# Model-Based Reasoning Methods within an Ambient Intelligent Agent Model

Tibor Bosse, Fiemke Both, Charlotte Gerritsen, Mark Hoogendoorn, and Jan Treur

Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
{tbosse, fboth, cg, mhoogen, treur}@few.vu.nl
http://www.few.vu.nl/~{tbosse, fboth, cg, mhoogen, treur}

**Abstract.** Ambient agents react on humans on the basis of their information obtained by sensoring and their knowledge about human functioning. Appropriate types of reactions depend on in how far an ambient agent understands the human. On the one hand, such an understanding requires that the agent has knowledge to a certain depth about the human's physiological and mental processes in the form of an explicitly represented model of the causal and dynamic relations describing these processes. On the other hand, given such a model representation, the agent needs reasoning methods to derive conclusions from the model and the information available by sensoring. This paper presents a number of such model-based reasoning methods. They have been formally specified in an executable temporal format, which allows for simulation of reasoning traces and automated verification in a dedicated software environment. A number of such simulation experiments and their formal analysis are described.

## 1 Introduction

Recent developments within Ambient Intelligence provide technological possibilities to contribute to personal care; cf. [1, 2, 18]. Such applications can be based on possibilities to acquire sensor information about humans and their functioning, but more substantial applications depend on the availability of adequate knowledge for analysis of information about human functioning. If knowledge about human functioning is explicitly represented in the form of computational models in ambient agents, these agents can show more understanding, and (re)act accordingly by undertaking actions in a knowledgeable manner that improve the human's wellbeing and performance. In recent years, human-directed scientific areas such as cognitive science, psychology, neuroscience and biomedical sciences have made substantial progress in providing an increased insight in the various physical and mental aspects involved in human functioning. Although much work still remains to be done, dynamic models have been developed and formalised for a variety of such aspects and the way in which humans (try to) manage or regulate them. From a biomedical angle, examples of such aspects are (management of) heart functioning, diabetes, eating regulation disorders, and HIV-infection; e.g., [5, 15]. From a psychological and social angle, examples are emotion regulation, attention regulation, addiction management, trust management, stress management, and criminal behaviour management; e.g., [6, 11, 16]. Such models can be the basis for dedicated model-based reasoning methods that allow an agent to derive relevant conclusions from these models and available sensor information.

This paper addresses the design of ambient agents that have knowledge about human behaviours and states over time in the form of explicitly represented models of the causal and dynamical relations involved. First it is shown how such models can be formally represented in a logical format that also integrates numerical aspects; cf. [9]. Next a number of logical reasoning methods are presented that are based on such models. These reasoning methods are represented in a temporal logical format according to the approach put forward in [14]. A number of simulation experiments to obtain reasoning traces are described. These traces have been formally analysed by a dedicated verification tool. The types of reasoning methods addressed cover a variety of phenomena such as causal and numerical simulation, qualitative reasoning and simulation, abductive reasoning [17], and explanation generation. The reasoning methods provide a conceptual and logical foundation for these phenomena. Moreover, they provide a solid basis for conceptual and detailed design of model-based ambient agents that need such capabilities.

Section 2 describes the formal modelling approach that is used throughout this paper. Next, in Section 3 and 4 the reasoning methods themselves are presented. Section 3 addresses uncontrolled methods for belief generation, and Section 4 addresses controlled methods for belief generation. Section 5 illustrates how these reasoning methods can be used, by performing simulation experiments in two example case studies. Section 6 provides a number of basic properties that may hold for model-based reasoning methods within ambient agents. Section 7 addresses verification of basic properties as introduced in Section 3 against simulation traces, and interlevel relations between properties at different aggregation levels. Section 8 concludes the paper with a discussion.

## 2 Modelling Approach

This section introduces the formal modelling approach that is used throughout this paper. Section 2.1 briefly describes the Temporal Trace Language (TTL) for specification of dynamic properties (and its executable sublanguage LEADSTO), and Section 2.2 briefly explains how reasoning methods are formalised in this paper.

### 2.1 The Temporal Trace Language TTL

In order to execute and verify human-like ambience models, the expressive language TTL is used [7]. This predicate logical language supports formal specification and analysis of dynamic properties, covering both qualitative and quantitative aspects. TTL is built on atoms referring to states, time points and traces. A *state* of a process for (state) ontology Ont is an assignment of truth values to the set of ground atoms in the ontology. The set of all possible states for ontology Ont is denoted by STATES(Ont). To describe sequences of states, a fixed *time frame* T is assumed which is linearly ordered. A *trace* $\gamma$ over state ontology Ont and time frame T is a mapping $\gamma : T \rightarrow$ STATES(Ont), i.e., a sequence of states $\gamma_t$ ($t \in T$) in STATES(Ont). The set of *dynamic properties* DYNPROP(Ont) is the set of temporal statements that can be formulated with respect to traces based on the state ontology Ont in the following manner. Given a trace $\gamma$ over state ontology Ont, the state in $\gamma$ at time point t is denoted by state($\gamma$, t). These states can be related to state properties via the formally defined satisfaction relation |=, comparable to the Holds-predicate in the Situation Calculus: state($\gamma$, t) |= p denotes that state property p holds in trace $\gamma$ at time t. Based on these statements, dynamic properties can be formulated in a sorted first-order predicate logic, using quantifiers over time and traces and the usual first-order logical connectives such as $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\forall$, $\exists$. A special software environment has been developed for TTL, featuring both a Property Editor for building and editing TTL properties and a Checking Tool that enables formal verification of such properties against a set of (simulated or empirical) traces.

**Executable Format** To specify simulation models and to execute these models, the language LEADSTO [8], an executable sublanguage of TTL, is used. The basic building blocks of this language are causal relations of the format $\alpha \twoheadrightarrow_{e, f, g, h} \beta$, which means:

| | |
|---|---|
| if | state property $\alpha$ holds for a certain time interval with duration g, |
| then | after some delay (between e and f) state property $\beta$ will hold for a certain time interval of length h. |

where $\alpha$ and $\beta$ are state properties of the form ‘conjunction of literals’ (where a literal is an atom or the negation of an atom), and e, f, g, h non-negative real numbers.

## 2.2 Temporal Specification of Reasoning Methods

In this paper a dynamic perspective on reasoning is taken, following, e.g.. [14]. In practical reasoning situations usually different lines of reasoning can be generated, each leading to a distinct set of conclusions. In logic semantics is usually expressed in terms of models that represent descriptions of conclusions about the world and in terms of entailment relations based on a specific class of this type of models. In the (sound) classical case each line (trace) of reasoning leads to a set of conclusions that are true in all of these models: each reasoning trace fits to each model. However, for non-classical reasoning methods the picture is different. For example, in default reasoning or abductive reasoning methods a variety of mutually contradictory conclusion sets may be possible. It depends on the chosen line of reasoning which one of these sets fits.

