Why Bad Coffee? Explaining Agent Plans with Valuings

Michael Winikoff[1](✉), Virginia Dignum[2], and Frank Dignum[3]

[1] University of Otago, Dunedin, New Zealand
michael.winikoff@otago.ac.nz
[2] Delft University of Technology, Delft, The Netherlands
M.V.Dignum@tudelft.nl
[3] Utrecht University, Utrecht, The Netherlands
F.P.M.Dignum@uu.nl

Abstract. An important issue in deploying an autonomous system is how to enable human users and stakeholders to develop an appropriate level of trust in the system. It has been argued that a crucial mechanism to enable appropriate trust is the ability of a system to explain its behaviour. Obviously, such explanations need to be comprehensible to humans. We argue that it makes sense to build on the results of extensive research in social sciences that explores how humans explain their behaviour. Using similar concepts for explanation is argued to help with comprehensibility, since the concepts are familiar. Following work in the social sciences, we propose the use of a folk-psychological model that utilises beliefs, desires, and "valuings". We propose a formal framework for constructing explanations of the behaviour of an autonomous system, present an (implemented) algorithm for giving explanations, and present evaluation results.

1 Introduction

This paper addresses the problem of how an autonomous system can explain itself by developing a computational mechanism that provides explanations for why a particular action was performed. It has been argued [6,8,19] that in a range of domains, a key factor in humans being willing to trust autonomous systems is that the systems need to be able to *explain* why they performed a certain course of action. Note that this is not the same as explaining system recommendations, since we are explaining a *course of action* (taken over time, in an environment), not a (static) recommendation.

Explanation is relevant to AI safety for a number of reasons. Firstly, explanation can reduce the opaqueness of a system, and support understanding its behaviour, and its limitations. Secondly, in situations where things do go wrong, a post-mortem analysis, using some sort of "black box" (as those used in airplanes) can use explanation techniques to help investigators understand what went wrong.

B. Gallina et al. (Eds.): SAFECOMP 2018 Workshops, LNCS 11094, pp. 521–534, 2018.
https://doi.org/10.1007/978-3-319-99229-7_47

In developing such an explanation mechanism, it is important to be mindful that the explanations have to be comprehensible, and useful, to a human, and therefore we should consider relevant social sciences literature [12]. According to Miller [12] explanations should be *contrastive* i.e. answer questions of the form "why did you do X ...instead of Y?"; *selected*, i.e. select relevant factors and present those; and, *social*, i.e. presented relative to what the explainer believes the listener (i.e. explainee) knows. That is, explanations, being in fact conversations, should follow Grice's maxims of quality, quantity, manner and relevance [7].

In our work we consider in particular the work of Malle [11], which argues that humans use folk psychological constructs (e.g. beliefs, desires) to explain behaviour. This leads us to adopt a model that includes desires and beliefs, specifically the well-known BDI (Beliefs, Desires, Intentions) model [1,2,13]. We contend that providing explanations in terms of the same concepts used in human-to-human explanations will help enable explanations to be comprehensible.

Malle identifies three types of reasons in explaining behaviour: desires, beliefs, and what he terms *valuings*, defined as things that "*directly indicate the positive or negative affect toward the action or its outcome*". We therefore extend the BDI model with valuings, following recent work by Cranefield *et al.* [5].

2 Formal Setting

In this paper, we assume a BDI model based on goal trees and we also assume that the listener assumes such a goal tree as the deliberation mechanism of the agent[1].

A *goal tree* is a tuple (N, G) of a name N, and either an action[2] (A), or a combination of sub-goals (N_i, G_i), which can be in sequence (Seq), unspecified order (And), or a choice (Or) where each option $O_i = (C_i, (N_i, G_i))$ has a sub-goal and a condition C_i indicating in which situations that sub-goal can be selected to realise the parent goal. Each action A has an associated pre-condition (denoted $\mathit{pre}(A)$) and post-condition ($\mathit{post}(A)$), both of which are viewed as sets of propositions. We define $\mathcal{B}(N)$ to be the beliefs held just prior to executing the goal N. We write (G_{1-n}) (resp. (O_{1-n})) to abbreviate $((N_1, G_1), \ldots, (N_n, G_n))$ (resp. $(O_1, \ldots, O_n)$). We also sometimes abbreviate (N, G) to G_N for readability, and, where the name is not important, just write G for G_N. Formally:

$$G ::= A \mid \textsc{Seq}(G_{1-n}) \mid \textsc{And}(G_{1-n}) \mid \textsc{Or}(O_{1-n})$$

Figure 1 shows a running example, along with a goal tree for this example, including the pre- and post-conditions (the V_i are valuings, explained below).

[1] Note that using a BDI model does not necessarily require the system to be designed or implemented as BDI agents. It is in principle possible to use a BDI model to provide explanations of a system's behaviour even if the system does not use BDI concepts.

