FORMAL ANALYSIS OF INTELLIGENT AGENTS FOR MODEL-BASED MEDICINE USAGE MANAGEMENT

Mark Hoogendoorn, Michel Klein, Zulfiqar Memon, and Jan Treur
Vrije Universiteit Amsterdam, Department of Artificial Intelligence,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{mhoogen, m.klein, za.memon, treur}@few.vu.nl

Abstract: A model-based agent system model for medicine usage management is presented and formally analysed. The model incorporates an intelligent ambient agent model that has an explicit representation of a dynamical system model to estimate the medicine level in the patient's body by simulation, is able to analyse whether the patient intends to take the medicine too early or too late, and can take measures to prevent this.

1 INTRODUCTION

A challenge for medicine usage management is to achieve in a non-intrusive manner that patients for whom it is crucial that they take medicine regularly, indeed do so. Examples of specific relevant groups include independently living elderly people, psychiatric patients or HIV-infected persons. One of the earlier solutions reported in the literature provides the sending of automatically generated SMS reminder messages to a patient's cell phone at the relevant times; e.g., (Safren, Hendriksen, Desousa, Boswell, and Mayer, 2003). A disadvantage of this approach is that patients are disturbed often, even if they do take the medicine at the right times themselves, and that due to this after some time a number of patients start to ignore the messages.

A more sophisticated approach can be based on a recently developed automated medicine box that has a sensor that detects whether a medicine is taken from the box, and can communicate this to a server; cf. SIMpill (Green, 2005). This enables to send SMS messages only when at a relevant point in time no medicine intake is detected. A next step is to let a computing device find out more precisely what relevant times for medicine intake are. One way is to base this on prespecified prescription schemes that indicate at what time points medicine should be taken. However, this may be inflexible in cases that a patient did not follow the scheme precisely. To obtain a more robust and flexible approach, this paper explores and analyses possibilities to use an automated medicine box in combination with model-based intelligent agents to dynamically determine the (estimated) medicine level over time.

The agent-based model for medicine usage management discussed was formally specified in an executable manner and formally analysed using dedicated tools. The system incorporates a model-based intelligent agent that includes an explicitly represented dynamic system model to estimate the medicine level in the patient's body by simulation. Based on this it is able to dynamically determine at what point in time the patient should take medicine, and given that, to analyse whether the patient intends to take medicine too early or too late, and to take measures to prevent this.

In this paper, Section 2 describes the multi-agent system introduced, whereas Section 3 present detailed information about the specific agents. Furthermore, Section 4 presents simulation results, and Section 5 formal analysis of these results. Finally, Section 6 is a discussion.

2 OVERVIEW OF THE SYSTEM

Figure 1 presents an overview of the entire system as considered. The top right corner shows the patient, who interacts with the medicine box, and communicates with the patient cell phone. The Medicine Box detects whether medicine is taken out

of the medicine box. The Medicine Box Agent (MBA) observes this medicine box. In case, for example, the patient intends to take the medicine too soon after the previous dose, it finds out that the medicine should not be taken at the moment (i.e., the sum of the estimated current medicine level plus a new dose is too high), and communicates a warning to the patient by a beep. Furthermore, all information obtained by this agent is passed on to the Usage Support Agent (USA). All information about medicine usage is stored in the patient database by this agent. If the patient tries to take the medicine too early, a warning SMS with a short explanation is communicated to the cell phone of the patient, in addition to the beep sound already communicated by the Medicine Box Agent.
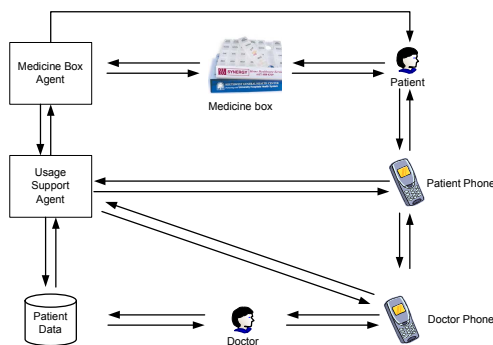

Figure 1. Multi-Agent System: Overview

On the other hand, in case the Usage Support Agent finds out that the medicine is not taken early enough (i.e., the medicine concentration is estimated too low for the patient and no medicine was taken yet), it can take measures as well. First of all, it can warn the patient by communicating an SMS to the patient cell phone. This is done soon after the patient should have taken the medicine. In case that after some time the patient still does not take medicine, the agent can communicate an SMS to cell phone of the appropriate doctor. The doctor can look into the patient database to see the medicine usage, and in case the doctor feels it is necessary to discuss the state of affairs with the patient, he or she can contact the patient via a call using the doctor cell phone to the patient cell phone.

