April 2022

What can I do to help you?

A formal framework for agents reasoning about behavior change support for people

Myrthe L TIELMAN [a,1], M. Birna VAN RIEMSDIJK [b] and Michael WINIKOFF [c]
[a] *Delft University of Technology, Netherlands*
[b] *University of Twente, Netherlands*
[c] *Victoria University of Wellington, New Zealand*

Abstract. Changing one's behavior is difficult, so many people look towards technology for help. However, most current behavior change support systems are inflexible in that they support one type of behavior change and do not reason about how that behavior is embedded in larger behavior patterns. To allow users to flexibly decide what they desire to change, a system needs to represent and reason about that desire. Moreover, we argue that reasoning about the context of a behavior could improve an agent's support. Therefore, we propose a formal framework for a reasoning agent to represent and reason about the personal behavioral context of desired user changes. This framework models an individual's possible and current behavior, their desire for change, as well as other relevant changes that a system could use to support a desired change. In a user survey we show that people feel these other relevant changes would be useful in more flexibly supporting their desired change in behavior. This work provides a foundation for more flexible personalized behavior change support.

Keywords. behavior change, formal framework, behavior trees.

1. Introduction

There are many circumstances in which people want or need to change their behavior. Such behavioral changes include health-related changes, such as getting more exercise or eating healthier, spending more time with family, or spending time more effectively. However, changing behavior is difficult and assistance is, therefore, very welcome [2]. Technology such as AI is increasingly used to provide such assistance, since, unlike human coaches, it is relatively cheap, and always available to help [24].

Behavior change support systems (BCSS) are developed for many behaviors and show great promise [16,10], especially with advances in AI. However, a BCSS is typically aimed at a specific behavior (such as living more healthily [17,12] or saving energy [11]), and does not incorporate any representation of the personal context in which that behavior occurs. This inflexibility and lack of representation has several possible consequences. Firstly, the support might not be as effective, since different people have different needs, desires and preferences [22]. Some people might wish to be more sustainable by taking shorter showers, while others would choose to eat less meat. Secondly,

[1] Corresponding Author: M. Tielman, m.l.tielman@tudelft.nl.

people also prefer diverse and personalized support and might, therefore, not use inflexible systems [8]. Thirdly, opportunities might be missed when a system does not reason about the consequences of a change. Some people might be assisted in biking more by convincing them to drive less. But that only works if they typically use a car. Finally, this inflexibility also means systems typically are not capable of supporting people with multiple different goals, such as both healthy eating and a more active lifestyle.

In order to equip a behavior change support system with the means to flexibly support various behaviors, it needs to model both the users' existing actions and the consequences of changes to those actions. Most BCSS follow two main steps: identifying the obstacle to change (e.g. motivation, or knowledge), and selecting an appropriate intervention method (e.g. convincing, or informing) [13]. These steps are typically done by the designers of the system. However, for a flexible system the behavior to change is not known in advance and the system itself will have to choose how to best support a change.

We argue that for an AI support system to make this choice, we need to add two steps at the start. Firstly, we need to model (i.e. represent) what behavior the user wishes to change, as well as the context that this behavior exists in (e.g. what alternatives exist). Actions do not typically exist in isolation, and stopping something might only be possible if you start something else, or stop doing other things as well. Secondly, we propose that modeling what other behavior changes could help bring about the desired change is important. This allows for a system to support in more flexible ways. For example, supporting biking could be useful when a user wishes to drive less if those are the two options. This type of reasoning requires a model of both the behavioral context of a change, as well as of the causes of and possible obstacles to a specific change. The example scenario illustrates these two steps we propose for a specific use case.

Example scenario: *Alice and Bob want to change their behavior to cycle to work more often, and use a BCSS. Step one is identifying their possible behavior. Alice has the option to go by bus, walk or bike, while Bob has the option to drive or to bike to work. This step includes identifying what actions are done to perform these options. Both must first find the keys for the bicycle lock, but also unlock the shed. For Bob driving involves finding car keys and driving. For Alice the bus involves walking to the bus stop, getting on and getting off and walking involves walking the route to work. Next, we need to model other possible support. Alice normally takes the bus, while Bob drives. Because they cannot both bike and take the car or bus, a system could support with stopping these behaviors. However, for Bob stopping to drive would always involve biking, while Alice has walking as an alternative as well. So for her, stopping to take the bus will not necessarily help her bike. Additionally, if Alice and Bob cannot find their bike keys they will not be able to bike. So if finding their keys is a problem, a system could also assist with this action.*

Only after these things are clear could a system support the desired changes by identifying the obstacles to change, selecting an intervention technique appropriate for the individual user, and executing this, for instance following [13].

In this paper, we present a formal framework which allows a BCSS to represent an individual's behavior and wish for change, as well as to reason about the causes of and possible obstacles to those changes. We identify two main contributions. Firstly, our framework is capable of **modelling a wish for change**, including how this is embedded in an individual's possible behavior. Secondly, based on this information it allows **reasoning about exactly what other changes could be relevant to a wish for change**.

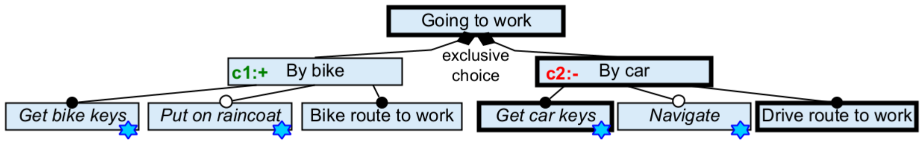

Figure 1. Behavior diagram of going to work. Black circles are mandatory and white circles optional. Bold outlined actions are being done, and actions in italics with stars are ambiguous. We consider two change models: c1 (starting to bike to work; green "+") and c2 (stopping driving to work; red "−").