The general idea underlying the approach followed here, and inspired by [14] is that a particular reasoning trace can be formalised by a sequence of *information states* $M_0, M_1, \ldots\ldots$ . Here any $M_t$ is a description of the (partial) information that has been derived up to time point t. From a dynamic perspective, an inference step, performed in time duration D is viewed as a transition $M_t \rightarrow M_{t+D}$ of a current information state $M_t$ to a next information state $M_{t+D}$. Such a transition is usually described by application of a deduction rule or proof rule, which in the dynamic perspective on reasoning gets a temporal aspect. A particular reasoning line is formalised by a sequence $(M_t)_{t \in T}$ of subsequent information states labelled by elements of a flow of time $T$, which may be discrete, based on natural numbers, or continuous, based on real numbers.

An information state can be formalised by a set of statements, or as a three-valued (false, true, undefined) truth assignment to ground atoms, i.e., a partial model. In the latter case, which is followed here (as in [14]), a sequence of such information states or reasoning trace can be interpreted as a partial temporal model. A transition relating a next information state to a current one can be formalised by temporal formulae the partial temporal model has to satisfy. For example, a modus ponens deduction rule can be specified in temporal format as:

$$\text{derived(I)} \wedge \text{derived(implies(I, J))} \twoheadrightarrow \text{derived(J)}$$

So, inference rules are translated into temporal rules thus obtaining a temporal theory describing the reasoning behaviour. Each possible reasoning trace can be described by a linear time model of this theory (in temporal partial logic).

In this paper, this dynamic perspective on reasoning is applied in combination with facts that are labelled with temporal information, and models based on causal or temporal relationships that relate such facts. To express the information involved in an agent's internal reasoning processes, the ontology shown in Table 1 is used.

| **Predicate** | **Description** |
|---|---|
| belief(I:INFO_EL) | information I is believed |
| world_fact(I:INFO_EL) | I is a world fact |
| has_effect(A:ACTION, I:INFO_EL) | action A has effect I |
| **Function to INFO_EL** | **Description** |
| leads_to_after(I:INFO_EL, J:INFO_EL, D:REAL) | state property I leads to state property J after duration D |
| at(I:INFO_EL, T:TIME) | state property I holds at time T |

**Table 1.** Generic Ontology used within the Ambient Agent Model

As an example belief(leads_to_after(I:INFO_EL, J:INFO_EL, D:REAL)) is an expression based on this ontology which represents that the agent has the knowledge that state property I leads to state property J with a certain time delay specified by D. An example of a kind of dynamic modus ponens rule can be specified as

$$\text{belief(at(I, T))} \wedge \text{belief(leads\_to\_after(I, J, D))} \rightarrow \text{belief(at(J, T+D))}$$

This temporal rule states that if it is believed (by the agent) that I holds at T and that I leads to J after duration D, then it will be believed that J holds at T + D. This representation format will be used to formalise this and other types of model-based reasoning methods, as will be shown more extensively in Sections 3 and 4.

## 3 Model-Based Reasoning Methods for Belief Generation

Two types of reasoning methods to generate beliefs can be distinguished:

- *Forward reasoning methods for belief generation*

These are reasoning methods that follow the direction of time and causality, deriving from beliefs about properties at certain time points, new beliefs about properties at later time points.

- *Backward reasoning methods for belief generation*
  These are reasoning methods that follow the opposite direction of time and causality, deriving from beliefs about properties at certain time points, new beliefs about properties at earlier time points.

In Section 3.1 the forward reasoning methods for belief generation are discussed, in Section 3.2 the backward reasoning methods.

### 3.1 Forward reasoning methods for belief generation

Forward reasoning methods are often used to make predictions on future states, or on making an estimation of the current state based on information acquired in the past. The first reasoning method is one that occurs in the literature in many variants, in different contexts and under different names, varying from, for example, computational (numerical) simulation based on difference or differential equations, qualitative simulation, causal reasoning, execution of executable temporal logic formulae, and forward chaining in rule-based reasoning, to generation of traces by transition systems and finite automata. The basic specification of this reasoning method can be expressed as follows.

**Belief Generation based on Positive Forward Simulation**

If it is believed that I holds at T and that I leads to J after duration D, then it is believed that J holds after D.

∀I,J:INFO_EL ∀D:REAL ∀T:TIME

belief(at(I, T)) ∧ belief(leads_to_after(I, J, D)) ↠ belief(at(J, T+D))

If it is believed that I1 holds at T and that I2 holds at T, then it is believed that I1 and I2 holds at T.

belief(at(X1,T)) ∧ belief(at(X2, T)) ↠ belief(at(and(X1, X2), T))

Note that, if the initial beliefs are assumed correct, belief correctness holds for leads to beliefs, and positive forward correctness of leads to relationships holds, then all beliefs generated in this way are correct. A second way of belief generation by forward simulation addresses the propagation of negations. This is expressed as follows.

**Belief Generation based on Single Source Negative Forward Simulation**

If it is believed that I does not hold at T and that I leads to J after duration D, then it is believed that J does not hold after D.

∀I,J:INFO_EL ∀D:REAL ∀T:TIME

belief(at(not(I), T)) ∧ belief(leads_to_after(I, J, D)) ↠ belief(at(not(J), T+D)))

If it is believed that I1 (resp. I2) does not hold at T, then it is believed that I1 and I2 does not hold at T.

belief((at(not(I1),T))) ↠ belief(at(not(and(I1, I2)), T))

belief(at(not(I2),T)) ↠ belief(at(not(and(I1, I2)), T))

Note that this only provides correct beliefs when the initial beliefs are assumed correct, belief correctness holds for leads to beliefs, and single source negative forward correctness holds for the leads to relationships.

**Belief Generation based on Multiple Source Negative Forward Simulation**

If for any J and time T, for every I that is believed to lead to J after some duration D, it is believed that I does not hold before duration D, then it is believed that J does not hold.

∀I,J:INFO_EL ∀D:REAL ∀T:TIME

∀I, D [ belief(leads_to_after(I, J, D)) → belief(at(not(I), t-D) ] ↠ belief(at(not(J), T))

If it is believed that I1 (resp. I2) does not hold at T, then it is believed that I1 and I2 does not hold at T.
belief(at(not(I1),T)) $\twoheadrightarrow$ belief(at(not(and(I1, I2)), T))
belief(at(not(I2),T)) $\twoheadrightarrow$ belief(at(not(and(I1, I2)), T))

This provides correct beliefs when the initial beliefs are assumed correct, belief correctness holds for leads to beliefs, and multiple source negative forward correctness holds for the leads to relationships.

### 3.2 Backward reasoning methods for belief generation

The basic specification of a backward reasoning method is specified as follows.

**Belief Generation based on Modus Tollens Inverse Simulation**

If it is believed that J does not hold at T and that I leads to J after duration D, then it is believed that I does not hold before duration D.
∀I,J:INFO_EL ∀D:REAL ∀T:TIME
belief(at(not(J), T)) ∧ belief(leads_to_after(I, J, D)) $\twoheadrightarrow$ belief(at(not(I), T-D))
If it is believed that not I1 and I2 holds at T and that I2 (resp. I1) holds at T, then it is believed that I1 (resp. I2) does not hold at T.
belief(at(not(and(I1, I2), T)) ∧ belief(at(I2, T)) $\twoheadrightarrow$ belief(at(not(I1), T))
belief(at(not(and(I1, I2), T)) ∧ belief(at(I1, T)) $\twoheadrightarrow$ belief(at(not(I2), T))

**Belief Generation based on Simple Abduction**

If it is believed that J holds at T and that I leads to J after duration D, then it is believed that I holds before duration D.
∀I,J:INFO_EL ∀D:REAL ∀T:TIME
belief(at(J, T)) ∧ belief(leads_to_after(I, J, D)) $\twoheadrightarrow$ belief(at(I, T-D))
If it is believed that I1 and I2 holds at T, then it is believed that I1 holds at T and that I2 holds at T.
belief(at(and(I1, I2), T)) $\twoheadrightarrow$ belief(at(I1,T)) ∧ belief(at(I2, T))

As another option, an abductive causal reasoning method can be internally represented in a simplified form as follows.

