Representing human habits: towards a habit support agent

Pietro Pasotti[1], **M. Birna van Riemsdijk**[2], **Catholijn M. Jonker**[3]

Abstract. Human behaviour is constrained by obligations on the one hand, by the routines and habits that constitute our *normal* behaviour on the other. In this paper, we present the core knowledge structures of *HabInt*, a Socially Adaptive Electronic Partner that supports its user in trying to adopt, break or maintain habitual behaviours. We argue that *HabInt*'s role is best conceived of as that of an extended mind of the user. Hence, we pose as requirements that *HabInt*'s representation of the relevant aspects of the user and her world should ideally correspond to that of the user herself, and use the same vocabulary. Furthermore, the knowledge structures of *HabInt* should be flexible and explicitly represent both its user's actual habitual behaviours and her desired habitual behaviours. This paper presents knowledge structures that satisfy the aforementioned requirements. We interleave their syntactic specification with a case study to show their intended usage as well as their expressive power.

1 Introduction

Man is a creature of habit. While people display a fascinating variety of behaviours even across relatively simple domains, it is also true that from day to day most people are quite fixed in their ways. Carrying out habitual activities is mostly unproblematic and even desirable ([29]). However at times unforeseen circumstances make our habitual choices unavailable or their outcomes undesirable. Other times we wish to adopt or break a habit, and both are difficult enterprises. While many of us normally have little or no difficulty in dealing with these challenges ([29]), the actual amount of nuisance is subjective. To some, even small disruptions of daily routines may cause anxiety and distress ([12, 18]), whereas for others, such as people suffering from depression, breaking habits can be beneficial ([24]). In this paper, we take the first steps in developing a concrete implementation of *HabInt*, a Socially Adaptive Electronic Partner ([31]) to support habit formation and breaking.

Our working definition of *habit* is the shared view among social psychologists in the tradition of Hull ([16]). They stress the Pavlovian nature of habits as goal-independent *learned associations between responses and features of performance contexts*.[4]

As argued in [32], a *habit* is not merely a *frequently performed behaviour*. A habit is best seen as a mental object *resulting from* repeatedly choosing the same behaviour when faced with the same choice in a stable context. A habit is thus an association between some fixed environmental (and temporal) cues and a learned response. The more a habitual behaviour consolidates, the more it acquires features including: a degree of *automaticity*; less need for attention/focus, so that it can be performed *concurrently* with other tasks; smaller emotional involvement; and finally a habit is *not goal-aware*: while typically consistent with one's goals, the goal it was originally directed at is no longer consciously pursued. (cfr. [35])

The two knowledge structures of *HabInt* that form the core of this paper are the Actual Behaviour Model (ABM) and the Desired Behaviour Model (DBM). The ABM encodes a set of overlapping chains of user activities and the ways in which they are typically performed. The DBM describes a 'contextually ideal' version of the ABM. The nodes of ABM are associated with information about the values they promote or demote, so that *HabInt* can keep track of the motivations behind the goals and construct a model of the user's preferences.

Section 2 describes the core aspects of *HabInt* architecture, and formulates the requirements for the knowledge structures representing actual and desired behaviour. The ABM is presented in § 3 and the DBM in § 4. In § 5 we give an overview of how *HabInt* can user the ABM and DBM information to monitor for various types of user anomalies. The related work is discussed in § 6. Finally, § 7, § 8 summarize our findings and point out directions for future work.

2 Habit support agent: what and how

This paper focuses on those structures of *HabInt* that represent the actual and desired behaviours of the user. The data they contain is accessed by a monitoring component which locates anomalies and hands them over to a module that determines what should the agent do to support the user, thereby closing the interaction loop (see Figure 1). By interacting with the user and monitoring her behaviour,

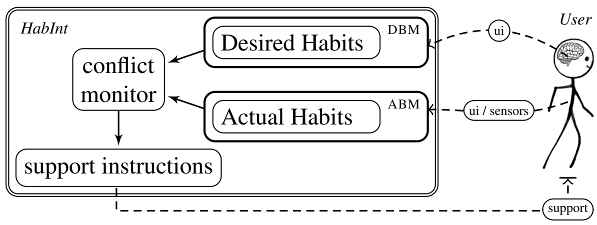

Figure 1. *HabInt*'s specific knowledge structures.

HabInt builds a model of the actual habits and typical activities of the user and a model of the user's desired habits. Desired habits are descriptions of behaviours which the user wants to turn into habits. These can be entirely new behaviours or changes to existing ones.

The monitoring module compares the user's desired and actual behaviour to detect conflicts: these are situations in which the user may need support. As a conflict is detected, a separate module that contains support instructions, previously provided by the user, is invoked. This module determines how should the agent intervene.

[1] TU-Delft,The Netherlands, email: P.Pasotti@tudelft.nl
[2] TU-Delft,The Netherlands, email: M.B.vanRiemsdijk@tudelft.nl
[3] TU-Delft,The Netherlands, email: c.m.jonker@tudelft.nl
[4] Cfr. [35] for an overview and further literature.

The ultimate goal of *HabInt* is to help the user achieve her goals and promote her values. In this sense, an implementation of a *HabInt* is best conceived of as part of the *extended mind* (see [9]) of its user. This means that it must be *trustworthy, reliable* and *accessible* (cfr. [9]) and so must be its knowledge. To make *HabInt* accessible and trustworthy, we must provide knowledge structures that are as transparent for the user as possible. The knowledge must be readily available for the user not only to use, but also to expand, contract and otherwise modify in a way that matches the way behaviours are discovered, explored, abandoned by people. To further enhance trust and reliance, we limit the agent's proactiveness to only those actions that are explicitly requested by the user. *Accessibility*, for one, means that the information/knowledge in the system must be easily accessible by the user. Consequently the *HabInt* has to store and manipulate its user model explicitly (unlike, for example, a neural network). For another thing, the user model should match the one the user has of herself as closely as possible, i.e. it should be a *shared mental model* (see [17]). Thus *HabInt* builds the vocabulary of goals, values, activities and actions from the user's wording. Summing up, *HabInt*'s knowledge structures should be:

adaptable: obtained by interacting with the user. This entails that they need to tolerate runtime updates and be built incrementally, while remaining meaningful at all intermediate stages of the construction process.

shared: correspond as much as possible to the user's conceptual structure and use the same vocabulary as the user does.[5]

explainable: *HabInt* needs to be able to carry out reasoning and explain the reasons that led to its current beliefs, in a dialogue referring to goals, values, and situational aspects ([30]). For example, *HabInt* should be able to model and then explain back to the user as requested which values are positively or negatively affected by some activities, to which goals activities contribute, and which values motivate which goals.

expressive: the structures need to accommodate uncertain, incomplete, and even inconsistent information. Finally, they must express the (context-dependent) behaviour enactment likelihood, for that is how *HabInt* can tell whether a behaviour is a habit or not, or whether it is becoming or ceasing to be one.

To show *HabInt*'s intended usage and the expressive power of its knowledge structures, we introduce a few snapshots of the life of a woman, Alice, as she interacts with *Hal*, her *HabInt*. Throughout the paper we will refer back to these scenarios and show how they are dealt with behind the scenes by *Hal*.