3 AGENT PROPERTIES

The model used for the Usage Support Agent (USA) makes (re)use of elements of the Generic Agent Model GAM described in (Brazier, Jonker, and Treur, 2000). In addition, it makes use of an explicitly represented dynamical model to for the medicine level over time within the patient. Moreover, the model for the Usage Support Agent includes a reasoning method (based on simulation) to estimate the current medicine level based on the dynamical model and information on medicine taking in the past.

To express the agent's internal states and processes, a state ontology partly shown in Table 1 was specified. An example of an expression that can be formed by combining elements from this ontology is

belief(leads_to_after(I:INFO_EL, J:INFO_EL, D:REAL))

which expresses that the agent has the knowledge that state property I leads to state property J with a certain time delay specified by D. This type of expression is used to represent the agent's knowledge of a dynamical model of a process. Using the ontology, the functionality of the agent has been specified by generic and domain-specific temporal rules.

Formalisation	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
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. Ontology for the Usage Support Agent Model

Note the formal form of the agent properties has been omitted below for the sake of brevity. Generic rules specify that incoming information (by observation or communication) from a source that is believed to be reliable is internally stored in the form of beliefs. When the sources are assumed always reliable, the conditions on reliability can be left out:

IB(X) From Input to Beliefs
If agent X observes some world fact, then it will believe this.
If X gets information communicated, then it will believe this.
Execution of a dynamical model is specified by:

SE(X) Simulation Execution
If it is believed that I holds at time T, and it is believed that I leads to J after time duration D, then it is believed that J holds at time T+D.

This temporal rule specifies how a dynamic model that is represented as part of the agent's knowledge can be used by the agent to extend its beliefs about the world at different points in time.

Domain-specific rules for the Usage Support Agent are shown below. The Usage Support Agent's specific functionality is described by three sets of temporal rules. First, the agent maintains a dynamic model for the concentration of medicine in the patient

over time in the form of a belief about a *leads to after* relation.

USA1: Maintain dynamic model
The Usage Support Agent believes that if the medicine level for medicine M is C, and the usage effect of the medicine is E, then after duration D the medicine level of medicine M is C+E minus G*(C+E)*D with G the decay value. Formally:

In order to reason about the usage information, this information is interpreted (USA2), and stored in the database (USA3).

USA2: Prepare storage usage
If the agent has a belief concerning usage of medicine M and the current time is T, then a belief is generated that this is the last usage of medicine M, and the intention is generated to store this in the patient database.

USA3: Store usage in database
If the agent has the intention to store the medicine usage in the patient database, then the agent performs this action.

Finally, temporal rules were specified for taking the appropriate measures. Three types of measures are possible. First, in case of early intake, a warning SMS is communicated (USA4). Second, in case the patient is too late with taking medicine, a different SMS is communicated, suggesting to take the medicine (USA5). Finally, when the patient does not respond to such SMSs, the doctor is informed by SMS (USA6).

USA4: Send early warning SMS
If the agent has the belief that an intention was shown by the patient to take medicine too early, then an SMS is communicated to the patient cell phone that the medicine should be put back in the box, and the patient should wait for a new SMS before taking more medicine.

USA5: SMS to patient when medicine not taken on time
If the agent has the belief that the level of medicine M is C at the current time point, and the level is considered to be too low, and the last message has been communicated before the last usage, and at the current time point no more medicine will be absorbed by the patient due to previous intake, then an SMS is sent to the patient cell phone to take the medicine M.

USA6: SMS to doctor when no patient response to SMS
If the agent has the belief that the last SMS to the patient has been communicated at time T, and the last SMS to the doctor was communicated before this time point, and furthermore, the last recorded usage is before the time point at which the SMS has been sent to the patient, and finally, the current time is later than time T plus a certain delay parameter for informing the doctor, then an SMS is communicated to the cell phone of the doctor that the patient has not taken medicine M.

USA7: Communicate Current Concentration
If the agent has the belief that the level of medicine M is C at the current time point then the agent informs the medicine box agent about this level.

The Medicine Box Agent has functionality concerning communication to both the patient and the Usage Support Agent. Generic temporal rules are included as for the Usage Support Agent (see above).