**Belief Generation based on Multiple Effect Abduction**

If for any I and time T, for every J for which it is believed that I leads to J after some duration D, it is believed that J holds after duration D, then it is believed that I holds at T.
∀I:INFO_EL ∀T:TIME
∀J [belief(leads_to_after(I, J, D)) → belief(at(J, T+D)) ] $\twoheadrightarrow$ belief(at(I, T))
If it is believed that I1 and I2 holds at T, then it is believed that I1 holds at T and that I2 holds at T.
belief(at(and(I1, I2), T)) $\twoheadrightarrow$ belief(at(I1,T)) ∧ belief(at(I2, T))

**Belief Generation based on Context-Supported Abduction**

If it is believed that J holds at T and that I2 holds at T and that I1 and I2 leads to J after duration D, then it is believed that I1 holds before duration D.
∀I,J:INFO_EL ∀D:REAL ∀T:TIME
belief(at(J, T)) ∧ belief(at(I2, T-D)) ∧ belief(leads_to_after(and(I1, I2), J, D)) $\twoheadrightarrow$ belief(at(I1, T-D))
If it is believed that I1 and I2 holds at T, then it is believed that I1 holds at T and that I2 holds at T.
belief(at(and(I1, I2), T)) $\twoheadrightarrow$ belief(at(I1,T)) ∧ belief(at(I2, T))

## 4 Controlling Belief Generation

An uncontrolled belief generation approach may easily lead to a combinatorial explosion of generated beliefs, for example, based on all conjunctions that can be formed. Therefore, controlled approaches where selection is done in some stage of the process are usually more effective. Often more specific knowledge is available based on which belief generation can leave out of consideration some (or most) of the possible beliefs that can be generated. To incorporate such selections, the following three approaches are possible: *selection afterwards overall*, *selection afterwards step by step*, *selection before*. Each of these options is discussed in more detail. Furthermore, it is discussed what selection criteria can be used to make such a selection.

## Belief Generation Selection

### Selection Afterwards Overall

In this approach first (candidate) beliefs are generated in an uncontrolled manner, and after that a selection process is performed based on some selection criterion. Two examples, one for a forward belief generation form and one for a backward belief generation form are as follows.

**Controlled Belief Generation based on Positive Forward Simulation by Selection Afterwards Overall**

If it is believed that I holds at T and that I leads to J after duration D, then it is believed that J holds after D.
∀I,J:INFO_EL ∀D:REAL ∀T:TIME
belief(at(I, T)) ∧ belief(leads_to_after(I, J, D)) ↠ belief(at(J, T+D))
If it is believed that I1 holds at T and that I2 holds at T, then it is believed that I1 and I2 holds at T.
belief(at(I1,T)) ∧ belief(at(I2, T)) ↠ belief(at(and(I1, I2), T))
If I is a belief and selection criterion s is fulfilled, then I is a selected belief.
belief(at(I, T)) ∧ s ↠ selected_belief(at(I, T))

**Controlled Belief Generation based on Multiple Effect Abduction by Selection Afterwards Overall**

If for any I and time T, for every J for which it is believed that I leads to J after some duration D, it is believed that J holds after duration D, then it is believed that I holds at T.
∀I:INFO_EL ∀T:TIME
∀J [belief(leads_to_after(I, J, D)) → belief(at(J, T+D)) ] ↠ belief(at(I, T))
If it is believed that I1 and I2 holds at T, then it is believed that I1 holds at T and that I2 holds at T.
belief(at(and(I1, I2), T)) ↠ belief(at(I1,T)) ∧ belief(at(I2, T))
If I is a belief and selection criterion s is fulfilled, then I is a selected belief.
belief(at(I, T)) ∧ s ↠ selected_belief(at(I, T))

This approach to control can only be applied when the number of beliefs that is generated in an uncontrolled manner is small. Otherwise more local approaches are better candidates to consider.

### Selection Afterwards Step by Step

The step by step variant of selection afterwards performs the selection immediately after a belief has been generated. By such a local selection it is achieved that beliefs that are not selected can not be used in further belief generation processes, thus limiting these processes. The approach uses the temporal selection rule given above together with a slightly adapted form of specification to generate beliefs. Again two examples, one for a forward belief generation form and one for a backward belief generation form are as follows.

**Controlled Bel. Generation based on Positive Forward Simulation by Selection Aft. Step by Step**

If it is believed that I holds at T and that I leads to J after duration D, then it is believed that J holds after D.
∀I,J:INFO_EL ∀D:REAL ∀T:TIME
selected_belief(at(I, T)) ∧ belief(leads_to_after(I, J, D)) ↠ belief(at(J, T+D))
If it is believed that I1 holds at T and that I2 holds at T, then it is believed that I1 and I2 holds at T.
selected_belief(at(I1,T)) ∧ selected_belief(at(I2, T)) ↠ belief(at(and(I1, I2), T))
If I is a belief and selection criterion s is fulfilled, then I is a selected belief.
belief(at(I, T)) ∧ s ↠ selected_belief(at(I, T))

**Controlled Belief Generation based on Multiple Effect Abduction by Selection Aft. Step by Step**

If for any I and time T, for every J for which it is believed that I leads to J after some duration D, it is believed that J holds after duration D, then it is believed that I holds at T.

∀I:INFO_EL ∀T:TIME

∀J [belief(leads_to_after(I, J, D)) → selected_belief(at(J, T+D)) ] →→ belief(at(I, T))

If it is believed that I1 and I2 holds at T, then it is believed that I1 holds at T and that I2 holds at T.

selected_belief(at(and(I1, I2), T)) →→ belief(at(I1,T)) ∧ belief(at(I2, T))

If I is a belief and selection criterion s is fulfilled, then I is a selected belief.

belief(at(I, T)) ∧ s →→ selected_belief(at(I, T))

This selection approach may be much more efficient than the approach based on selection afterwards overall.

**Selection Before**

The approach of selection afterwards step by step can be slightly modified by not selecting the belief just after its generation, but just before. This allows fo
r a still more economic process of focus generation. Again two examples, one for a forward belief generation form and one for a backward belief generation form are as follows.

**Controlled Belief Generation based on Positive Forward Simulation by Selection Before**

If it the belief that I holds at T was selected and it is believed that I leads to J after duration D, and selection criterion s1 holds, then the belief that J holds after D is selected.