SCENARIOS: Alice and *Hal*

S1 Alice has a new job and would like to form a robust routine for travelling there. Also she would like to stop oversleeping. To help her with these issues Alice buys an *HabInt*, which she calls *Hal*. After booting it, Alice explains that she has two goals: first, to *'wake up'* and then to *'get to work'*. *Hal* asks what the options regarding the two goals are. It discovers that while there are a number of ways to get to the workplace, there is only one way of waking up, which requires remembering to set an alarm.

Alice explains to *Hal* that the main ways of getting to work are 1) by car, and 2) by bike. Furthermore, one can go by bike in two ways, 2.1) via the fast but risky Route A, or 2.2) via the safer, but longer Route B.

S2 Alice sometimes takes a cab to work. She feels no need for support in doing so, so when *Hal* reminds her to check the weather as she is leaving for work, she just says "well, actually today I'm going to work in some other way, so I won't need it. You don't need to worry about this." *HabInt* does not know how Alice is going to work that day.

S3 Alice now has the habit of setting the alarm every single day. However, exceptionally, on Mondays she forgets to set the alarm almost every other week (Probably this relates to her Sunday night's Vodka Tasting Club meetings).

S4 Alice tells *Hal* that of the two options to go to work by bike, she prefers the safe route (2.2) over the fast one (2.1). She explains that being fast is not as important to her as being safe.

S5 Alice asked *Hal* to help her grow the habit of going to work by bike. Years later, however, Alice decides to stop biking to work and go by car instead. Thus specific habitual behaviours part of her previous biking-to-work-routine are no longer necessary. She tells *Hal* the following: "(Instead of going by bike) now I'd like to go work by car", "I'll also need to stop taking the raincoat as I go to work."

S6 Alice long ago told *Hal* that she dislikes *'smoking'*, an action, because it demotes *'health'*, which she greatly values. Consequently, she has not smoked a single cigarette for 10 years now. However, one day *Hal* learns that Alice is smoking. After inquiring, *Hal* is told simply: "I *want* to start smoking."

3 The Actual Behaviour model (ABM)

There are habits regarding *what activities* we carry out daily; i.e. habits regarding, once something is done, what do we do *next* (`next`-*habits*). We model such activity patterns by capturing the sequential activation patterns of the goals that they purport to achieve.

Second, there are habits regarding *the way* in which we carry each activity out. We will call them `conc`-*habits*, for *concretisation habits*. The intuition is that just like the goal *get home by car* is intuitively more concrete that the goal *get home*, the activity of *driving home*, which achieves the former, is more concrete than the activity of *going home*, which achieves the latter. Achieving the former goal entails achieving the latter, but the converse does not hold. This *is-a-way-of* relation between goals is what we intend to capture with the notion of *concretisation*: we model habits regarding the way in which we do things by modelling the underlying goal concretisation patterns.

Finally, there are habits regarding *what actions* we perform as part of carrying out an activity (in a particular way): we call them *Action-habits*. For every activity, we represent the actions the user can perform when she tries to achieve its goal, and capture the likelihood that they are in fact performed.

In § 3.1 and § 3.2 we describe a knowledge representation language based on these three notions. Finally, exploiting our representation of the actions' consequences, we can express the values that they affect, and hence talk about the motivation and preferences that underlie behaviour choice and change. This is done in § 3.3.

The common basis of the language that the ABM is built upon is a language of alphanumeric strings. *HabInt* parses the User's messages at the level of propositional logic operators and treats the remaining uninterpreted strings as atoms. For example, "[the user is] not eating" becomes $\neg$*'eating'*. A propositional language over Strings[6] $\mathcal{L}_{\mathtt{str}}$ is the basis of the knowledge structures we define next. A logical consequence relation is defined on formulae of $\mathcal{L}_{\mathtt{str}}$ in the standard way. We use $\mathtt{a}, \mathtt{b}, \mathtt{l}$ as variables ranging over $\mathcal{L}_{\mathtt{str}}$.

3.1 Activities: what we do and how we do it

Abstracting away the temporal features for the sake of simplicity, an *Activity* is informally understood as *something which the user does to modify the current state of affairs*. Most of our daily activities

[5] I.e. if the user refers to its habit of 'brushing teeth after every meal', then that, literally, is the name *HabInt* stores.

[6] We capitalise technical terms, to avoid confusion with common concepts.

are carried out with a purpose, which we call a Goal. Our working definition of Goal is: *a declarative description of the state of affairs which the user would like to achieve by carrying out an Activity*.

HabInt's most fundamental knowledge structure represents the user's daily activities' underlying goals, and what goals are concretisations of what other goals. We call this knowledge the *Goal Base*.

The relation conc defines a branching structure of Goals that represents the way the user conceptualises her daily goals in terms of more concrete versions of themselves. This is captured by the binary relation conc. A special role is played by *toplevel Goals*, which are not a concretisation of any other Goal. In other terms, those for which the user sees no need to provide a higher goal. Examples for Alice include being awake early, having breakfast, getting to work and back home again. Toplevel Goals are then linked to one another by the relation next, forming a separate branching structure. This structure represents the user's potential Goal activation sequences: information about what she *might* do *after* doing something else.

Definition 1 (Goals and Goal Base) *The user's Goal Base is a triple* $\boldsymbol{G} := \langle G, \texttt{conc}, \texttt{next}\rangle$*, where:*

- $G \subseteq \mathcal{L}_{str}$ *is the current set of Goals of the user, with typical variables* g *and* g'.
- $\texttt{conc} : G \times G$ *is a* directed and acyclic concretisation relation*, such that* $\forall g, g' \in G : \texttt{conc}(g, g')$ *iff* g' *is a concretisation of* g*.*
- $\texttt{next} := \texttt{top}_{\boldsymbol{G}} \times \texttt{top}_{\boldsymbol{G}}$ *is a* directed, acyclic *relation such that* $\texttt{next}(g, g')$ *if once she has satisfied goal* g*, the user may try to satisfy (i.e. adopt)* g' *next.*

Furthermore, $\texttt{top}_{\boldsymbol{G}}$ *is the set of* toplevel goals *of* $\boldsymbol{G}$*:*

$$\texttt{top}_{\boldsymbol{G}} := \{g \in G \mid \forall g' \in G(\neg\texttt{conc}(g', g))\}$$

Note that the user can specify as many Goals as she likes, and can leave gaps and blanks. So, even if she habitually smokes at home, her *HabInt* may never know. It is up to the user to inform *HabInt* of alternative activities for the goals she mentioned to it, even if the user leaves these *under*specified. Hence *HabInt* must assume that unknown, additional alternatives always exist.

