Why Bad Coffee? Explaining Agent Plans with Valuings

Michael Winikoff
University of Otago
New Zealand

Virginia Dignum
Delft University of Technology
The Netherlands

Frank Dignum
Utrecht University
The Netherlands

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 commonsense-psychology 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.

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 (EU 2016; Winikoff 2017; Gunning 2018) 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.

The problem this paper therefore addresses is how an autonomous system can provide an explanation for why it chose a particular course of action. Specifically, we develop a computational mechanism that provides explanations for why a particular action was performed.

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 (Miller 2017). According to Miller (2017) 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[1] (i.e. explainee) knows. That is, explanations, being in fact conversations, should follow Grice's maxims of quality, quantity, manner and relevance (Grice 1975).

In our work we consider in particular the work of Malle (2004), which argues that humans use commonsense psychological constructs (e.g. beliefs, desires) to explain behaviour. This leads us to adopt an agent model that includes desires and beliefs, specifically the well-known BDI model (Rao and Georgeff 1992; Bratman, Israel, and Pollack 1988; Bratman 1987). 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.* (2017) (cf. Section "Running Example"). This framework (described below) includes beliefs, desires, and valuings, and is the setting for our explanation algorithm. We contend that providing explanations in terms of the same concepts used in human-to-human explanations will help enable explanations to be comprehensible.

Formal Setting

In this paper, we assume that the listener assumes a goal tree model as the deliberation mechanism of the BDI agent.

A *goal tree*[2] is a tuple (N,G) of a name N, and either an action[3] (A, with associated pre and post conditions), or a combination of sub-goals (N_i, G_i), which can be in sequence (SEQ), unspecified order (AND), or a choice (OR) between options O_i, 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. We write $(G_{1\text{-}n})$ (resp. $(O_{1\text{-}n})$) 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 we define a goal tree G as:

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

[1] In the remainder of the paper, we use the term listener to refer to the one who is given the explanation.

[2] We use "goal" to be consistent with the literature. Space precludes a discussion of the subtle distinction between "desire" and "goal".

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

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 in the paper.

The formal semantics of a goal tree is obtained by mapping it to a sequence of actions. A goal tree can yield multiple such sequences, so formally $[\![(N, G)]\!]$ is a set of sequences of actions with the expectation that each sequence of actions achieves the goal.

We extend goal trees with *valuings*. The semantics of valuings is based on the theory of values as put forward by Schwartz (2012). In Weide (2011) it is shown how these abstract values can be connected to concrete aspects of action decision. Following Cranefield *et al.* (2017) 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. 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. In Cranefield *et al.* (2017) it is shown how these valuings can be kept consistent and work for large goal-plan trees. Due to space restrictions we do not repeat that part in this paper, but just assume the valuings to be present and indicating consistent preferences over alternatives.

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, as already can be seen by the fact that 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 weighting of the different aspects and thus the resulting valuings might not be fixed, but depend on the context. Also he might prefer the best coffee from the shop, but not want to spend much time to get it when he is finishing a paper for a deadline. Thus, in general a valuing (or preference) for an option is based on the values, but also on the current situation and practical considerations.

Finally, we note that not all of the information that we use would necessarily need to provided by the designer. For instance, action postconditions and preconditions could perhaps be learned from observation. By defining actions in terms of their pre and post conditions we can view them basically as black boxes.

Running Example: Getting Coffee

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). We assume that an observer, not aware of Jo's preferences, will require an explanation concerning Jo's actual choice.

As noted earlier, we follow (Cranefield et al. 2017) in capturing valuings as annotations. In this case 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 the office and kitchen are close to each other ("low" distance), and the shop is far from both kitchen and office ("high" distance).

Figure 1 shows a goal tree for obtaining coffee in the setting of this running example. The figure also shows the pre- and post-conditions, and the valuing annotations. We use $dist(L_1, L_2)$ to denote the distance between locations L_1 and L_2.

