we denote with $\sigma, \sigma', ... \in \prod_{X_i \in \textbf{VAR}} \sum_{X_i}$ specific combinations $\{\sigma_1, ..., \sigma_n\}, \{\sigma'_1, ..., \sigma'_n\}, ...$ of probability distributions over each random variables in **VAR**. Given a certain σ, every $\sigma_i \in \sigma$ corresponds to a set $\sigma_i = \{[(X_i = x_1) = a_1], ..., [(X_i = x_r) = a_r]\}$ of probability assignments $a_1, ..., a_r \in [0, 1]$ to each possible instantiations of the variable X_i. Now, we denote with $B = \bigcup_{\sigma_i \in \sigma} \sigma_i, B' = \bigcup_{\sigma'_i \in \sigma'} \sigma'_i, ...$ specific configurations of beliefs of the agent, and with $BEL = \{\bigcup_{\sigma_i \in \sigma} \sigma_i | \sigma \in \prod_{X_i \in \textbf{VAR}} \sum_{X_i}\}$ the set of all possible configurations of beliefs of the agent. Given a specific configuration of beliefs $B = \bigcup_{\sigma_i \in \sigma} \sigma_i$, we write $B(X_i = x_j) = a_j$ if and only if $[(X_i = x_j) = a_j] \in \sigma_i$. Thus, $B(X_i = x_j) = 0.4$ means that given the configuration of beliefs B the agent assigns probability 0.4 to the fact that variable X_i takes value x_j. We denote with $B(X_i = x_j)$ the number $a_j \in [0, 1]$ such that $B(X_i = x_j) = a_j$.

We also model motivational attitudes by denoting with INT the set of potential intentions of an agent. Here we suppose that every instantiation of a variable in **VAR** is a potential intention of the agent, that is, we suppose that $INT = Inst$. Thus every instantiation of a variable corresponds to a result that the agent can intend to achieve. We denote with $I, I', ... \in 2^{INT}$ specific sets of intentions of the agent. Given a specific set I of intentions of the agent, we denote with $I_{Var} = \{X_i \in \mathbf{VAR} | \exists x_i\ s.t.(X_i = x_i) \in I\}$ the subset of **VAR** which includes all intended variables, that is, all those variables which have (at least) one instantiation in I. As for intentions, we specify a set $DES = Inst$ of potential desires. We denote with $D, D', ... \in 2^{DES}$ specific sets of desires of the agent.

We specify a set MER of means-end rules and a set PR of planning rules. A means-end rule in MER is a desire-generation rule in the style of [12] of the form:

$\psi_1, ..., \psi_s | \lambda_1, ..., \lambda_j \implies \varphi_1, ..., \varphi_t$. Such a rule is responsible for generating t desires $\varphi_1, ..., \varphi_t$ when the agent has s beliefs $\psi_1, ..., \psi_s$ and j intentions $\lambda_1, ..., \lambda_j$. The set MER of means-end rules corresponds to the function $options : BEL \times 2^{INT} \mapsto 2^{DES}$. This function returns a specific set D of desires, given a specific configuration B of beliefs and a specific set I of intentions. A planning rule in PR is a plan-generation rule of the form: $\psi_1, ..., \psi_s | \lambda_1, ..., \lambda_j \Longrightarrow \varphi_1, ..., \varphi_t$. Such a rule is responsible for generating t plans $\varphi_1, ..., \varphi_t \in \mathit{\Pi}$, where $\mathit{\Pi}$ is the repertoire of actions of our agent, when the agent has s beliefs $\psi_1, ..., \psi_s$ and j intentions $\lambda_1, ..., \lambda_j$. The set PR of planning rules corresponds to a function: $plan : BEL \times 2^{INT} \mapsto 2^{\mathit{\Pi}}$. This function returns a set π of plans, given a specific set B of beliefs and specific set I of intentions.

To summarize, a mental state of an agent is defined in our abstract model as a tuple $(B, D, I, K, MER, PR, \mathit{\Pi})$, where each element in the tuple is defined as before.