The combination of motivation and formalisation is the core of this paper's contribution. Additionally to this motivation and formalization, we also present a user survey which evaluates the underlying intuitions about what additional relevant changes exist, given a user's main change wish.

In the remainder of this paper, we will use a running example involving ways of traveling to work (see Figure 1). Section 2 presents a framework for modeling the user's behavior and desired changes and Section 3 a method to identify other relevant changes. In Section 4 we present a user study that shows that with this framework, a behavior change support agent is able to identify other changes relevant to a main change which are intuitive to users. We finish (Section 5) with a discussion and conclusion on how our work contributes towards more flexible support for a behavior change support agent.

2. Behavior Model

To model the user's wish for change, we first need to model the structure of their possible behavior, specifically how different actions relate to each other. Given that behaviors can form hierarchies (e.g. smaller actions make up larger behaviors), we follow [9] and assume a basic tree structure (Definition 2.1) of actions that are annotated with various properties (Definition 2.2) to give a *behavior diagram*.

Definition 2.1 (Tree) *Let N be a set of actions and $E : N \times N$ a relation on N, where $(a,b) \in E$ denotes that a is the parent of b. We say that (N,E) is a* tree*, if E is antisymmetric, irreflexive, and such that for any a, b, $c \in N$, if $(a,b) \in E$ and $(c,b) \in E$, then $a = c$. We use r to denote the root of a tree, i.e., the node $m \in N$ such that there is no $n \in N$ with $(n,m) \in E$. There can be precisely one such root node in any tree.*

Each action $n \in N$ that is not a leaf in the tree is labelled as one of `xor` (it is done by doing exactly one of its children), `or` (it is done by doing one or more of its children), or `option` (it is done by doing all of its children that are in a `mandatory` relationship to it, and some (possibly none) of non-mandatory (i.e. `optional`) children).

To illustrate with our example (Figure 1), the action *going to work* can be performed in two different ways and is an exclusive choice (`xor`) between biking and driving. The action of *biking to work* ("By bike", `option`) needs to include the mandatory *Get bike keys* and *bike route to work*, and could also optionally include *put on raincoat*.

One difference between an `or`/`xor` relationship and an `option` relationship is that while for the first doing any child means you are doing the parent (e.g. biking to work means you are going to work), for the second the child is only a *part* of what defines the activity (e.g. getting keys is only a part of biking to work). Because of that, we say that `optional` and `mandatory` children could potentially be practically indistinguishable from actions performed in a different context (e.g. getting bike keys when tidying).

We, therefore, define *ambiguous* actions as being those that can occur in different behavioral contexts. Distinguishing between ambiguous and non-ambiguous actions is important, because when we look to support a behavior then we also consider other actions that, if supported, can contribute to the change. For example, we might support Alice cycling to work by assisting her to bike the route to work, if she has trouble finding the way. On the other hand, helping her find her bike keys is less helpful: because finding the keys is ambiguous, supporting it might not help with cycling to work specifically.

Definition 2.2 (Behavior Diagram) *A Behavior Diagram D is a structure* $D = (N, AMB, E, \lambda, \mu)$ *such that:*

- *N is the set of* nodes *representing actions;* (N,E) *is a tree;*
- $E \subseteq N \times N$ *is the set of* decomposition edges *and* $N^* \subseteq N$ *is the set of nodes that are not leaves (i.e.* $\forall a \in N^* \, \exists b \in N : (a,b) \in E$*);*
- $\lambda : N^* \to NT$ *is a labelling of the nodes, where* $NT = \{\texttt{or}, \texttt{xor}, \texttt{option}\}$ *is the set of* node types*;*
- $AMB \subset N$ *is those actions that are* ambiguous*, i.e. they can occur in different behavioral contexts (they must be the children of an* `option` *node:* $\forall n \in AMB : \exists (n', n) \in E : \lambda(n') = \texttt{option}$*);*
- *and finally, if we let* $N^{opt} \subseteq N$ *be the set of nodes with label* `option`*, i.e.,* $\{a \mid a \in N, \lambda(a) = \texttt{option}\}$*, and* E^{opt} *be the set of edges emerging from these nodes, i.e.,* $\{(a,b) \mid (a,b) \in E, a \in N^{opt}\}$*, then* $\mu : E^{opt} \to \{\texttt{mandatory}, \texttt{optional}\}$ *is a labelling of these edges.*

Given this type of behavior diagram, a behavior or *model* consists of a subset of the diagram's nodes. A model intuitively specifies one typical way in which the user performs the behavior specified by the diagram. A behavior is *practical* if it can practically occur according to the behavior diagram (Definition 2.3). This means that the behavior adheres to the specified node types of the diagram, and that a child activity is only included if its parent activity is also performed, unless the child activity is ambiguous.

Definition 2.3 (Model / Practical Model) *Let* $D = (N, AMB, E, \lambda, \mu)$ *be a Behavior Diagram. A* model *of D is a subset* $M \subseteq N$ *of the nodes of D. A* Practical Model*, denoted* $D \Vdash M$*, is a subset* $M \subseteq N$ *such that*

- *if* $n \in M$ *and* $\lambda(n) = \texttt{or}$*, then for at least one* $m \in N$ *with* $(n,m) \in E$*,* $m \in M$*;*
- *if* $n \in M$ *and* $\lambda(n) = \texttt{xor}$*, then for precisely one* $m \in N$ *with* $(n,m) \in E$*,* $m \in M$*;*
- *if* $n \in M$ *and* $\lambda(n) = \texttt{option}$*, then for all* $m \in N$ *with* $(n,m) \in E$ *and* $\mu((n,m)) = \texttt{mandatory}$*,* $m \in M$*;*
- *if* $m \in M$ *with* $m \neq r$*, and* $m \notin AMB$*, then also* $n \in M$ *for the unique n with* $(n,m) \in E$*.*

Note that this definition does not require that the root is in the model, because we can have a model where the type of behavior specified by the diagram is not performed at all ($M = \emptyset$) or where we observe some actions that are part of the behavior (e.g. *Get bike keys* $\in M$), but are ambiguous and hence can also appear in other behaviors, so we cannot conclude that the behavior is being performed in a specific context.