By monitoring and interacting with the user, *HabInt* learns and keeps track of the Activities that she carries out as she tries to achieve her goals, and of the sequences of actions that compose these Activities. All the actions *HabInt* is aware of are kept track of in the Action Base. The way we express Actions is standard practice:

Definition 2 (Actions and Action Base) $\mathcal{L}_{act}$ *is the language of actions, which are defined as formulae over* $\mathcal{L}_{str}$ *of the form*

$$\alpha \in \mathcal{L}_{act} := [\texttt{l}\text{: } \texttt{a} \twoheadrightarrow \texttt{b}]$$

where l *is the* name*,* a *the* precondition *and* b *the* postcondition *of the Action. The Action Base* $\boldsymbol{C} \subseteq \mathcal{L}_{act}$ *is the current set of Actions.*

By making the pre- and postcondition more detailed, the agent can represent each of the user's Actions in more or less detail, as well as specify the way in which they can be sequentially executed. We assume in what follows that *HabInt* has a planning module enabling it to reason about how to chain Actions together based on this (technicalities omitted due to lack of space).

Activities group up actions that can be performed as part of achieving some goal, and assign a name to the full bundle. If the *Act* field of an Activity is empty, that means that either performing that Activity is obvious enough to the user (i.e., she needs no support on that) or that she does not know yet. In that case, all an Activity does is associate a declarative goal with an informal (and meaningless to *HabInt*) description of a possible way to achieve it. All known Activities are stored in the Activity Base.

Definition 3 (Activities and Activity Base) *Given a set of Actions from the Action Base* $Act \subseteq \boldsymbol{C}$*, a goal* g *from the Goal Base* $\boldsymbol{G}$*, and a name* $\texttt{l} \in \mathcal{L}_{str}$*, an* Activity $\mathcal{A} \in \mathcal{L}_{uac}$ *is a tuple of the form:* $\langle \texttt{l}, g, Act\rangle$*.* l *is the name,* g *the goal, and* Act *the set of actions of the Activity. The* Activity Base $\boldsymbol{A} \subseteq \mathcal{L}_{uac}$ *is the current set of Activities.*

Through the Actions that compose them, Activities, too, can be made more or less fine-grained. Activities whose Goals are toplevel encode those activities that the user perceives as being self-justified or motivated by some of her values.

These Goal-Activity structures can be viewed as a variant of Goal-Plan Trees [27] where the conc-relation corresponds to OR-nodes, AND-nodes are left implicit, and distinct GPTs can be connected by the next relation.

> **[Behind the scenes of S1]** *Hal* performs natural language analysis and determines that Alice's utterances mean the following: Alice wants support with two activities: going from home to work and waking up. The corresponding Goals are *'is awake'*, ①, and *'is at work'*, ②, respectively. Furthermore, *'go by bike'* and *'go by car'*, are names of activities whose corresponding declarative goals are *'is at work'* ∧ *'biked to work'*, ③, and *'is at work'* ∧ *'drove to work'*, ④. It has also recorded how going by bike/by car are ways of going to work, but going to work seems not to be a way to do something else, and is thus toplevel. The ABM is now as in Figure 2.
>
> When Alice mentions how waking up requires having set the alarm, an Action α achieving ② (i.e. with ② as postcondition) is specified, which requires *'alarm set'* to be true. Formally, α = [*'wake up'*: *'alarm set'* ↠ *'is awake'*]. Now $\mathcal{A}$ is ⟨*'waking up'*, *'is awake'*, {α}⟩.

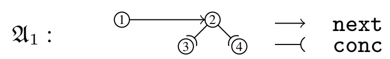

Figure 2. Hal's ABM of Alice.

In a nutshell, the construction process of the ABM is as follows: first, the user specifies a number of goals and whether each goal stands in a conc or next relation to some other known goal. Secondly, the user gives the names of activities that can achieve that goal, and, finally, she can describe the relevant actions that take part in carrying out each activity. In this way, the user determines what is an appropriate amount of specificity.

> **[Behind the scenes of S2]** *Hal* learns that Alice does go to work, but neither by car nor bike. Therefore *Hal* records a new Activity $\mathcal{A}$, whose goal g_0 (a novel placeholder) is a concretisation of *'at work'*. $\mathcal{A}$ is named *'unknown alternative'*. Maybe one day Alice will tell *Hal* that she is going to work by *'take[ing] a cab'*. If so, *Hal* will update its knowledge structures.

The above shows that the knowledge structures are expressive, and explainable in that by manipulating directly the utterances (as strings) *HabInt* can maintain a model using the same vocabulary as the user. Furthermore, it is straightforward to define update operations such as splitting an Activity into two sub-Activities, adding/removing Actions, and splitting Actions into longer chains.

3.2 Habits: the way we normally do what we do

As we mentioned in § 1, a habit is not just a "frequent behaviour". However, frequency, automatism, ease of performance and other features of habitual behaviours are correlated. in particular frequency can serve as a *predictor* for the other features and it can be derived from observing and communicating with the user ([35]). Therefore, we chose to detect habits through the underlying behaviours' enactment likelihoods. We describe a user's day as a sequence of toplevel goals (given by next). Each of those can then be concretised in different ways (as described by conc), and each goal can be assigned, via an Activity, a set of Actions that can be executed whilst achieving it. This is information about what the user is known to sometimes do: it defines the space of possible *behaviours*. Each one of these may in practice be enacted rarely or never, and both their content and their performance frequency can change over time. Consequently, we keep the representation of what the user knows she may do (next, conc, Activities) separate from the expectations regarding what she *will* do.

We have seen in § 1 that habits are cued by contexts. Hence we must keep track of those parts of the context that are believed by the user (or by some internal learning algorithm) to cue some behavioural response. We call them *Triggers*. Let $\mathbf{T} \subseteq \mathcal{L}_{\mathtt{str}}$ be a *finite set of known Trigger*. Let τ range over $\mathbf{T}$.

However, reacting to Triggers is not automatic: even in the presence of a Trigger the cued behaviour may not follow. Then we must record, given the presence of a Trigger, the likelihood of the associated behaviour occurring. This is captured by prob. Formally, prob is a function of type $(\mathbf{T} \times G \times (G \cup \mathcal{L}_{\mathtt{act}})) \rightarrowtail [0, 1]$. Intuitively:

$\mathtt{prob}(g'|\tau, g) = x$, for toplevel goals g and g', means that *given* that g has been *just achieved* and that the Trigger τ holds, then the user adopts g' with likelihood x. If g' is not toplevel, then it means that *while* she tries to achieve g, and given Trigger τ the user is expected to *adopt* g' with likelihood $x \in [0, 1]$.

$\mathtt{prob}(\alpha|\tau, g) = x$ means that if the goal g is adopted and τ holds, then the user is expected to execute action α with likelihood x.

Some behaviours' Triggers can be unknown, or so frequent to be irrelevant. In that case the Trigger is true.

If, given a Trigger, the enactment likelihood of some behaviour is above a certain threshold $t \in [0, 1]$, *HabInt* infers that the behaviour is a *habit*. We call t the user's *habit threshold*, which is the performance likelihood above which the user feels confident in calling something a *habit*. Given existing research (e.g. [36]), it is reasonable to assume that $t > 0.5$. The return values of prob are estimated based on information from the user and/or sensor data (cf. [20]). Now we must keep in mind a key property of the notion of Goal we employ here: Goals that are concretisations of the same goal cannot be adopted concurrently. While this is not generally true for next-related Goals, here for simplicity we assume it is.[7] For example, after waking up, Alice can *go to work* or *go to the beach*, not both. Also, as she goes to work, Alice can go *'by bike'* or *'by car'*, not both. Therefore, no matter how prob is calculated, the likelihoods must sum up to one on all outgoing next paths and on all outgoing conc paths too. Actions are executed independently from one another, so there is no such constraint there.