3 From the Abstract Model to the Experimental Scenario

Our experimental scenario is represented by the 8×8 grid in Fig. 1a. An agent moves in the grid being driven by the goal of finding fruits of a certain color, according to the ongoing season. Indeed, agents look for fruits of different colors in different seasons of the year. We suppose that there are three different seasons and related colors of fruits and trees: the red season, the blue season and the green season. Agents are intrinsically motivated to look for and to eat red fruits during the red season, blue fruits during the blue season and green fruits during the green season. Environmental dynamics are characterized by periodic season cycles: after s_t rounds the season changes on the basis of a periodic function and the intrinsic motivation of an agent changes accordingly. Fruits of any color occupy cells (i, j) (with $1 \leq i \leq 16$ and $1 \leq j \leq 4$), whilst trees of any color occupy macro areas i of size 2×2 (with $1 \leq i \leq 16$) in the grid depicted in Fig. 1a. We suppose that at each moment for every color there is exactly one fruit and tree of that color in the grid. We suppose an objective dependence between trees and fruits in the grid. Indeed, a fruit of a certain color is a sign of the presence of a fruit of the same color in the immediate neighborhood. Agents exploit these signs during their search of fruits. We suppose that a tree of any color is randomly placed in a macro area i of size 2×2. Given a tree of a certain color in a macro area i of size 2×2, a fruit of the same color is randomly placed by the environment simulator in one of the four cells inside the macro area i. For example, if a red tree is in the macro area 1 of the grid then for each cell $(1, 1)$, $(1, 2)$, $(1, 3)$ and $(1, 4)$ there is 0.25 of probability that a red fruit is located in that cell. Fruits and trees change periodically their positions in the grid. More

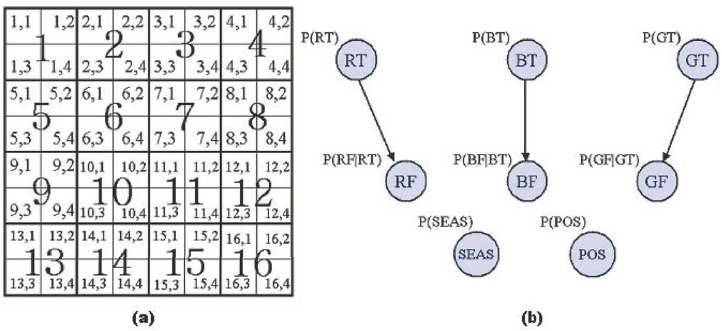

Fig. 1. (a) The Environment grid; (b) The Bayesian Network

precisely, the dynamism factor δ indicates how many seasons have to pass before a tree location changes. We impose constraints on the perceptual capabilities of agents and the related set Γ of perceived data by supposing that an agent sees only those fruits which are in the cells belonging to the same macro-area in which the agent is. For example, if the agent is in cell $(6, 1)$, he only see those fruits which are in the cells belonging to the macro area 6. Moreover we suppose that an agent sees only those trees which are situated in the same macro-area in which the agent is or in the four neighbouring macro areas on the left, right, up or down. For example, if the agent is in cell $(6, 1)$, he only see those trees which are in macro areas $2, 5, 7, 10$.