[2] For actions we assume that the name of the goal tree node and the name of the action coincide, i.e. that $A = N$.

Jo is an academic visiting colleagues at another University. Like many academics, he requires coffee. There are a number of possible sources of coffee: The little kitchen near Ann's office has coffee-like-substance freely available, but this machine requires a staff card to operate. Ann has in her office a coffee machine which converts pods into nice coffee. There is also a coffee shop a few buildings away, where good coffee can be obtained, at a (financial) cost. Jo prefers coffee to coffee-like substances, which is the over-riding preference. Less-important preferences are to save money, and to use the nearest coffee source. Therefore the three relevant quality attributes are (in order): quality (coffee preferred to coffee-like), money (free preferred to expensive), and location (smallest distance from starting location).

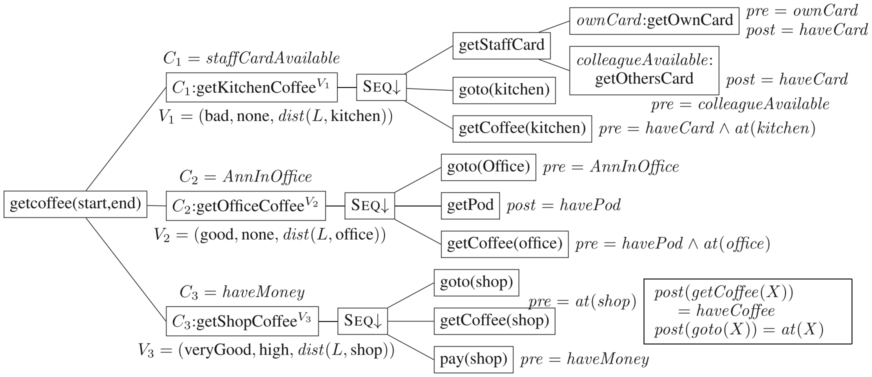

getcoffee(start,end)
$C_1 = staffCardAvailable$
C_1:getKitchenCoffee^{V_1}
SEQ↓
V_1 = (bad, none, $dist(L$, kitchen))
getStaffCard
$ownCard$:getOwnCard
$pre = ownCard$
$post = haveCard$
$colleagueAvailable$: getOthersCard
$post = haveCard$
$pre = colleagueAvailable$
goto(kitchen)
getCoffee(kitchen)
$pre = haveCard \wedge at(kitchen)$
$C_2 = AnnInOffice$
C_2:getOfficeCoffee^{V_2}
SEQ↓
V_2 = (good, none, $dist(L$, office))
goto(Office)
$pre = AnnInOffice$
getPod
$post = havePod$
getCoffee(office)
$pre = havePod \wedge at(office)$
$C_3 = haveMoney$
C_3:getShopCoffee^{V_3}
SEQ↓
V_3 = (veryGood, high, $dist(L$, shop))
goto(shop)
getCoffee(shop)
$pre = at(shop)$
pay(shop)
$pre = haveMoney$
$post(getCoffee(X)) = haveCoffee$
$post(goto(X)) = at(X)$

Fig. 1. Running Example (goals are OR-decomposed unless indicated by SEQ↓)

Intuitively, a goal tree is executed as follows. If the tree is simply an action, then the action is performed (assuming its preconditions hold). If the tree is an AND or SEQ decomposition, then all of the sub-goals are executed, either in the specified sequential order (SEQ), or in some, unspecified, order. Finally, if the tree is an OR decomposition, then an applicable option (i.e. one whose condition C_i is believed to hold in the current situation) is selected and executed. Many BDI platforms provide a way to handle failure, which we discuss later. Formally, the semantics of a goal tree is obtained by mapping it to a set of possible sequences of actions.

The semantics of valuings is based on the theory of values as put forward by Schwartz [14]. In Weide [18] it is shown how these abstract values can be connected to concrete aspects of action decision. Following Cranefield *et al.* [5] we incorporate them by annotating nodes in the goal tree with an abstract evaluation of key aspects of their effects. By "key aspects" we mean those that are relevant to the agent evaluating which options it prefers, that is, its valuings. For instance, in the running example, each annotation V_i is of the form (coffee quality, cost, distance), respectively drawn from {veryGood, good, bad}, {none, low, high}, and {none, low, medium, high} where $dist(A, B)$ denotes the distance between A and B computed as follows: the office and kitchen are close to each other ("low" distance), and the shop is far from both kitchen and office ("high" distance). We write $\{V_1, \ldots, V_n\} \prec V$ to denote that the value annotation V is preferred to each of the V_i, and we define $\{\} \prec V_i$ to be equivalent to $\top$.

