Incorporating Human Aspects in Ambient Intelligence and Smart Environments

Tibor Bosse
Vrije Universiteit Amsterdam, The Netherlands

Mark Hoogendoorn
Vrije Universiteit Amsterdam, The Netherlands

Michel Klein
Vrije Universiteit Amsterdam, The Netherlands

Rianne van Lambalgen
Vrije Universiteit Amsterdam, The Netherlands

Peter-Paul van Maanen
Vrije Universiteit Amsterdam, The Netherlands

Jan Treur
Vrije Universiteit Amsterdam, The Netherlands

Abstract
In this chapter, we propose to outline the scientific area that addresses Ambient Intelligence applications in which not only sensor data, but also knowledge from the human-directed sciences such as biomedical science, neuroscience, and psychological and social sciences is incorporated. This knowledge enables the environment to perform more in-depth, human-like analyses of the functioning of the observed humans, and to come up with better informed actions. A structured approach to embed human knowledge in Ambient Intelligence applications is presented an illustrated using two examples, one on automated visual attention manipulation, and another on the assessment of the behaviour of a car driver.

1 Human Knowledge in Ambient Intelligence

Ambient Intelligence provides possibilities to contribute to more personal care; e.g., (Aarts, Harwig, Schuurmans, 2001; Aarts, Collier, Loenen, Ruyter, 2003; Riva, Vatalaro, Davide, Alcañiz, 2005). Acquisition of sensor information about humans and their functioning is an important factor, but without adequate knowledge for analysis of this information, the scope of such applications is limited. However, devices in the environment possessing such knowledge can show a more human-like understanding and base personal care on this understanding. For example, this may concern elderly people, patients depending on regular medicine usage, surveillance, penitentiary care, psychotherapeutical / selfhelp communities, but also, for example, humans in highly demanding tasks such as warfare officers, air traffic controllers, crisis and disaster managers, and humans in space missions; e.g., (Green, 2005; Itti and Koch, 2001).

Within human-directed scientific areas, such as cognitive science, psychology, neuroscience and biomedical sciences, models have been and are being developed for a variety of aspects of human functioning. If such models of human processes are represented in a formal and computational format, and incorporated in the human environment in devices that monitor the physical and mental state of the human, then such devices are able to perform a more in-depth analysis of the human's functioning. This can result in an environment that may more effectively affect the state of humans by undertaking actions in a knowledgeable manner that improve their wellbeing and performance. For example, the workspaces of naval officers may include systems that, among others, track their eye movements and characteristics of incoming stimuli (e.g., airplanes on a radar screen), and use this information in a computational model that is able to estimate where their attention is focussed at. When it turns out that an officer neglects parts of a radar screen, such a system can either indicate this to the person, or arrange on the background that another person or computer system takes care of this neglected part. Note that for a radar screen it would also be possible to make static design changes, for example those that improve situation awareness (e.g. picture of the environment, Wickens, 2002). However, as different circumstances might need a different design, the advantage of a dynamic system is that the environment can be adapted taking both the circumstances and the real-time behaviour of the human into account.

In applications of this type, an ambience is created that has a better understanding of humans, based on computationally formalised knowledge from the human-directed disciplines. The use of knowledge from these disciplines in Ambient Intelligence applications is beneficial, because it allows taking care in a more sophisticated manner of humans in their daily living in medical, psychological and social respects. In more detail, content from the domain of human-directed sciences, among others, can be taken from areas such as medical physiology, health sciences, neuroscience, cognitive psychology, clinical psychology, psychopathology, sociology, criminology, and exercise and sport sciences.

2 Framework for Reflective Coupled Human-Environment Systems

One of the challenges is to provide frameworks that cover the class of Ambient Intelligence applications showing human-like understanding and supporting behaviour; see also (Treur, 2008). Here human-like understanding is defined as understanding in the sense of being able to analyse and estimate what is going on in the human's mind (a form of mindreading) and in his or her body (a form of bodyreading). Input for these processes are observed information about the human's state over time, and dynamic models for the human's physical and mental processes. For the mental side such a dynamic model is sometimes called a Theory of Mind (e.g., Baron-Cohen, 1995; Dennett, 1987; Gärdenfors, 2003; Goldman, 2006) and may cover, for example, emotion, attention, intention, and belief. Similarly for the human's physical processes, such a model relates, for example, to skin conditions, heart rates, and levels of blood sugar, insulin, adrenalin, testosterone, serotonin, and specific medicines taken. Note that different types of models are needed: physiological, neurological, cognitive, emotional, social, as well as models of the physical and artificial environment.