The knowledge of our agents is encoded by means of 8 random variables **VAR** $= \{SEASON, POS, RF, BF, GF, RT, BT, GT\}$. RF, BF, GF take values from the sets $Val_{RF} = Val_{BF} = Val_{BF} = \{(i, j)|1 \leq i \leq 16, 1 \leq j \leq 4\}$, whilst RT, BT, GT take values from the set $Val_{RT} = Val_{BT} = Val_{BT} = \{i|1 \leq i \leq 16\}$. Finally, $Val_{SEAS} = \{red, blue, green\}$ and $Val_{POS} = \{(i, j)|1 \leq i \leq 16, 1 \leq j \leq 4\}$. Variables RF, BF, GF specify respectively the position of a red/blue/green fruit in the grid depicted in Fig. 1a. Variables RT, BT, GT specify respectively the position of a red/blue/green tree in the grid. For example, $RT = 13$ means "there is a red tree in the macro area 13". Variable $SEAS$ specifies the current season. For example, $SEASON = blue$ means "it is time to look for blue fruits!". Finally, Variable POS specifies the position of the agent in the grid. We suppose that the variables in **VAR** are organized in the bayesian network K as follows: $par(POS) = \{\emptyset\}$, $par(SEAS) = \{\emptyset\}$, $par(RT) = \{\emptyset\}$, $par(BT) = \{\emptyset\}$, $par(GT) = \{\emptyset\}$, $par(RF) = \{RT\}$, $par(BF) = \{BT\}$, $par(GF) = \{GT\}$. This leads to the bayesian network K depicted in Fig. 1b. Since there are 64 possible positions of a fruit in the grid and 16 possible positions of a tree in the grid, each conditional probability table associated with $P(RF|RT)$, $P(BF|BT)$ and $P(GF|GT)$ has $64 \times 16 = 1024$ entries. We suppose that the knowledge of an agent about the dependencies between trees and fruits perfectly maps the objective dependencies between trees and fruits. Hence, we only specify for each tree of a certain color (RT, BT or GT) and arbitrary macro area $i \in \{1, ..., 16\}$ in the grid in which a tree can appear, the 4 conditional probabilities that a fruit of the same color appears in one cell in that macro area. We suppose for each of them the same value 0.25. All other conditional probabilities have value 0, that is, given a tree of certain color which appears in an arbitrary macro area $i \in \{1, ..., 16\}$, the probability that there is a fruit of the same color outside that macro area is zero. More precisely, we have that:

for all $1 \leq i \leq 16$ $P(RF = (i, 1)|RT = i) = P(RF = (i, 2)|RT = i) = P(RF = (i, 3)|RT = i) = P(RF = (i, 4)|RT = i) = 0.25$;
for all $1 \leq i, j \leq 16$ if $j \neq i$ then $P(RF = (j, 1)|RT = i) = P(RF = (j, 2)|RT = i) = P(RF = (j, 3)|RT = i) = P(RF = (j, 4)|RT = i) = 0$.

Means-end rules in MER are exploited by agents for solving the general task of finding a fruit of a certain color in the grid. Agents are endowed with three general classes of means-end rules. The first class includes means-end rules of the following form. For $i \in Val_{SEAS}$: $[(SEAS = i) = 1] \Longrightarrow SEAS = i$.

Such means-end rules are responsible for changing the intrinsic motivation of an agent, according to the season change, that is: if an agent is certain that it is time to look for fruits of kind i (red, blue or green), then he should form the desire to look for fruits of kind i.[3] The second class includes means-end rules of the following form. For

$1 \leq i \leq 16$: $[(RT = i) = 1] \, |SEAS = red \Longrightarrow RT = i$,
$[(BT = i) = 1] \, |SEAS = blue \Longrightarrow BT = i$,
$[(RT = i) = 1] \, |SEAS = green \Longrightarrow GT = i$.

Such means-end rules are responsible for orienting the search of an agent towards a certain macro area, according to the current season (i.e. an intention to find fruits of a certain color) and his beliefs about the position of trees in the grid. For example, if an agent is certain that there is a red tree in the macro area 3 of the grid (i.e. $[(RT = 3) = 1]$) and intends to find a red fruit (i.e. $SEAS = red$), then he should form the desire to reach that position of a red tree (i.e. $RT = 3$). Finally, agents are endowed with means-end rules of the following form. For $1 \leq i \leq 16$ and $1 \leq j \leq 4$:

$[(RF = (i, j)) = 1] \, |SEAS = red \Longrightarrow RF = (i, j)$,
$[(BF = (i, j)) = 1] \, |SEAS = blue \Longrightarrow BF = (i, j)$,
$[(RF = (i, j)) = 1] \, |SEAS = green \Longrightarrow GF = (i, j)$.