∀I,J:INFO_EL ∀D:REAL ∀T:TIME

selected_belief(at(I, T)) ∧ belief(leads_to_after(I, J, D)) ∧ s1 →→ selected_belief(at(J, T+D))

If the beliefs that I1 holds at T and that I2 holds at T were selected, and selection criterion s2 holds, then the conjunction of I1 and I2 at T is a selected belief.

selected_belief(at(I1,T)) ∧ selected_belief(at(I2, T)) ∧ s2 →→ selected_belief(at(and(I1, I2), T))

**Controlled Belief Generation based on Multiple Effect Abduction by Selection Before**

If for any I and time T, for every J for which it is believed that I leads to J after some duration D, the belief that J holds after duration D was selected, and selection criterion s1 holds, then it the belief that I holds at T is a selected belief.

∀I:INFO_EL ∀T:TIME

∀J [belief(leads_to_after(I, J, D)) → selected_belief(at(J, T+D)) ] ∧ s1 →→ selected_belief(at(I, T))

If the beliefs that I1 and I2 holds at T were selected, and selection criterion s2 holds then the belief that I1 holds at T is a selected belief.

selected_belief(at(and(I1, I2), T)) ∧ s2 →→ selected_belief(at(I1,T))

If the beliefs that I1 and I2 holds at T were selected, and selection criterion s2 holds then the belief that I2 holds at T is a selected belief

selected_belief(at(and(I1, I2), T)) ∧ s3 →→ selected_belief(at(I2, T))

## 4.2 Selection Criteria in Reasoning Methods for Belief Generation

Selection criteria needed for controlled belief generation can be specified in different manners. A simple manner is by assuming that the agent has knowledge which beliefs are relevant, expressed by a predicate in_focus. If this assumption is made, then any selection criterion s can be expressed as in_focus(I), where I is the property for which a belief is considered. The general idea is that if a belief can be generated, it is selected (only) when it is in focus. For example, for the two methods for selection afterwards, the temporal rule will be expressed as:

belief(at(I, T)) ∧ in_focus(I) →→ selected_belief(at(I, T))

For the method based on selection before, based on focus information the temporal rules will be expressed for the forward example by:

$\forall$I,J:INFO_EL $\forall$D:REAL $\forall$T:TIME
selected_belief(at(I, T)) $\wedge$ belief(leads_to_after(I, J, D)) $\wedge$ in_focus(J) $\twoheadrightarrow$ selected_belief(at(J, T+D))
selected_belief(at(I1,T)) $\wedge$ selected_belief(at(I2, T)) $\wedge$ in_focus(and(I1, I2)) $\twoheadrightarrow$ selected_belief(at(and(I1, I2), T))

For the backward example of the method based on selection before, the temporal rules will be expressed by:

$\forall$I:INFO_EL $\forall$T:TIME
$\forall$J [belief(leads_to_after(I, J, D)) $\rightarrow$ selected_belief(at(J, T+D)) ] $\wedge$ in_focus(I) $\twoheadrightarrow$ selected_belief(at(I, T))
selected_belief(at(and(I1, I2), T)) $\wedge$ in_focus(I1) $\twoheadrightarrow$ selected_belief(at(I1,T))
selected_belief(at(and(I1, I2), T)) $\wedge$ in_focus(I2) $\twoheadrightarrow$ selected_belief(at(I2, T))

It is beyond the scope of this paper whether such foci may be static or dynamic and how they can be determined by an agent. For cases that such general focus information is not available, the selection criteria can be specified in different manners.

# 5 Simulation

This section illustrates for a number of the reasoning methods provided in the previous sections how they can be used within ambient agents that perform model-based reasoning. This is done by means of two example case studies, each involving an ambient system that uses a causal dynamic model to represent the behaviour of a human, and uses the reasoning methods to determine the state of the human in a particular situation. Section 5.1 focuses on a system that monitors the state of car drivers in order to avoid unsafe driving. Section 5.2 addresses an ergonomic system that monitors the stress level of office employees. Both case studies have been formalised and, using the LEADSTO simulation software [8], have been used to generate a number of simulation traces. In this section, for each model one example simulation trace is shown. More simulation traces can be found in the Appendix on[1].

## 5.1 Ambient Driver Model

The example model used as an illustration in this section is inspired by a system designed by Toyota which monitors drivers in order to avoid unsafe driving. The system can basically measure drug level in the sweat of a driver (e.g., via a sensor at the steering wheel, or at an ankle belt), and monitor steering operations and the gaze of the driver. Note that the system is still in the experimental phase. The model used in this paper describes how a high drug intake leads to a high drug level in the blood and this leads to physiological and behavioural consequences: (1) physiological: a high drug level (or a substance relating to the drug) in the sweat, (2) behavioural: abnormal steering operation and an unfocused gaze. The dynamic model is represented within the ambient agent by the following beliefs (where D is an arbitrary time delay):

belief(leads_to_after(drug_intake_high, drug_in_blood_high, D))
belief(leads_to_after(drug_in_blood_high, drug_in_sweat_high, D))

[1] http://www.cs.vu.nl/~mhoogen/reasoning/appendix-rm-ami.pdf

belief(leads_to_after(drug_in_blood_high, abnormal_steering_operation, D))
belief(leads_to_after(drug_in_blood_high, unfocused_gaze, D))

Figure 1 shows this dynamical model in a graphical form.

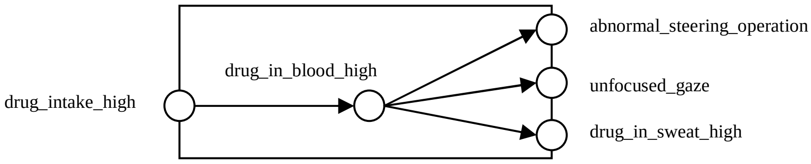


**Fig. 1.** Graphical representation of the dynamical model

By applying the different reasoning methods specified in Section 3 and 4, the state of the driver and the expected consequences can be derived. In the simulations below the controlled belief generation method has been used based on selection before beliefs are generated; every temporal rule requires that certain selection criteria are met and that the belief to be derived is in focus. In the following simulations, for the sake of simplicity all information is desired, therefore all derivable beliefs are in focus. The selection criteria involve knowledge about the number of effects and sources that are required to draw conclusions. The knowledge used in this model is the following.

sufficient_evidence_for(and(abnormal_steering_operation, unfocused_gaze), drug_in_blood_high)
sufficient_evidence_for(drug_in_sweat_high, drug_in_blood_high)
sufficient_evidence_for(drug_in_blood_high, drug_intake_high)
in_focus(drug_intake_high); in_focus(drug_in_blood_high); in_focus(drug_in_sweat_high);
in_focus(abnormal_steering_operation); in_focus(unfocused_gaze)

Here, the predicate sufficient_evidence_for(P, Q) represents the belief that expression P is sufficient evidence for the system to derive Q. An example simulation trace is shown in Figure 2. In the Figure, the left side shows the atoms that occur during the simulation, whereas the right side represents a time line where a grey box indicates an atom is true at that time point, and a light box indicates false. In this trace, it is known (by observation) that the driver is steering abnormally and that the driver's gaze is unfocused. Since these two beliefs are sufficient evidence for a high drug level in the blood, using the reasoning method Belief Generation based on Multiple Effect Abduction, at(drug_in_blood_high, 1) becomes a selected belief at time point 3. Given this derived belief, the belief can be deduced that the drug level in the sweat of the driver is high, using Positive Forward Simulation. At the same time (time point 4), the reasoning method Simple Abduction determines the belief that the drug intake of the driver must have been high.