A framework can be used as a template for the specific class of Ambient Intelligence applications as described. The structure of such an ambient software and hardware design can be described in an agent-based manner at a conceptual design level and can be given generic facilities built in to represent knowledge, models and analysis methods about humans, for example (see also Figure 1):

- human state and history models
- environment state and history models
- profiles and characteristics models of humans

- ontologies and knowledge from biomedical, neurological, psychological and/or social disciplines
- dynamic process models about human functioning
- dynamic environment process models
- methods for analysis on the basis of such models

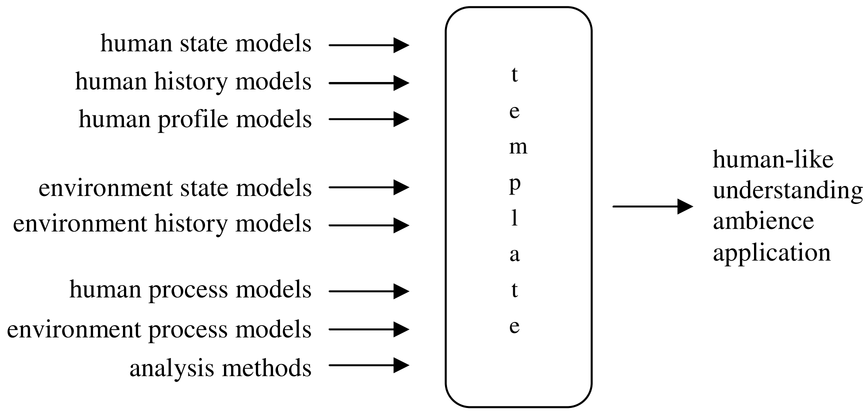

Figure 1. Framework to combine the ingredients

Examples of useful analysis methods are voice and skin analysis with respect to emotional states, gesture analysis, and heart rate analysis. The template can include slots where the application-specific content can be filled to get an executable design for a working system.

This specific content together with the generic methods to operate on it, provides a reflective coupled human-environment system, based on a tight cooperation between a human and an ambient system to show human-like understanding of humans and to react from this understanding in a knowledgeable manner. In this sense, 'coupled' means mutually interacting. For the specific type of applications considered here, however, the coupling takes two different forms; see also Figure 2.

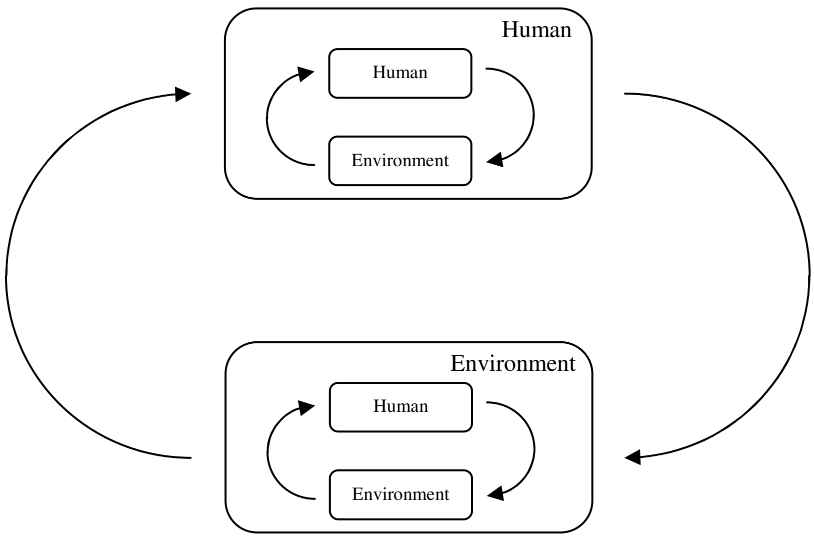

Figure 2. Reflective coupled human-environment systems

- On the one hand the coupling takes place as interaction between human and environment, as in any Ambient Intelligence application:
  - the environment gets information generated by the human as input, and
  - the human gets information generated by the environment as input.
- In addition, coupling at a more deep, reflective level takes place due to the fact that
  - the environment has and maintains knowledge about the functioning of the human, the environment and their interaction, and
  - the human has and maintains knowledge about functioning of him or herself, the environment, and their interaction