Such means-end rules are responsible for orienting the search of an agent towards a certain cell, according to the current season (i.e. an intention to find fruits of a certain color) and his beliefs about the position of fruits in the grid. For example, if an agent intends to find a blue fruit (i.e. $SEAS = blue$) and he is certain that there is a blue fruit in cell $(10, 1)$ of the grid (i.e. $[(BF = 10, 1) = 1]$), then he should form the desire to move towards that position of the blue fruit (i.e. $BF = (10, 1)$). Our agents have a reduced repertoire of actions $\Pi = \{MoveDown, MoveUp, MoveLeft, MoveRight\}$. Indeed, at each round they can only move from one cell to the next one. Planning rules encode approaching policies which depend on the agent's current intentions and his actual position in the grid. Agents have both planning rules for reaching macro areas in the grid (given their current positions) and planning rules for reaching cells in the

[3] In our experimental setting agents are always notified of the fact the season has changed. Therefore, at the beginning of a new season, an agent is certain that it is time to look for fruits of a different color and forms the desire to look for fruits of a different color.

grid (given their current positions). The latter planning rules are exploited for the local search of a fruit of a certain color inside a macro area. Examples of these planning rules are the following:

$[(POS = (15, 1)) = 1]\,|RT = 3 \Longrightarrow MoveUp$,
$[(POS = (10, 2)) = 1]\,|RF = 10, 4 \Longrightarrow MoveDown$.

For instance, according to first planning rule, if an agent intends to reach position 3 of a red tree and is certain to be in cell $(15, 1)$ then he should form the plan to move up.[4].

4 Surprise-Based Filter of Belief Update

Our general aim in this section is to model two different typologies of agents. The first type of agent corresponds to a standard BDI agent whose control loop is described in the right column of Table 1. The second type of agent, whose control loop is described in the left column of Table 1, is a BDI agent endowed with a surprise-based filter of belief update. We call this second type of agent $BDIS$ agent. The formal description of the control loop of the standard BDI agent is similar to [7,8]. In lines 1-2 the beliefs and intentions of the agent are initialized. The main control loop is in lines 3-10. In lines 4-5 the agent perceives some new facts Γ and updates his beliefs according to a function bu. In line 6 the agent generates new desires by exploiting his means-end rules. In line 7 he deliberates over the new generated desires and his current intentions according to the function $filter$.[5] Finally, in lines 8-9 the agent generates a plan for achieving his

Table 1. The two typologies of agents

$\mathcal{BDIS}$ **agent control loop**	$\mathcal{BDI}$ **agent control loop**
1. $B := B_0$;	1. $B := B_0$;
2. $I := I_0$;	2. $I := I_0$;
3. while (true) do	3. while (true) do
4. get new percept Γ;	4. get new percept Γ;
5. if $\mathbf{S}(I, \Gamma, B) > \Delta$ then	5. $B := bu(\Gamma, B)$;
6. $B := bu^*(\Gamma, B, I)$;	6. $D := options(B, I)$;
7. end-if	7. $I := filter(B, D, I)$;
8. $D := options(B, I)$;	8. $\pi := plan(B, I)$;
9. $I := filter(B, D, I)$;	9. $execute(\pi)$;
10. $\pi := plan(B, I)$;	10. end-while
11. $execute(\pi)$;	
12. end-while	

[4] In our experimental setting agents have always access to their current position in the grid.