The data structures we have defined so far allow us to express the types of habits described at the beginning of this section: next-, conc- and action-habits. These correspond to transitions between next-related goals, conc-related goals, and $\langle g, \alpha \rangle$ pairs respectively.

[7] So at any given moment the user can adopt at most one toplevel goal g, and an arbitrarily long conc chain of Goals with g at one end.

Definition 4 (Habits) *$\forall\, g, g' \in \boldsymbol{G}$, $\alpha \in \mathcal{L}_{act}$, $\tau \in \boldsymbol{T}$, and given a habit threshold $t \in [0, 1]$, we define:*

$$\mathtt{hab}(g'|\tau, g) \iff \begin{cases} \mathtt{next}(g, g') \text{ and } \mathtt{prob}(g'|\tau, g) > t \quad \textit{or} \\ \mathtt{conc}(g, g') \text{ and } \mathtt{prob}(g'|\tau, g) > t \end{cases}$$

$$\mathtt{hab}(\alpha|\tau, g) \iff \mathtt{prob}(\alpha|\tau, g) > t$$

Intuitively, if g' is toplevel, $\mathtt{hab}(g'|\tau, g)$ means that the user habitually *adopts* g' given that g has been *achieved* immediately before, and that τ holds: a next-habit. If g' is not toplevel, $\mathtt{hab}(g'|\tau, g)$ means that given that g is *adopted* and τ holds, the user habitually *adopts* g' *too*. In other words, as g' is a concretisation of g, the user *habitually tries to achieve* g *by achieving* g': a conc-habit. $\mathtt{hab}(\alpha|\tau, g)$, finally, means that the user habitually executes α *given that she has adopted* g and τ holds: an Action-habit.

[Behind the scenes of S3] Suppose that *Hal* knows that Alice's habit threshold t is 0.89. *Hal* then updates its ABM to reflect how her now-established habit of setting the alarm (the Action α) is endangered if the Trigger *'monday'* is present: while under no Trigger the behaviour has likelihood 0.9, when *'monday'* is the case it goes down to 0.6 (*HabInt* sets these values through interaction or monitoring).

$$\mathtt{prob}(\alpha|\mathtt{true}, \textit{'wake up'}) = 0.9$$

$$\mathtt{prob}(\alpha|\textit{'monday'}, \textit{'wake up'}) = 0.6$$

3.3 Values: why we do what we do as we do it

Even though a *HabInt* having only the above structures can already be of use, in our opinion it needs to understand the motivations (based on values) for the choices the user makes to best support her. The user may need support in making satisfactory choices regarding her behaviour, which involves comparing competing Actions, based on their outcomes; competing Activities, based on the Actions they are associated with; and competing Goals, based on the Activities that they can be achieved by. For supporting the user, *HabInt* needs to understand and reason with the motives of the user's behaviour, and thus needs to know and understand her values. Paraphrasing [19],

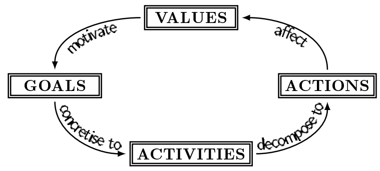

Figure 3. How values close the concretisation loop.

we understand *values* as *a hierarchy of desirable, abstract, cross-situational goals*. Ultimately, any activity is motivated by the pursuit of values. Still, all actions that we take as part of any activity end up affecting the same values (see Figure 3). So the user interface must be capable of value-based argumentation, and it is therefore natural to store also these knowledge structures in the unified world model we are describing here. As our *HabInt* is a *personal* support agent, here we assume that every user has her own (hierarchy of) values and hence we ignore their often-alleged universality ([23]). With the help of the user, *HabInt* learns what values she has, how important they are relative to each other, and what world features (literals from $\mathcal{L}_{\mathtt{str}}$) can affect them. *HabInt* reasons about Values using value-based argumentation frameworks (cfr. [5]).

Definition 5 (Values and Value Base) *We define* $\boldsymbol{V} := \langle V, \lhd, \texttt{pro} \rangle$ *to be the Value Base of the user, where*

- $V \subseteq G$ *denotes the set of given Values of the user.*
- $\lhd \subseteq V \times V$ *is a preorder, such that* $\forall v, v' \in V : v \lhd v'$ *holds if* v is less important than v'.
- $\texttt{pro} := \mathcal{L}_{str} \times V \rightarrowtail \{\downarrow, \nmid, \uparrow\}$ *is an* injective *function encoding the way literals* a *from* $\mathcal{L}_{str}$ *promote (*$\uparrow$*), demote (*$\downarrow$*) or not affect (*$\nmid$*) the user's values.*

Note that the default return value of the function pro is $\nmid$: we assume that the user does not know or does not care, until she says otherwise.

When the user and her *HabInt* are reasoning about the best course of action to take, the postconditions of the actions involved play the fundamental role. Each postcondition expresses not only the goal its Action achieves but (in conjunctive normal form) a list of its effects. Exploiting this fact, *HabInt* can infer from the Value Base the way Actions first, then Activities affect Values.

As abstract goals of activities, values may well be unspecified and in the background.[8] But when it comes to evaluating the effects of the concrete Actions that together form an Activity, the importance and visibility of values become greater. Actions can be said to promote and demote values *by bringing about their postcondition* and, through the Actions that *habitually achieve them*, so can Activities. While Actions' outcomes are stable, habits dictate which Actions are executed when carrying out an Activity. Therefore, to determine what values are affected by an Activity, one must factor in habits.

With the Value Base, all parts of the ABM have been discussed.

Definition 6 (Actual Behaviour model (ABM)) *The* Actual Behaviour Model *is the tuple* $\mathfrak{A} := \langle \boldsymbol{V}, \boldsymbol{G}, \boldsymbol{A}, \boldsymbol{C}, \boldsymbol{T}, \texttt{prob}, t \rangle$*, where the elements are respectively, the Value Base, the Goal Base, the Activity Base, the Action Base, the set of Triggers, the conditional likelihood function, and the habit threshold.*

Given pro, which tells how $\mathcal{L}_{\texttt{str}}$ literals affect Values, we generalise it to $\texttt{pro}^*$, which also tells how Actions and Activities do.