The agent's valuings, i.e. which options it appreciates more or less, are specific to a given situation. They are founded on the agent's values, which are the underlying drivers. For example, an agent might value good coffee, saving money, and saving time. These aspects are the measurable criteria indicating whether a certain value is promoted by a course of action. However, since we have multiple aspects (thus creating a kind of multi-criteria optimization), they do not completely determine the agent's valuing. E.g. an agent might prefer good coffee over bad coffee, but decide to get bad coffee for free at the end of the month when his salary runs out and get good coffee once his salary is in. So, the weighing of the different aspects and thus the resulting valuings is not fixed, but depends on the context. Thus, in general a valuing (or preference) for an option is based on the values, but also on the current situation and practical considerations.

In Cranefield *et al.* [5] it is shown how these valuings can be kept consistent and work for large goal-plan trees. Here, we therefore assume the valuings to be present and indicating consistent preferences over alternatives.

3 Generating Explanations

As discussed in the introduction, an explanation is given in terms of reasons which can be desires (goals), beliefs, or valuings. More precisely, an explanation is either $\bot$ (representing that the question does not make sense, e.g. "why did you do X?" when X was not done), or a set of reasons. Reasons can be beliefs that were held, desires that were pursued, and valuings. Valuings are explained as "I preferred V to $\{V_1, \ldots, V_n\}$". We also have forward-looking reasons of the form "I did N_1 in order to be able to later do N_2" ($N_1 \mapsto N_2$). Finally, as discussed towards the end of this section, one possible type of reason is an indication that a particular option was attempted but failed. For example, "I chose to get coffee from the kitchen because I tried to buy it from the shop but failed" (e.g. shop was closed). Finally, we also define $\top$ to be an explanation that carries no information. Clearly, $\top$ is not a useful explanation to a user, but it is used in the formal definitions below where some parts of the process do not provide any useful information.

The definition of the explanation function E is with respect to the goal-tree. Specifically, $E_N^T(G_{N'})$ is "explain N using the tree (N', G) and trace T". We define $n(G)$ as denoting the set of all node names occurring in the tree rooted at G. We define $T^{\prec N}$ to be the part of the trace T that occurs before N. Note that if $N \notin T$ then we simply define $E_N^T(G) = \bot$, otherwise the rest of the definitions below apply.

$$\begin{aligned}
E_N^T(G_{N'}) &= \bot, \text{if } N \notin T \\
E_N^T(A_{N'}) &= \begin{cases} \{\} & \text{if } \mathit{pre}(A) = \top \\ \{\mathit{pre}(A)\} & \text{otherwise} \end{cases} \\
E_N^T(\textsc{And}(G_{1-n})_{N'}) &= \Theta \\
E_N^T(\textsc{Seq}(G_{1-n})_{N'}) &= \Theta \\
E_N^T(\textsc{Or}(O_{1-n})_{N'}) &= \begin{cases} \mathit{pref}(O_i, \{O_1, \ldots, O_n\}) \cup \Theta, & \text{if } N \in n(G_i) \\ \Theta, & \text{otherwise} \end{cases} \\
\text{where } \Theta &= \bigcup_{G_i : n(G_i) \cap T^{\prec N} \neq \emptyset} E_N^T((N_i, G_i))
\end{aligned}$$

The function E collects reasons by traversing the relevant parts of the goal tree. A part of the goal tree is relevant if it occurs in the execution trace before beginning the process of executing the node N that is being explained. Simply, if something occurs before N, then it can affect N. This relevance condition is checked in the definition of Θ: $G_i : n(G_i) \cap T^{\prec N} \neq \emptyset$ finds all sub-goals G_i of the current node which contain beneath them at least some node that appears in the prefix of the trace T before N (viewing the trace prefix as a set).

In the case of an action A the explanation collected is the action's precondition as this affects the execution of the action, and consequently, of whatever comes after it. In the case for Seq and And the explanation collected is simply the explanation associated with the sub-goals. In the case for Or there is an additional explanation relating to why the particular option taken was chosen. This is defined by the function *pref* which provides an explanation for why the selected option, G_i, is preferred to the other options. The definition of *pref* is complex. Intuitively, given a choice-point $(N, \textsc{Or}(O_1, \ldots, O_n))$, where G_i was selected, the explanation consists of three parts:

1. the condition of the selected sub-goal being true ("C_i");
2. for each condition C_j $(j \neq i)$ that is false at the decision point, the explanation includes that the condition was false: $\bigcup_{C_j : \mathcal{B}(N) \not\models C_j} \neg C_j$; and
3. for each condition C_j $(j \neq i)$ that is true at the decision point, but that was not selected, the annotations of those sub-goals, and an indication that the selected sub-goal was preferred to these other available sub-goals in the current situation: $\{V_j \mid j \neq i \wedge \mathcal{B}(N) \models C_j\} \prec V_i$.