Domain-specific temporal rules are both shown below. First of all, the observed usage of medicine is communicated to the Usage Support Agent in case the medicine is not taken too early, as specified in MBA1.

MBA1: Medicine usage communication
If the Medicine Box Agent has a belief that the patient has taken medicine from a certain position in the box, and that the particular position contains a certain type of medicine M, and taking the medicine does not result in a too high medicine concentration of medicine M within the patient, then the usage of this type of medicine is communicated to the USA.

In case medicine is taken out of the box too early, a warning is communicated by a beep and the information is forwarded to the Usage Support Agent (MBA2 and MBA3).

MBA2: Too early medicine usage prevention
If the Medicine Box Agent has the belief that the patient has taken medicine from a certain position in the box, that this position contains a certain type of medicine M, and taking the medicine results in a too high medicine concentration of medicine M within the patient, then a warning beep is communicated to the patient.

MBA3: Early medicine usage communication
If the Medicine Box Agent has a belief that the patient was taking medicine from a certain position in the box, and that the particular position contains a certain type of medicine M, and taking the medicine would result in a too high concentration of medicine M within the patient, then this is communicated to the Usage Support Agent.

4 SIMULATION

In order to show how the above presented system functions, the executable properties have been implemented in a dedicated software environment that can execute such specifications (Bosse, Jonker, Meij, and Treur., 2007). To enable creation of simulations, a patient model is used that simulates the behaviour of the patient in a stochastic manner. The model specifies four possible behaviours of the patient, each with its own probability: (1) too early intake, (2) correct intake (on time), (3) responding to an SMS warning that medicine should be taken, and (4) responding to a doctor request by phone. Based upon such probabilities, the entire behaviour of the patient regarding medicine usage can be simulated. In the following simulations, values of respectively 0.1, 0.8, 0.9 and 1.0 have been used.

Figure 2 shows an example of a simulation trace whereby the medicine support system is active. The figure indicates the medicine level over time as estimated by the agent based on its dynamic model. Here the x-axis represents time whereas the y-axis represents the medicine level.
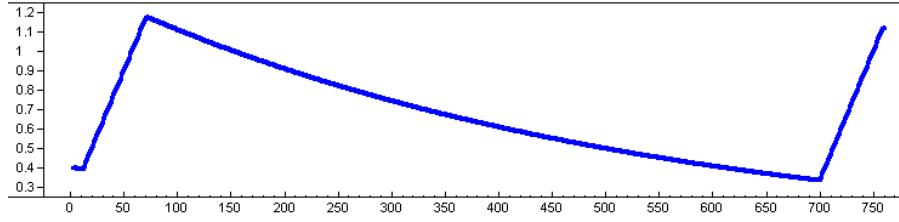
Figure 2. Medicine level over time

5 FORMAL ANALYSIS

Note that in this case, the minimum level of medicine within the patient is set to 0.35 whereas the maximum level is 1.5. These numbers are based on the medicine half-life value, that can vary per type of medicine. For more details on the formal properties behind the simulation, and a more elaborate discussion of the results, see (Hoogendoorn, Klein, and Treur, 2007).

5 FORMAL ANALYSIS

When a model such as the one described above, has been specified, it is easy to produce various simulations based on different settings, initial conditions and external events offered. Moreover, it is possible to incorporate nondeterministic behaviours by temporal rules that involve probabilistic effects (cf. Bosse, *et al.,* 2007). Thus large sets of traces can be generated. When such a set is given, it is more convenient to check them on relevant properties automatically, than going through them by hand. Furthermore, it may also be useful when insight is provided how dynamic properties of the multi-agent system as a whole depend on dynamic properties of the agents within the system, and further on, how these relate to properties of specific components within the agents. This section shows how this can be achieved. To this end a number of *dynamic properties* have been specified for different aggregation levels of the multi-agent system, cf. (Jonker and Treur 2002; Bosse, Jonker, Meij, Sharpanskykh, and Treur, , 2006). The main property considered for the system as a whole is: will the medicine level in the patient be maintained between the required minimum and maximum level? This desired situation is called 'S'. That a value V of a variable P should be within a specific range between the lower threshold TL and the upper threshold TU, is specified as follows:

has_value(P,V) ∧ (V > TU ∨ V < TL)   (S)

This has been applied to the variable 'medicine_level'.

GP1 At any point in time the medicine level is between TL and TU.

∀T:TIMEPOINT, V:REAL:
    state(M, T) |= has_value(P,V) ⇒ (V ≤ TU ∧ V ≥ TL)

Here M is a trace, and state(M, T) denotes the state in this trace at time T. Moreover, state(M, T) |= p denotes that state property p holds in state state(M, T).