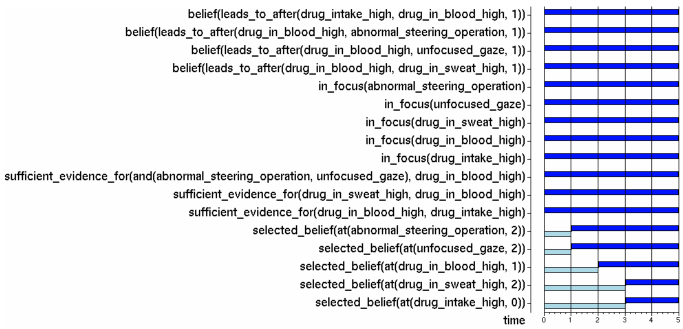


**Fig. 2.** Simulation Trace: abnormal steering and unfocused gaze detected

### 5.2 Ambient Stress Model

The example model used in this section is inspired by ergonomic systems that monitor the activities of office employees in their workspace, e.g., in order to avoid RSI (for example, WorkPace, see http://workpace.com/). Such systems may measure various types of information. In this section, three types of measurable (sensor) information are taken into account, namely *actions* (e.g., mouse clicks or key strokes), *biological aspects* (e.g., heart beat, temperature, or skin conductivity), and *activities* (e.g., incoming e-mails, telephone calls, or electronic agenda items). The model considered here describes how (the observation of) a certain activity can lead to a high level of stress and this leads to biological/physiological and behavioural consequences: (1) biological: called here 'high biological aspect' (e.g., increased heart rate) (2) behavioural: changed action (e.g., high number of keystrokes per second). The dynamical model is represented within the ambient agent by the following beliefs:

```
belief(leads_to_after(activity, observes(activity), D))
belief(leads_to_after(observes(activity), preparedness_to_act, D))
belief(leads_to_after(observes(activity), stress(high), D))
belief(leads_to_after(preparedness_to_act, stress(high), D))
belief(leads_to_after(stress(high), preparedness_to_act, D))
belief(leads_to_after(preparedness_to_act, action, D))
belief(leads_to_after(stress(high), biological_aspect, D))
```

Figure 3 shows this dynamical model in a graphical form.

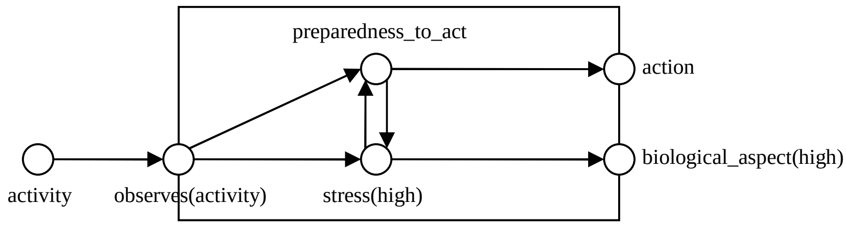


**Fig. 3.** Graphical representation of the dynamical model

Similar to Section 5.1, by applying the different reasoning methods specified earlier, the expected consequences for the state of the human and can be derived. Below, a number of simulation traces are shown, each with different settings for the selection criteria:

```
sufficient_evidence_for(biological_aspect(high), stress(high))
sufficient_evidence_for(observes(activity), activity)
sufficient_evidence_for(preparedness_to_act, stress(high))
sufficient_evidence_for(preparedness_to_act, observes(activity))
sufficient_evidence_for(stress(high), preparedness_to_act)
sufficient_evidence_for(stress(high), observes(activity))
sufficient_evidence_for(action, preparedness_to_act)

in_focus(action); in_focus(biological_aspect(high); in_focus(stress(high));
in_focus(observes(activity)); in_focus(activity)
```

In other words, by selecting different combinations of these criteria, different reasoning steps will be performed. Notice that the model considered here contains a cycle (see Figure 3). Therefore it is possible to derive an infinite number of beliefs for different time points. For example, if at(preparedness_to_act, 8) is believed, then by simple Positive Forward Simulation also at(stress(high), 9) would be derived, after which at(preparedness_to_act, 10) would be derived, and so on. However, it is not conceptually realistic, nor desirable that an agent attempts to derive beliefs about time points very far in the future. Therefore, by means of the in_focus predicate, an indication of a focus time interval has been specified, for example by statements like in_focus(at(preparedness_to_act, 8)).

An example simulation trace is shown in Figure 4. This trace uses as foci all possible information between time point 0 and 10. These foci have been derived using the following rule:

$$\text{in\_focus}(I) \wedge 0 \leq T \leq 10 \twoheadrightarrow \text{in\_focus}(\text{at}(I, T))$$

The only initially available knowledge that is present in this trace is at(action, 5). As shown in the figure, both Positive Forward Simulation and Simple Abduction are performed several times, eventually leading to all possible derivable information between time point 0 and 10.

