Model-Based Ambient Analysis of Human Task Execution

Fiemke Both, Mark Hoogendoorn, Jan Treur

Vrije Universiteit Amsterdam, Department of Artificial Intelligence, Agent Systems Research Group
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
Email: {fboth, mhoogen, treur}@cs.vu.nl URL: http://www.cs.vu.nl/~{ fboth, mhoogen, treur}

ABSTRACT
One of the challenges for ambient intelligent agents to support a human in demanding tasks, is to find out and be aware of what the human is exactly doing, and how much progress is made. Of course, in principle it would be possible to interact with the human to discover what this human is doing, but this communication can potentially slow down or even endanger task performance. In this paper an ambient agent model is presented that is able to obtain such an awareness of the human's progress in task execution by performing model-based analysis using available workflow models and available observation information. The design of the model is based on a component-based generic ambient agent model. Simulation experiments for a case study are discussed, and evaluated by automated formal verification.

Categories and Subject Descriptors
H.1.2 User/Machine Systems, Human factors. I.2.11 Distributed Artificial Intelligence, Intelligent agents.

General Terms
Design, Human Factors.

Keywords
Ambient agent, workflow model, task progress, awareness, model-based reasoning.

1. INTRODUCTION
To support a human in demanding tasks using an ambient system (e.g. [3; 2; 13]), it is useful if this system is aware of exactly what the human is doing. Using this knowledge, the ambient system can adapt the support given to the human in a dedicated way. For instance, if an ambient system is aware that a human is currently performing a very intensive task during which it is better not to disturb the human, the system might not forward certain communications coming in from other humans. Similarly, in case a human is performing a task for which the ambient system can deliver certain support, this support can be given automatically when the system is aware that the task is being performed. Sometimes a registration system may be available where it is documented on the fly at which points in time a certain (sub)task is started and at which time it is finished. In other cases a direct communication about this with the human may be possible. However, in many cases such options are not realistic as they would disturb the human's process too much. One of the challenges for ambient intelligent agents is to obtain such information about a human's task execution progress in more indirect manners, by an analysis using (possibly partial) observation information.

In some other approaches the use of objects in activities is exploited; detecting an object reveals information about the activity of the human; see, e.g., [12], [15]. In contrast, the approach put forward in the current paper does not assume any use of objects, but exploits available work flow models.

Often workflow models are nondeterministic, and the human's choice for a certain branch may depend on circumstances and on his or her preferences, which may be unknown. Therefore it may not be clear at forehand which of the branches of such a workflow model are actually followed, and at least some observation information is needed. In this paper an ambient agent model is presented that is able to perform model-based analysis using available workflow models and available observation information, to obtain more detailed information on the human's progress in task execution.

The model has been designed as a specialisation of the generic agent model for human-like ambience presented in [5], and incorporates formally specified model-based reasoning methods as presented in [4]. As part of these reasoning methods, a focussing approach was applied to control the reasoning. This allows the ambient agent to limit the amount of processing and information within itself. An important aspect of the approach developed is that model-based reasoning methods, as known within Artificial Intelligence and usually applied to causal models, are now applied to workflow models. To this end a representation format for workflow models was chosen that unifies with usual formats for model-based reasoning.

This paper is organized as follows. In Section 2 the case study is introduced. Thereafter, Section 3 introduces the formal modelling language that has been used, whereas Section 4 uses this language to express the reasoning method used. Section 5 addresses how to use focusing mechanisms, and in Section 6 a simulation model is presented. Simulation results are shown in Section 7, and verified in Section 8. Finally, Section 9 is a discussion.

2. CASE STUDY
In order to get a better understanding of the idea, the case study is introduced first. It concerns two mechanics on board of a naval vessel, namely Michael and Pete. The ambient agent of Michael has knowledge of the workflow Michael is currently in, as shown in Figure 1. Hereby, the nodes indicate states whereas the arrows represent possible transitions. Furthermore, the specific control mechanism Michael uses to go through the workflow is represented by the box in the figure, connected to the workflow via dashed arrows. It is assumed that this control mechanism is a black box, i.e. it is not known by the ambient agent. The mechanic Michael is currently trying to solve an alarm about a critical system when Pete wants to call him to ask for advice about his

own task. That triggers Michael's ambient agent to start reasoning (using the workflow) about what Michael is doing at the moment in order to determine whether to disturb him with the call or not. The current time is 10:04; the last time the ambient agent received input on what Michael was doing 9 minutes ago at 09:55, when Michael accepted the alarm and started working on it (subtask B1 in the figure below). The average time in minutes it takes Michael to perform the different processes are shown in Figure 1. This leaves two possible positions in the workflow: O5 (path 1: B1, O1, B3, O3, B5, O5) and B6 (path 2: B1, O1, B2, O2, B4, O4, B6). Furthermore, the ambient agent knows that Michael cannot be disturbed during tasks B2, B3, B4 and O5 (shown in figure 1 by the grey nodes).