Formally $\mathit{pref}(O_i, \{O_1, \ldots, O_n\})$ is defined to be: $\{C_i\} \cup \left(\bigcup_{C_j : \mathcal{B}(N) \not\models C_j} \{\neg C_j\}\right) \cup \{\{V_j \mid j \neq i \wedge \mathcal{B}(N) \models C_j\} \prec V_i\}$.

Consider as an example the situation in which C_2 is false, and the other C_i are true. Then the preference explanation for why C_3 was chosen is[3]: $\{C_3, \neg C_2, \{V_1\} \prec V_3\}$. Rendered in English (which can be done by applying a simple pattern[4]) this reads: "I chose to get coffee from the shop because I had money, and Ann was not in her office, and I prefer V_3 to V_1 in this situation".

[3] All explanations given in this section were produced by the implementation.

[4] This has subsequently been implemented.

On the other hand, in a situation where all C_i are true and C_3 is selected, the explanation would take the form: $\{C_3, \{V_1, V_2\} \prec V_3\}$. In English: "I chose to get coffee from the shop because I had money, and I prefer V_3 to both V_1 and V_2 in this situation".

Note that these explanations just present the set of annotations, indicating an overall preference between them. However, we could provide more precise explanations by taking into account the known priorities of factors, e.g. that coffee quality is the overriding factor, followed by money, then distance. So, for example, for the first example above, we could explain more precisely that the reason why V_3 was preferred to V_1 is that it yields better quality coffee. Similarly, for the second example, we could explain that V_3 was preferred to both V_1 and V_2 because the coffee quality was better (despite V_2 being good coffee and cheaper than V_3).

On the other hand, suppose that the office coffee was selected, even though all three C_i were true. In order to explain why $\{V_1, V_3\} \prec V_2$ we would need to explain that V_2 was preferred to V_1 because it had better coffee, and, perhaps, that it was preferred to V_3 because cost was a factor.

3.1 Adding Preparatory Actions

We now extend the definition to also include preparatory actions. For example, an explanation for "why did you go to the kitchen?" could also be "because I need to be in the kitchen in order to get coffee". This is where an action's post condition is (part of) the precondition of a future action. Specifically, a preparatory reason applies to explain an action A when (i) the post-condition of A is required in order for the pre-condition of another action A' to hold, and (ii) A' occurs after A. We assume that $\mathit{before}(A, B)$ formalises that it is possible for B to occur after A in a trace, but not for A to occur after B.

Turning to the first condition, an obvious formalisation is simply $\mathit{post}(A) \rightarrow \mathit{pre}(A')$. But A's post condition may be only *part* of the pre-condition. For example, the action getPod only achieves havePod, so $\mathit{post}(\text{getPod}) \not\rightarrow \mathit{pre}(\text{getCoffee(office)})$. We therefore formalise "required" as "without it, things don't work", i.e. if A's post-condition fails to hold, then the pre-condition of A' also must fail to hold: $(\neg \mathit{post}(A)) \rightarrow (\neg \mathit{pre}(A'))$. This assumes that $\mathit{post}(A) \neq \top$. In our setting, where pre and post conditions are conjunctions of positive atoms, this is equivalent (viewing the conjunctions as sets) to $\mathit{post}(A) \neq \emptyset \wedge \mathit{post}(A) \subseteq \mathit{pre}(A')$.

We then extend the explanation with preparatory action explanations: when explaining an action A given goal tree G and trace T, we add to $E_A^T(G)$ the set of links $A \mapsto A'$ where $A' \in n(G) \wedge \mathit{before}(A, A') \wedge \mathit{post}(A) \neq \emptyset \wedge \mathit{post}(A) \subseteq \mathit{pre}(A')$. So, for example, an alternative explanation for why the agent performed the action getPod is that it was required for the subsequent getCoffee(office) action. Finally, in order to consider preparatory actions between *goals*, we follow previous work on summary information [15–17], and extend pre and post conditions to intermediate goals, inferring them.

3.2 Adding Motivations

Finally, in addition to the explanation function E, which yields beliefs and valuings, and the link function, we also add explanations in terms of parent goals: these are desires that explain why the current course of action is being pursued.

This reason is simple: we also include in the explanation all the ancestors of the node being explained. However, we do not include ancestors that are OR refined, since these are not helpful. In explaining why a particular option was done, for instance why getOwnCard was done, it is not helpful to refer to the parent, getStaffCard, because the parent is less specific.