Just a model of possible behavior is not enough if we wish to support people in changing their behavior. To do this, we need to know three additional things. Firstly, what a person is currently doing (i.e. how they normally do things), secondly, what they wish to change, and thirdly, for which of the desired changes support is sought. To express this quadruple of possible behavior (D), actual behavior (AB), desired changes (DC), and sought support (S) we use a *behavior model* (BM, Definition 2.4).

We assume that the user's desired changes satisfy two conditions. Firstly, it cannot already hold. Secondly, a desired change must be *achievable*[2]. These conditions align with literature on goals which defines that you cannot want to achieve something that is already the case [7,6], and that a goal must be achievable (e.g. [23]).

Definition 2.4 (Behavior Model) *Let $D = (N, AMB, E, \lambda, \mu)$ be a Behavior Diagram. We then define a Behavior Model $BM = (D, AB, DC, S)$ where:*

- *$AB \subseteq N$ such that $D \Vdash AB$ (i.e. AB is a Practical model of D)*
- *$DC : N \to \{\mathsf{change}, \mathsf{keep}\}$ is a (partial) function that maps some of the actions $a \in N$ to either* change *or* keep. *These respectively indicate that a is desired to* change *(either start or stop), or be kept unchanged (*keep*). If a is not mapped by DC (formally written $a \notin domain(DC)$) then it indicates that the DC is agnostic about a. Below we write DC as a set of pairs, i.e. $DC \subset N \times \{\mathsf{change}, \mathsf{keep}\}$.*
- *DC is not achieved, i.e., there is at least one $n \in N$ that is mapped to* change. *Formally: $\{n \mid (n, \mathsf{change}) \in DC\} \neq \emptyset$.*
- *DC is achievable with respect to D and AB (see Definition 2.6).*
- *S is a set of nodes ($S \subseteq N$) that are in DC where support is desired. It must be a non-empty subset of the things in DC (formally: $S \subseteq domain(DC) \wedge S \neq \emptyset$). In many cases, the simplest S is simply all of those things that are desired to change, i.e. $S = \{n \mid DC(n) = \mathsf{change}\}$. However, there may be situations where we want to change something, but do not need support for it. There may also be situations where we want to have support to keep doing (rather than changing) a particular behavior.*

We now turn to defining what it means to apply a change DC to a user's actual behavior AB. We define this behavior change in two steps. Firstly, we define a function Δ that applies a desired change directly by making the corresponding change to the model. This is done by adding those actions a that are changing and are not already being done ($a \notin AB$) and removing actions that are changing and already being done. However, Δ can yield an impractical model. For example, if we want to start cycling, then just adding this yields a model with both cycling and driving to work. We therefore extend Δ into a function, Δ^*, that applies the desired changes, and then applies further changes to make the model practical. These further changes are drawn from the actions which the desired change is agnostic about[3], which we name "repair options", and, in fact, we only consider *minimal* repair options[4] (denoted MRO).

[2] Although people might in practice wish non-achievable things, e.g. biking and driving to work at the same time, a system trying to help them achieve conflicting things would not be very useful. Instead, a system should be able to point out that this is the case, which our framework could do by knowing what achievability means.

[3] More precisely, that DC does not indicate that a particular change n should be kept or changed. However, in an informal sense, some of these changes will be desired changes, since they are consequences of the DC. For instance, adding mandatory children of a node that is desired to be added to the behavior.

[4] Minimality is checked by $\nexists MRO' \subset MRO . D \Vdash \Delta(BM'_M)$ in Definition 2.5

For each possible *MRO*, the function Δ^* constructs a behavior model BM_M that adds the *MRO* to the desired changes. It then uses Δ to apply these changes, and checks that the resulting model is Practical ($D \Vdash \Delta(BM_M)$). Since there may be more than one possible minimal repair option, $\Delta^*(BM)$ is a set of (Practical) models (unlike $\Delta(BM)$, which is a single, possibly non-Practical, model).

Definition 2.5 (Behavior Change) *Let* $BM = (D, AB, DC, S)$ *be a behavior model. We define functions* $\Delta(BM)$ *and* $\Delta^*(BM)$*.*

$$\Delta(BM) = (AB \cup \{a \mid (a, \mathsf{change}) \in DC \wedge a \notin AB\}) \setminus \{a \mid (a, \mathsf{change}) \in DC \wedge a \in AB\}$$
$$\Delta^*(BM) = \{\Delta(BM_M) \mid MRO \subseteq \{(n, \mathsf{change}) \mid n \notin domain(DC)\} \wedge$$
$$D \Vdash \Delta(BM_M) \wedge \nexists MRO' \subset MRO. D \Vdash \Delta(BM'_M)\}$$
$$\textit{where } BM_M = (D, AB, MRO \cup DC, S) \textit{ and } BM'_M = (D, AB, MRO' \cup DC, S)$$

We define a behavior model to be *achievable* iff its desired changes can be realised, i.e. when we apply *DC* to *AB*, then the resulting model can be made Practical by further adjustments (which do not undo any of the changes from *DC*).