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 explanatory factors. Factors can be beliefs that were held (i.e. logical formulae, *Condition*), desires that were pursued, and valuings. Valuings are explained as "I preferred V to $\{V_1, \ldots, V_n\}$". We also have forward-looking explanatory factors 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 explanatory factor 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. Formally an explanation X can be defined as:

$$
\begin{aligned}
X \quad &::= \quad \bot \mid \{X'_1, \ldots, X'_n\} \\
X' \quad &::= \quad \mathit{Condition} \mid \{V_1, \ldots, V_n\} \prec V \mid \textsf{Desire}\ N \\
&\qquad\ \mid N_1 \mapsto N_2 \mid \textsf{Tried}\ N \mid \top
\end{aligned}
$$

We now define an explanation function which explains why a particular action was done. The definition of the explanation function E is with respect to the goal-tree and observed behaviour trace. Specifically, $E^T_N(G_{N'})$ 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

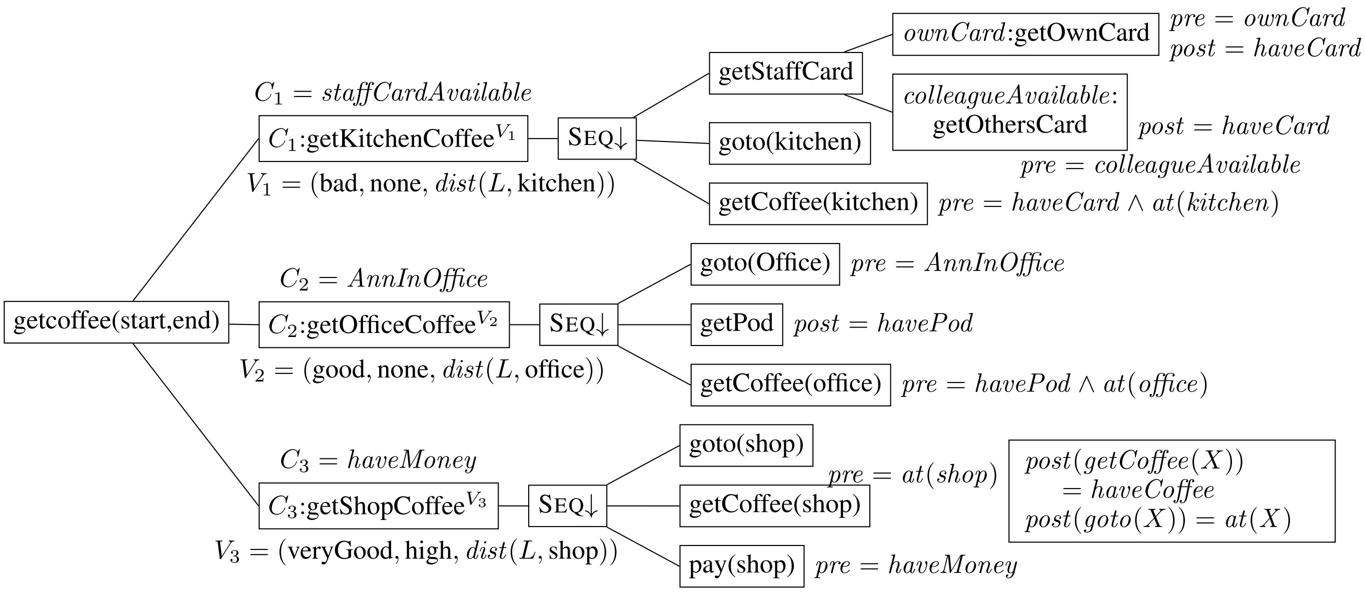
getcoffee(start,end)
$C_1 = \mathit{staffCardAvailable}$
C_1:getKitchenCoffee^{V_1}
$V_1 = (\text{bad}, \text{none}, \mathit{dist}(L, \text{kitchen}))$
SEQ↓
getStaffCard
$\mathit{ownCard}$:getOwnCard
$\mathit{pre} = \mathit{ownCard}$
$\mathit{post} = \mathit{haveCard}$
$\mathit{colleagueAvailable}$: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

Figure 1: Running Example

that occurs before N, and we overload set membership to operate on the trace T which is a sequence of names. Note that if $N \notin T$ then we simply define $E_N^T(G) = \bot$, otherwise the rest of the definitions below apply.