Pulling all the pieces together, the overall explanation function is then:

$$\begin{aligned}\mathcal{E}_N^T(G_{N'}) = E_N^T(G_{N'}) \;\cup\; & \\ & \{N \mapsto N' \mid N' \in n(G) \wedge \mathit{link}(N, N')\} \;\cup \\ & \{\mathsf{Desire}(N') \mid \mathit{ancestor}(N', N) \wedge \neg \mathit{isOR}(N')\}\end{aligned}$$

For example, given the scenario described, in a situation where C_1 and C_3 hold, but not C_2, the possible reasons that could be used to explain why the agent did "$\mathit{goto}(\mathit{shop})$" are: {haveMoney, ¬AnnInOffice, $\{\langle$bad, none, high$\rangle\} \prec$ $\langle$veryGood,high, none$\rangle$, goto(shop) $\mapsto$ getCoffee(shop), Desire: getShopCoffee}. In English, these are: I had money, Ann was not in her office, I preferred V_3 to V_1 (perhaps because it yields better quality coffee), I needed to go to the shop in order to do getCoffee(shop), and I desired to getShopCoffee.

3.3 Adding Failure Handling

We now extend the explanation mechanism to handle failure handling. Informally, actions can fail, and the failure of a node is handled by considering its parent. If the parent is a Seq or a And then it too is considered to be failed, and failure handling moves to consider that node's parent. When an Or node is reached, failure is handled by trying an alternative plan (if one exists, otherwise the Or node is deemed to have failed). We assume that we know which actions in the trace are failed (denoted $\mathit{failed}_A(A, T)$). Then the condition under which a non-leaf node is considered to be failed can be easily derived from the tree, and is denoted $\mathit{failed}(G)$.

Extending the explanation to account for the possibility of previous failures is done by defining an extended *pref* function. Note that the definition of the explanation function E is unchanged, except that in the definition of the recursive call Θ we exclude failed nodes.

Recall that the definition of *pref* has three components: the condition of the selected sub-goal being true, the conditions of those (other) sub-goals that are false, and, for those other sub-goals that have true conditions, a preference indication. We modify the second and third components by only considering those sub-goals that have not yet been attempted. We then add a fourth component that explains those things that have been previously attempted. Intuitively, this is of the form "...and I already unsuccessfully tried doing X". Formally:

$\{\mathsf{Tried}(G_j) \mid j \neq i \wedge \mathit{failed}(G_j)\}$ where i is the index of the sub-goal that was selected.

To illustrate this definition, consider a situation where Jo has decided to getOfficeCoffee, but by the time he reaches Ann's office, Ann has had to leave for a meeting. The plan therefore fails, and Jo then recovers by electing to go to the shop. In response to the query "why did you getShopCoffee?" the explanation given is "$\{$haveMoney, $\{\langle$bad, none, low$\rangle\} \prec \langle$veryGood, high, high$\rangle$, Tried:getOfficeCoffee$\}$" which can be rendered in English as "because I have money, I prefer good coffee to bad coffee, and because I tried (and failed) to get pod coffee".

4 Evaluation

There are two broad questions that concern evaluation of this work. The first is whether the explanations provided are comprehensible and *useful* to a human user. The second is whether the approach is sufficiently *efficient*.

In order to assess the comprehensibility and *usability* of the explanations generated, as well as provide guidance to future work on selecting explanations, we conducted a preliminary human participant evaluation. Note that we focussed on evaluating E_N^T, and did not include in the explanations either preparatory actions (links) or parent goals (except for the fifth explanation - see below).

Participants were recruited on Mechanical Turk and paid US$0.50 for an estimated 5 min survey. Each participant was provided with a brief description of the coffee scenario and an indication of what behaviour was observed. Participants were divided into three cohorts, each of which was given a different observed behaviour. The allocation to cohorts was random. We obtained 109 responses, comprising 42 in Cohort 1, 37 in Cohort 2, and 30 in Cohort 3. Participants were 28 females and 81 males. Their highest level of education was high school (22), bachelors (63), and master/graduate degree (21). One person had not completed high school and two respondents had PhDs. Finally, around 35% had some experience with programming (38 out of 109).

Each cohort was given five possible explanations for the observed behaviour. The explanations were created manually, following the corresponding explanation method. The first explanation combined valuings and beliefs, and corresponds to the E_N^T function defined earlier (indicated with "V+B" below). The second and third explanations are solely in terms of valuings: one is abstract (AV), just saying "*This is the best possible coffee available*", and the second is concrete (V), with a specific explanation (see below). The fourth candidate explanation provides only relevant beliefs (B). The fifth candidate explanation gives the goal, and the beliefs that enabled the specific behaviour that was selected, which is the explanation mechanism proposed by Harbers [9] (G+B).