So, in such a more specific human-environment system, being coupled does not only mean that the human and its environment interact, but also that they have knowledge, understanding and awareness of each other, themselves and their interaction. This entails two types of awareness:

- *Human awareness*: awareness by the human about the human and environmental processes and their interaction
- *Technological awareness*: awareness by the environment about the human and environmental processes and their interaction

A general approach for embedding knowledge about the interaction between the environment and the human in Ambient Intelligence applications is to integrate dynamic models of this interaction (i.e. a model of the *domain*) into the application. This integration takes place by embedding domain models in certain ways within agent models of the Ambient Intelligence application. By incorporating domain models within an agent model, the Ambient Intelligence agent gets an understanding of the processes of its surrounding environment, which is a solid basis for knowledgeable intelligent behaviour. Three different ways to integrate domain models within agent models can be distinguished. A most simple way is to use a domain model that specifically models human behaviour in the following manner:

- *domain model directly used as agent model*
  In this case a domain model that describes human processes and behaviour is used directly as an agent model, in order to simulate human behaviour. Note that here the domain model and agent model refer to the same agent.

Such an agent model can be used in interaction with other agent models, in particular with *ambient agent models* to obtain a test environment for simulations. For this last type of (artificial) agents domain models can be integrated within their agent models in two different ways, in order to obtain one or more of the following (sub)models; see Figure 3. Here the solid arrows indicate information exchange between processes (data flow) and the dotted arrows the integration process of the domain models within the agent models.

- *analysis model*
  To perform analysis of the human's states and processes by reasoning based on observations (possibly using specific sensors) and the domain model.
- *support model*
  To generate support for the human by reasoning based on the domain model.

Note that here the domain model that is integrated refers to one agent (the human considered), whereas the agent model in which it is integrated refers to a different agent (the ambient software agent).