Related to this, global properties can be defined that specify that the total number of violations of the threshold values is smaller than some maximum number, or that the total duration of the violation is smaller than some maximum period. In these definitions Σ case(p, 1, 0) is a special construct in the language that calculates the sum of timepoints for which a state holds.

GP2 The total number of times that the medicine level falls below TL or raises above TU is smaller than MAX_OCCURANCES.

∀M:TRACE: ∀T1, T2:TIMEPOINT:
  Σ case(T1≤T & T ≤T2 & state(M, T) |= S &
      state(M, T+1) |= ¬S, 1, 0) < MAX_OCCURANCES

GP3 The total time period that the medicine level is not between TL and TU is smaller than MAX_DURATION.

∀M:TRACE ∀T1, T2:TIMEPOINT:
  Σ case(T1≤T & T ≤T2 & state(M, T) |= ¬S, 1, 0) < MAX_DURATION

5.1 Evaluation of Traces

The formal properties have been used to evaluate the usefulness of the medicine usage management system. For this, 60 simulation traces of the medicine level within a patient have been generated, with a length of 36 hours. In half of the traces the medicine usage management system was supporting the patient, in the other half the system did not take any action, but was still monitoring the medicine usage. As a consequence of the stochastic patient model (the probabilities used are the ones mentioned in Section 5), this resulted in 60 different simulation traces.

For all traces it has been checked whether, how often and how long, the medicine level is between the required values of 0.5 and 1.35. It has also been checked whether this is the case for *preferred* values. It can be assumed that, in addition to the required range for the medicine level, there is also an optimal or preferred range. Table 2 lists the total number of violations in the 30 traces with support of the system and the 30 traces without support for different maximum and minimum levels. Table 3 shows the

duration of the violations time in minutes for the same set of traces.

number of violations	with support	without
above 1.5 (required)	2	8
below 0.35 (required)	5	18
total required	**7**	**26**

Table 2. Total number of violations of the threshold values (used in property GP2).

duration (in minutes)	with support	without
above 1.5 (required)	0.8	25.36
below 0.35 (required)	3.53	179.13
total required	**4.33**	**204.49**
above 1.2 (preferred)	161.6	328.20
below 0.5 (preferred)	218.1	327.63
total preferred	**379.7**	**655.83**

Table 3. Total duration of violations of the threshold values (used in property GP3).

From the figures in tables it is immediately apparent that the medicine level is much more often and for a much longer time between the required or preferred values. However, it is also clear that even with support of the system the medicine level is sometimes outside the required range. In fact, property GP1 did not hold in 5 out of the 30 simulation traces in which the system was active. An analysis of these traces revealed that this is a side-effect of the specification of the simulation: as every communication between agents and between components within agents costs at least one time-step, it takes a number of time-steps before a message of the system has reached the patient (in fact 24 minutes). In between sending and receiving a message, it could happen that the medicine level has dropped below the minimum value, or that the patient has taken a pill already. For violations of the minimum level it is possible to compensate for this artificial delay by allowing an additional decrease that is equivalent to the decay of the medicine during the time of the delay. This means that medicine level should not drop below the 0,3335 if the delay is taken into account. Table 4 shows the duration of the violations for the corrected minimum level. In this case, there are no violations of the lower threshold in traces where the system is active.

Unfortunately, a similar correction for violations of the maximum level is not possible, as these violations are caused by taking two pills within a short period, which is a non-monotonic effect.

duration (in minutes)	with support	without
below 0.3335	**0**	**164.77**

Table 4. Corrected values for duration the violations of the minimum level.

5.2 Relating Global Properties to Executable Properties

Besides the verification of properties against simulation traces, the correctness of the entire model can also be proven (given certain conditions). This proves that for all possible outcomes of the model the global properties are indeed achieved under these specific conditions. Such correctness of a model can be proved using the SMV model checker (MacMillan, 1995). In order to come to such a proof, an additional property level is introduced, namely the external behavioural properties for the components within the system. Thereafter, the relationship between the executable properties of the Medicine Box Agent, and the Usage Support Agent are related to these behavioural properties, and furthermore, these behavioural properties are shown to entail the top-level properties.