For example, in the case where the colleague's machine was selected (Cohort 1), the five explanations given are:

(E1) "This is the best possible coffee available; I had no money." (V+B)
(E2) "This is the best possible coffee available." (AV)

(E3) "This coffee is better than the kitchen and cheaper than in the shop." (V)
(E4) "I've no money; Ann was in her room." (B)
(E5) "I wanted coffee; Ann was in her room." (G+B)

	Cohort 1			Cohort 2			Cohort 3		
	Believ.	Accept.	Comprehen.	Believ.	Accept.	Comprehen.	Believ.	Accept.	Compr.
E1	3.929 3	4.071 2	3.976 2	3.649 3	3.811 2	4.027 3	3.933 1	4.200 1	3.867 2
E2	3.095 5	3.286 5	3.714 4	3.892 1	3.892 1	4.054 2	2.500 5	2.767 5	3.200 5
E3	4.190 1	4.238 1	4.286 1	3.865 2	3.811 2	4.243 1	3.933 1	4.167 2	4.167 1
E4	3.857 4	3.500 4	3.690 5	2.973 5	3.000 5	3.108 5	3.567 4	3.600 4	3.567 3
E5	3.976 2	3.857 3	3.976 2	3.541 4	3.595 4	3.568 4	3.600 3	3.733 3	3.533 4
$p =$	0.00026	0.000013	0.038	0.0013	0.0047	0.00013	.000029	0.000025	0.034

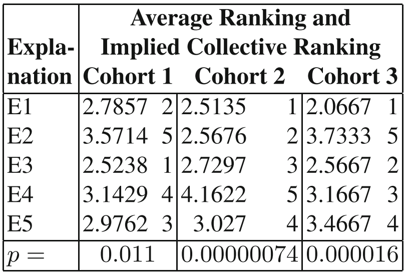
Fig. 2. Believability, acceptability, and comprehensibility scores (1 = very bad, 5 = excellent) for the three Cohorts and five explanations.

For each possible explanation, the participants were asked to *score* the explanation in terms of three criteria: *believability* ("I can imagine someone giving this answer"), *acceptability* ("This is a valid explanation of Jo's choice"), and *comprehension* ("I understand the text of this explanation"). Each score was on a five-point Likert scale from"very bad" (1) to "excellent" (5). Participants were also asked to *rank* the five candidate explanations by order of preference, from most preferred (rank 1) to least preferred (rank 5). Finally, participants were also asked whether they felt that further explanation was required, and, if so, what form it should take (e.g. providing source code, entering a dialogue with the system).

Figure 2 shows for each cohort and each explanation the average score for each of the three criteria. The figure also shows the implied ranking. For example, for Cohort 1 and Believability, the third explanation (E3) had the best (highest) average score, and therefore collectively E3 is ranked best for Believability by this cohort. For each of the three criteria and three cohorts a statistical test[5] confirms there is a difference ($p < 0.05$) amongst the explanations for that cohort, and post-hoc tests with Holm adjustment find that some of the pairwise differences are significant.

We now turn to analysing the responses on the ranking question, in which participants ranked the explanations from most preferred (1) to least preferred (5). Figure 3 shows for each explanation (E1 to E5) and for each cohort the *average ranking*, which is the average of each explanation's ranking for that cohort. So, for example, if half of the participants in a given cohort were to rank E1 as their most preferred (1), and the other half were to rank it as their second-most preferred (2), then it would have an average ranking of 1.5 for that cohort. The table also shows for each explanation and cohort the preferred order of explanations that is implied by the average ranking (i.e. the implied

[5] Kruskal-Wallis, since data is not expected to be normally distributed.



Expla-	Average Ranking and					
	Implied Collective Ranking					
nation	Cohort 1		Cohort 2		Cohort 3	
E1	2.7857	2	2.5135	1	2.0667	1
E2	3.5714	5	2.5676	2	3.7333	5
E3	2.5238	1	2.7297	3	2.5667	2
E4	3.1429	4	4.1622	5	3.1667	3
E5	2.9762	3	3.027	4	3.4667	4
$p =$	0.011		0.00000074		0.000016	

Fig. 3. Rankings for the three Cohorts and five explanations.

collective ranking). For example, for cohort 1, explanation 3 had the best (lowest) average ranking, and is therefore the most preferred explanation. A statistical test confirms that there are differences between the explanations' scores for each of the cohorts (as before, all p values are <0.05). Post-hoc tests (Mann-Whitney, with Holm correction), find that the ranking differences are significant between E1-E2, E2-E3 (Cohort 1), E4 and all other explanations (Cohort 2), and between E1-E2, E1-E4, E1-E5, E2-E3, E3-E5 (Cohort 3).

Considering the question of whether the explanation given would be adequate, or whether additional information would be desired, 69% of Cohort 1 indicated that no further explanation would be required (with the remaining responses asking for a dialogue (19%) or source code (12%)). For Cohort 2 these figures were respectively 54% (no further explanation), 22% (dialogue), 19% (source code), and for Cohort 3 they were 63%, 20% and 17%.