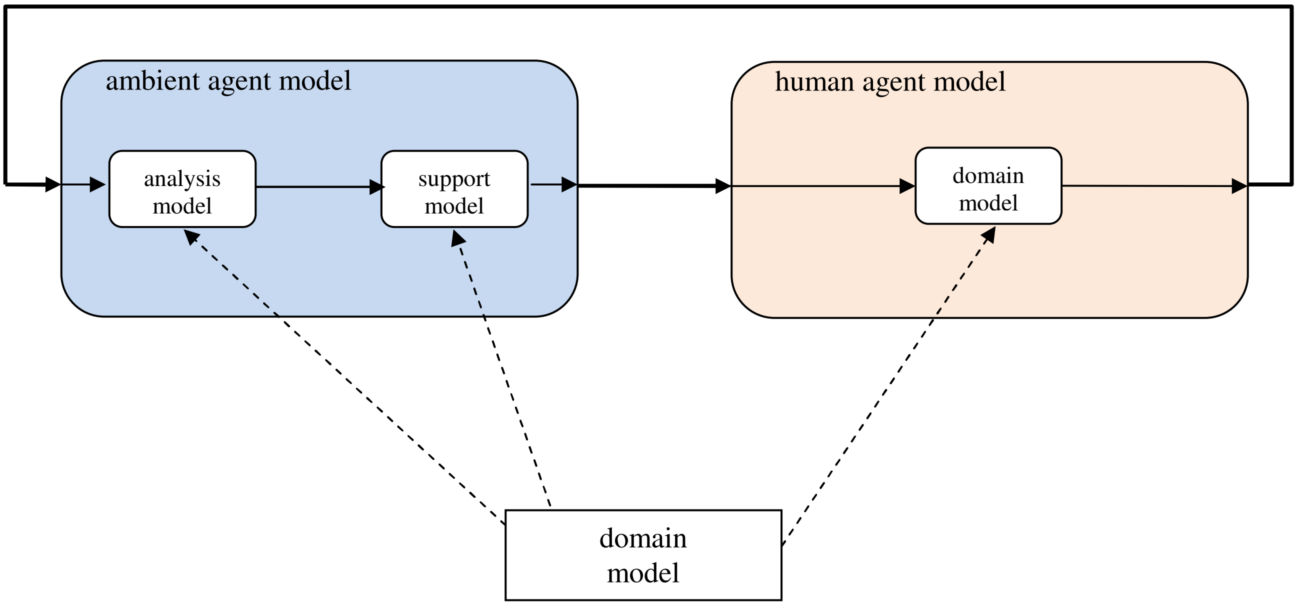

Figure 3. Three ways to integrate a domain model within an agent model

In the next sections of this chapter, two examples of Ambient Intelligence applications are presented, and the role of human knowledge in these applications is discussed. In the examples, each of the three types of models are discussed. Section 3 addresses a case study on automated visual attention manipulation, and Section 4 addresses assessment of driver behaviour.

3 Automated Visual Attention Manipulation

For people that have to perform complex and demanding tasks, it is very important to have sufficient *attention* for the various subtasks involved. This is particularly true for tasks that involve the continuous inspection of (computer) screens. For example, a person that has to inspect the images of a surveillance camera can not permit him- or herself to miss part of the events that occur. The same holds for an air traffic controller that inspects the movements of aircraft, or a naval operator that monitors the movements of hostile vessels on a radar screen. In such situations, a person may be supported by an ambient agent that keeps track of where his or her attention is, and provides some assistance in case the attention is not where it should be. To the end, the agent should maintain a *domain model* of the state of attention of the human, as well as an *analysis model* to reason about this domain model, and a *support model* to determine actions based on this analysis. This section introduces such models, and shows how they can be used to perform simulations. To make the models a bit more concrete, they are explained the context of a case study about a naval operator, which is introduced below.

Case study: a naval operator
In the domain of naval warfare, it is crucial for the crew of the vessels involved to be aware of the situation in the field. Examples of important questions that should be addressed continuously are “in which direction are we heading?”, “are we currently under attack?”, “are there any friendly vessels around?”, and so on. To assess such issues, one of the crew

members is usually assigned the Tactical Picture Compilation Task (TPCT): the task to identify and classify all entities in the environment (e.g., Heuvelink and Both, 2007). This is done by monitoring several radar screens in the control room for radar contacts, and reasoning with the available information in order to determine the type and intent of the contacts on the screens. For example, available information on the speed and direction of a radar contact can be used to determine the threat of this contact. However, due to the complex and dynamic nature of the environment, the person assigned to the TPCT has to deal with a large number of tasks in parallel. Often the radar contacts are simply too numerous and dynamic to be adequately monitored by a single human, which compromises the performance of the task.

For these reasons, it may be useful to offer this human operator some support from an intelligent ambient system, consisting of software agents that assist him in the execution of the Tactical Picture Compilation Task. For example, in case the human is directing its attention on the left part of a radar screen, but ignores an important contact that just entered the radar screen from the right, such a system may alert him about the arrival of that new contact. To be able to provide this kind of intelligent support, the system somehow needs to maintain a model of the cognitive state of the human: in this case the human's focus of attention. It should have the capability to attribute mental, and in particular attentional (e.g., Itti and Koch, 2001) states to the human, and to reason about these. In this section, an example of such a system is described, based on (Bosse, Maanen, and Treur, 2006; Bosse, Maanen, and Treur, 2009); see also (Bosse, Lambalgen, Maanen and Treur, 2011). Two types of sensor information are assumed, namely: information about the human's gaze (e.g., measured by an eye tracker), and characteristics of stimuli (e.g., the colour and speed of airplanes on radar screens, or of the persons on surveillance images).

3.1 Domain Model

This section introduces the domain model for attentional processes. In Section 3.1.1, the main aspects of the model and their relations are introduced. In Section 3.1.2, the detailed model is provided, and Section 3.1.3 presents an example simulation trace.

3.1.1 Analysis of the Main Aspects and Their Relations

In this section, a persons "state of attention" is defined as a distribution of values over different locations in the world. Assuming that the person is facing a certain screen, these locations are described by segments of the screen (which, for the case of the naval operator, may contain radar contacts). A high value for a certain location means that the person has much attention for that location. Note that "attention" is not (purely) defined by "visual attention": a person may currently not be looking at a certain location, but may still have attention for it. Rather, our notion of "attention" of an object should be seen as "presence in working memory".