5.2.1 External Behavioural Properties

First a number of external properties for the Usage Support Agent (USA) are introduced. The first property specifies that the patient should be warned that medicine should be taken in case the medicine level is close to being too low (EUSA1). Secondly, property EUSA2 specifies that the USA should warn the doctor in case such a warning has been sent to the patient, but there has been no response. Moreover, the storage of the usage history is specified in EUSA3, and the sending of an early warning message is addressed in EUSA4. Finally, EUSA5 describes that the USA should communicate the current medicine level within the patient. Note that in all properties a parameter e is used, which specifies the maximum delay after which these communications or actions should occur. Such a parameter can vary per property, and is used throughout this section for almost all non-executable properties.

EUSA1: Usage Support Agent Behaviour External View Patient Warning Take Medicine
If the Usage Support Agent received communicated medicine intake, and based on these, the estimated accumulated concentration is C, and C<TL, then it communicates an SMS to the Patient Cell Phone that medicine should be taken.

$\forall$t:TIME, γ:TRACE, M:MEDICINE, C:REAL
history_implied_value(γ, input(USA), t, M, C) & C < TL $\Rightarrow$
$\exists$t' t$\leq$ t'$\leq$ t+e & state(γ, t', output(USA)) |=
    communication_from_to(sms_take_medicine(M),
    usage_support_agent, patient_cell_phone)

EUSA2: Usage Support Agent Behaviour External View Doctor Warning
If the Usage Support Agent sent out a warning message to the patient, and the patient did not take medicine within X time steps after the warning, the Usage Support Agent sends a message to the Doctor Cell Phone.

∀t:TIME, γ:TRACE state(γ, t, output(USA)) |=
        communication_from_to(sms_take_medicine(M),
        usage_support_agent, patient_cell_phone) &
¬∃t2:TIME [t2 ≥ t & t2 < t + X & state(γ, t2, input(USA)) |=
        communicated_from_to(medicine_used(M),
        medicine_box_agent, usage_support_agent)]
⇒ ∃t':TIME t + X ≤ t'≤ (t + X) +e state(γ, t', output(USA)) |=
        communication_from_to(sms_not_taken_medicine(M),
        usage_support_agent, doctor_cell_phone)

EUSA3: Usage Support Agent   Behaviour External View Store Information in Database

If the Usage Support Agent receives a communication that medicine has been taken, then the agent stores this information in the patient database.

∀t:TIME, γ:TRACE, M:MEDICINE   state(γ, t, input(USA)) |=
    communicated_from_to(medicine_used(M), medicine_box_agent,
    usage_support_agent) &
⇒ ∃t':TIME t ≤ t'≤ t + e [state(γ, t', output(USA)) |=
        performing_in(store_usage(M, t), patient_database)]

EUSA4: Usage Support Agent   Behaviour External View Store Send Early Warning Message to Patient

If the Usage Support Agent receives a communication that the patient attempted to take medicine too early, then the agent sends an SMS to the patient cell phone.

∀t:TIME, γ:TRACE, M:MEDICINE   state(γ, t, input(USA)) |=
        communicated_from_to(too_early_intake_intention,
        medicine_box_agent, usage_support_agent) &
⇒ ∃t':TIME t ≤ t'≤ t + e [state(γ, t', output(USA)) |=
    communication_from_to(put_medicine_back_and_wait_for_signal,
    usage_support_agent, patient_cell_phone)]

EUSA5: Usage Support Agent Behaviour External View Send Approximated Concentration to Medicine Box Agent

If the history of medicine usage implies a certain medicine level, then the Usage Support Agent communicates this value to the Medicine Box Agent.

∀t:TIME, γ:TRACE, M:MEDICINE, C:REAL
history_implied_value(γ, input(USA), t, M, C) ⇒
∃t' t ≤ t'≤ t+e &   state(γ, t', output(USA)) |=
        communication_from_to(medicine_level(M, C),
        usage_support_agent, medicine_box_agent)

Besides the Usage Support Agent, the Medicine Box Agent (MBA) plays an important role within the system as well. From an external perspective, the MBA has three behavioural properties. The first (EMBA1) expresses that the MBA should communicate to the Usage Support Agent that medicine has been taken by the patient. This only occurs when the medicine is not taken too early. Properties EMBA2 and EMBA3 concern the communication in case of an early intake. First of all, the MBA should sound a beep (EMBA2), and furthermore, the MBA should communicate this information to the USA (EMBA3). Again, a parameter *e* is used to specify the maximum delay for these properties. Note that these properties are later referred to as EMBA, which is the conjunction of the three properties specified below.