The next sections describe a formal approach how the ambient agent can reason to derive what Michael has been doing.

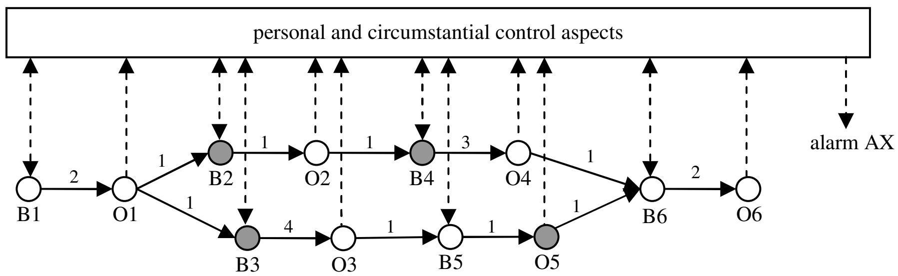

Figure 1. Representation of a workflow. The numbers above the arrows indicate the average time it takes to complete one task and start the next. The grey nodes represent subtasks during which the operator cannot be disturbed.

3. TEMPORAL TRACE LANGUAGE

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* γ over state ontology Ont and time frame T is a mapping γ : T $\rightarrow$ STATES(Ont), i.e., a sequence of states γ_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 γ over state ontology Ont, the state in γ at time point t is denoted by state(γ, t). These states can be related to state properties via the formally defined satisfaction relation $\models$, comparable to the Holds-predicate in the Situation Calculus: state(γ, t) $\models$ p denotes that state property p holds in trace γ 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 α holds for a certain time interval with duration g,
then after some delay (between e and f) state property β will hold for a certain time interval of length h.

where α and β 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.

4. MODEL-BASED REASONING

A first step in deriving what task an agent is currently doing is by specifying the workflow of the agent. The workflow is represented in the ambient agent as a collection of rules that state which node follows which node. The set of rules are of the form

belief(leads_to_after(node1, node2, duration)).

When the ambient agent believes that the human has performed a specific subtask at some time point, it has a belief of the form belief(at(node, time)). A set of generic reasoning methods has been developed to derive more beliefs based on the leads_to_after rules and one or more beliefs about active nodes; cf. [4]. Below is a summary of these methods.

4.1 Forward reasoning methods

Reasoning methods that reason forward in time 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. This varies 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 (note that for the sake of clarity the subscript below the

LEADSTO arrow has been omitted since the same values for e,f,g, and h are used namely 0,0,1,1).

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

4.2 Backward reasoning methods

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

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)) ↠ 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)) ↠ belief(at(I1,T)) ∧ belief(at(I2, T))

5. FOCUSSED REASONING

When the ambient agent uses these methods to derive more beliefs, the number of beliefs can quickly get out of control. For example, in the scenario above the ambient agent could derive beliefs about all nodes in the workflow that follow the first node, because there is no reason to select one path and not the other. Therefore, this section introduces a selection mechanism that can control which beliefs are derived and which beliefs are not. For the belief generation reasoning methods this means that an antecedent is added stating which selection criteria must be met.

This idea is shown for the reasoning method *positive forward simulation*.

> If 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))

Selection criteria needed for controlled belief generation can be specified in different manners. A simple manner is by assuming that the ambient agent has knowledge about which beliefs are relevant, expressed by a predicate in_focus. If this assumption is made, then the selection criterion s1 in the example above 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. This section explains how foci can be generated dynamically within an ambient agent.