In order to calculate this state of attention, obviously, the person's *gaze direction* is an important factor: where is the person currently looking to? Next, this information should be combined with the *locations of objects*: if the location of an object (e.g., a radar contact) is close to the location of the person's gaze, then the person will have more attention for that object. However, these are not the only factors. Also different *characteristics of objects* (such as brightness and size) are important: usually people have more attention for objects that are bright and large, compared to objects that are dark and small. Together, these three factors can be combined in order to calculate the *current attention contribution* of a certain location:

to what extent is the location currently attracting the person's attention? If this current contribution is known for all locations of the screen, then also a *normalised attention contribution* can be calculated for each location, i.e., the relative attention contribution in comparison with all other locations. One step further, this normalised attention contribution of a location can be used to calculate the actual *attention level* that a person has for that location. In order to do this, also the previous attention level plays a role. For example, in case a person recently had much attention for location X, then currently (s)he will still have some attention, even if the current contribution is very low. Finally, the attention level plays a role in the behaviour of the person: the more attention a person has for a certain location, the more likely (s)he is to act upon that location. In this section, it is assumed that a high attention level for a location leads to a *mouse click* on that location (for example, because clicking on a location helps a naval operator identifying the radar track on that location).

In sum, the following concepts are needed to describe the dynamics of a person's attention:

- gaze direction
- locations of objects
- characteristics of objects
- current attention contribution of locations
- normalised attention contribution of locations
- (old and current) attention level for locations
- mouse clicks on locations

These dynamic relationships that can be identified on the basis of these concepts are visualised in Figure 4. Note that the difference between current attention level and old attention level is not visualised: both are represented by the same node.

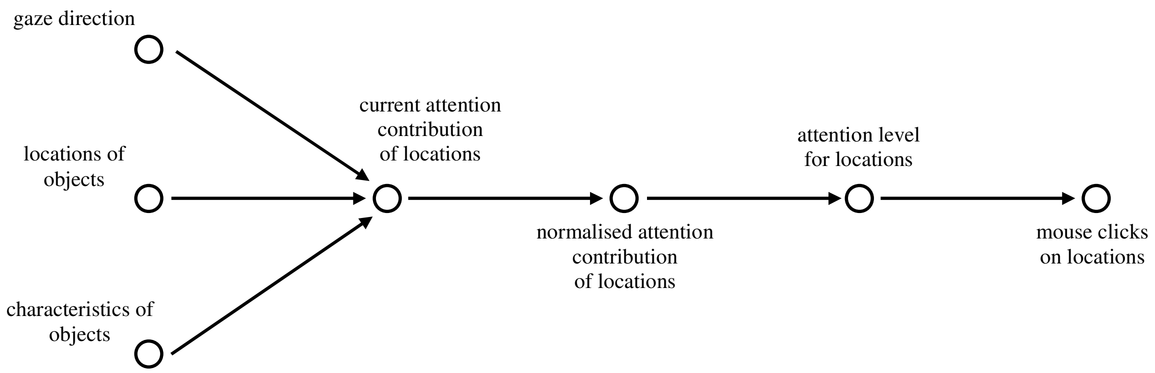

Figure 4. Overview of Domain Model for Attentional Processes

3.1.2 The Detailed Model

For the formalization of the conceptual model described above, the LEADSTO language (Bosse et al, 2007) is used. This is a specification language with the specific purpose to develop simulation models in a declarative manner. It is based on the assumption that dynamics can be described as an evolution of states over time. The notion of state as used here is characterised on the basis of an ontology defining a set of relevant facts (properties) that do or do not hold at a certain point in time. A specific state is characterised by dividing the set of state properties into those that hold, and those that do not hold in the state. In LEADSTO, direct temporal dependencies between two state properties in successive states are modelled by executable dynamic properties. The LEADSTO format is defined as follows. Let α and β be state properties as defined above. Then, the notation $\alpha \twoheadrightarrow_{e, f, g, h} \beta$ means:

If state property α holds for an interval with duration g, then after some delay between e and f, state property β will hold for an interval with duration h.

To formalise the concepts introduced in the previous section for each of them a logical atom is introduced; see Table 1.