EMBA1: Medicine Box Agent   Behaviour External View Communicate Usage Non-Early Intake

When the medicine box agent observes medicine is taken from position X,Y in the box and the medicine is of type M, and the medicine level of M communicated to the agent is C, and furthermore, the medicine level plus a dose does not exceed the overall maximum, then the medicine box agent outputs medicine has been taken to the Usage Support Agent.

∀γ:TRACE, t:TIME, X, Y:INTEGER, C:REAL, M:MEDICINE
state(γ, t, input(medicine_box_agent) |=
  observed_result_from(medicine_taken_from_position(
  x_y_coordinate(X, Y)), medicine_box) &
state(γ, t, input(medicine_box_agent) |=
  communicated_from_to(medicine_level(M, C),
  usage_support_agent, medicine_box_agent) &
C+DOSE <= MAX_MEDICINE_LEVEL &
medicine_at_location(X, Y, M)
⇒ ∃t':TIME t ≤ t'≤ t + e [state(γ, t', output(medicine_box_agent) |=
  communication_from_to(medicine_used(M), medicine_box_agent,
  usage_support_agent)]

EMBA2: Medicine Box Agent   Behaviour External View Communicate Beep when Early Intake

When the medicine box agent observes medicine is taken from position X,Y in the box and the medicine is of type M, and the medicine level of M communicated to the agent is C, and furthermore, the medicine level plus a dose exceeds the overall maximum, then the medicine box agent outputs a beep to the Patient.

∀γ:TRACE, t:TIME, X, Y:INTEGER, C:REAL, M:MEDICINE
state(γ, t, input(medicine_box_agent) |=
  observed_result_from(medicine_taken_from_position(
  x_y_coordinate(X, Y)), medicine_box) &
state(γ, t, input(medicine_box_agent) |=
  communicated_from_to(medicine_level(M, C),
  usage_support_agent, medicine_box_agent) &
C+DOSE > MAX_MEDICINE_LEVEL &
medicine_at_location(X, Y, hiv_slowers)
⇒ ∃t':TIME t ≤ t'≤ t + e [state(γ, t', output(medicine_box_agent) |=
  communication_from_to(sound_beep, medicine_box_agent, patient)]

EMBA3: Medicine Box Agent   Behaviour External View Communicate Early Intake to Usage Support Agent

When the medicine box agent observes medicine is taken from position X,Y in the box and the medicine is of type M, and the medicine level of M communicated to the agent is C, and furthermore, the medicine level plus a dose exceeds the overall maximum, then the medicine box agent outputs a communication concerning this early intake intention to the Usage Support Agent.

∀γ:TRACE, t:TIME, X, Y:INTEGER, C:REAL, M:MEDICINE
state(γ, t, input(medicine_box_agent) |=
  observed_result_from(medicine_taken_from_position(
  x_y_coordinate(X, Y)), medicine_box) &
state(γ, t, input(medicine_box_agent) |=
  communicated_from_to(medicine_level(M, C),
  usage_support_agent, medicine_box_agent) &
C+DOSE > MAX_MEDICINE_LEVEL &
medicine_at_location(X, Y, hiv_slowers)
⇒ ∃t':TIME t ≤ t'≤ t + e [state(γ, t', output(medicine_box_agent) |=
  communication_from_to(too_early_intake_intention,
  medicine_box_agent, usage_support_agent)]

Furthermore, a number of properties are specified for the external behavior of the other components. These properties include basic forwarding of information by the patient cell phone (PCP), the doctor cell phone (DCP), communication between various communicating components (CP). Furthermore, it includes the specification of the observation results of performing actions in the medicine box (EMD), the storage of information in the database (PDP), and the transfer of those actions and observations (WP).

Finally, in order for such external behavioral properties to establish the global property, certain assumptions need to be made concerning the

behavior of the doctor and of the patient. For the proof, the minimal behavior of the patient is used. This minimal behavior is specified by stating that the patient should at least respond to the doctor communication. Of course, most likely would be that the patient already responds to the SMS being sent, but using the minimal behavior it can already be proven that this is not even required, as long as the patient responds to the doctor communication. This ensures, that even if the patient does not respond on an SMS his medicine level will still remain within the boundaries set. The behavior of the patient is expressed in property PB. Besides the patient, also the doctor needs to respond to the SMS of the system in a particular way, namely that he immediately contacts the patient, as expressed in property DB.

PB: Respond to Doctor request

When a patient receives a doctor warning that the patient should take medicine M, then the patient takes the medicine from the appropriate place in the box.