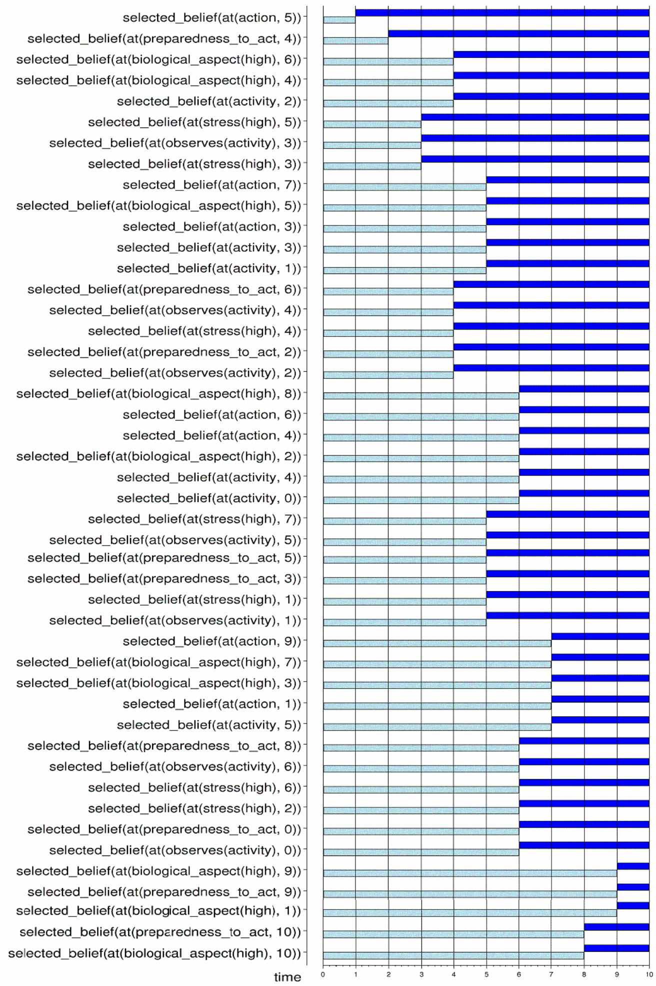


**Fig. 4.** Simulation Trace: Employee performs active behaviour (to be continued on next page)

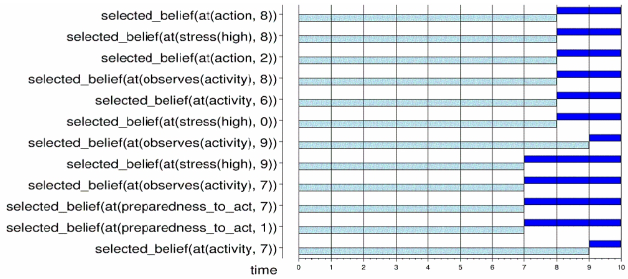


**Fig. 4**. Simulation Trace: Employee performs active behaviour (continued from previous page)

# 6 Basic Properties of World Facts, Beliefs and Leads To Relations

This section provides a number of basic properties that may hold for model-based reasoning methods within ambient agents. Section 6.1 addresses properties of world facts and beliefs; Section 6.2 addresses properties of LEADSTO relations.

## 6.1 Properties of world facts and beliefs

The following basic assumptions concerning two-valued world facts may hold:

**Consistency of world facts** In any state, it never happens that a world fact and its negation both hold.
not [ state(γ, t) |= world_fact(I) & state(γ, t) |= world_fact(not(I)) ]
**Completeness of world facts** In any state, for any world fact it holds or its negation holds.
state(γ, t) |= world_fact(I) | state(γ, t) |= world_fact(not(I))
**Consistency and completeness of world facts** In any state, for any world fact it holds if and only if its negation does not hold
state(γ, t) |= world_fact(I) ⇔ not state(γ, t) |= world_fact(not(I))
**Belief consistency** In any state, it never happens that a fact and its negation are both believed.
not [ state(γ, t) |= belief(I) & state(γ, t) |= belief(not(I)) ]
**Belief correctness** In any state, when a fact is believed it holds as a world fact.
state(γ, t) |= belief(at(I, t')) ⇒ state(γ, t') |= world_fact(I)
**Belief persistence** In any state, if a fact is believed, it will be believed at any later time point, unless its negation is believed at that time point.
∀t, t'≥t [ state(γ, t) |= belief(I) & not state(γ, t') |= belief(not(I)) ⇒ state(γ, t') |= belief(I) ]
∀t, t'≥t [ state(γ, t) |= belief(not(I)) & not state(γ, t') |= belief(I) ⇒ state(γ, t') |= belief(not(I)) ]
**Belief completeness** For any state, any fact is believed or its negation is believed.
state(γ, t) |= belief(I) | state(γ, t) |= belief(not(I))
**Belief coverage** In any state, any true world fact is believed.
state(γ, t) |= world_fact(I) ⇒ state(γ, t) |= belief(I)

In the general form, where a universal quantifier is assumed over I, belief completeness and belief coverage will usually not hold. However, it may hold for a specific class of information I. For example, sometimes it is assumed that the agent has complete beliefs about leads to relationships.

### 6.2 Properties of leads to relationships

The leads_to_after relationship expresses the conceptual core of a wide class of dynamic modelling concepts that occur in the literature in different contexts and under different names; see also [10]. Examples of such dynamical modelling concepts are, computational numerical modelling by difference or differential equations, qualitative dynamic modelling, causal relationships, temporal logic specifications, rule-based representations, Petri net representations, transition systems and finite automata. Often, either explicitly or implicitly the general assumption is made that when facts are true in the world, the facts to which they lead are also true in the world. This property is expressed as follows, also formulated by contraposition into a logically equivalent one:

**Positive forward correctness** If a world fact I holds in a state and it leads to another world fact J after duration D, then in the state after duration D this J will hold

state(γ, t) |= world_fact(I) & state(γ, t) |= world_fact(leads_to_after(I, J, D)) ⇒ state(γ, t+D) |= world_fact(J)

**Negative backward correctness** If a world fact J does not hold in a state and another world fact I leads to J after duration D, then in the state before duration D this I will not hold

state(γ, t) |= world_fact(not(J)) & state(γ, t) |= world_fact(leads_to_after(I, J, D)) ⇒
state(γ, t-D) |= world_fact(not(I))

Sometimes, also the more specific assumption is made that a world fact can be true *only* when a world fact preceding it via a leads to relation is true. This assumption can be seen as a temporal variant of a Closed World Assumption.

**Negative forward correctness (single source)** If a world fact I doers not hold in a state and it leads to another world fact J after duration D, then in the state after duration D this J will not hold

state(γ, t) |= world_fact(not(I)) & state(γ, t) |= world_fact(leads_to_after(I, J, D)) ⇒
state(γ, t+D) |= world_fact(not(J))

**Positive backward correctness (single source)** If a world fact J holds in a state and another world fact I leads to J after duration D, then in the state before duration D this I will hold

state(γ, t) |= world_fact(J) & state(γ, t) |= world_fact(leads_to_after(I, J, D)) ⇒ state(γ, t-D) |= world_fact(I)

The latter property can be formulated by contraposition into a logically equivalent property of the former one. These properties play a role in abductive reasoning methods, and automated explanation generation (in particular for why-explanations: answers on questions such as ‘Why does J hold?’). The latter two properties may not be fulfilled in cases that two (or multiple) non-equivalent world facts I1 and I2 exist that each lead to a world fact J. If I1 holds, and it leads to the truth of J, then it may well be the case that I2 never was true. A more complete property to cover such cases is the following.

**Negative forward correctness (multiple sources)** If for a world fact J, for every world fact I which leads to J after a duration D it does not hold in the state before duration D, then in the state after duration D this J will not hold

∀I, D [ state(γ, t-D) |= world_fact(leads_to_after(I, J, D)) ⇒ state(γ, t-D) |= world_fact(not(I)) ]
⇒ state(γ, t) |= world_fact(not(J))

**Positive backward correctness (multiple sources)** If a world fact J holds in a state, then there exists a world fact I which leads to J after a duration D which holds in the state before duration D.

state(γ, t) |= world_fact(J)
⇒ ∃I, D [ state(γ, t-D) |= world_fact(leads_to_after(I, J, D)) & state(γ, t-D) |= world_fact(I) ]

To obtain a logical foundation for a temporal variant of the Closed World Assumption in such situations in the context of executable temporal logic, in [13] the notion of temporal completion was introduced, as a temporal variant of Clark’s completion in logic programming.

## 7 Formal Analysis of Dynamic Properties

This section shows how it can be verified that the reasoning methods introduced in Section 3 and 4 (and simulation traces generated on the basis of these methods) satisfy certain basic properties as introduced in Section 6. This is done by establishing logical (inter-level) relationships between a *global property* (GP) of reasoning methods on the one hand, and the basic reasoning steps (or *local properties*, LP's) on the other hand, in such a way that the combination of reasoning steps (logically) entails the global property. In order to establish such inter-level relationships, also certain *intermediate properties* (IP's) are constructed, which can be used as intermediate steps in the proof. Here, the focus is on one particular property from Section 6, namely the Belief Correctness property. This global property for belief generation is expressed below in GP1 and states that all beliefs should be correct. This should hold for all reasoning intervals within the trace (i.e. starting at an observation interval, and the reasoning period thereafter without new observation input). Note that all variables γ that are not explicitly declared are assumed to be universally quantified. Moreover, E is assumed to be the duration of a reasoning step.