Focussing the reasoning is useful when, for example, the ambient agent has a belief about one node being true at some time point and it only needs to know what has happened after that time point and not before that time point. Another example is when two nodes are believed to have been true. Reasoning about one of the nodes may lead to two different possible paths, while reasoning about the other node within a specific path may lead to only one possibility. In principle the ambient agent only needs to derive the beliefs necessary for fulfilling its desire (e.g. interrupting the human as in the case study described in Section 2, or giving advice).

Many types of information provide possible selection criteria. The ambient agent can use these to determine the focus. As an example, using *competences* of the human is one type of information that can be used (and has actually been used in the case study). It can be used to focus the reasoning on a path that fits most to the competence profile of the human. It can also be used to decide if and when the human may need additional information about the problem he/she is solving. A human might, for instance, be competent to use a certain set of tools, but be incompetent to use other tools.

Given that the ambient agent wants to know what the human (i.e., the mechanic) is doing now and that it knows some state in the past, there are several possible reasoning strategies to find out where the human is at the moment. The first strategy is focusing on the existing belief about the past and reasoning forward towards the current time. For this method, for example, the selection criterion competence can be used to choose between different paths. Since the starting point is a belief about a specific node and a specific time, the competences possible for this node can be taken as an indication. All nodes with the same competences can be in focus.

A second reasoning strategy is starting from the current time point and reasoning backwards until the ambient agent arrives at the belief about the past. Assumptions must be made about the possible states that the human is in now, therefore this method is useful when the number of possibilities is limited.

Beliefs about the competences of the human and about the allowed competences for the tasks are required, for example:
belief(competence_level_operator(0.8))
belief(allowed_competence(B4,0.8))
belief(allowed_competence(B5,0.4))

The value for competence represents the ability of the human to perform the action without help. Using these beliefs as selection criteria for focus generation:

```
if      belief(allowed_competence(I, X)) &
        belief(competence_level_operator(X)
then    in_focus(I, HIGH)

if      selected_focus_level(Y) &
        in_focus(I, Y)
then    in_selected_focus(I)
```

Again, any reasoning method can be used to actually derive a belief about the node in selected focus.

6. SPECIFICATION OF THE MODEL

To assess the human's task execution state, the ambient agent performs model-based reasoning of the type discussed in Section 4. However, it does so in a focused manner, where foci are determined using criteria of the type discussed in Section 5. This means that the overall reasoning pattern is an alternation of generation of a focus, and generation of beliefs within a given focus. As the criteria for focus generation have a heuristic character, it is not guaranteed that deriving beliefs within a focus will lead to a successful outcome. Therefore sometimes an earlier generated focus set has to be abandoned and a new focus set has to be generated. This is modelled by distinguishing different levels of foci. First the highest level is chosen, but if after model-based reasoning to generate beliefs this does not lead to an appropriate outcome, the highest level is disqualified and the one but highest level is chosen, and so on. More specifically, the following process is followed to model the overall reasoning process.

```
1   set highest focus level

2   while there are no selected foci
3       generate focus levels for all nodes
4       while there is no selected focus level
5           if the highest focus level is disqualified
6               selected focus level = next focus level
7           else
8               selected focus level = highest focus level
9           end if
10      end while
11      select foci for selected focus level
12  end while

13  derive beliefs based on selected foci and the model

14  if there is an observed belief inconsistent with the current foci
15      disqualify selected focus level and start focus generation
16  end if
```

The first line of the reasoning process description is applied only once at the start of the simulation. The second part, lines 2 to 12, describes the process of generating foci, selecting the highest not disqualified focus level, and selecting the foci corresponding to the selected focus level. The third part of the process, line 13, covers all generic belief generation methods described in Section 4. The last part, lines 14 to 16, continuously checks for (new) observed beliefs that are inconsistent with the current selected focus set. If there is such a belief, the current focus level is disqualified. Since selected foci are only generated when there is a selected focus level, all selected foci do not hold any longer and the reasoning process continues with line 2.