$$\begin{aligned}
E_N^T(G_{N'}) &= \bot\text{ , if } N \notin T \\
E_N^T(A_{N'}) &= \begin{cases} \{\} & \text{if } \mathit{pre}(A) = \top \\ \{\mathit{pre}(A)\} & \text{otherwise} \end{cases} \\
E_N^T(\text{AND}(G_i)_{N'}) &= \Omega \\
E_N^T(\text{SEQ}(G_i)_{N'}) &= \Omega \\
E_N^T(\text{OR}(O_i)_{N'}) &= \begin{cases} \Omega \cup \Theta & \text{if } N \in n(G_i) \\ \Omega & \text{otherwise} \end{cases}
\end{aligned}$$

$$\text{where } \Omega = \bigcup_{G_i : n(G_i) \cap T^{\prec N} \neq \emptyset} E_N^T((N_i, G_i))$$

$$\text{and } \Theta = \mathit{pref}(O_i, \{O_1, \ldots, O_n\})$$

The function E collects explanation factors 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 Ω: $G_i : n(G_i) \cap T^{\prec N} \neq \emptyset$ finds all sub-goals G_i which contain at least some node that appears in the prefix of the trace T before N.

In the case of an action A the explanation collected is the action's precondition. This is because whether the precondition holds or not 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 $(N, \text{OR}(O_1, \ldots, O_n))$, where G_i was selected the explanation consists of three parts (recall that $O_j = (C_j, G_j)$ where C_j is a condition):

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[4], the explanation includes that the condition was false:
$$\bigcup_{C_j : \mathcal{B}(N) \not\models C_j} \neg C_j$$
3. for each condition C_j ($j \neq i$) that is true at the decision point, the annotations of those sub-goals, and an indication that the selected sub-goal was preferred to the other available sub-goals[5] in the current situation:
$$\{V_j \mid j \neq i \wedge \mathcal{B}(N) \models C_j\} \prec V_i$$

We then define $\mathit{pref}(O_i, \{O_1, \ldots, O_n\})$ as the union of the above three parts: $\mathit{pref}(O_i, \{O_1, \ldots, O_n\}) = \{C_i\} \cup \left(\bigcup_{C_j : \mathcal{B}(N) \not\models C_j} \{\neg C_j\}\right) \cup \{\{V_j \mid j \neq i \wedge \mathcal{B}(N) \models C_j\} \prec$

[4] $\mathcal{B}(N)$ stands for the beliefs of the agent just *prior* to the execution of the goal tree (N, G). Note that recording, and retrieving, a history of beliefs during execution can be done efficiently (Koeman, Hindriks, and Jonker 2017).

[5] This is one place where the details of the BDI model matter. If the children of OR nodes are considered in a particular order and the BDI model is being used post-hoc, then instead of considering $j \neq i$ we only consider options that appear earlier, i.e. $j < i$. If the BDI model is being used to both generate and explain behaviour then, assuming the implementation captures which options were considered when O_i was selected, we would only consider those options that were considered in the explanation.

$V_i\}$ where we define $\{\} \prec V_i$ to be equivalent to $\top$, in other words, we do not generate that part of the explanation if there actually was no alternative option.

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: $\{C_3, \neg C_2, \{V_1\} \prec V_3\}$. in other words: "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". 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 other words: "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 second option (office coffee) was selected, even though all three C_i were true. In this situation, in order to explain why $\{V_1, V_3\} \prec V_2$ we would need to use two factors. We could note 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 at this point in time.

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 before A'. We now need to formalise these two conditions.

For the first condition, i.e. that the post-condition of A is required for the pre-condition of A' to hold, an obvious formalisation is simply $post(A) \rightarrow pre(A')$. But A's post condition may be only *part* of the pre-condition. For example, the action getPod only achieves havePod, so $post(\text{getPod}) \not\rightarrow pre(\text{getCoffee(office)})$. 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: $(\neg post(A)) \rightarrow (\neg pre(A'))$. This assumes that $post(A) \neq \top$. In our setting, where pre and post conditions are conjunctions of positive atoms, this is equivalent (viewing the conjunctions as sets) to $post(A) \neq \emptyset \wedge post(A) \subseteq pre(A')$.