**GP1 (Belief Correctness)**

For all time points t1 and t2 later than t1 whereby at t1 a observations are observed, and between t1 and t2 no new observations are received, GP1(t1, t2) holds.

GP1 ≡
∀t1, t2 ≥ t1
[state(γ, t1) |= observation_interval &
¬state(γ, t2) |= observation_interval &
∀t' < t2 & t' > t1 [state(γ, t2) |= ¬observation_interval] ]
⇒ GP1(t1, t2)

The specification of the global property for an interval is expressed below.

**GP1(t1, t2) (Belief Correctness from t1 to t2)**

Everything that is believed to hold at T at time point t' between t1 and t2, indeed holds at that time point T.

GP1(t1, t2) ≡
∀I, T, t' ≥ t1 & t' ≤ t state(γ, t') |= belief(at(I, T)) ⇒ state(γ, T) |= world_fact(I)

In order to prove that property GP1 indeed holds, a proof by means of induction is used. The basis step of this proof is specified in property LP1, whereby the beliefs during the observation interval need to be correct.

**LP1(t) (Belief Correctness Induction Basis)**

If time point t is part of the observation interval, then everything that at time point t is believed to hold at time point T, indeed holds at time point T.

LP1(t) ≡
state(γ, t) |= observation_interval ⇒
[ ∀I,T state(γ, t) |= belief(at(I, T)) ⇒ state(γ, T) |= world_fact(I) ]

Furthermore, the induction step includes that if the global property holds from a time point t to the same time point, then the property should also hold between t and t + E.

**IP1 (Belief Correctness Induction Step)**

For all time points t, if GP1(t, t) holds, then also GP1(t, t+E) holds.

IP1 ≡
∀t GP1(t, t) ⇒ GP1(t, t+E)

In order to prove that this induction step indeed holds, the following three properties are specified: IP2, LP2, and LP3. First of all, the *grounding* of the belief generation (IP2) which states that for all beliefs that have not been generated since the last observation interval, they should either have been derived by means of forward reasoning, or by means of abduction.

**IP2 (Belief Generation Grounding)**

For all time points t+E, if information element J is believed to hold at time point T and J was not believed during the last observation interval, then either this was derived by applying a forward leadsto rule, or by means of abduction.

IP2 ≡
∀t,t0,J,T
[ state(γ, t) |= belief(at(J, T)) & last_observation_interval(t, t0) & ¬state(γ, t0) |= belief(at(J, T))
 ⇒ ∃I,t2, D
[ state(γ, t2) |= belief(at(I, T-D)) & state(γ, t2) |= belief(leads_to_after(I, J, D)) |
 state(γ, t2) |= belief(at(I, T+D)) & state(γ, t2) |= belief(leads_to_after(J, I, D)) ]

Property LP2 expresses the correctness of the model believed, that should correspond with the model present in the world.

**LP2 (Model Representation Correctness)**

For all time points t, if it is believed that I leads to J after duration D, then I indeed leads to J after duration D.

LP2 ≡
∀t,I,J,D
state(γ, t) |= belief(leads_to_after(I, J, D)) ⇒ state(γ, t) |= world_fact(leads_to_after(I, J, D))

The correctness of the derivations within the world is expressed in LP3.

**LP3 (Positive Forward Correctness)**

For all time points t, if information element I holds and I leads to J after duration D, then at time point t+D information element J holds.

LP3 ≡
∀t,I,J,T,D
state(γ, t) |= world_fact(I) & state(γ, t) |= world_fact(leads_to_after(I, J, D)) ⇒ state(γ, t+D) |= world_fact(J)

The final properties specified (LP4 and LP5) are used to ground property IP2. LP4 expresses that if a certain belief concerning an information element holds, and from this belief another belief concerning an information element can be derived, then this is the case at some time point t2.

**LP4 (Belief Generation based on Positive Forward Simulation)**

For all time points t, if information element I is believed to hold at time point T and it is believed that I leads to J after duration D, then there exists a time point t2 information element J is believed to hold at time point T+D.

LP4 ≡
∀t1,t2,I,J,T,D
state(γ, t1) |= belief(at(I, T)) & state(γ, t1) |= belief(leads_to_after(I, J, D)) ⇒ state(γ, t2) |= belief(at(J, T+D))

Property LP5 specifies how beliefs can be generated based on abduction.

**LP5 (Belief Generation based on Abduction)**

For all time points t, if information element J is believed to hold at time point T and it is believed that I leads to J after duration D, then there exists a time point t2 information element I is believed to hold at time point T-D.

LP4 ≡
∀t1,t2,I,J,T,D
state(γ, t1) |= belief(at(J, T)) & state(γ, t1) |= belief(leads_to_after(I, J, D)) ⇒ state(γ, t2) |= belief(at(I, T-D))

Figure 5 depicts the relations between the various properties by means of an AND tree. Here, if a certain property is connected to properties at a lower level, this indicates that the properties at the lower level together logically imply the higher level property. Note: LP4G and LP5G are the *grounding*[2] variant of LP4 and LP5 respectively, which is why they are depicted in grey.

---

[2] The grounding variant of an executable property states that there is no other property with the same consequent. For example, the grounding variant of A ⇒ B states that there is no other property with B in its consequent, thus B ⇒ A can be derived.

Figure 5 shows that global property GP1 can be related (by logical relations, as often used in mathematical proof) to a set of local properties (LPs) of the reasoning methods put forward in Section 3 and 4. Note that it is not claimed here that GP1 holds for all reasoning methods, but that it holds for those methods that satisfy the lower level properties (LP1, LP4G, LP5G, LP2, and LP3). Such inter-level relations can be useful for *diagnosis of dysfunctioning* of a reasoning process. For example, suppose for a given reasoning trace (obtained either by simulation, such as in Section 5, or by other means, e.g. based on empirical material of an existing ambient system) that the dynamic property GP1 does not hold, i.e., not all beliefs are correct. Given the AND-tree structure in Figure 5, at least one of the children nodes of GP1 will not hold, which means that either LP1 or IP1 will not hold. Suppose by further checking it is found that IP1 does not hold. Then the diagnostic process can be continued by focusing on this property. It follows that either IP2, LP2, or LP3 does not hold. This process can be continued until the cause of the error is localised.

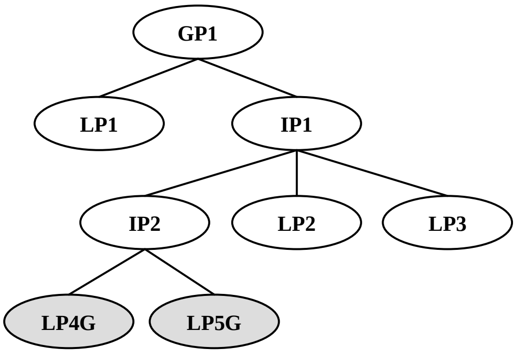


**Fig. 5.** Proof of GP1 depicted by means of an AND tree

The process mentioned above is based on the assumption that it is possible to (automatically) check any property against a trace. To this end, the TTL Checker Tool [5] can be used (and has indeed been used). For the traces presented in Section 5 all properties shown in Figure 5 were checked, and turned out to hold.