Definition 7 (Promote) *Given an* ABM $\mathfrak{A}$*, the function* $\texttt{pro}^* := ((\mathcal{L}_{str} \cup \mathcal{L}_{act} \cup \mathcal{L}_{uac}) \times V) \rightarrowtail \{\uparrow, \downarrow, \nmid\}$ *is defined as follows:*

- *If* a *is a literal from* $\mathcal{L}_{str}$*, then* $\texttt{pro}^*(a, v) = \texttt{pro}(a, v)$.
- *If* $\varphi \in \mathcal{L}_{str}$*, then we require all the disjuncts to 'agree' on* v*:*

$$\texttt{pro}^*(\varphi \vee a, v) = \begin{cases} \texttt{pro}(a, v) & \text{if } \texttt{pro}^*(\varphi, v) = \texttt{pro}(a, v) \\ \nmid & \text{otherwise} \end{cases}$$

- *Let* α *be an action* $[\mathtt{l}\text{: } a \twoheadrightarrow b]$*, and cnf*($\alpha$) *denote the* set of b's conjunctive normal form's conjuncts *(with* $\Box \in \{\uparrow, \downarrow, \nmid\}$*). Given:*

$$C^{\alpha}_{\Box v} \quad := \quad \{\varphi \in \mathit{cnf}(\alpha) \,:\, \texttt{pro}^*(\varphi, v) = \Box\}$$
$$\texttt{pro}^*(\alpha, v) = \uparrow \quad \mathit{iff} \quad |C^{\alpha}_{\uparrow v}| > |C^{\alpha}_{\downarrow v}| \qquad (1)$$

similar to [33], we say that α *promotes a Value* v *(*$\texttt{pro}^*(\alpha, v) = \uparrow$*) if it brings about more* v*-promoting than* v*-demoting postcondition. The conditions for* $\texttt{pro}^*(\alpha, v) = \downarrow$ *or* $= \nmid$ *are very similar: change '*$>$*' to '*$<$*' and '*$=$*' in* (1) *respectively.*
- *Let* $\mathcal{A} = \langle \mathtt{l}, g, \mathit{Act} \rangle$*, and* $h(\mathcal{A}) := \{\alpha \in \mathit{Act} \mid \exists \tau, g : \texttt{hab}(\alpha | \tau, g)$ *holds in* $\mathfrak{A}\}$*; then*

$$D^{\mathcal{A}}_{\Box v} \quad := \quad \{\alpha \in h(\mathcal{A}) \,:\, \texttt{pro}^*(\alpha, v) = \Box\}$$
$$\texttt{pro}^*(\mathcal{A}, v) = \uparrow \quad \mathit{iff} \quad |D^{\mathcal{A}}_{\uparrow v}| > |D^{\mathcal{A}}_{\downarrow v}|$$

$\texttt{pro}^*(\mathcal{A}, v) = \uparrow$ *means that the activity* $\mathcal{A}$ *promotes* v*. The conditions for demoting or not affecting* v *are again very similar.*

This is crucial for *HabInt* to represent inconsistencies between what the user does, or wishes to do, and his Values (cf. § 5 for an example).

> **[Behind the scenes of S4]** *Hal* learns that Alice considers *'be safe'* (v_1) and *'be fast'* (v_2) as Values. Hence, it adds them to its previously empty Value Base, which now is $\mathbf{V} = \langle \{v_1, v_2\}, \varnothing, \varnothing \rangle$. Then it learns that biking through Route A (an Activity $\mathcal{A}$) promotes *'safety'*, but it does not know what specific postcondition of what Action involved in the Activity promotes it. Hence, first *Hal* adds a dummy Action $\alpha = [$*'something'*: *'something'* $\twoheadrightarrow$ *'*a_0*'*$]$ to $\mathcal{A}$ (where *'*a_0*'* is a new atom), and then adds to its Value Base the fact that pro(*'*a_0*'*, *'safety'*) $= \uparrow$. Via the same process, it also records that Route B promotes *'be fast'*. From this, *Hal* can deduce that $\texttt{pro}^*(\mathcal{A}$, *'safety'*). Finally, it learns: *'be fast'* $\lhd$ *'be safe'*. (Actually things are a bit more complicated, as promoting *'safety'* seems to be a property the Activity *always has*, according to Alice. Hence, by chaining post- and pre-conditions appropriately, *Hal* must ensure that α is presumed executed by the user *every time* the Activity of *'biking through route A'* is performed.)

4 The Desired Behaviour model (DBM)

People that are not quite satisfied with their actual behaviour may tell their *HabInt* what is bothering them. Only then, can they describe how they would like to be supported in changing it.

While the ABM of a user describes what the user does (and how she does it) in specific situations, the Desired Behaviour Model (DBM) describes a set of *Desired Habits* to the ABM that reflect how the user would like her ABM to become. The key intuition here is that if conforming to a desired behaviour were not an issue under any circumstance, the user would not mention it to *HabInt*. Therefore, *HabInt* treats each Desired Habit as *a support request*, which still does not convey any information about how the agent can in practice support the user. Later on, each Desired Habit can be linked to one or multiple ways in which the agent can support the user: for instance, instructions of when and how to produce a reminder, initiate a conversation, monitor some environmental variable, or ask what is going on. However, we do not discuss these in this paper.

In what follows, $\mathfrak{A}$ is an ABM, τ is a Trigger, g, g' are Goals, and α is an Action (all from $\mathfrak{A}$). We consider Desired Habits of three types:

next-Desired Habits are structures of the form $\langle \tau, g, g' \rangle$, where $\texttt{next}(g, g')$ is part of the Goal Base of $\mathfrak{A}$. This Desired Habit type formalizes the user's desires concerning her toplevel goal sequences. When she talks about what she should or would prefer to habitually do after doing something else, *HabInt* will formalise that as a next-Desired Habit.

conc-Desired Habits are structures of the form $\langle \tau, g, g' \rangle$, where $\texttt{conc}(g, g')$. This Desired Habit formalizes habit change desires concerning the way the user achieves some goal (i.e. her concretisation patterns). conc-Desired Habits formalise, for example, the user's desired habitual way of achieving some toplevel Goal.

Action-Desired Habits are structures of the form $\langle \tau, g, \alpha \rangle$, where there is some activity $\mathcal{A} = \langle \mathtt{l}, g, \mathit{Act} \rangle$ in $\mathfrak{A}$'s Activity Base with $\alpha \in \mathit{Act}$. If the user wishes to change the actions she habitually performs as part of carrying out some Activity, that will be formalised as an Action-Desired Habit.

In a similar fashion we introduce *undesired* behaviours or the habits which the user wants to drop. We call them *Undesired Habits*. They are also expressed in $\mathcal{L}_{\texttt{amd}}$ but stored in a different set, Undhab. Each Undesired Habit encodes the user's *desire to habitually not* enact a behaviour (in some way) or perform an action (given some

[8] Think about the habitual activity of going back home (after a day of work). The user can, but does not need to specify which values that macroscopic activity promotes.

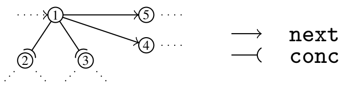

Figure 4. An example Goal structure. The Goal ① has two concretisations, ② and ③. Also, after achieving ①, the user can try to achieve either ④ or ⑤.

Trigger). The only constraint we impose is that `Undhab` and `Dhab` be *disjoint*, for obvious reasons.