Table 1. Formalisation of concepts used for domain model

concept	**formalisation**
gaze direction	gaze_at_loc(X:COORD, Y:COORD)
locations of objects	is_at_location(O:OBJECT, loc(X:COORD, Y:COORD))
characteristics of objects	has_value_for(O:OBJECT, V:REAL, A:ATTRIBUTE)
current attention contribution of locations	has_current_attention_contribution(loc(X:COORD, Y:COORD), V:REAL)
normalised attention contribution of locations	has_normalised_attention_contribution(loc(X:COORD, Y:COORD), V:REAL)
(old and current) attention level for locations	has_old_attention_level(loc(X:COORD, Y:COORD), V:REAL) has_attention_level(loc(X:COORD, Y:COORD), V:REAL)
mouse clicks on locations	mouse_click_on(loc(X:COORD, Y:COORD))

Note that some atoms make use of sorts. The specific sorts that are used in the presented model, and the elements that they contain, are shown in Table 2.

Table 2. Sorts used

sort	**description**	**elements**
COORD	coordinates of the screen used in the example	{1, 2, ..., max_coord}
REAL	the set of real numbers	the set of real numbers
OBJECTS	objects (e.g. radar contacts) used in the example	{contact05, contact22, contact39, nothing}
ATTRIBUTE	attributes of objects used in the example	{brightness, size}

In the detailed model, the following LEADSTO relationships are used:

DDR1 (Determining attention contributions))

If object O is at location loc(X1,Y1),
and the person's gaze is at location loc(X2,Y2),
and object O has value V1 for brightness,
and object O has value V2 for size,
then the current attention contribution of location loc(X1,Y1) is (V1*W1+V2*W2) / (1+ α * (X1-X2)^2+ (Y1-Y2)^2)

is_at_location(O, loc(X1, Y1)) $\wedge$ gaze_at_loc(X2, Y2) $\wedge$
has_value_for(O, V1, brightness) $\wedge$ has_value_for(O, V2, size)
$\twoheadrightarrow$ has_current_attention_contribution(loc(X1, Y1), (V1*W1+V2*W2) / (1+ α * ((X1-X2)^2+ (Y1-Y2)^2)))

This formula determines the current attention contribution of each location. For calculation of the current attention contribution, first the weighted sum of the attribute values of the location is calculated. Here, W1 and W2 (both real numbers between 0 and 1) denote weight factors for the attributes brightness and size, respectively. The weighted sum is divided by $1 + \alpha * r^2$, where r^2 is the *Euclidean distance* between the location loc(X1,Y1) of object O and the current location of a person's gaze loc(X2,Y2), defined by adding the squared value of X1-X2 to the squared value of Y1-Y2. The importance factor α (represented by a real number between 0 and 1) determines the relative impact of distance to the gaze location on the contribution of attention. Note that the formula only takes these two attributes of locations into account. In a more general form, the current attention contribution of a location can be calculated by $\Sigma_{A:attributes}$ V(A)*W(A) / $(1 + \alpha * R^2)$. The formula also assigns a value to empty locations (by using the special element nothing of sort OBJECT, which has value 0 for all attributes).

DDR2 (Normalising attention contributions)

If the current attention contribution of location loc(X,Y) is V,
then the normalised attention contribution of location loc(X,Y)
is V * A divided by the sum of the current attention contributions of all locations

has_current_attention_contribution(loc(X, Y), V) ∧
has_current_attention_contribution(loc(1, 1), V1) ∧
has_current_attention_contribution(loc(1, 2), V2) ∧
has_current_attention_contribution(loc(1, 3), V3) ∧
has_current_attention_contribution(loc(2, 1), V4) ∧
has_current_attention_contribution(loc(2, 2), V5) ∧
has_current_attention_contribution(loc(2, 3), V6) ∧
has_current_attention_contribution(loc(3, 1), V7) ∧
has_current_attention_contribution(loc(3, 2), V8) ∧
has_current_attention_contribution(loc(3, 3), V9)
↠ has_normalised_attention_contribution(loc(X, Y),V*A / (V1+V2+V3+V4+V5+V6+V7+V8+V9))

This formula determines the normalised attention contribution of each location, by dividing the current attention contribution of the location by the sum of the current attention contributions of all locations. Furthermore, the resulting value is multiplied by a constant a (represented by a real number between 0 and 1) which represents the total amount of attention that a person can spend. This usually depends on many factors that differ per person and situation (e.g., cognitive abilities, stress, concentration, and so on).