Overall, explanations 1 and 3 were considered as being better than the other explanations, and that, except for Cohort 2, explanation 2 was seen as being the worst. Since explanations 1 and 3 both include valuings, this finding supports the key thesis of this paper, that valuings are important to provide useful explanations. Furthermore, for Cohorts 2 and 3, E1 was preferred to E3, indicating that valuings alone were not sufficient.

We now turn to *efficiency*. We observe that the explanation has three components: the reasons calculated by the function $E_N^T(G)$, the links between nodes, and parent goals. The last is simple to compute, involving merely traversing the tree upwards from the node being queried (i.e. $O(\log N)$ where N is the number of nodes in the goal tree). The second, the links, only depend on the static structure of the tree (i.e. which nodes precede other nodes), and on the pre and post conditions, and therefore can be computed ahead of time. This does assume that pre and post conditions are specified ahead of runtime. If this is not the case, then a runtime calculation is required, which involves checking pre and post conditions for every pair of nodes that precede each other. Given a tree with N nodes, there are obviously at most $O(N^2)$ such pairs, and the check is $O(1)$ (we assume that each node's pre and post conditions do not become longer as the tree grows).

Turning now to the explanation function E, we observe that the function basically traverses the tree from root to leaves. For each non-leaf node it checks which of the child nodes contain at least one node that is in the trace prefix ($n(G_i) \cap T^{\prec N} \neq \emptyset$). This check could be implemented by first traversing the tree upwards, tagging each node G_i with its $n(G_i)$, and then checking for intersection between $n(G_i)$ and $T^{\prec N}$. Since for each node the size of $n(G_i)$ is a function of the number of nodes beneath it, i.e. $O(N)$, computing the intersection (assuming indexing on $T^{\prec N}$) for a single node is $O(N)$, and for the whole tree it would be[6] $O(N^2)$. Finally, for each Or node, there is an additional calculation of *pref* which is proportional to the number of children and the size of conditions, both of which we assume is effectively a constant, i.e. does not grow with N. Therefore calculating E_N^T is $O(N^2)$.

In order to empirically assess the actual runtime required, and the algorithm's scalability, we have conducted an experimental evaluation on generated trees. The generated trees have the following structure: $T^0 = A$ and $T^{d+1} = \textsc{Or}_N(O_{1-j})$ where $O_i = (c, \textsc{Seq}_{N_i}(T^d_{1-k}))$. In other words, a generated tree of depth 0, denoted T^0, is just an action A (with a new unique name), and a generated tree of depth $d+1$ is a disjunction of j options, where each option O_i has the same fixed condition c, and a sequential composition of k trees of depth d. All nodes have unique names. Note that the number of nodes in a tree with branching factors j and k and depth d can be calculated as: $n(j, k, 0) = 1$ and $n(j, k, (d+1)) = 1 + j + (j \times k \times n(j, k, d))$.

For the efficiency evaluation various values of j, k and d were systematically generated, and the number of nodes in the tree and the time taken to compute E_N^T were recorded. The experiments were done using the GHC Haskell implementation (version 8.2.1) running on a 3.2 GHz Intel Core i5 iMac with 16 GB RAM running OSX 10.10.3.

These experiments show that for relatively small trees (fewer than 1000 nodes) the explanation generation, even with an unoptimised Haskell prototype, is clearly fast enough to be practical (<0.1 s). It is worth noting that real goal trees are not necessarily large. For instance, the (real-world) application described by Burmeister *et al.* [3] has 57 nodes in its goal tree. Finally, we note that the core of the Haskell implementation is a direct transliteration of the equations earlier in this paper. While this ensures that the implementation matches the paper, there are clear, and substantial, opportunities to improve efficiency.

5 Related Work

In this section we briefly highlight closely related work. Harbers [9], like us, assumes that a goal tree is given, and defines a number of templates that can be used to explain observed behaviour. It is worth noting that our approach strictly generalises Harbers' approach, in that we include links, ancestor goals,

[6] However, the prototype implementation does not tag nodes, so it recomputes $n(G_i)$, leading to higher computational complexity.

and relevant beliefs. In other words, every reason that is included in explanations generated using Harbers templates is included in $\mathcal{E}$. Finally, we note that whereas Harbers just outlines the rules as brief templates, we provide full formal definitions that have been implemented.

Another approach is that of explanation as model reconciliation, where the assumption is that in realistic scenarios humans have domain and task models that differ significantly from that used by the agent [4]. This assumption is supported by psychological studies [10]. However, this approach does not link to the beliefs, desires and values/valuings of the user and is therefore less adequate to connect to the reasons behind the decisions taken in the process. Moreover, it assumes that the human's mental model is *known*, a fairly strong assumption that we do not make.