Definition 2.6 (Achievable) *Let* $BM = (D, AB, DC, S)$ *be a behavior model. Then we say BM is* achievable *iff* $\Delta^*(BM) \neq \emptyset$*.*

The ability to determine achievability also explains why our behavior model supports having a wish to keep a behavior the same. Because wanting to change one thing will often mean that other things need to be changed in order for the model to remain Practical. For instance, stopping to drive the route to work will also mean that you have to start going to work by bike, or stop going to work at all. So if you do not want to stop going to work, you need to have this explicitly in the behavior model, otherwise the system will assume that this change can be made in order to create a new Practical behavior model.

Having defined what it means to apply a change *DC* to a given behavior *AB* to obtain a new behavior AB', we now define what it means for a desired change *DC* to be *realised* with respect to *AB* and AB'. Informally, this is the case if: (i) anything that is desired to change is different in *AB* and AB', and (ii) everything that is desired to be kept, is the same in both *AB* and AB'.

Definition 2.7 (Change realised) *We define a desired change DC being realised by models AB (before) and* AB' *(after), denoted by* $AB \mapsto AB' \Vvdash DC$*, as:*

- $AB \mapsto AB' \Vvdash DC \equiv \forall c.(c, \mathsf{change}) \in DC : (c \in AB \iff c \notin AB')$

  $\wedge \forall c.(c, \mathsf{keep}) \in DC : (c \in AB \iff c \in AB')$

Theorem 2.1 (Δ and Δ^* realise *DC*) *Let* $BM = (D, AB, DC, S)$*, then (i) applying* Δ *to BM results in* AB' *such that DC is realised with respect to AB and* AB' *(formally:* $AB \mapsto \Delta(BM) \Vvdash DC$*); and (ii) applying* Δ^* *to BM results in a set of* AB^* *such that for each* $AB^* \in \Delta^*(BM)$*, DC is realised with respect to AB and* AB^* *(formally:* $\forall AB^* \in \Delta^*(BM) : AB \mapsto AB^* \Vvdash DC$*).*

Proof: *(i) The definition of* Δ *yields an* AB' *that changes things in DC that are specified as* change*. This means that those changes will be realised with respect to AB and* AB'*.*

Since Δ does not change anything else, those things that should be kept are also realised. (ii) The definition of Δ^* yields a set of elements where each $AB^* \in \Delta^*(BM)$ is constructed by applying Δ with a desired change set $MRO \cup DC$. This means that those things in DC that are specified as change *are changed and will be realised with respect to AB and AB^*. Furthermore, since MRO only contains items $n \notin domain(DC)$, we have that Δ^* does not change anything that is specified in DC as* keep*. Therefore, those things in DC that should be kept are also realised with respect to AB and any $AB^* \in \Delta^*(BM)$.* □

3. Relevant Change Model

The behavior model defines what changes the user wishes to make, and where they desire support. However, in addition to directly supporting a desired change, a system might also be able to *indirectly* support. For this reason, we introduce a *relevant change model*, which captures not only our desired changes, but also the additional changes that could help in bringing these about, which are *causes* and *possible obstacles*.

Intuitively, a *cause* (Definition 3.1) is something that, if changed, will *necessarily* bring about (i.e. cause) the desired change. Following our example, supporting me to start riding my bike would also necessarily lead me to stop taking the car if I want to keep going to work. Similarly, hiding the car keys would have the same effect, as I cannot drive without getting my car keys. More precisely, a cause C is a set[5] of action-change tuples such that: (i) if they are changed, assuming that desired changes not requiring support are respected, there exists at least one minimal repair action that yields a Practical model (if this is not the case then C is not viable); (ii) all the resulting Practical models realise the desired changes that require support (i.e. C *necessarily* leads to the desired change); and (iii) C is minimal.

The formal definition of $cau(BM)$ below begins by constructing possible candidate causes ($C \subseteq \ldots$). It defines DC_{ns} as the set of things that are desired but do not require support, DC_s as the set of things that are desired and do require support, and BM_C which takes the original behavior model BM and replaces DC with $C \cup DC_{ns}$. We assess condition (i) using $\Delta^*(BM_C) \neq \emptyset$: i.e. applying C, assuming DC_{ns} is respected, allows a minimal repair action. Condition (ii) is assessed in the following line ($\forall AB^* \in \Delta^*(BM_C)\ldots$), and the following lines assess condition (iii).

Definition 3.1 (Cause) *Given a behavior model $BM = (D, AB, DC, S)$, a cause C for a desired change DC is a set of action-change tuples denoted $cau(BM)$ defined as:*

- $cau(BM) = \{\ C \mid C \subseteq \{(a, \mathsf{change}) \mid a \notin domain(DC)\} \wedge$

$$\begin{aligned}
&\Delta^*(BM_C) \neq \emptyset\ \wedge \\
&\forall AB^* \in \Delta^*(BM_C) : AB \mapsto AB^* \Vdash DC_s\ \wedge \\
&\nexists C' \subset C :\ (\Delta^*(BM'_C) \neq \emptyset \wedge \forall AB^* \in \Delta^*(BM'_C) : AB \mapsto AB^* \Vdash DC_s\ )\ \} \\
&\textit{Where } DC_{ns} = \{(a,m) \mid (a,m) \in DC \wedge a \notin S\} \\
&\textit{and } DC_s = \{(a,m) \mid (a,m) \in DC \wedge a \in S\} \textit{ and } BM = (D, AB, DC, S)
\end{aligned}$$

[5]It is a set because we may need to consider more than one action. For example, consider the tree with a at its root, and with three OR-decomposed children, b, c and d, where we have $AB = \{a, c, d\}$ and want to start doing b, and keep doing a. Stopping d won't achieve anything, nor will stopping c, but stopping both c and d will force b to be done. Therefore, $\{c, d\}$ is a cause but neither $\{c\}$ nor $\{d\}$ are causes.

$$\text{and } BM_C = (D, AB, C \cup DC_{ns}, S) \text{ and } BM'_C = (D, AB, C' \cup DC_{ns}, S)$$

Turning to *possible obstacles*, these are things that are a *consequence* of the desired change. Possible obstacles are important because if we have trouble changing these things, then we would also have trouble with our desired changes. For example, not being able to find my bicycle keys will stop me from going by bike. Therefore, supporting the user to overcome a possible obstacle can help (indirectly) support a desired change by potentially removing an obstacle to the change.