Definition 8 (Desired Habits and Undesired Habits of the ABM) *Given an* ABM $\mathfrak{A} = \langle \boldsymbol{V}, \boldsymbol{G}, \boldsymbol{A}, \boldsymbol{C}, \boldsymbol{T}, \texttt{prob}, t\rangle$*, the set of* Desired Habits *is* $\texttt{Dhab} \subseteq \mathcal{L}_{dhab}$*, and the set of her* Undesired Habits *is* $\texttt{Undhab} \subseteq \mathcal{L}_{dhab}$*, where (*$g, g'$ *are Goals in* $\boldsymbol{G}$*,* $\tau \in \boldsymbol{T}$ *and* $\alpha \in \boldsymbol{C}$*):*

$$\mathcal{L}_{dhab} := \langle \tau, g, g' \rangle \mid \langle \tau, g, \alpha \rangle$$

With the difference between `next`- and `conc`-Desired Habits in mind, we can clarify their intended semantics by specifying the conditions under which they can be said to be complied with. A Desired Habit points out a behaviour which *should be habitual* under some trigger; hence a Desired Habit is complied with when that behaviour is indeed a habit (under that trigger). Similarly, a Undesired Habit is complied with when the corresponding behaviour is *not a habit*.

Definition 9 (Compliance) *Given a habit threshold* t *and an* ABM $\mathfrak{A}$*, we say that* $\mathfrak{A}$ *complies with*

$$\langle \tau, g, g' \rangle \in \texttt{Dhab} \quad \textit{iff} \quad \texttt{prob}(g' | \tau, g) > t \qquad (2)$$
$$\langle \tau, g, \alpha \rangle \in \texttt{Dhab} \quad \textit{iff} \quad \texttt{prob}(\alpha | \tau, g) > t \qquad (3)$$
$$\langle \tau, g, g' \rangle \in \texttt{Undhab} \quad \textit{iff} \quad \texttt{prob}(g' | \tau, g) < t \qquad (4)$$
$$\langle \tau, g, \alpha \rangle \in \texttt{Undhab} \quad \textit{iff} \quad \texttt{prob}(\alpha | \tau, g) < t \qquad (5)$$

Based on the known Triggers, *HabInt* keeps track of what behaviours the user wishes to change and stores them in its `Dhab` and `Undhab`. The `Dhab`, `Undhab` and ABM constitute the DBM.

Definition 10 (Desired Behaviour Model) *Given the* ABM $\mathfrak{A}$*, the Desired Habits* `Dhab`*, and the Undesired Habits* `Undhab`*, the* Desired Behaviour Model *is* $\langle \mathfrak{A}, \texttt{Dhab}, \texttt{Undhab} \rangle$.

[Behind the scenes of S5] Initially, `Undhab` is empty. However: $\texttt{Dhab} = \{\langle \texttt{true}, \textit{'at work'}, \textit{'biked'} \rangle\}$, because Alice originally wanted to form the habit of biking to work.

When Alice changes her mind, *Hal* firstly has to move $\langle \texttt{true}, \textit{'at work'}, \textit{'biked'} \rangle$ from `Dhab` to `Undhab`. Then, *Hal* formalizes Alice's desire to drive to work as the `conc`-Desired Habit: $\langle \textit{'rain'}, \textit{'at work'}, \textit{'drove'} \rangle$ and *adds it to* `Dhab`. Now *Hal* knows:

$$\texttt{Dhab} = \{\langle \texttt{true}, \textit{'at work'}, \textit{'drove'} \rangle\} \qquad (6)$$
$$\texttt{Undhab} = \{\langle \texttt{true}, \textit{'at work'}, \textit{'biked'} \rangle\} \qquad (7)$$

Since Alice now also wants to drop the habit of "getting the raincoat" as she leaves for work (an Action $\alpha = [\textit{'get raincoat'}\!: \textit{'at home'} \twoheadrightarrow \textit{'has raincoat'}]$), *Hal* has to further update `Undhab` to:

$$\texttt{Undhab} = \{\langle \texttt{true}, \textit{'at work'}, \textit{'biked'} \rangle, \langle \texttt{true}, \textit{'at work'}, \alpha \rangle\}$$

Other types of Desired Habits could in principle have been defined. For example, looking at Figure 4, one may wish to express `Dhab(true`, ②, ⑤). It could be read as requesting to form a habit of "instead of doing ① by means of ②, stop doing ① altogether and start doing ⑤ instead". But this is rather convoluted, and we see little added value. Rarely we say things like: "if you see me go to work *by bike*, remind me I should stay home instead". For similar reasons also the other possible `Dhab`-types require more far-fetched interpretations. So, we will not discuss them further.

5 Violation, anomaly, and inconsistency monitor

The structures we have described so far capture (un)desired habits, one-off behaviours, existing habits, and also the user's values, and *all can be at odds with one another*. Hence many types of conflict can be expressed in their language. Here we describe three: the most crucial ones to monitor for habit support. Namely we show that, given the DBM and ABM, *HabInt* can monitor whether an actual behaviour is *anomalous*, *inconsistent* with the user's value-based preferences or whether it *violates* an existing Desired Habit. The examples point out how *HabInt*'s monitoring module can check the user's ABM and DBM for such conflicts (all examples refer to Figure 4).

behavioural anomaly: when the user does something unusual (or in an unusual way). For example, when `hab`(τ, ①, ⑤), but the agent believes that the user is now doing ④ instead of ⑤.

When a behavioural anomaly is detected, *HabInt* can e.g. be instructed to investigate, remind the user of her habitual behaviour, or alert a supervisor. The ABM knowledge alone is sufficient for expressing this anomaly. Both ABM and DBM are needed, on the other hand, to express the following state of *violation*: when a Desired Habit is not a habit (or vice versa, when an Undesired one is).

$$\langle \tau, g, g' \rangle \in \texttt{Dhab} \wedge \langle \tau, g, g'' \rangle \notin \texttt{Dhab} \wedge \langle \tau, g, g'' \rangle \in \texttt{hab}$$

undesired behaviour: when the user does something (in a way) she declared she does not want to (or should not). For example, if ⟨τ, ①, ④⟩ ∈ `Dhab` ∧ ⟨τ, ①, ⑤⟩ ∉ `Dhab`, but the agent believes that the user habitually does ⑤ after ①, when τ.

When *undesired behaviour* is detected, this means that the user is doing something she declared she wanted support in *not doing* (or vice versa). Many kinds of support can be associated with violations of this type. For example the user may ask to be reminded of the values she invoked when she set the Desired Habit she is about to violate or to talk once more about the consequences of her behaviour. The same holds for one-off behaviours in place of habits.

Furthermore, using the notions introduced in § 3.3, *HabInt* can reason about which Values are affected by an Activity and know, for example, if an Activity $\mathcal{A}$ for the goal g demotes the user's most important values: $\forall v (\nexists v' (v \lhd v') \Rightarrow \texttt{pro}^*(\mathcal{A}, v) = \downarrow)$

If the user mentions that she is carrying out $\mathcal{A}$, or $\langle \tau, g', g \rangle \in \texttt{Dhab}$, then her *HabInt* will detect a *value inconsistency*:

value inconsistency: when the user's Actions, Activities, or (Un)Desired Habits are not in line with her preferences. For example, when an Action demotes an important Value.

[Behind the scenes of S6] When *Hal* perceives Alice smoking, its *behavioural anomaly* handling would instruct it to ask: "what is going on?". But at the same time, *Hal* thought that Alice disliked smoking, "because smoking demotes *'health'*", so this also categorises as an *undesired behaviour* (i.e. she might be falling into old bad habits) and has to be dealt with differently. So *Hal* asks instead: "is everything all right?" When Alice tells *Hal*: "I *want* to start smoking. Every day after lunch, for a start.", then *Hal* will have to handle a *value inconsistency*: either *'health'* is not *that* important (any more), or maybe the user has forgotten the values behind her previous choice.