This process has been formally modelled in the directly executable LEADSTO format. Examples of formalisations for some of the lines above are:

Focus Generation based on Competence Level (line 3)

If it is believed that I needs competence level X and that the focus level for competence level X is Y and that the current phase is focus generation, then I is in focus with focus level Y

$$\forall \text{I:INFO\_EL}\ \forall \text{X:COMPETENCE\_LEVEL}\ \forall \text{Y:FOCUS\_LEVEL}$$
$$\text{belief(competence\_for(I,X))} \wedge \text{competence\_focus\_level(X,Y)} \wedge \text{focus\_generation} \twoheadrightarrow \text{in\_focus(I,Y)}$$

Selected Focus Generation based on Selected Focus Level (line 11)

If it is believed that I needs competence level X and that the focus level for competence level X is Y and that the current phase is focus generation, then I is in focus with focus level Y.

$$\forall \text{I:INFO\_EL}\ \forall \text{X: FOCUS\_LEVEL}$$
$$\text{in\_focus(I,X)} \wedge \text{selected\_focus\_level(X)} \wedge \text{focus\_generation} \twoheadrightarrow \text{in\_selected\_focus(I)}$$

Focus Level Disqualification upon Inconsistent Belief (lines 14-16)

If it is believed that I holds at T and that I is in focus at level X and that the selected focus level is Y and that X is not equal to Y and that I is not in focus at level Y, then X is disqualified and focus generation is true.

$$\forall \text{I:INFO\_EL}\ \forall \text{X,Y:FOCUS\_LEVEL}\ \forall \text{D:REAL}\ \forall \text{T:TIME}$$
$$\text{observed\_belief(at(I, T))} \wedge \text{in\_focus(I, X)} \wedge \text{selected\_focus\_level(Y)} \wedge X \neq Y \wedge \text{not(in\_focus(I, Y)} \twoheadrightarrow \text{disqualified(X)} \wedge \text{focus\_generation}$$

7. SIMULATION RESULTS

A number of scenarios have been simulated based on the case study described in Section 2. In the first scenario, the ambient agent of Michael receives a communication request from the ambient agent of Pete at time point 6. The agent has one belief about a subtask of the human at time point 1. Using this belief and the competence levels of the two possible paths, the agent first derives the bottom path: it believes the current task is B3. Since B3 is a task that cannot be interrupted, the ambient agent takes no action. After the reasoning process, the beliefs of the ambient agent are confirmed by an observation about task B3.

In scenario 2, the ambient agent receives the communication request at time point 10 and has a belief about subtask B1 at time point 1. The ambient agent focuses on the path with the highest competence level and derives beliefs about the bottom path. The conclusion is that Michael's current task is O5. Because O5 is not an interruptible task, the ambient agent does not forward the call. At some later time point, a new observation is made involving subtask O4. The agent revises its beliefs and foci, and derives the top path: it believes the current task is B6. Since B6 is a task that can be interrupted, the ambient agent transfers a new attempt by Peter to call Michael.

In scenario 3, Pete tries to contact Michael at a later time point. Therefore, the ambient agent starts reasoning at time point 12, at which the belief about subtask O4 already exists. In this scenario, the agent soon disqualifies the highest focus level, because the belief about O4 is inconsistent with the bottom path. The ambient agent derives the top path first and concludes that the human is currently performing O6. The call is transferred by the agent, because Michael can be disturbed during execution of task O6.