In addition to requiring that a possible obstacle is a consequence of the desired change, we also need an additional condition: in the case where not all desired changes require support, we only want to consider (a, change) to be a possible obstacle if it is a consequence of the *supported* desired changes. The reason is simple: if a change does not need to be supported, then we do not want to include its possible obstacles as support options. In the definition below, the first condition ($\forall AB^*$) checks that a changing is a necessary consequence of DC. The second condition[6] ($\exists AB'$) checks that this change is related to a desired change that requires support[7].

Definition 3.2 (Possible obstacles) *Given a behavior model* $BM = (D, AB, DC, S)$ *we define the set of all possible obstacles* $pob(BM)$*:*

- $pob(BM) = \{c \mid c \in \{(a, \mathsf{change}) \mid a \notin domain(DC)\} \wedge$

  $\forall AB^* \in \Delta^*(BM) : AB \mapsto AB^* \Vdash \{c\} \wedge$
  $\exists AB' \in \Delta^*(BM_s) : AB \mapsto AB' \Vdash \{c\}\ \}$
  Where $DC_s = \{(a,m) \mid (a,m) \in DC \wedge a \in S\}$
  and $BM = (D, AB, DC, S)$, $BM_s = (D, AB, DC_s, S)$

In considering what changes may help the user to achieve DC (i.e. the *relevant change model*, Definition 3.3), we therefore have three cases. Firstly, we could *directly* support a desired change $c \in DC$. Secondly, we could support a cause, which would then lead to the desired change. Thirdly, we could support a possible obstacle, which may be hindering the desired change. The second and third cases are indirect support, and the third is weaker in that it may not actually always help.

Definition 3.3 (Relevant change model) *Given an achievable behavior model* $BM = (D, AB, DC, S)$*, we define a* relevant change model $RCM(BM)$ *as being the smallest set such that:*

- $(\{c\}, \textsf{direct}) \in RCM(BM)$ *if* $c \in DC$*; and*
- $(C, \textsf{indirect-cau}) \in RCM(BM)$ *if* $C \in cau(BM)$*; and*
- $(\{c\}, \textsf{indirect-pob}) \in RCM(BM)$ *if* $c \in pob(BM)$.

Where direct*,* indirect-cau *and* indirect-pob *are labels.*

The next two theorems show that when changing either the causes or possible obstacles, the main desired change is achievable.

[6] This condition is an exists because the desired changes that do not require support are being ignored, which may lead to additional possible models.

[7] If $DC_s = DC$ (i.e. all desired changes require support) then this condition reduces to the desired change being achievable (since the first condition implies that any model resulting from applying DC satisfies $\{c\}$) which we assume to be the case (Definition 2.4).

Theorem 3.1 *For all* $(C, \textsf{indirect-cau}) \in RCM(BM)$ *we have that* $\Delta^*(BM_C) \neq \emptyset$ *where* $BM_C = (D, AB, C \cup DC, S)$ *.*
Proof: *We know that* $(C, \textsf{indirect-cau}) \in RCM(BM)$ *iff* $C \in cau(BM)$*. By Definition 3.1, we know that applying* $C \cup DC_{ns}$ *(in* $\Delta^*(BM_C)$*) results in a state* AB^* *where all the desired changes in* DC_s *are met. But* $DC = DC_{ns} \cup DC_s$*, and therefore in this state we also have that* $C \cup DC$ *is met, and hence is possible.* □

Theorem 3.2 *For all* $(\{c\}, \textsf{indirect-pob}) \in RCM(BM)$ *we have that* $\Delta^*(BM_c) \neq \emptyset$ *where* $BM_c = (D, AB, \{c\} \cup DC, S)$ *.*
Proof: *We know that* $(\{c\}, \textsf{indirect-pob}) \in RCM(BM)$ *iff* $c \in pob(BM)$*. By Definition 3.2 we know that applying DC results in a state in which the change* $\{c\}$ *holds. Therefore* $\{c\} \cup DC$ *is possible.* □

The definitions of Sections 2 & 3 have been implemented in Prolog to test the complex reasoning around support, causes, and possible obstacles, to ensure that the definitions yield the answers expected, and that they do not have missing cases or parts. This implementation, including an example, can be found at [21].

4. User survey

The goal of the relevant change model is to identify behaviors that, if supported to change, could be relevant to supporting the desired change. The definitions of causes and possible obstacles are based on the formal consequences and antecedents to a change. However, this does not mean that they are also seen as intuitively relevant by users seeking help with a change in behavior. Therefore, a user study was done. Participants were first asked about their own intuitions for what should be in the relevant change model, and, subsequently, were asked to assess sets with causes and possible obstacles for helpfulness in realizing the (primary) behavior change goal for different scenarios.

4.1. Methodology

We recruited 120 participants via Amazon Mechanical Turk (55% female, mean age: 36.68 (sd: 9.70)) and paid for their time. The study design was approved by the human-research ethics committee of the institution of the first author (ID=975). We began by giving participants an introduction to what behavior trees look like, including an explanation of the different types of relationships. These were always explicitly worded in the example pictures (e.g. 'exclusive choice' between two `xor` children). Additionally, all behavior change scenarios were described in text. The full survey can be found in [21].