The second condition (A before A') holds exactly when A and A' have a common ancestor that is a SEQ node, where the sub-tree containing A occurs before the sub-tree containing A'. Formally:

$$\begin{aligned} \mathit{before}(A, A') \quad &\equiv \quad \exists N = \textsc{Seq}(G_{1-n}) \; : \\ &\qquad A \in n(G_i) \wedge A' \in n(G_j) \wedge i < j \end{aligned}$$

Combining, we therefore have:

$$\begin{aligned} \mathit{link}(A, A') \quad &\equiv \quad \mathit{before}(A, A') \wedge \\ &\qquad post(A) \neq \emptyset \wedge post(A) \subseteq pre(A') \end{aligned}$$

We then extend the explanation with preparatory action explanations: when explaining an action A given goal tree G and trace T, we add to the explanation the set of links $A \mapsto A'$ where $A' \in n(G) \wedge \mathit{link}(A, A')$. 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 (Thangarajah, Padgham, and Winikoff 2003a; 2003b; Visser et al. 2016), and extend pre and post conditions to intermediate goals, inferring them (details omitted due to space).

Adding Motivations

Finally, we also add explanations in terms of parent goals: these are desires that explain why the current course of action is being pursued.

This factor 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.

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

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

For example, given the scenario described, in a situation where C_1 and C_3 hold, but not C_2, the possible factors that could be used to explain why the agent did goto(shop) are (assuming a starting location at the shop): {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.

Adding Failure Handling

We now extend the explanation mechanism to handle failure handling. Informally[6], 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

[6] Space precludes a full formal definition.

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 $\mathit{failed}^T(A)$). Then the condition under which a non-leaf node is considered to be failed is defined as:

$$\begin{aligned}
\mathit{failed}^T(\textsc{And}(G_{1-n})) &= \bigvee_{1\leq i\leq n} \mathit{failed}^T(G_i)\\
\mathit{failed}^T(\textsc{Seq}(G_{1-n})) &= \bigvee_{1\leq i\leq n} \mathit{failed}^T(G_i)\\
\mathit{failed}^T(\textsc{Or}(O_{1-n})) &= \bigwedge_{1\leq i\leq n} \mathit{failed}^T(G_i)
\end{aligned}$$

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, Ω, we exclude failed nodes:

$$\Omega = \bigcup_{G_i : n(G_i)\cap T^{\prec N}\neq\emptyset \wedge \neg \mathit{failed}^T(G_i)} E_N^T((N_i, G_i))$$

Turning to *pref* 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. So, instead of the second component being $\bigcup_{C_j:\mathcal{B}(N)\not\models C_j} \neg C_j$ we modify it to $\bigcup_{C_j:\mathcal{B}(N)\not\models C_j \wedge \neg \mathit{failed}^T(G_j)} \neg C_j$ Similarly, we modify the third component to:

$$\{V_j \mid j \neq i \wedge \mathcal{B}(N) \models C_j \wedge \neg \mathit{failed}^T(G_j)\} \prec V_i$$

Finally, we 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 G_j". Formally we have:

$$\{\mathsf{Tried}(G_j) \mid j \neq i \wedge \mathit{failed}^T(G_j)\}$$

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, {⟨bad, none, low⟩} ≺ ⟨ veryGood, high, high⟩, 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".

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*.

Evaluation of Usability

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.

However, before proceeding to present the evaluation, we make three observations. Firstly, and most importantly, our mechanism is informed by extensive research in social sciences, and uses concepts that we know humans use when explaining behaviour. That our explanations are couched in terms of familiar concepts suggests that the explanations are likely to be comprehensible. Secondly, we gave earlier a number of explanations generated for the running example. We argue that these explanations are both comprehensible and useful. In particular, we note that explanations are not excessively long, and, furthermore, that explanations are not excessively complex. What these examples do not, and cannot, show, is the extent to which explanations remain compact and comprehensible for larger examples. However, we note that we have not yet developed a selection mechanism. We know (Miller 2017) that humans give selected explanations, rather than complete explanations. Providing a selection mechanism would help ensure that explanations remain compact even with larger domains. Thirdly, as it can be seen in the evaluation described below, we use natural language sentences. The implementation of the algorithm described in the previous section was used to generate the complete explanation, including all factors, (beliefs, valuings, goals). These were manually converted to natural language. The focus of current work is the generation of formalized explanations from a goal-plan tree, and not generation of the corresponding sentences in natural language. Given the formal syntax of the explanations generated by the algorithm, it is straightforward to use patterns to map explanations to natural language.