DDR3 (Determining attention levels)

If the normalised attention contribution of location loc(X,Y) is V1,
and the old attention level of location loc(X,Y) is V2,
then the new attention level of location loc(X,Y) is d*V2 + (1-d*V1

has_normalised_attention_contribution(loc(X, Y),V1) ∧
has_old_attention_level(loc(X, Y), V2)
↠ has_attention_level(loc(X, Y), D*V2 + (1-D)*V1)

This formula determines the actual attention level of each location, based on the old attention level and the current (normalised) attention contribution. Here, d is a persistence parameter (represented by a real number between 0 and 1) that determines the relative impact of the old attention level on the new attention level.

DDR4 (Time shift)

If the attention level of location loc(X,Y) is V,
then the old attention level of location loc(X,Y) becomes V

has_attention_level(loc(X, Y), V)
↠ has_old_attention_level(loc(X, Y), V)

DDR4 is a simple formula that converts the current attention level to the old attention level after each time step.

DDR5 (Determining mouse clicks)
If the attention level of location loc(X,Y) is V,
and V > th,
then there will be a mouse click on location loc(X,Y)

has_attention_level(loc(X, Y), V) $\wedge$ V > th
$\twoheadrightarrow$ mouse_click_on(loc(X, Y))

This formula determines when the person will click on a certain location. For simplicity, this is assumed to occur in all cases that the person has an attention level V for that location that exceeds a certain threshold th (again represented by a real number between 0 and 1).

3.1.3 Example Simulation

This section presents an example simulation trace that was generated on the basis of the domain model for attentional processes. Figure 5 displays (a selection of) the symbolic concepts involved, and Figure 6 displays (a selection of) the numerical concepts. The settings used for the parameters of the model (see previous section) are as follows: W1=0.8, W2=0.5, α=0.9, A=0.9, d=0.3, th=0.25, max_coord=3.

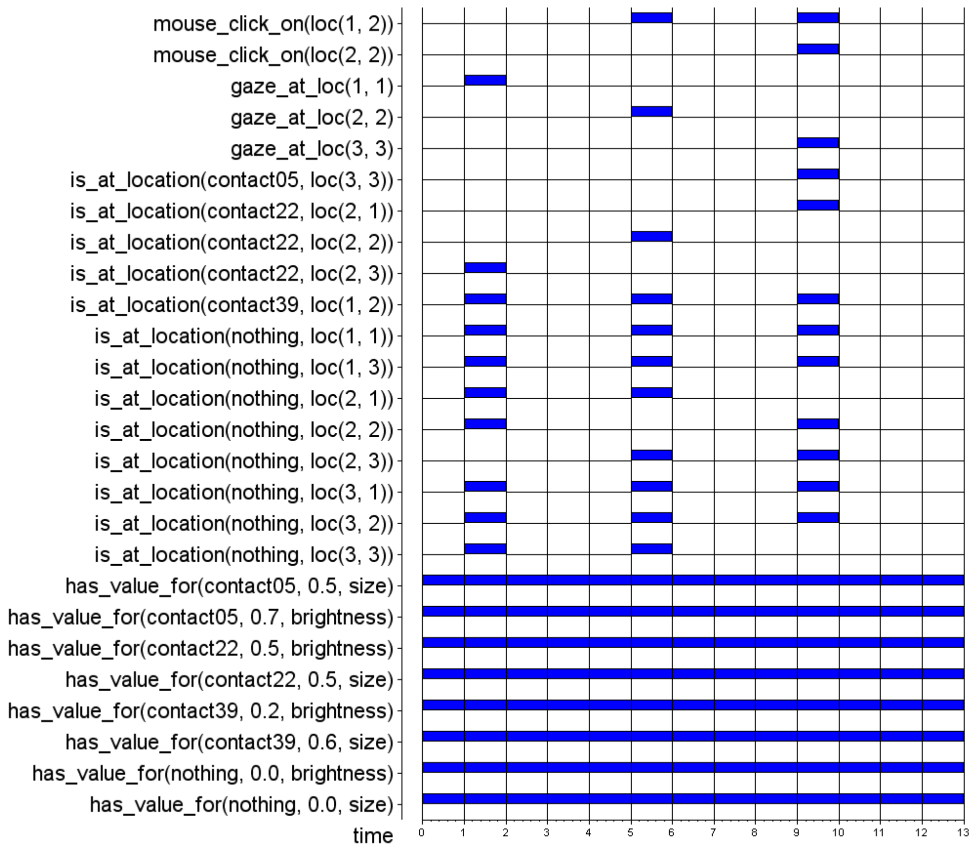

Figure 5. Example simulation trace for domain model - symbolic concepts.