Next, participants were each shown 6 (randomly) out of a possible 22 small behavior scenarios, along with a tree picture. The full set of trees can be found in [21]. The scenarios included trees of depth 2, combinations of the relationships (with `or`, `xor` and `option`, including both optional and mandatory children), and stopping and starting both parent and child actions (2 or 3 of them in the tree). Participants were asked: *"Aside from directly supporting [main action] (for instance motivating you to do so), which other changes could the system help you with to indirectly help you achieve your goal?"* All actions other than the main change were given as multi-select options.

Finally, participants were shown three scenarios, one with `option` as main change one with `xor`, and one with `or`). Each had two desired changes (one stopping, one starting) in two different trees both with a root and at least two children. They were then shown four possible support options: one with the causes, one with the possible obstacles, one with both, and one empty set. They were asked to first rank them: *"Below, are four options on additional changes the system could help you with to reach your main goals. Please rank these options as to which makes most sense to you."*. Then they were asked to rate them: *"Additionally, for each of these options, please indicate how helpful it would be for you to achieve your main goals of [2 main goals] if the system were to help you with them."*.

4.2. Survey Results & Discussion

In this section, we will first discuss how intuitive the sets of causes, possible obstacles and combination were to support a main change. Secondly, we will discuss participant's own intuitions on what should be in a relevant change model. Note that this order (chosen for clarity of exposition) is the opposite of the order in the experiment.

For all types of scenarios (`xor`, `or`, `option` nodes as main change), a linear regression[8] shows that the content of the relevant change set (i.e empty, possible obstacles, causes, or both) was a significant predictor of helpfulness to the main change (`xor`: F(3, 484)=24.14 ,$p < 0.001$, `or`: $F(3,484) = 64.46$, $p < 0.001$, `option`: $F(3,484) = 72.69$, $p < 0.001$ resp.). Table 1 shows the results of a single measures t-test (comparing to neutral 0) indicating whether the set of changes was considered significantly helpful. Additionally, it shows the significant results of a linear regression with pairwise comparisons between the sets on helpfulness scores. These indicate which sets were considered more helpful than which others within their category (`xor`, `or`, `option`).

For the `or` and `option` scenarios, the results show that **the relevant change model (Definition 3.3) contains only helpful changes (including the causes and possible obstacles), and that it makes sense to not distinguish between causes and possible obstacles in importance** for these scenarios. However, in the `xor` scenario we see that the set with only causes was neither helpful nor detrimental. This is surprising, but can be explained when examining the scenario. The cause included in this case involved stopping the parent of the action to be stopped, something which seems to have been counter-intuitive. E.g. if you want to stop driving to work, it does not seem right to stop going to work at all. We return to this point below, proposing an adjustment to the behavior model in Definition 4.1.

We also studied how closely aligned the causes and possible obstacles are to people's own intuitions on a relevant change model. Our experimental results show a lot of individual differences, with on average 5.14 different combinations of changes for a single scenario. If we define *consensus* to mean at least a 10% difference between the most common and second-most common answer, consensus was reached on only half of the scenarios (6 out of 7 `xor` scenarios, 5/8 `or`, 0/7 `option`). This might be because for the `option` relationship there are more possibilities for the precise relationship between parent and child actions. (e.g. some might be ambiguous and others not[9]). Moreover,

[8] All data was analyzed with R version 4.2.1.

[9] Our survey did not include the notion of actions being ambiguous; including individual intuition on which actions are ambiguous might increase the consensus.

Table 1. Statistical results for the three types of scenarios, showing both deviation from the neutral 0 on the helpfulness score, and which sets were found to be significantly more helpful than others. (PO = Possible obstacles, C=Causes, B=Both, N=Neither *=p < 0.05, **=p<0.01)

Scen.		Helpful? (score>0)	More helpful than
XOR	***PO***	m=6, t=16.77, p<0.001**	C:(F(1.242)=70.13)**, B:(F(1.242)=43.02)** N:(F(1.242)=46.62)**
	C	m= -0.48, t=-0.69, p=0.49	
	B	m=1.26, t=2.03, p<0.05*	C:(F(1.242)=3.99)**
	N	m=1.66, t=3.15, p<0.01**	C:(F(1.242)=6.46)*
OR	***PO***	m=6.26, t=24.67, p<0.01**	N:(F(1.242)=83.21)**
	C	m=5.9, t=18.56, p<0.01**	N:(F(1.242)=66.98)**
	B	m=7.45, t=23.48, p<0.01**	C:(F(1.242)=12.77)**, PO(F(1.242)=9.16)**, N:F(1.242)=111.9)**
	N	m=0.46, t=0.78, p=0.44	
Opt.	***PO***	m=6.7, t=24.49, p<0.01**	N:(F(1.242)=86.68)**
	C	m=5.98, t=19.15, p<0.01**	N:(F(1.242)=65.19)**
	B	m=7.86, t=32.86, p<0.01**	C:(F(1.242)=23.83)**, PO:(F(1.242)=9.93)**, N:(F(1.242)=131.7)**
	N	m=0.66, t=1.12, p=0.26	

there is often an implicit temporal ordering. On the other hand, the `or` and `xor` relationships have a clearer meaning. This potentially explains why there is more diversity, and less consensus, for the `option` relationship. We observed that the consensus always included all possible obstacles, but not always all causes. This was either for semantic reasons (preferring one option over another despite identical relationships[10]), or could be traced back to the intuition that when stopping an `or`/`xor` child you do not want to also stop the parent, echoing our findings for the helpfulness ratings.