Our evaluation took the coffee scenario described and administered a survey. Note that we focused 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 minute survey. Each participant was provided with a brief description of the 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 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

	Cohort 1						Cohort 2						Cohort 3					
	Believ.		Accept.		Compr.		Believ.		Accept.		Compr.		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	

Figure 2: Believability, Acceptability, and Comprehensibility average scores (1=very bad, 5=excellent).

(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 (Harbers 2011) (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)

In the case where the coffee shop is selected (Cohort 2), the five explanations are:

(E1) "This is the best possible coffee available; I had money." (V+B)

(E2) "This is the best possible coffee available." (AV)

(E3) "This is the best quality coffee." (V)

(E4) "I've money; Ann was away." (B)

(E5) "I wanted coffee; I had money." (G+B)

In the case where the kitchen is selected (Cohort 3), the five explanations are:

(E1) "This is the cheapest coffee; Ann was away." (V+B)

(E2) "This is the best possible coffee available." (AV)

(E3) "This coffee is cheaper than in the shop." (V)

(E4) "I've a card; Ann was away." (B)

(E5) "I wanted coffee; I've a card." (G+B)

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). Finally, we also collected demographic information.

Expla-nation	**Average Ranking and Implied Collective Ranking**					
	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	

Figure 3: Rankings for the three Cohorts and five Explanations.

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[7]) 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[8] confirms there is a difference amongst the explanations for that cohort[9].

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. Note that the most preferred rank is 1, and the least preferred is 5, so a *lower* average ranking is a *more* preferred explanation. The table also shows for each explanation and cohort the preferred order of explanations implied by the average ranking (the implied collective ranking). For example, for cohort 1, explanation 3 had the 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

[7] Recall that 1 = very bad, and 5 = excellent.

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

[9] All p values are < 0.05 and hence significant, space precludes presentation of the post-hoc tests, a sequence of pair-wise Mann-Whitney tests, with Holm adjustment to reduce Type I errors, which find that some of the pairwise differences are significant.

(Cohort 1), E4 and all other explanations (Cohort 2), and between E1-E2, E1-E4, E1-E5, E2-E3, E3-E5 (Cohort 3). Space precludes detailed discussion.

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.

Evaluation of Efficiency

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[10] $O(N^2)$. 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.

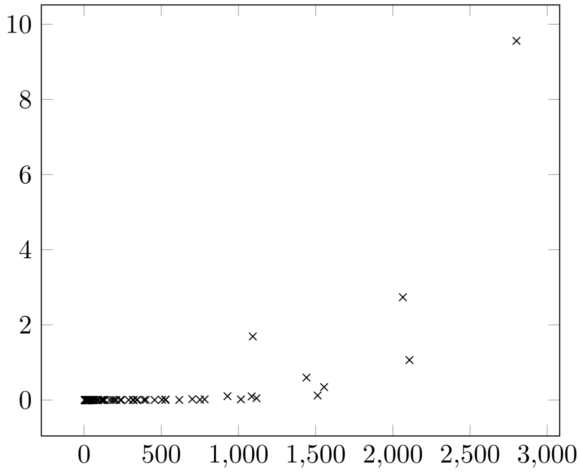

Figure 4: Time in seconds (Y) vs. number of nodes (X)

Figure 4 shows a scatter plot[11] of runtime (Y axis, in seconds) against the number of nodes in the tree (X axis). It is worth emphasising 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.

As can be seen, for relatively small trees (fewer than 1000 nodes) the explanation generation, even with an unoptimised Haskell prototype, is clearly fast enough to be practical. It is worth noting that real goal trees are not necessarily large. For instance, the (real-world) application described by Burmeister *et al.* (Burmeister et al. 2008) has 57 nodes in its goal tree.