## 8 Discussion

When ambient agents need to have knowledge about human behaviours and states over time, it is useful when they possess explicitly represented causal and dynamical models about the human's processes. Once an ambient agent has such a model, a number of logical reasoning methods can be based on such a model, and formally specified as part of the agent design, as shown in this paper. The reasoning methods included cover, for example, causal and numerical simulation, qualitative reasoning and simulation, and abductive reasoning. In a number of simulation experiments example reasoning patterns were shown based on this, thus showing reusability of the ambient agent design obtained. These simulation traces have been formally analysed and verified.

In the general abductive reasoning framework, integrity constraints can be specified (see e.g. [3, 12]). Such constraints can also be specified using the approach specified in this paper, namely by incorporating these by means of the focus mechanism specified in Section 4.2. Note that the notion of a focus is not only meant to avoid integrity constraints not being satisfied, but is also meant as a way to direct the reasoning process in an appropriate and efficient way.

In [4] temporal reasoning is combined with an Active Database (ADB) for the detection of complex events in Smart Homes. The focus of that research is the combination of ADB and temporal reasoning. There is no selection mechanism in that paper as in the current work: the focus mechanism. Another example of temporal reasoning in Ambient Intelligence [19] developed a multi-agent system based on a knowledge-goal-plan (KGP) agent for transparent communication between users and an Ambient Intelligence device. They have based their reasoning model on well-known reasoning techniques such as Abductive Logic Programming and Logic Programming with Priorities. In the current work however, the focus is at developing the underlying reasoning methods that are useful in Ambient Intelligence applications.

Although the proposed reasoning methods have been applied successfully in two case studies, the examples addressed were modelled at an abstract, conceptual level. In future work, more complex and realistic case studies will be performed. In these case studies, the possibilities to incorporate the proposed reasoning methods in real artefacts in the environment will be explored. A specific question that will be addressed is to what extent the reasoning methods are able to deal with dynamic learning of new knowledge.

## References


1. Aarts, E., Collier, R., van Loenen, E., and Ruyter, B. de (eds.) (2003). *Ambient Intelligence. Proc. of the First European Symposium, EUSAI 2003*. Lecture Notes in Computer Science, vol. 2875. Springer Verlag, 2003, pp. 432.
2. Aarts, E., Harwig, R., and Schuurmans, M. (2001), Ambient Intelligence. In: P. Denning (ed.), *The Invisible Future*, McGraw Hill, New York, pp. 235-250.
3. Abdennadher, S., and Christiansen, H. (2000). An Experimental CLP Platform for Integrity Constraints and Abduction, Fourth International Conference on Flexible Query Answering Systems, FQAS 2000, Warsaw, Poland.
4. Augusto, J., and Nugent, C. (2004). The use of temporal reasoning and management of complex events in smart homes. In de Mantaras, R.L., Saitta, L., eds.: *Proceedings of European Conference on Artificial Intelligence (ECAI 2004)*, IOS Press (Amsterdam, The Netherlands) p. 778–782. August, 22-27.
5. Bosse, T., Delfos, M.F., Jonker, C.M., and Treur, J. (2006). Analysis of Adaptive Dynamical Systems for Eating Regulation Disorders. *Simulation Journal (Transactions of the Society for Modelling and Simulation)*, vol. 82, 2006, pp. 159-171.
6. Bosse, T., Gerritsen, C., and Treur, J. (2007). Integration of Biological, Psychological and Social Aspects in Agent-Based Simulation of a Violent Psychopath. In: Shi, Y., Albada, G.D. van, Dongarra, J., and Sloot, P.M.A. (eds.), *Computational Science II, Proceedings*

*of the Seventh International Conference on Computational Science, ICCS'07, Part II.* Lecture Notes in Computer Science, vol. 4488. Springer Verlag, 2007, pp. 888-895.

7. Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A, and Treur, J. (2006). Specification and Verification of Dynamics in Cognitive Agent Models. In: Nishida, T. et al. (eds.), *Proceedings of the Sixth International Conference on Intelligent Agent Technology, IAT'06*. IEEE Computer Society Press, 2006, pp. 247-254.
8. Bosse, T., Jonker, C.M., Meij, L. van der, and Treur, J. (2007). A Language and Environment for Analysis of Dynamics by Simulation. *International Journal of Artificial Intelligence Tools*, vol. 16, 2007, pp. 435-464.
9. Bosse, T., Sharpanskykh, A., and Treur, J. (2007). Integrating Agent Models and Dynamical Systems. In: Baldoni, M., Son, T.C., Riemsdijk, M.B. van, and Winikoff, M. (eds.), Proceedings of the Fifth International Workshop on Declarative Agent Languages and Technologies, DALT'07, 2007, pp.1-16.
10. Bosse, T., and Treur, J., (2007). Higher-Order Potentialities and their Reducers: A Philosophical Foundation Unifying Dynamic Modelling Methods. In: M.M. Veloso (ed.), *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence, IJCAI'07*. AAAI Press, 2007, pp. 262-267.
11. Brazier, F.M.T., Jonker, C.M., and Treur, J. (2000). Compositional Design and Reuse of a Generic Agent Model. *Applied Artificial Intelligence Journal,* vol. 14, 2000, pp. 491-538.
12. Endriss, U., Mancarella, P., Sadri, F., Terreni, G. and Toni, F. (2004). Abductive logic programming with ciff: Implementation and applications. In Proceedings *of the Convegno Italiano di Logica Computazionale CILC-2004*. University of Parma.
13. Engelfriet, J., Jonker, C.M., and Treur, J. (2002). Compositional Verification of Multi-Agent Systems in Temporal Multi-Epistemic Logic. *Journal of Logic, Language and Information,* vol. 11, 2002, pp. 195-225.
14. Engelfriet J., and Treur J. (2000). Specification of Nonmonotonic Reasoning. *Journal of Applied Non-Classical Logics*, vol. 10, 2000, pp. 7-27.
15. Green D. J. (2005). Realtime Compliance Management Using a Wireless Realtime Pillbottle - A Report on the Pilot Study of SIMPILL. In: *Proc. of the International Conference for eHealth, Telemedicine and Health, Med-e-Tel'05*, 2005, Luxemburg.
16. Gross, J.J. (ed.) (2007). *Handbook of emotion regulation*. New York: Guilford Press.
17. Josephson, J.R., and Josephson, S.G. (eds.) (1996), Abductive Inference: Computation, Philosophy, Technology. New York: Cambridge University Press.
18. Riva, G., F. Vatalaro, F. Davide, M. Alcañiz (eds.) (2005). *Ambient Intelligence*. IOS Press, 2005.
19. Stathis, K., and Toni, F. (2004). Ambient Intelligence using KGP Agents. *Lecture Notes in Artificial Intelligence*, 3295:351–362.