To incorporate this intuition into our framework we follow Grice's maxim of quality [5], which refers to people tending to make their utterances exactly as informative as necessary, not more. It is implied, therefore, that when you say you want to stop a more specific child action you do *not* want to stop the parent. For example, indicating wanting to stop driving to work as a desired change implies that one does *not* want to stop going to work (its parent). We account for this by adjusting the behavior model (Definition 4.1): whenever *DC* includes stopping a child of an `or`- decomposition or a `xor`-decomposition, and the parent is not in *DC*, then we add the parent to the *DC* as a keep.

Definition 4.1 (Grice-adjusted behavior model) *Let* $D = (N, AMB, E, \lambda, \mu)$ *be a Behavior Diagram and BM be a Behavior Model* $BM = (D, AB, DC, S)$*. We define a Grice-adjusted behavior Model* BM^g *as* $BM^g = (D, AB, DC', S)$ *where* $DC' = DC \cup \{(n', \mathsf{keep}) \mid n \in AB \wedge (n, \mathsf{change}) \in DC \wedge (n', n) \in E \wedge \lambda(n) \in \{\texttt{or}, \texttt{xor}\} \wedge n' \notin \mathrm{domain}(DC)\}$.

[10]For example for the scenario: *"Imagine a situation where you go to visit your cousin every weekend. You have three different ways of getting there: by bike, bus, or by taxi. Currently, you always go by taxi. You want the system to help you stop taking the taxi to visit your cousin, since you want to save cost."* People had a preference for 'start biking' over 'start taking the bus' or both, probably reasoning that biking was cheaper than the bus. This type of semantic information is not captured in our formalization.

5. Discussion & Conclusion

The main contribution of this work is a formal framework with which behavior change support AI can represent and reason about a user's wish for change in the context of their behavior. This framework adds the reasoning capabilities required to reason about not only which changes are desired, but also what additional changes would be relevant to achieve this. Our relevant change model is built on the premise that the causes of and possible obstacles to a change are relevant, which is confirmed by a user survey. This work provides a foundation for more flexible, personalized behavior change support. Rather than a tool to help with one thing, we envision behavior change support systems which can support a range of different changes that users might wish to make.

Process of flexible support: Though modeling the wish for change is a crucial aspect of flexible support, it is not the only step in behavior change. In the following paragraphs, we will contextualize our contribution within the larger change process. The trans-theoretical model of behavior change models the different stages of change [18]. Our framework is specifically aimed at modeling the change goal of people who require help to get from determination to the maintenance stage via action. This means that people already are sure about a change, but need help putting this into practise. However, technology might also itself try to understand what the user wishes to change (for instance based on values, as in [3,20]).

After understanding comes the actual support. We argue that it would be possible to take the output of our framework and in a next step automatically identify how to support these changes. There are several elements which contribute to a change in behavior such as motivation and capability [4]. These contributors to change are individual and, therefore, would need to be elicited from a user for each desired change. Michie *et al.* show that such contributors to change can be linked to the types of interventions likely to be successful [15,14]. Based on the models that our framework provides, a system could elicit the contributors to change for the modelled changes and select the corresponding options for support based on the theories described. A next step would be to monitor whether the change actually happens, for instance through behavior monitoring [17], and temporal behavior reasoning [9].

The type of flexible support as described in this paper relies on the assumption that the information modeled can be acquired. We assume that usually the user knows their own and possible behavior, as well as what they wish to change. Therefore, asking the user about their actions and behaviors might be a good and transparent solution, e.g. in [1]. However, some aspects of current behavior might also be learned, e.g. [19]. Uncertainty or vagueness might be incorporated as in [9]. The model might also not be static, but rather have to be updated over time if possible behavior (e.g. after buying a car) or desired changes evolve.

Conclusion: In conclusion, the framework presented in this paper provides a way for an AI agent to model a user's wish for behavior change in the context of their possible and actual behavior. Additionally, we provide a formalisation of the possible obstacles to and causes of a change. In a user survey, we show that this method identifies intuitive relevant changes. This work provides a foundation for behavior change systems to adapt their behavior to the specific needs of each individual user.

April 2022

Acknowledgements This work is part of the research programme CoreSAEP, with project number 639.022.416, which is financed by the Netherlands Organisation for Scientific Research (NWO).

References

[1] BERKA, J., BALATA, J., JONKER, C. M., MIKOVEC, Z., VAN RIEMSDIJK, M. B., AND TIELMAN, M. L. Misalignment in semantic user model elicitation via conversational agents: A case study in navigation support for visually impaired people. *International Journal of Human–Computer Interaction 0*, 0 (2022), 1–17.

[2] BERKMAN, E. T. The neuroscience of goals and behavior change. *Consult Psychol J. 70* (2018), 28–44.

[3] CRANEFIELD, S., WINIKOFF, M., DIGNUM, V., AND 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, Melbourne, Australia, August 19-25, 2017* (2017), C. Sierra, Ed., ijcai.org, pp. 178–184.

[4] FOGG, B. J. A behavior model for persuasive design. In *Persuasive Technology, Fourth International Conference, PERSUASIVE 2009, Claremont, California, USA, April 26-29, 2009. Proceedings* (2009), S. Chatterjee and P. Dev, Eds., vol. 350 of *ACM International Conference Proceeding Series*, ACM, p. 40.

[5] GRICE, H. P. Logic and conversation. In *Syntax and Semantics volume 3: Speech Acts*, P. Cole and J. Morgan, Eds. Academic Press, New York, 1975.

[6] HINDRIKS, K. V., DE BOER, F. S., VAN DER HOEK, W., AND MEYER, J. C. Agent programming with declarative goals. In *Intelligent Agents VII. Agent Theories Architectures and Languages, 7th International Workshop, ATAL 2000, Boston, MA, USA, July 7-9, 2000, Proceedings* (2000), C. Castelfranchi and Y. Lespérance, Eds., vol. 1986 of *Lecture Notes in Computer Science*, Springer, pp. 228–243.