[5] Space restrictions prevent a formal description of the function $filter$ here (see [7] for a detailed analysis). Let us only note that this function is responsible for updating the agent's intentions with his previous intentions and current beliefs and desires (i.e. $filter : BEL \times 2^{INT} \times 2^{DES} \mapsto 2^{INT}$).

intentions by exploiting his planning rules and he executes an action of the current plan. The main difference between the standard BDI agent and the $BDIS$ agent is the belief update part in the control loop. We suppose that a process of belief update is triggered in the $BDIS$ agent only if the agent perceives a fact and evaluates this as incompatible with respect to the knowledge he has about the things he intends to achieve (line 5 in the control loop of the $BDIS$ agent). In this sense, the $BDIS$ is endowed with a cognitive mechanism of surprise-based belief change. In fact, this mechanism filters out all perceived facts that are irrelevant with respect to the current intentions. Thus, the $BDIS$ agent only updates his beliefs by inputs which are surprising and relevant with respect to his current intentions. Differently, at each round the standard BDI agent updates his beliefs indiscriminately: for any fact he perceives, he updates his beliefs whether the perceived fact is relevant with respect to his intentions or not. In order to model the triggering role of surprise in the $BDIS$ agent, we specify a local surprise function noted by $s(Y = y, \Gamma, B)$. Suppose that $(Y = y) \in I$ then:

$$s(Y=y,\Gamma,B)=\begin{cases} 1 - B(\Gamma(Y)) \\ Condition\ A : [if\ Y \in \Gamma_{Var}] \\ \\ |B(Y=y) - P(Y = y|\ \{X_i = x_i | X_i \in par(Y), X_i = x_i \in \Gamma\})| \\ Condition\ B : [if\ par(Y) \subseteq \Gamma_{Var}\ and\ Y \notin \Gamma_{Var}] \\ \\ 0 \\ Condition\ C : [if\ par(Y) \nsubseteq \Gamma_{Var}\ and\ Y \notin \Gamma_{Var}] \end{cases} \tag{1}$$

According to this function, the degree of local surprise due to the percept Γ and intended fact $Y = y \in I$ is: a) equal to the degree of unexpectedness of the percept Γ, when the intended variable Y is also a perceived variable in Γ_{Var} (i.e. there exists an instantiation of Y which is an element of Γ);[6] b) equal to the degree of discrepancy between the intended fact $Y = y$ and the percept Γ, defined by the absolute value of the difference between the probability assigned to $Y = y$ (i.e. $B(Y = y)$) and the conditional probability that $Y = y$ is true given that the perceived instantiations of the parents of Y are true (i.e. $P(Y = y|\ \{X_i = x_i | X_i \in par(Y), X_i = x_i \in \Gamma\})$), when the intended fact $Y = y$ is not an instantiation of a perceived variable in Γ_{Var} and the parents of Y in the bayesian network K are perceived variables in Γ_{Var};[7] c) 0, when the intended fact $Y = y$ is not an instantiation of a perceived variable in Γ_{Var} and not all Y's parents in the bayesian network K are perceived variables in Γ_{Var}. This third condition corresponds to the irrelevance of the incoming input Γ with respect to the agent's intention $Y = y$. Under this third condition, the agent simply ignores the input, hence he is not surprised by what he perceives.

We define a global surprise function $\mathbf{S}(I, \Gamma, B)$ which returns the maximum value of local surprise for each intended fact $Y = y \in I$.

$$\mathbf{S}(I, \Gamma, B) = \max_{Y=y \in I} s(Y = y, \Gamma, B) \tag{2}$$

[6] According to this function, the degree of unexpectedness of the percept Γ is inversely proportional to the probability assigned by the agent to the perceived instantiation of the intended variable Y, i.e. $B(\Gamma(Y))$. This is similar to the notion of unexpectedness studied in [13].

[7] This corresponds to a sort of surprise based on an inferential process.

This function is used in the control loop of the $BDIS$ agent: if the new percept Γ is responsible for generating a degree of global surprise higher than Δ (with $\Delta \in [0, 1]$) then a process of belief update is triggered and the $BDIS$ agent adjusts his beliefs with the perceived data Γ according to a function bu^*. The belief update function bu^* of the $BDIS$ agent takes in input the set of intentions I, the belief configuration B and the percept Γ and returns an update belief configuration B', that is $bu^* : 2^{Inst} \times BEL \times 2^{INT} \mapsto BEL$. More precisely, suppose that $bu^*(\Gamma, B, I) = B'$ then for all $Y \in$ **VAR**:

1. if $Y \in I_{Var}$ and $Y \in \Gamma_{Var}$ then
   $B'(\Gamma(Y)) = 1$ and $\forall(Y = x_i) \in Inst_Y/\Gamma(Y), B'(Y = x_i) = 0$
2. if $Y \in I_{Var}$ and $par(Y) \subseteq \Gamma_{Var}\ and\ Y \notin \Gamma_{Var}$ then
   $\forall(Y = y) \in Inst_Y, B'(Y = y) = P(Y = y| \{X_i = x_i|X_i \in par(Y), X_i = x_i \in \Gamma\})$
3. otherwise,
   $\forall(Y = y) \in Inst_Y, B'(Y = y) = B(Y = y)$

According to the previous formal characterization of the function bu^*, the $BDIS$ agent only reconsiders the probability distributions over intended random variable $Y \in I_{Var}$. In fact, we suppose that the $BDIS$ agent only reconsiders those beliefs which are directly related with his intentions, since he allocates his attention on the current task he is trying to solve. More precisely: if Y is both an intended random variable in I_{Var} and a perceived variable in Γ_{Var}, then the updated probability distribution over Y assigns probability 1 to the perceived instantiation $\Gamma(Y)$ of variable Y and probability 0 to all the other instantiations of variable Y (condition 1); if Y is an intended random variable in I_{Var}, it is not a perceived variable in Γ_{Var} but its parents in the bayesian network are perceived variables in Γ_{Var}, then the updated probability distribution over Y assigns to each instantiations $Y = y$ of variable Y a probability which is equal to the conditional probability that $Y = y$ is true given that the perceived instantiations of the parents of Y are true (i.e. $P(Y = y| \{X_i = x_i|X_i \in par(Y), X_i = x_i \in \Gamma\})$) (condition 2). In all other cases the probability distribution over Y is not updated.

Space restrictions prevent a formal description of the belief update function bu of the standard BDI agent. Let us only say that function bu (differently from the function bu^* of the $BDIS$ agent) updates indiscriminately all beliefs of the agent, that is, at each round the standard BDI agent reconsiders the probability distributions over all random variables $Y \in$ **VAR** (even those variables which are not intended).[8]

5 Experimental Results and Experimental Setting

In order to compare traditional and surprise driven strategies for belief update, we run the standard BDI agent and the $BDIS$ agent in simulative experiments in the foraging scenario. Each reported experiment consists of 10 runs using different randomly generated initial conditions in a discrete world. Season length s_t is set to 15 rounds. Random initial placements of agents and entities (fruits, trees) are used for all experiments. The

[8] Function bu has the same three conditions of function bu^* specified above. The only difference is that in the three conditions of bu the requirement $Y \in I_{Var}$ is not specified.

threshold Δ of belief update in the $BDIS$ agent is set to 1. Thus, the $BDIS$ agent revises his beliefs only if a tree or a fruit of a certain color is perceived in a completely unexpected position in the grid.

Given that environmental dynamics are independent from the agent activities, we expect that to higher dynamism correspond higher costs of belief change (and, on the contrary, to lower dynamism correspond lower costs of belief change). More than absolute performance relying on the agent score (i.e. number of eaten fruits), we are interested in monitoring the ratio between belief update costs and the absolute performance in terms of eaten fruits. As in [14,15], in our experiments we evaluate the computational efforts for epistemic activities. For each trial we define the *belief change cost* of an agent as the total amount of belief change operations performed by the agent (i.e. the total number of modifications of the belief base of the agent during the all trial). Obviously, if the input belief set and the output belief set of the belief update function bu (viz. bu^*) are the same, that is $bu(\Gamma, B) = B$ (viz. $bu^*(\Gamma, B, I) = B$), then this does not count as a belief change operation. We define the *cost ratio* c of an agent in terms of *belief change cost* divided by the total amount of achieved task (number of eaten fruits). Namely, c represents the unit of cost spent for each achieved goal.