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

[11] The time taken is affected not just by the number of nodes, but also by the shape of the tree, so there can be a number of trees with the same number of nodes and different time taken to compute the explanation.

Related Work

Miller (2017) surveys a broad range of work in the social sciences. His key argument is that work on explainable AI should take account of the work that has been done on human explanation of behaviour. We have already, in the introduction, summarised his three key findings (that explanations are contrastive, selected, and social). However, the paper does not itself propose any specific explanation mechanisms, instead it provides a broad survey. In doing so, it raises questions, poses challenges to the field of explainable AI, and identifies various factors that affect the design of explanation mechanisms. For example, that humans give explanations of the behaviour of intentional entities in terms of folk psychological constructs. Another example is the importance of abnormality in explanations: a factor that is unexpected, or abnormal, is more important in an explanation, even though other explanatory factors might be closer in time.

Harbers (2011), like us, assumes that a goal tree is given, and defines a number of templates that can be used to explain observed behaviour. Harbers' templates provide a range of possible explanations, e.g. explaining an action in terms of its parent goal, or its grand-parent goal, or the *next* action (corresponding to our links). She also conducted an empirical evaluation of the different templates in a given scenario. It is worth noting that our approach strictly generalises Harbers' approach, in that we include links, ancestor goals, and relevant beliefs. In other words, every factor that is included in explanations generated using Harbers' templates is included in $\mathcal{E}$. Furthermore, Harbers does not take into account other available alternatives, i.e. it explains why X was done solely in terms of what allowed X to be done, and does not mention anything relating to other available options. Finally, we note that whereas Harbers just outlines the rules as brief templates, we provide full formal definitions that have been implemented.

An approach closer to our work 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 (Chakraborti et al. 2017). This assumption is supported by psychological studies that observed that explanations are *"typically contrastive... the contrast provides a constraint on what should figure in a selected explanation..."* (Lombrozo 2012). However, this approach does not link to the values/valuings, beliefs and desires of the human in the loop and is therefore less adequate to connect to the reasons behind the decisions taken in the process. Additionally, it assumes that we *know* the human's mental model, which is a fairly strong assumption, and one that we do not make.

Finally, it is worth noting that the term "explaining" can be applied to other things. In this paper we tackle the problem of an autonomous system explaining its *behaviour* to a human observer. There is also work (e.g. (Milliez et al. 2016)) that considers an autonomous system (specifically a robot) explaining a *plan* with a human collaborator. The problem being addressed is different. In the work of Milliez *et al.* there is a plan, and the explanation is of the next steps that the human needs to perform. In our work the explanation is of the observed behaviour, i.e. the past actions of the software, rather than the immediate future actions of the human.

Discussion

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 that "*Among the mental states that function as reasons, beliefs and desires are most common, and there is a third class that we might call* valuings" (Malle 2004, Section 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, and, indeed, the option that corresponds to (part of) our explanation function was collectively ranked as the best explanation by two of the evaluation cohorts, and as second-best by the remaining cohort.

However, there is scope for further empirical evaluation. This would include using more scenarios, including the other explanatory factors, and assessing 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* explanatory factors. However, this is only part of the solution to the problem of explaining behaviour. We know that humans select parts of the explanation (Miller 2017). The next step in this research is to define means for *selecting* parts of the possible explanation. This would need to take into account what information is available about the human asking for the explanation, for instance, when a contrastive question of the form "why did you do X rather than Y?" is asked, we can interpret Y as the expected behaviour, and look for factors that specifically explain the difference between the implied expected behaviour of Y and the actual behaviour of X.

There is also scope to extend the representation (goal-plan trees) in various ways, and to consider related notations such as Behaviour Trees (Colledanchise and Ögren 2017), Hierarchical Task Networks (Alford et al. 2016) (which extend Hierarchical Goal Networks (Shivashankar 2015)), or Geib & Goldman's Plan Trees (Geib and Goldman 2009).

Finally, note that in our work we have assumed that explanations are *honest*, in that they attempt to provide an accurate representation of the reasons for the observed behaviour. An alternative, which we eschew, is to consider *dishonest* explanations that seek to present the agent in the best possible light.