$\forall\gamma$:TRACE, t :TIME, M:MEDICINE, X, Y:INTEGER
state(γ, t, input(patient)) |=
communicated_from_to(doctor_request_take_medicine(M),
patient_cell_phone, patient)
medicine_at_location(X, Y, M)
$\Rightarrow \exists$t':TIME t $\leq$ t'$\leq$ t + e
state(γ, t', output(patient)) |=
performing_in(take_medicine_from_position(x_y_coordinate(X, Y)),
medicine_box)

DB: Warn patient after SMS

When the doctor has received an SMS concerning a patient that has not used medicine, then the doctor uses its cell phone communicating that the patient to take medicine.

$\forall\gamma$:TRACE, t :TIME, M:MEDICINE
state(γ, t, input(doctor)) |=
            communicated_from_to(sms_not_taken_medicine(M),
            doctor_cell_phone, doctor)
$\Rightarrow \exists$t':TIME t $\leq$ t'$\leq$ t + e [state(γ, t', output(doctor)) |=
communication_from_to(doctor_request_take_medicine(M), doctor,
doctor_cell_phone)]

5.2.2   Relating Properties

The relation between the external behavioral properties introduced in Section 6.2.1 and the top-level global property is shown in Figure 4. The figure also shows the relationship between the executable properties of the Usage Support Agent, and the Medicine Box Agent and the external behavioral

properties thereof. Note that the external level of both the Usage Support Agent as well as the Medicine Box Agent has been represented in the figure as EUSA and UMBA respectively. This is simply the conjunction of the external properties of these agents. Furthermore, the external behavioral properties of the other components directly translate to executable properties in the specification. The following relations hold from the local to the external level.

Usage Support Agent External Behavior
IB & SE & USA1 & USA5 $\Rightarrow$ EUSA1
IB & USA2 & USA5 & USA6 $\Rightarrow$ EUSA2
IB & USA2 & USA3 $\Rightarrow$ EUSA3
IB & USA4 $\Rightarrow$ EUSA4
IB & SE & USA1 & USA7 $\Rightarrow$ EUSA5

Medicine Box Agent External Behavior
IB & MBA1 & MBA2 & MBA3 $\Rightarrow$ EMBA

Furthermore, the global property GP1 has the following relationship with the external properties of the various components.

Global Behavior
EMBA & EUSA & PCP & DCP & CP & EMD & PDP & WP & PB & DB $\Rightarrow$ GP1

In order to prove the above specified relationships, the relations have been checked within the SMV model checker. Using the following parameters, the inter-level relations within the tree expressed in Figure 4 are indeed shown to hold using SMV: The initial medicine level is set to 60, the reduction level per step is set to 98 (i.e. 0.98), the maximum medicine level is set to 150 and the minimum level to 30. Furthermore, the warning level is set to 60 (after which an SMS is sent to the patient). Finally, the delay for a doctor message is set to 10 time steps. Note that in the case of the SMV checks no communication delay is specified. Therefore, this formal verification shows that without this delay the model indeed entails the global property.

6 DISCUSSION

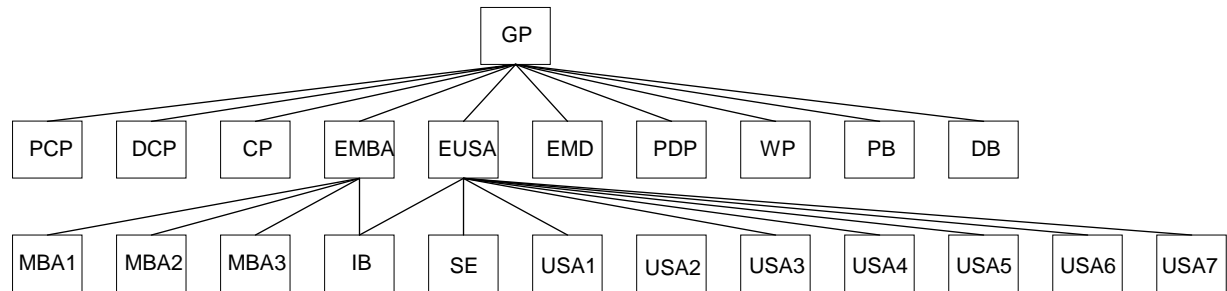In this paper, possible support in the domain of