Because of the distributions, the *cost ratio* of an agent presents a fluctuating course before converging, hence each individual trial has to be sufficiently long for the effectiveness to become stable. In order to measure the effectiveness in function of time, we define a standard trial length of 600 rounds. We define the characterization of an agent by averaging his cost ratio progresses for 10 trials. Experiments are conducted in environments with three different levels of dynamism.

Static World: Fig. 2a shows the cost ratios of the two typologies of agents in a static environment ($\delta = 3$, a tree changes its location every 3 seasons, 45 rounds). The standard BDI agent attains an average of 25.9 eaten fruits on each trial, while $BDIS$ achieves an average performance of 21.7. Both agents show a comparable progress in terms of cost ratio. On the long term they stabilize their knowledge through a low frequency of belief change activities. Considering the low dynamism, once agents have overcome their transitory progress the result of effectiveness converges towards a value c = 0.2.

Medium World: Fig. 2b shows the cost ratios of the two typologies of agents in an environment with medium dynamism ($\delta = 2$, a tree changes its location every 2 seasons, 30 rounds). In terms of eaten fruits the BDI agent attains better performances (31.9) than the $BDIS$ (25.7). On the long term, cost performance converges to a value lower than 0.3 for both agents, even if the BDI wastes more resources for belief change. Despite of a lower number of eaten fruits, the $BDIS$ agent is able to maintain a better cost ratio along the experiments.

Dynamic World: Fig. 2c shows the cost ratios of the two typologies of agents in a highly dynamic environment ($\delta = 1$, a tree changes its location at each season change, 15 rounds). Due to his epistemic activity, the BDI agent is able to maintain a more consistent and complete knowledge of the environment. In so doing, he strongly overcomes the $BDIS$ agent in terms of achieved goals (25.9 average number of eaten fruits against 21.7) but, accordingly, he faces with higher epistemic costs, even beyond the transitory phase. The cost ratios of the two agents highlight a difference in performance:

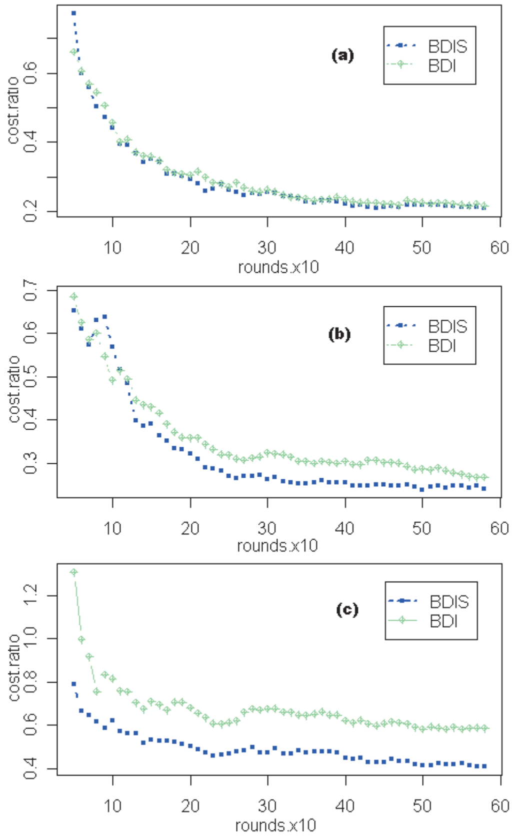

Fig. 2. Cost performance measured in static (a), medium (b) and dynamic (c) environments

the $BDIS$ cost ratio converges to a value of about 0.4 which is two orders of magnitude lower than the BDI cost ratio (0.6). The results of the experiments show that the more an agent spends his resources for belief change, the more his beliefs will be correct thereby enabling the agent to eat more fruits. On the other side, the results of the experiments show that in a very dynamic environment, the higher costs sustained for belief update are not compensated by an enhancement of the performance (i.e. number of eaten fruits).