For *HabInt*, anomalies, violations and inconsistencies mean either that the user is in trouble, or that its information is outdated. If an undesired behaviour violation is detected, then a sought-for habit change process may not be going smoothly, and the agent can e.g.

deliver a warning, as previously instructed by the user. The *value inconsistency* type of anomaly can be a symptom of inconsistencies in the user's motivation/intention/action structure, or irrational behaviour. To find out which one it is, *HabInt* can be instructed to initiate communication with the user.

6 Related work

Research on human-computer interaction has explored many ways in which technology can be used to aid behaviour change (e.g. [25]) and support habit formation and breaking. In contrast with these approaches, our focus is not on the psychological aspects of behaviour change in a specific domain and how to support this through technology. Rather, our conception of *HabInt* is as an extended mind of the user: we focus on developing generic knowledge structures that allow a *HabInt* to represent and construct user habits in a way that corresponds to the her conception of her activities.

The field of Activity Recognition has developed machine learning approaches to deduce what an observed human being is doing (and her behavioural patterns too), based on raw sensor data. For examples and further references, see [25, 11, 20]. These techniques will be used in the monitoring component, to update the `prob` function and automatically determine what the user is doing. This reduces the amount of information we need to get directly from the user. Research such as [8, 10] on (often neural network-like) learners that mimic the acquisition and monitoring of routine sequential action in humans is related, but does not satisfy most of the requirements of § 2 due to its different purpose. Our knowledge structures are at a higher level of abstraction, and capture the relations between activities as well as their motivations to enable user support on the basis of these higher-level concepts. However, the `prob`-based transition system is inspired to Markov Models [13].

Another area has investigated agents that form habits and routines of their own (see for example [2, 15]). Instead, our interest is in an agent that *supports* a human user in dealing with her habits. Knowledge structures oriented towards habit-learning agents need not be shared or explainable, and consequently the models are not explicit.

The challenge of developing support agents capable of dynamically interacting with humans in complex environments is not new (e.g., [4, 6, 37]). We share with them the general vision of a support agent, but the domain of support for dealing with individual habits was still unexplored. Secondly, while existing support agents in this tradition build on the notion of an agent whose primary goal is to take over some of the tasks a human has, the *HabInt* we propose has no such purpose. As a consequence, we face some different issues. For example, the issue of Adjustable Autonomy, as in [22], disappears, because *HabInt* does not take over human tasks or responsibilities but simply automates some of them *when requested*. On the other hand, the question of how to time the interventions remains.

In the area of multi-agent systems, knowledge structures for the representation of goals and actions have been extensively studied (e.g. [7]). In this paper we show how such structures can be used as a basis for the representation of habits. In future work we will investigate to what extent BDI languages can be used as a basis for implementing *HabInt*. A difference between *HabInt* and BDI agents is that agents *execute* plans themselves while *HabInt* represents a *user's* goals and activities in order to support the user in executing them. Thus a core challenge towards *HabInt*'s development will be to study how to these knowledge structures can be constructed in interaction with the user, and to develop further notions and reasoning techniques for habit support on their basis.

7 Discussion

The literature agrees on habits being no longer consciously goal-oriented: *awareness* of the goal has been lost in the process of habit formation and is no longer an explicit motive, but at most a latent justification. While in other situations the motive needs to be present in order to motivate action, habitual behaviours often lose sight of the motives as soon as they are no longer needed.[9] Clearly, turning carefully deliberated-upon choices into habits is a value-driven endeavour of its own, since in so doing we free up time and effort-consuming deliberations by crystallising them into automatic cue-response mechanisms (e.g. [21]). But by obscuring the values the behaviours' goals used to promote or demote, the process of habit formation risks making us blind to better choices and pitfalls alike. *HabInt* can help to counter this phenomenon by recording explicit representations of the values that are promoted or demoted by certain behaviours. The habit-formation process weakens awareness of how actions affect values and how values motivate goals (cfr. Figure 3) *HabInt* can help to close the loop, so that the values that are the motivation and purpose of habitual behaviour can be made visible again, revealing perhaps its normative aspects.

The dictionary definition of the word *norm* ([1]) includes: ***a)*** *A principle of right action binding upon the members of a group and serving to guide, control, or regulate proper and acceptable behaviour* ***b)*** *A widespread or usual practice, procedure, or custom.*

This paper focuses on habits, which fall under the second meaning of *norm*. We find it interesting that habits often conflict with norms of the first kind (e.g. see [28]), abiding by which often requires conscious effort and self-control. This is at odds with the absent-minded, automatic way we carry out our daily routines. We believe that the common ground of both meanings of *norm* can be found in *values*.

We also remark that the link between the two kinds of norms is even stronger than it may initially seem when observed under the perspective we have outlined in this paper. *HabInt* has a model of what *is* actually the case, the ABM, and a model of what *should be* the case, the DBM, which is the user's own idea of a better self (one that wakes up in time, brushes her teeth, etc...). This idea echoes the normative-descriptive dichotomy of economists and psychologists ([26, 3]) on which deontic logics can be built, as shown e.g. in [14].

8 Conclusion and future work

While performing habitual behaviours is characteristically easy, forming and breaking habits can be difficult. Our aim is to develop a *HabInt* agent that is able to support humans in such efforts. We conceive of such an agent as an extended mind of the user. In this paper we have presented the foundations for developing this agent by outlining the vision and requirements, as well as providing knowledge structures for performing the necessary reasoning and support tasks. A case study illustrates the envisaged use of the framework.

The knowledge structures model habits on the basis of a representation of goals, activities and actions relevant to the user. The paper shows how these concepts can be linked to personal values which is essential for helping the user to choose desired behaviours that are in line with her underlying motivations. We have put forward the notion of a desired behaviour model as the basis for supporting a user in modifying habits. We have proposed several types of non-compliance based on the actual and desired behaviour models which can be used by the *HabInt* to monitor whether the user's actual behavior is in line

[9] This is a simplification: see [34] for a more complete account.

with her desired behavior. In this way, *HabInt* will be able to determine when and how to give the user timely advice.

A key feature of *HabInt* is that it adheres strictly to the user's vocabulary for expressing goals, activities, actions and values, and that the fine-grainedness of the concretisation relation and of the actions involved in the user's activities are tailored to the user's needs. If a reminder to "go by bike" is enough for the user to know what to do, then no additional information is stored by *HabInt*.

In future work we intend to investigate the resemblance to David Lewis' *perfect worlds* semantics for deontic logic. We would like to study whether deontic logic techniques would allow the type of reasoning needed to differentiate between actual behaviours, already formulated desired behaviours, and tentative attempts of the user to formulate her actual or desired behaviours.