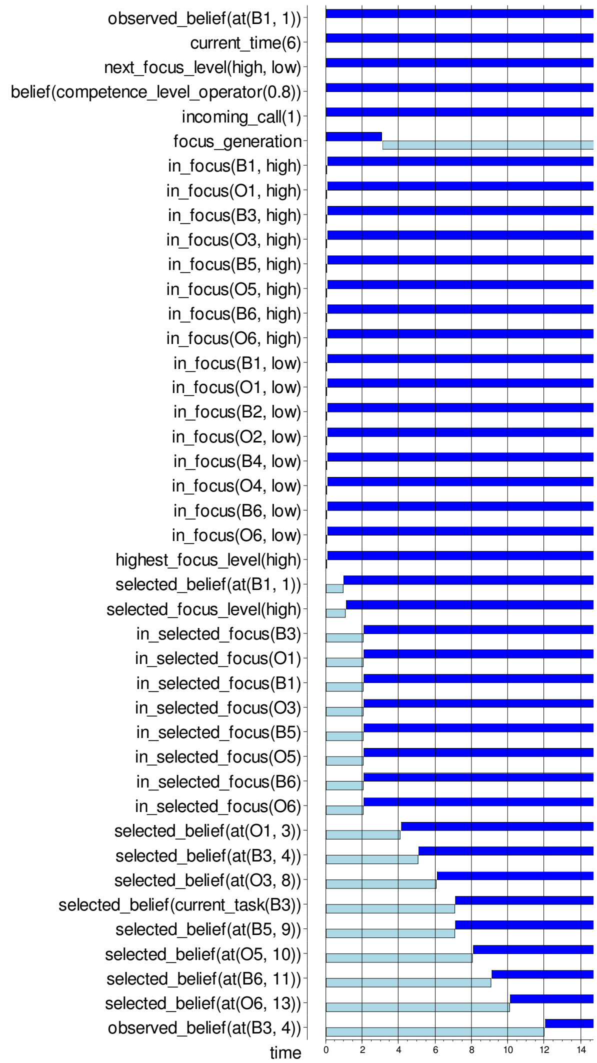

Figure 2. Example simulation trace 1.

The trace resulting from applying the presented reasoning method on the first scenario is shown in Figure 2. In the Figure, the left side indicates the atoms that occur over time, whereas the right side of the figure indicates a timeline where a dark box indicates that the atom is true, and a grey box indicates false. Note that the time on the x-axis is the simulation time which is not related to the time points in the atoms. The current time in the scenario is 6 (current_time(6)), which is an atom that is known during the entire simulation (simulation time 0 to 14). The agent starts with an observation about subtask B1 at time 1 (observed_belief(at(B1, 1))). The agent thereafter starts with the focus generation phase. In this case study, two paths exist in the workflow. Using the set of beliefs about the allowed competence levels of the two paths and a belief about the competence level of the mechanic, the ambient agent derives two sets of foci (in_focus(O1, high) and in_focus(O1, low)) for the two focus levels: high and low. Since the level 'high' is the highest and not disqualified, it is selected (selected_focus_level(high)). At the same time, the observed belief about task B1 becomes a selected belief. Now that a focus level is selected, the foci can be used to derive selected foci matching the selected focus level: all nodes in the bottom path (in_selected_focus(B1), etc.). The belief generation methods described in Section 4 are used to derive selected beliefs. When the time points of selected beliefs match the current time, the ambient agent derives that B3 is the current task of the human (selected_belief(current_task(B3))). After this derivation, the ambient agent observes that the mechanic was indeed working on subtask B3 (observed_belief(at(B3, 4))). Since B3 is a task during which Michael cannot be disturbed, the call of Pete is not forwarded to Michael.

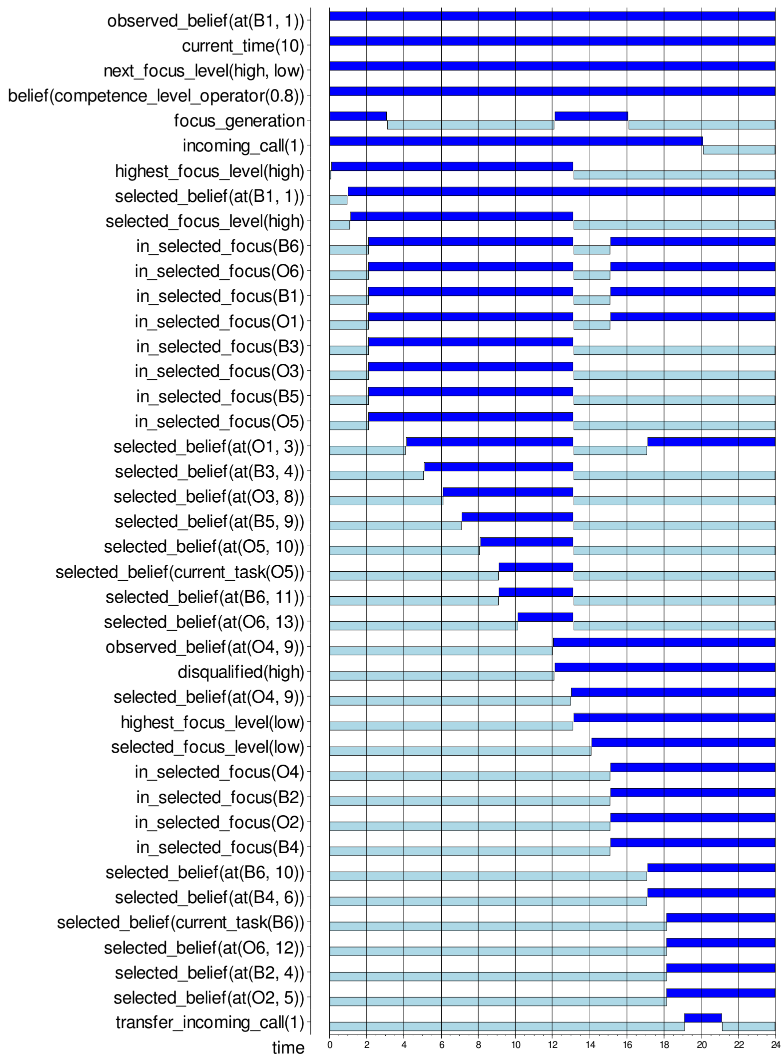

Figure 3. Example simulation trace 2.