6 Discussion and Future Works

The mechanism of surprise-based belief update modeled in this paper enables agents to process perceived data according to their ongoing intentions. Hence, agents acquire the capability to divide the overall set of perceived data in a *relevant* subset and a *irrelevant* one. The possibility to build agents which can filter out all irrelevant information they get will be a critical issue for forthcoming cognitive systems (e.g. agents engaged in a information retrieval task in the context of open system applications). We are actually working on a generalization of the model by introducing a more sophisticated belief and expectation processing. As in [16], our aim is to have *uncertainty* in deliberation by using prediction models (i.e. forward models) and introducing a quantitative dimension of goal importance (i.e. utilities of the expected outcome). Besides, we think that the model presented in this paper provides a novel understanding of the issue of intention reconsideration [14]. Since the persistence of an intention over time depends on the persistence of those beliefs which support this intention (i.e. beliefs are reasons for intending [2]), a surprised-based filter of belief update should affect persistence of intentions in an indirect way, that is, an agent should revise his intentions only if he is surprised by some perceived facts (since only in condition of surprise the agent's beliefs change). We would like to explore such an intriguing issue in a future work.

References

1. Cherniak, C.: Minimal rationality. MIT Press, Cambridge (1986)
2. Bratman, M.: Intentions, plans, and practical reason. Harvard University Press, Cambridge (1987)
3. Meyer, W.U., Reisenzein, R., Schützwohl, A.: Towards a process analysis of emotions: The case of surprise. Motivation and Emotion 21, 251–274 (1997)
4. Lorini, E., Castelfranchi, C.: The cognitive structure of surprise: looking for basic principles. Topoi: an International Review of Philosophy 26(1) (2007)
5. Itti, L., Baldi, P.: Bayesian surprise attracts human attention. In: Advances in Neural Information Processing Systems, pp. 1–8. MIT Press, Cambridge (2006)
6. Ortony, A., Norman, D.A., Revelle, W.: Effective functioning: A three level model of affect, motivation, cognition, and behavior. In: Fellous, J.M., Arbib, A. (eds.) Who Needs Emotions? The Brain Meets the Machine, Oxford University Press, New York (2005)
7. Wooldridge, M.: An Introduction to Multiagent Systems. John Wiley & Sons, Chichester (2002)
8. Rao, A.S., Georgeff, M.: An abstract architecture for rational agents. In: Proc. of the 3th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'92) (1992)
9. Pereira, D., Oliveira, E., Moreira, N., Sarmento, L.: Towards an architecture for emotional BDI agents. In: Bento, C., Cardoso, A., Dias, G. (eds.) EPIA 2005. LNCS (LNAI), vol. 3808, Springer, Heidelberg (2005)
10. Alchourron, C., Gardenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. The Journal of Symbolic Logic 50, 510–530 (1985)
11. Casati, R., Pasquinelli, E.: How can you be surprised? the case for volatile expectations. Phenomenology and the Cognitive Sciences 6(1-2), 171–183 (2006)
12. Kaelbling, L.P., Rosenschein, S.J.: Action and planning in embedded agents. In: Maes, P. (ed.) Designing autonomous agents, pp. 35–48. MIT Press, Cambridge (1990)
13. Ortony, A., Partridge, D.: Surprisingness and expectation failure: Whats the difference? In: Proc. of the 10th Int. Joint Conf. on Artificial Intelligence (IJCAI) (1987)
14. Schut, M.C., Wooldridge, M., Parsons, S.D.: The theory and practice of intention reconsideration. Journal of Experimental and Theoretical Artificial Intelligence 16(4), 261–293 (2004)
15. Kinny, D., Georgeff, M., J., H.: Experiments in optimal sensing for situated agents. In: Proc. of the 2nd Pacfic Rim. Int. Conf. on AI (PRICAI-92) (1992)
16. Piunti, M., Castelfranchi, C., Falcone, R.: Surprise as shortcut for Anticipation: clustering Mental States in reasoning. In: Proc. of IJCAI-07 (2007)