[7] HINDRIKS, K. V., VAN DER HOEK, W., AND VAN RIEMSDIJK, M. B. Agent programming with temporally extended goals. In *8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Budapest, Hungary, May 10-15, 2009, Volume 1* (2009), C. Sierra, C. Castelfranchi, K. S. Decker, and J. S. Sichman, Eds., IFAAMAS, pp. 137–144.

[8] KHAN, M., FERNANDES, G., AND MAES, P. Users want diverse, multiple, and personalized behavior change support: Need-finding survey. In *Persuasive Technology* (Cham, 2021), R. Ali, B. Lugrin, and F. Charles, Eds., Springer International Publishing, pp. 245–255.

[9] KLIESS, M. S., STOELINGA, M., AND VAN RIEMSDIJK, M. B. From good intentions to behaviour change. In *PRIMA 2019: Principles and Practice of Multi-Agent Systems* (2019), Springer International Publishing, pp. 354–369.

[10] KOLENIK, T., AND GAMS, M. Intelligent cognitive assistants for attitude and behavior change support in mental health: State-of-the-art technical review. *Electronics 10*, 11 (2021).

[11] KOROLEVA, K., MELENHORST, M., NOVAK, J., HERRERA GONZALEZ, S. L., FRATERNALI, P., AND RIZZOLI, A. E. Designing an integrated socio-technical behaviour change system for energy saving. *Energy Informatics 2*, 1 (2019), 30.

[12] KULYK, O. A., OP DEN AKKER, R., KLAASSEN, R., AND GEMERT-PIJNEN, L. V. Personalized virtual coaching for lifestyle support: Principles for design and evaluation. *Advances in life sciences 6* (2014), 300–309.

[13] MICHIE, S., ATKINS, L., AND WEST, R. *The Behaviour Change Wheel: A Guide To Designing Intervention*. Silverback Publishing, 2014.

[14] MICHIE, S., RICHARDSON, M., JOHNSTON, M., ABRAHAM, C., FRANCIS, J., HARDEMAN, W., ECCLES, M. P., CANE, J., AND WOOD, C. E. The behavior change technique taxonomy (v1) of 93 hierarchically clustered techniques: Building an international consensus for the reporting of behavior change interventions. *Annals of Behavioral Medicine 46* (2013), 81–95.

[15] MICHIE, S., VAN STRALEN, M. M., AND WEST, R. The behaviour change wheel: A new method for characterising and designing behaviour change intervention. *Implementation Science 6* (2011), n.42.

[16] OINAS-KUKKONEN, H. Behavior change support systems: A research model and agenda. In *Persuasive Technology, 5th International Conference, PERSUASIVE 2010, Copenhagen, Denmark, June 7-10, 2010. Proceedings* (2010), T. Ploug, P. F. V. Hasle, and H. Oinas-Kukkonen, Eds., vol. 6137 of *Lecture Notes in Computer Science*, Springer, pp. 4–14.

[17] PRIOLEAU, T., II, E. M., AND GHOVANLOO, M. Unobtrusive and wearable systems for automatic dietary monitoring. *IEEE Transactions on Biomedical Engineering 64* (2017), 2075–2089.

[18] PROCHASKA, J. O., AND DICLEMENTE, C. C. The transtheoretical approach. In *Handbook of Psychotherapy Integration*, J. Norcross and M. Goldfried, Eds. Oxford University Press, 2005, pp. 147–171.

[19] SCHÜTZ, N., KNOBEL, S. E., BOTROS, A., SINGLE, M., PAIS, B., SANTSCHI, V., GATICA-PEREZ, D., BULUSCHEK, P., URWYLER, P., GERBER, S. M., MÜRI, R. M., MOSIMANN, U. P., SANER, H., AND NEF, T. A systems approach towards remote health-monitoring in older adults: Introducing a zero-interaction digital exhaust. *npj Digital Medicine 5*, 0 (2022), 116.

[20] TIELMAN, M. L., JONKER, C. M., AND VAN RIEMSDIJK, M. B. What should I do? deriving norms from actions, values and context. In *Proceedings of the Tenth International Workshop Modelling and Reasoning in Context co-located with the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018) and the 23rd European Conference on Artificial Intelligence (ECAI 2018), Stockholm, Sweden, July 13, 2018* (2018), J. Cassens, R. Wegener, and A. Kofod-Petersen, Eds., vol. 2134 of *CEUR Workshop Proceedings*, CEUR-WS.org, pp. 35–40.

[21] TIELMAN, M. L., VAN RIEMSDIJK, M. B., AND WINIKOFF, M. Dataset belonging to the paper: What can i do to help you? Tech. rep., 2023. doi: https://www.doi.org/10.4121/22099832.

[22] VAN RIEMSDIJK, M. B., JONKER, C. M., AND LESSER, V. R. Creating socially adaptive electronic partners: Interaction, reasoning and ethical challenges. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015* (2015), G. Weiss, P. Yolum, R. H. Bordini, and E. Elkind, Eds., ACM, pp. 1201–1206.

[23] WINIKOFF, M., PADGHAM, L., HARLAND, J., AND THANGARAJAH, J. Declarative & procedural goals in intelligent agent systems. In *Proceedings of the Eights International Conference on Principles and Knowledge Representation and Reasoning (KR-02), Toulouse, France, April 22-25, 2002* (2002), D. Fensel, F. Giunchiglia, D. L. McGuinness, and M. Williams, Eds., Morgan Kaufmann, pp. 470–481.

[24] YARDLEY, L., CHOUDHURY, T., PATRICK, K., AND MICHIE, S. Current issues and future directions for research into digital behavior change interventions. *American Journal of Preventive Medicine 51*, 5 (2016), 814 – 815.