The current model does not address the temporal dimension of habits. As timely interventions are crucial, the temporal aspects will be addressed in a future paper. Furthermore, here we defined `prob` as a probability function, whereas we would like *HabInt* to reason qualitatively, using notions such as 'always, often, at least biweekly...'. This challenge is to be addressed in future work, as will the other components of the agent architecture, the implementation and the verification of the validity, scalability and robustness of the approach.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

[1] http://www.merriam-webster.com/dictionary/. [accessed 15-05-2016].
[2] Philip E. Agre, 'The dynamic structure of everyday life', Technical report, MIT, (1988).
[3] David E. Bell, Howard Raiffa, and Amos Tversky, eds. *Decision Making: Descriptive, Normative, and Prescriptive Interactions*. Cambridge University Press, 1988.
[4] V. Bellotti, B. Dalal, N. Good, P. Flynn, D.G. Bobrow, and N. Ducheneaut, 'What a to-do: Studies of task management towards the design of a personal task list manager', in *Proceedings of CHI'04*, pp. 735–742, (2004).
[5] Trevor J.M. Bench-Capon, 'Persuasion in practical argument using value-based argumentation frameworks', *Journal of Logic and Computation*, **13**(3), (2003).
[6] P. Berry, K. Conley, M. Gervasio, B. Peintner, T. Uribe, and N. Yorke-Smith, 'Deploying a personalized time management agent', in *Proceedings of AAMAS'06*, pp. 1564–1571, (2006).
[7] Rafael H. Bordini, Mehdi Dastani, Jürgen Dix, and Amal El Fallah Seghrouchni, *Multi-Agent Programming: Languages, Tools and Applications*, Springer, Berlin, 2009.
[8] Matthew M. Botvinick and David C. Plaut, 'Doing without schema hierarchies: A recurrent connectionist approach to normal and impaired routine sequential action', *Psychological Review*, (2004).
[9] Andy Clark and David Chalmers, 'The extended mind', *Analysis*, **58**(1), 7–19, (1998).
[10] Richard P. Cooper, Nicolas Ruh, and Denis Mareschal, 'The goal circuit model: A hierarchical multi-route model of the acquisition and control of routine sequential action in humans', *Cognitive Science: A Multidisciplinary Journal*, (2013).
[11] Thi V. Duong, Hung H. Bui, Dinh Q. Phung, and Svetha Venkatesh, 'Activity recognition and abnormality detection with the switching hidden semi-markov model', in *CVPR 2005 : Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 838–845, (2005).
[12] Kerry Fairbrother and James Warn, 'Workplace dimensions, stress and job satisfaction', *Journal of Managerial Psychology*, **18**(1), 8–21, (2003).
[13] Shai Fine, Yoram Singer, and Naftali Tishby, 'The hierarchical hidden markov model: Analysis and applications', *Machine Learning*, (1998).
[14] Holly S. Goldman, 'David Lewis' semantics for deontic logic', *Mind*, **86**(342), 242–248, (1977).
[15] Henry H. Hexmoor, *Representing and Learning Routine Activities*, Ph.D. dissertation, New York State University, 1995.
[16] Clark L. Hull, *Principles of behavior: an introduction to behavior theory*, Appleton-Century, 1943.
[17] Catholijn M. Jonker, M.Birna van Riemsdijk, and Bas Vermeulen, 'Shared mental models - a conceptual analysis', in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, ed., van der Hoek, (2010).
[18] *Model of Human Occupation: Theory and Application*, ed., Gary Kielhofner, Lippincott Williams & Wilkins, 2008.
[19] Ariel Knafo and Shalom H. Shwartz, *Cultural transmission: Psychological, developmental, social, and methodological aspects*, chapter Accounting for parent-child value congruence: Theoretical considerations and empirical evidence., 240–268, Culture and psychology, Cambridge University Press, 2009.
[20] Óscar D. Lara and Miguel A. Labrador, 'A survey on human activity recognition using wearable sensors', *IEEE Communications Surveys and Tutorials*, **15**(3), 1192–1209, (2013).
[21] Eva Lindbladh and Carl H. Lyttkens, 'Habit versus choice: the process of decision-making in health-related behaviour', *Social Science & Medicine*, **55**(3), 451–465, (August 2002).
[22] David V. Pynadath and Milind Tambe, *Socially Intelligent Agents*, volume 3 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*, chapter Electric Elves: Adjustable Autonomy in Real-World Multi-Agent Environments, 101–108, Springer, 2002.
[23] Meg J. Rohan, 'A rose by any name? the value construct', *Personality and Social Psychology Review*, **4**(3), 255–277, (2000,).
[24] Neil S. Jacobson, Christopher R. Martell, and Sona Dimidjian, 'Behavioral activation treatment for depression: Returning to contextual roots', *Clinical Psychology: Science and Practice*, **8**(3), 255–270, (September 2001).
[25] Katarzyna Stawarz, Anna L. Cox, and Ann Blandford, 'Beyond self-tracking and reminders: Designing smartphone apps that support habit formation', in *CHI '15: Conference on Human Factors in Computing Systems, Seoul, Republic of Korea, April 18 - 23, 2015.*, pp. 2653 – 2662, (2015).
[26] Carroll U. Stephens and Jon M. Shepard, *Wiley Encyclopedia of Management*, chapter Normative/Descriptive.
[27] John Thangarajah, *Managing the Concurrent Execution of Goals in Intelligent Agents*, Ph.D. dissertation, Royal Melbourne Institute of Technology, 2005.
[28] David Trafimow, 'Habit as both a direct cause of intention to use a condom and as a moderator of the attitude-intention and subjective norm-intention relations.', *Psychology and Health*, **15**, 383–393, (2000).
[29] Frank Trentmann, *Time, Consumption and Everyday Life: Practice, Materiality and Culture*, chapter Disruption is Normal: Blackouts, Breakdowns and the Elasticity of Everyday Life, Berg, 2009.
[30] M. van Lent, W. Fisher, and M. Mancuso, 'An explainable artificial intelligence system for small-unit tactical behavior', in *Proc. of the Sixteenth Conference on Innovative Applications of Artificial Intelligence*, (2004).
[31] M.Birna van Riemsdijk, Catholijn M. Jonker, and Victor Lesser, 'Creating socially adaptive electronic partners', in *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, (2015).
[32] Bas Verplanken, 'Beyond frequency: Habit as a mental construct', *British Journal of Social Psychology*, **45**, 639–656, (2006).
[33] Wietske Visser, Koen V. Hindricks, and Catholijn M. Jonker, 'Argumentation-based qualitative preference modelling with incomplete and uncertain information', *Group Decision and Negotiation*, (1), 99–127, (2012).
[34] Wendy Wood and David T. Neal, 'A new look at habits and the habit-goal interface', *Psychological Review*, **114**(4), 843– 863, (2007).
[35] Wendy Wood and Dennis Rünger, 'Psychology of habit', *Annual Review of Psychology*, (2015).
[36] Wendy Wood, Leona Tam, and Melissa Witt G., 'Changing circumstances, disrupting habits', *Journal of Personality and Social Psychology*, (2005).
[37] Neil Yorke-Smith, Shahin Saadati, Karen L. Myers, and David N. Morley, 'The design of a proactive personal agent for task management', *International Journal on Artificial Intelligence Tools*, **21**(1), (2012).