6 Conclusion

We have argued that explaining the behaviour of autonomous software could be done using the same concepts as are used by humans when explaining their behaviour. Specifically, we have followed the findings of Malle and based explanations on beliefs, desires and *valuings* [11, Sect. 4.2.4]. This paper has proposed a formal framework, using BDI-style goal-trees, augmented with value annotations. This formal framework is then used to define an explanation function, which has been implemented. A human subject evaluation has highlighted that, as expected based on the literature, valuings are seen as being of value in explaining behaviour. Further empirical evaluation is needed, using more scenarios and including the other types of reasons, to assess not just the believability, acceptability and comprehensibility of explanations, but more broadly assessing their effect on trust in the autonomous system.

Stepping back to consider the bigger picture, we have provided a mechanism for *generating* reasons. However, this is only part of the solution to the problem of explaining behaviour. We know that humans select parts of the explanation [12]. The next step in this research is to define means for *selecting* parts of the possible explanation.

Finally, we contend that providing usable explanations of autonomous systems requires the use of human-oriented models, such as our extended BDI model. One area for future work is to develop ways of using our work for autonomous systems that are based on machine learning techniques. Such systems are known for their opacity. In our future work, we will research the possibilities of complementing such systems with the reasoning we propose in this paper. For instance, can an extended BDI model be developed in parallel and maintained to correspond to a behaviour that is learned? Can BDI models be derived automatically from learned behaviours?

References

1. Bratman, M.E., Israel, D.J., Pollack, M.E.: Plans and resource-bounded practical reasoning. Comput. Intell. **4**, 349–355 (1988)
2. Bratman, M.E.: Intentions, Plans, and Practical Reason. Harvard University Press, Cambridge (1987)
3. Burmeister, B., Arnold, M., Copaciu, F., Rimassa, G.: BDI-agents for agile goal-oriented business processes. In: Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS) [Industry Track], pp. 37–44. IFAAMAS (2008)
4. Chakraborti, T., Sreedharan, S., Zhang, Y., Kambhampati, S.: Plan explanations as model reconciliation: moving beyond explanation as soliloquy. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, pp. 156–163 (2017). https://doi.org/10.24963/ijcai.2017/23
5. Cranefield, S., Winikoff, M., Dignum, V., Dignum, F.: No pizza for you: value-based plan selection in BDI agents. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, pp. 178–184 (2017). https://doi.org/10.24963/ijcai.2017/26
6. EU: EU General Data Protection Regulation, April 2016. http://tinyurl.com/GDPREU2016 (see articles 13-15 and 22)
7. Grice, H.P.: Logic and conversation. In: Cole, P., Morgan, J. (eds.) Syntax and Semantics Volume 3: Speech Acts. Academic Press, New York (1975)
8. Gunning, D.: Explainable Artificial Intelligence (XAI) (2018). https://www.darpa.mil/program/explainable-artificial-intelligence
9. Harbers, M.: Explaining Agent Behavior in Virtual Training. SIKS dissertation series no. 2011-35, SIKS (Dutch Research School for Information and Knowledge Systems) (2011)
10. Lombrozo, T.: Explanation and abductive inference. In: Oxford Handbook of Thinking and Reasoning, pp. 260–276 (2012)
11. Malle, B.F.: How the Mind Explains Behavior: Folk Explanations, Meaning, and Social Interaction. The MIT Press, Cambridge (2004). ISBN 0-262-13445-4
12. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. CoRR abs/1706.07269 (2017). http://arxiv.org/abs/1706.07269
13. Rao, A.S., Georgeff, M.P.: An abstract architecture for rational agents. In: Rich, C., Swartout, W., Nebel, B. (eds.) Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning, pp. 439–449. Morgan Kaufmann Publishers, San Mateo (1992)
14. Schwartz, S.: An overview of the Schwartz theory of basic values. Online Read. Psychol. Cult. **2**(1) (2012). https://doi.org/10.9707/2307-0919.1116
15. Thangarajah, J., Padgham, L., Winikoff, M.: Detecting and avoiding interference between goals in intelligent agents. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI), pp. 721–726 (2003)
16. Thangarajah, J., Padgham, L., Winikoff, M.: Detecting and exploiting positive goal interaction in intelligent agents. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 401–408. ACM Press (2003)
17. Visser, S., Thangarajah, J., Harland, J., Dignum, F.: Preference-based reasoning in BDI agent systems. Auton. Agents Multi-Agent Syst. **30**(2), 291–330 (2016). https://doi.org/10.1007/s10458-015-9288-2

18. van der Weide, T.: Arguing to motivate decisions. Dissertation, Utrecht University Repository (2011). https://dspace.library.uu.nl/handle/1874/210788
19. Winikoff, M.: Towards Trusting Autonomous Systems. In: Fifth Workshop on Engineering Multi-Agent Systems (EMAS) (2017)