In Figure 3, the trace of the second scenario is shown. The current time is 9 minutes after the time of the first belief (current_time(10) and observed_belief(at(B1, 1))). The first half of this trace is the same as trace 1, except for the current time and current task: because the current time is 10, the ambient agent believes that the current task is the interruptible task O5. The two sets of foci the agent derives are not shown in this figure. The ambient agent does not forward the call from Pete to Michael at time 10.

At simulation time 12, a new observation is made by the agent that subtask O4 was executed at time 9 (observed_belief(at(O4, 9))). This belief is inconsistent with the selected focus level, because the competence level for O4 is low. Therefore, the focus level 'high' is disqualified and all beliefs and foci based on this level do not hold any longer. The focus generation phase starts again, the other focus level is selected, new selected foci are derived and the agent reasons about the top path. The ambient agent now derives that the current task of the human is B6. Michael can be interrupted during execution of B6, therefore the agent transfers the call from Pete. The trace of the third scenario has been omitted for the sake of brevity.

8. VERIFICATION OF PROPERTIES

For the model a number of overall properties have been identified, formally specified and automatically verified against the simulation traces of the three scenarios:

P1: Observed facts will become selected beliefs

If at time point t a new observed belief exists, and the current set of derived beliefs does not comply to this observed belief, then eventually the set of derived beliefs will change such that it complies to the observed beliefs.

∀γ:TRACE, t:TIME, I:INFO_EL, T:TIME
[[state(γ, t) |= observed_belief(at(I, T)) &
¬state(γ, t-1) |= observed_belief(at(I, T)) &
¬state(γ, t) |= selected_belief(at(I, T))]
⇒ ∃t2:TIME ≥ t state(γ, t2) |= selected_belief(at(I, T))]

This property is satisfied for all three traces considered.

P2: Derived beliefs are within one path in the workflow model

When a set of beliefs is derived, then this set contains at most one execution path of the workflow.

∀γ:TRACE, t:TIME, I1, I2:INFO_EL, T1, T2:TIME
[state(γ, t) |= selected_belief(at(I1, T1)) &
state(γ, t) |= selected_belief(at(I2, T2)) &
I1 ≠ I2 & T2 > T1]
⇒ path_between(γ, I1, I2)

Hereby (with te the last time point of the trace),

path_between(γ, I1, I2) ≡ state(γ,te) |= path_between(I1, I2)

Here state properties path_between(I1, I2) can be generated in the trace as additional information (i.e., a form of trace enrichment), by:

belief(leads_to_after(I1, I2, D)) ↠ path_between(I1, I2)
belief(leads_to_after(I1, I2, D)) ∧ path_between(I2, I3)
↠ path_between(I1, I3)

This information is easily derived during or after the simulation using the predicates already present in the trace.

This property distinguishes traces without revision from traces with revision. Property 2 is satisfied for trace 1 and 3, but not for all time points in trace 2. Due to the occurrence of a new observation result, temporary a node in another path is selected as a belief, resulting in the property not being satisfied. At the last time point however, the property is satisfied, since then the inconsistencies are no longer present.

P3: Most plausible beliefs are generated first

If one path is more likely to be followed than another path according to the background information, and there are no observed beliefs that distinguish either path, then this path will be explored first by deriving beliefs along that path.