Figure 3. Property hierarchy in the form of an AND tree

medicine usage management was addressed. A multi-agent system model that supports the users of medicine in taking their medicine at the appropriate time was discussed and formally analysed by simulation and verification. The system has been specified using a formal modeling approach which enables the specification of both quantitative as well as qualitative aspects (Jonker and Treur 2002; Bosse, Jonker, Meij, and Treur, 2006). To specify the model, both generic and domain specific temporal rules have been used, enabling reuse of the presented model. The analysis of the model has been conducted by means of (1) generation of a variety of simulation runs using a stochastic model for patients; (2) specification of dynamic properties at different aggregation levels; (3) specification of interlevel relations between these properties; (4) automated verification of properties specified in (2) against traces generated in (1), and (5) automated verification of the interlevel relations specified in (3). The simulation execution in (1) has been achieved making use of the LEADSTO software environment (cf. Bosse et al. 2007). Specification of properties in (2) and interlevel relations in (3) have been performed using the TTL software environment (cf. Bosse, *et al*, 2006), as has automated verification of such properties against set of simulation traces in (4). Verification of interlevel relations in (5) has been performed using the SMV model checking environment (MacMillan, 1995). Evaluation of the system with actual users is part of future work.

The presented analysis fits well in the recent developments in Ambient Intelligence (Aarts, Collier, Loenen, Ruyter, 2001; Aarts, Harwig, and Schuurmans, 2003; Riva *et al*., 2005). Furthermore, it also shows that multi-agent system technology can be of great benefit in health care applications, as also acknowledged in (Moreno and Nealon, 2004). More approaches to support medicine usage of patients have been developed. Both in (Greene, 2005) as well as (Floerkemeier and Siegemund, 2003) models are presented that do not simply always send an SMS that medicine should be taken such as proposed by (Safren, *et al.*, 2003). Both approaches only send SMS messages in case the patient does not adhere to the prescribed usage. The model presented in this paper however adds an additional dimension to such a support system, namely the explicit representation and simulation of the estimated medicine level inside the patient. Having such an explicit model enables the support agent to optimally support the patient; see also (Bosse, Memon, and Treur, 2007) for the use of such a model for mental processes of another agent.

REFERENCES

Aarts, E.; Collier, R.; van Loenen, E.; Ruyter, B. de (eds.) (2003). *Ambient Intelligence. Proc. EUSAI 2003*. LNCS vol. 2875. Springer Verlag, 2003, pp. 432.

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

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 IAT'06*. IEEE Computer Society Press, 2006, pp. 247-254.

Bosse, T., Jonker, C.M., Meij, L. van der, and Treur, J. (2007). A Language and Environment for Analysis of Dynamics by Simulation. *Int. Journal of Artificial Intelligence Tools*, vol. 16, 2007, pp. 435-464.

Bosse, T., Memon, Z.A., and Treur, J. (2007). A Two-level BDI-Agent Model for Theory of Mind and its Use in Social Manipulation. In: P. Olivier, C. Kray (eds.), *Proceedings of AISB'07*. AISB Publications, 2007, pp. 335-342.

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.

Floerkemeier, C., Siegemund, F. (2003). Improving the Effectiveness of Medical Treatment with Pervasive Computing Technologies. *Workshop on Ubiquitous Computing for Pervasive Healthcare Applications at Ubicomp 2003*, Seattle, October 2003

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.

Hoogendoorn, M., Klein, M., and Treur, J., Formal Design and Simulation of an Ambient Multi-Agent System Model for Medicine Usage Management. In: *Proceedings of the International Workshop on Ambient Assisted Living (AAL 2007)*, to appear, 2007.

Jonker, C.M., and Treur, J. (2002). Compositional Verification of Multi-Agent Systems: a Formal Analysis of Pro-activeness and Reactiveness. *Intl. J. of Coop. Inf. Systems*, vol. 11, 2002, pp. 51-92.

McMillan, K.L., (1993). *Symbolic Model Checking: An Approach to the State Explosion Problem*. PhD thesis, School of Computer Science, CMU, Pittsburgh, 1992. Published by Kluwer Academic Publishers, 1993.

Moreno, A., Nealon, J.L. (eds.), *Applications of Software Agent Technology in Health Care Domain*. Birkhäuser Basel, 2004.

Riva, G., F. Vatalaro, F. Davide, M. Alcañiz (eds). (2005). *Ambient Intelligence*. IOS Press, 2001.

Safren, S.A., Hendriksen, E.S., Desousa, N., Boswell, S.L., Mayer, K.H., (2003). Use of an on-line pager system to increase adherence to antiretroviral medications. In: *AIDS CARE*, vol. 15, pp. 787 – 793.