References

Alford, R.; Shivashankar, V.; Roberts, M.; Frank, J.; and Aha, D. W. 2016. Hierarchical planning: Relating task and goal decomposition with task sharing. In Kambhampati, S., ed., *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, 3022–3029. IJCAI/AAAI Press.

Bratman, M. E.; Israel, D. J.; and Pollack, M. E. 1988. Plans and resource-bounded practical reasoning. *Computational Intelligence* 4:349–355.

Bratman, M. E. 1987. *Intentions, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press.

Burmeister, B.; Arnold, M.; Copaciu, F.; and Rimassa, G. 2008. BDI-agents for agile goal-oriented business processes. In *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS) [Industry Track]*, 37–44. IFAAMAS.

Chakraborti, T.; Sreedharan, S.; Zhang, Y.; and Kambhampati, S. 2017. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 156–163.

Colledanchise, M., and Ögren, P. 2017. Behavior trees in robotics and AI: an introduction. *CoRR* abs/1709.00084.

Cranefield, S.; Winikoff, M.; Dignum, V.; and Dignum, F. 2017. No Pizza for You: Value-based Plan Selection in BDI Agents. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 178–184.

EU. 2016. EU General Data Protection Regulation. http://tinyurl.com/GDPREU2016 (see articles 13-15 and 22).

Geib, C. W., and Goldman, R. P. 2009. A probabilistic plan recognition algorithm based on plan tree grammars. *Artif. Intell.* 173(11):1101–1132.

Grice, H. P. 1975. *Logic and conversation*. Academic Press, New York.

Gunning, D. 2018. Explainable Artificial Intelligence (XAI). https://www.darpa.mil/program/explainable-artificial-intelligence.

Harbers, M. 2011. *Explaining Agent Behavior in Virtual Training*. SIKS dissertation series no. 2011-35, SIKS (Dutch Research School for Information and Knowledge Systems).

Koeman, V. J.; Hindriks, K. V.; and Jonker, C. M. 2017. Omniscient debugging for cognitive agent programs. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 265–272.

Lombrozo, T. 2012. Explanation and abductive inference. *Oxford handbook of thinking and reasoning* 260–276.

Malle, B. F. 2004. *How the Mind Explains Behavior: Folk Explanations, Meaning, and Social Interaction*. The MIT Press. ISBN 0-262-13445-4.

Miller, T. 2017. Explanation in artificial intelligence: Insights from the social sciences. *CoRR* abs/1706.07269.

Milliez, G.; Lallement, R.; Fiore, M.; and Alami, R. 2016. Using human knowledge awareness to adapt collaborative plan generation, explanation and monitoring. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, HRI '16, 43–50. Piscataway, NJ, USA: IEEE Press.

Rao, A. S., and Georgeff, M. P. 1992. An abstract architecture for rational agents. In Rich, C.; Swartout, W.; and Nebel, B., eds., *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, 439–449. San Mateo, CA: Morgan Kaufmann Publishers.

Schwartz, S. 2012. An Overview of the Schwartz Theory of Basic Values. *Online Readings in Psychology and Culture* 2(1).

Shivashankar, V. 2015. *Hierarchical Goal Networks: Formalisms and Algorithms for Planning and Acting*. Ph.D. Dissertation, University of Maryland, College Park, MD, USA.

Thangarajah, J.; Padgham, L.; and Winikoff, M. 2003a. Detecting and avoiding interference between goals in intelligent agents. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, 721–726.

Thangarajah, J.; Padgham, L.; and Winikoff, M. 2003b. Detecting and exploiting positive goal interaction in intelligent agents. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 401–408. ACM Press.

van der Weide, T. 2011. Arguing to motivate decisions. Dissertation, Utrecht University Repository https://dspace.library.uu.nl/handle/1874/210788.

Visser, S.; Thangarajah, J.; Harland, J.; and Dignum, F. 2016. Preference-based reasoning in BDI agent systems. *Autonomous Agents and Multi-Agent Systems* 30(2):291–330.

Winikoff, M. 2017. Towards Trusting Autonomous Systems. In *Fifth Workshop on Engineering Multi-Agent Systems (EMAS)*.