∀γ:TRACE, t:TIME, I, J:INFO_EL,
T:TIME, D:DURATION, V:VALUE
[state(γ, t) |= selected_belief(at(I, T)) &
state(γ, t) |= belief(leads_to_after(I, J, D)) &
state(γ, t) |= belief(competence_for(J, V)) &
consistent_with_observations(γ, t, J, T+D) &
¬∃J2:INFO_EL, D2:DURATION, V2:VALUE
[state(γ, t) |= belief(leads_to_after(I, J2, D2)) &
state(γ, t) |= belief(competence_for(J2, V2) & V2 > V &
consistent_with_observations(γ, t, J2, T+D2)]
⇒ ∃t2:TIME state(γ, t2) |= selected_belief(at(J2, T+D))]

Here:
consistent_with_observations(γ:TRACE, t:TIME,
I:INFO_EL, T:TIME) ≡
[∀I2:INFO_EL, T2:TIME state(γ, t) |= observed_belief(at(I2, T2)) ⇒
[[I = I2 & T = T2] OR path_between(γ, I2, I) OR
path_between(γ, I, I2)]

This property is satisfied for all traces

P4: Only beliefs consistent with observations are derived

If beliefs are derived, then these beliefs need to be consistent with the observations.

∀γ:TRACE, t:TIME, I:INFO_EL, T:TIME
state(γ, t) |= selected_belief(at(I, T)) ⇒
consistent_with_observation(γ, t, I, T)

Also this property distinguishes traces without revision from traces with revision. This property is satisfied for trace 1 and 3, but not for all time points in trace 2, due to the updating process of beliefs as mentioned for property P2. Before the observation is received, and after the reasoning has been conducted, the property is satisfied.

P5: Immediate correct derivation

If a set of beliefs is derived at time t, then there does not exist a time t' > t where the derived belief is not consistent with the observations.

∀γ:TRACE, t:TIME, I:INFO_EL, T:TIME
[state(γ, t) |= selected_belief(at(I, T))
⇒ ∀t2:TIME > t consistent_with_observation(γ, t2, I, T)]

Again, this property distinguishes traces without revision from traces with revision. It is satisfied for traces 1 and 3, but not for trace 2 since in that case a conflicting belief comes in that requires revision.

9. DISCUSSION

This paper addressed one of the challenges for ambient intelligent agents (e.g., [3; 2; 13]), to support a human in demanding tasks, namely to be aware of which (sub)tasks the human is doing, and how much progress is made, without direct communication about this. Some other work on subjects related to this theme can be found in, for example, [11], [12], [14], [15].

A formally specified executable ambient agent model was introduced that is able to perform model-based analysis using available workflow models and available observation information. Thus it obtains awareness of the human's progress in task execution. Some simulation experiments for a case study concerning the question what the human is doing at the current time point have been discussed and evaluated. This information has been used by the ambient agent to decide whether the human can be disturbed at this time point or not. In case it was derived that the human was working on a task during which it could not be disturbed, calls were not forwarded whereas in the other cases the calls were forwarded. Another question that the ambient agent can ask itself in the context of the case study is: what has the human been doing until now? The ambient agent might want to find out which path the human has taken to reach his/her goal in order to store this information as experience, for example, or to store this as information for future reference (when the same workflow needs to be executed advice can be given about the previous solution). Yet another question that can be addressed is: what should the human be doing in the future? If the ambient agent needs to determine the shortest path between two states, it can use a breadth-first search mechanism. The ambient agent determines the next location that can be reached in all paths after a fixed amount of time (1 minute for example). The path that reaches the last node first is the shortest one.

The model was designed as a specialisation of the generic agent model for human-like ambience presented in [5], which also was taken as a point of departure for an ambient agent model for assessment of driving behaviour; cf. [6]. Furthermore, it includes formally specified model-based reasoning methods adopted from [4]. In the current approach it is assumed that the observations are deterministic. Another option would be to obtain observations with a certain probability. The approach presented in this paper can handle such probabilistic observations by simply assuming the one with the highest probability. More advanced methods are future work.

Specification of the workflow of a human can be done using a variety of methods, e.g. [9] and [10]. For a comparison of these methods, see [1]. The advantage of the representation used throughout the current paper is that it is a rich specification format, enabling the usage of the generic model-based reasoning methods as specified in this paper.

In other work such as [11] and [14] also temporal relationships between activities are exploited to recognize plan execution states, based on relational markov networks and causal networks, respectively. These papers address the theme from a probabilistic angle, whereas the current paper addresses it from a logical reasoning perspective. For future work it will be interesting to explore how such probabilistic approaches can be integrated within a logical reasoning approach. Other approaches exploit the use of objects in activities; e.g., [12], [15]. In contrast, the approach put forward in the current paper does not assume any use of objects, but exploits available work flow models. However, when additional information is available via the use of objects, it can be integrated within the reasoning approach as observations.

In further future work, extensions will be made on the model and the application of it. For example, different focus generation methods will be explored. Moreover, within focus generation different domain aspects will be incorporated, such as statistical information about how often in the past a certain path was chosen. For example, if one path was followed in 90% of the cases, the ambient agent can use that information to focus on that path first. A challenge here is the ambient learning part: for the ambient agent to learn which path is often taken, without disturbing the human with questions.

10. REFERENCES

[1] Aalst, W.M.P. van der, Hofstede, A.H.M. ter, Kiepuszewski, B., and Barros, A.P. Workflow Patterns. QUT Technical report FIT-TR-2002-02, Queensland University of Technology, Brisbane, 2002.

[2] Aarts, E.; Collier, R.; van Loenen, E.; 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.

[3] Aarts, E., Harwig, R., and Schuurmans, M. (2001). Ambient Intelligence. In: P. Denning (ed.), *The Invisible Future*, McGraw Hill, New York, pp. 235-250.

[4] Bosse, T., Both, F., Gerritsen, C., Hoogendoorn, M., and Treur, J., Model-Based Reasoning Methods within an Ambient Intelligent Agent Model. In: *Proceedings of the First International Workshop on Human Aspects in Ambient Intelligence*. Published in: M. Mühlhäuser, A. Ferscha, and E. Aitenbichler (eds.), *Constructing Ambient Intelligence: AmI-07 Workshops Proceedings*. Communications in Computer and Information Science (CCIS), vol. 11, Springer Verlag, 2008, pp. 352-370.

[5] Bosse, T., Hoogendoorn, M., Klein, M., and Treur, J., (2008a). An Agent-Based Generic Model for Human-Like Ambience. In: *Proceedings of the First International Workshop on Model Driven Software Engineering for Ambient Intelligence Applications*. Published in: M. Mühlhäuser, A. Ferscha, and E. Aitenbichler (eds.), *Constructing Ambient Intelligence: AmI-07 Workshops Proceedings*. Communications in Computer and Information Science (CCIS), vol. 11, Springer Verlag, 2008, pp. 93-103.

[6] Bosse, T., Hoogendoorn, M., Klein, M., and Treur, J., (2008b) A Component-Based Ambient Agent Model for Assessment of Driving Behaviour. In: *Proceedings of the 5th International Conference on Ubiquitous Intelligence and Computing*. Lecture Notes in Computer Science, Springer Verlag, 2008, to appear.

[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.), *Proc. 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] Ellis, C.A. and Nutt, G.J.. Modelling and Enactment of Workflow Systems. In M. Ajmone Marsan, editor, Application and Theory of Petri Nets 1993, Lecture Notes in Computer Science, vol. 691. Springer Verlag, Berlin, 1993, pp. 1-16.

[10] Georgakopoulos, D., Hornick, M., and Sheth, A.. An Overview of Workow Management: From Process Modeling to Workow Automation Infrastructure. Distributed and Parallel Databases, 3:119-153, 1995.

[11] Liao, L., Fox, D., and Kautz, H., Location-Based Activity Recognition using Relational Markov Networks. In: Proceedings of the Nineteenth International Conference on Artificial Intelligence, IJCAI'05, 2005.

[12] Patterson, D. Fox, D. Kautz, H. and Philipose, M., Fine-Grained Activity Recognition by Aggregating Abstract Object Usage. In: Noma, H., Kogure, K., Nakajima, Y., Shimonomura, H., Ohsuga, M. (eds.), Wearable Computers 2005, Proc. of the IEEE 9th International Symposium on Wearable Computers, ISWC'05. IEEE Computer Society Press, 2005, pp. 44-51.

[13] Riva, G., F. Vatalaro, F. Davide, M. Alcañiz (eds.) (2005). *Ambient Intelligence*. IOS Press.

[14] Qin, X., and Lee, W., Attack Plan Recognition and Prediction Using Causal Networks. In: Proc. of the 20th Annual Conference on Computer Security Applications, 2004, pp. 370- 379.

[15] Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., and Rehg, J.M., A Scalable Approach to Activity Recognition based on Object Use. In: Proc. of the 11th Int. IEEE Conference on Computer Vision, ICCV'07, 2007, pp. 1-8.