As shown in Figure 5, there are 9 locations involved (namely loc(1,1), ..., loc(3,3)) and three objects (namely radar contacts contact05, contact22, and contact39). Three rounds of attention generation are shown. During the first round, the person's gaze is at location loc(1,1), later it shifts to loc(2,2) and finally to loc(3,3). Furthermore, contact22 moves from location loc(2,3) to loc(2,2) and to loc(2,1), contact39 is always at location loc(1,2), and contact05 only comes into play during the last round, when it appears at location loc(3,3). The attribute values of the different contacts are shown in the bottom lines of the picture.

The dynamics of the person's attention based on these events are shown in Figure 6 (note that the figure only focuses on three locations, and that intermediate steps that calculate current and normalised attention contribution are not shown). As shown in this picture, initially the person's attention is spread equally over the locations. However, due to the appearance of contact39 at location loc(1,2), the attention for this location strongly increases in the next round. In the rounds that follow, the attention of this location converges back to the initial value.

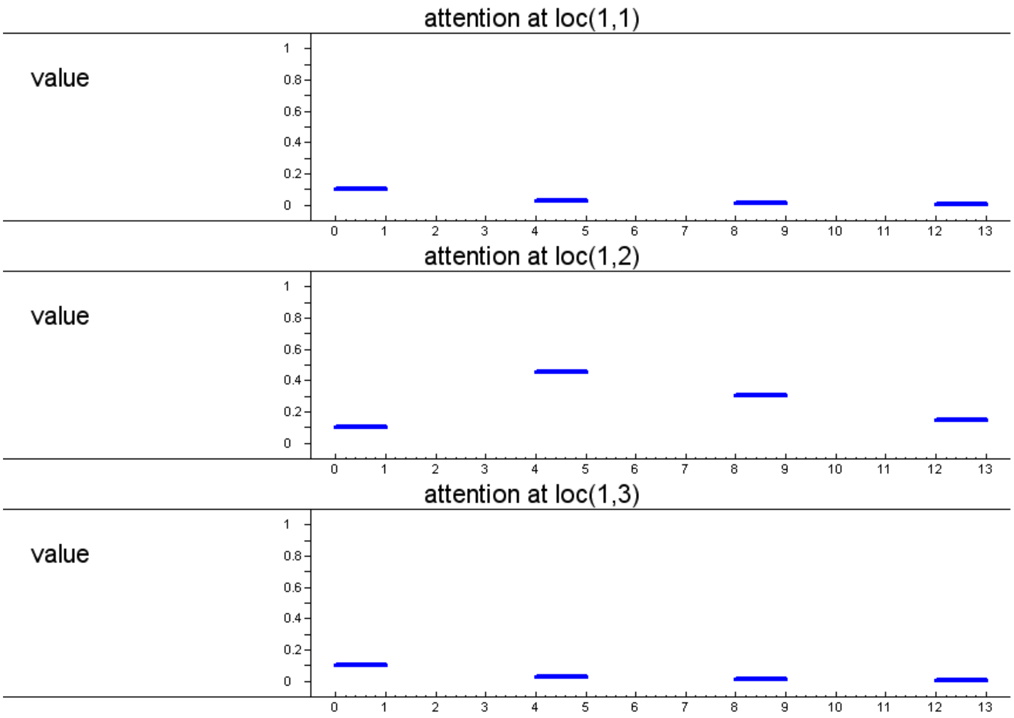

Figure 6. Example simulation trace for domain model - numerical concepts.

3.2 Model for Analysis

This section introduces the model for analysis of attentional processes. In Section 3.2.1, the main aspects of the model and their relations are introduced. In Section 3.2.2, the detailed model is provided, and Section 3.2.3 presents an example simulation trace.

3.2.1 Analysis of the Main Aspects and Their Relations

To be able to analyse the dynamics of a human's attention, an ambient agent should, in principle, be equipped with the domain model introduced in Section 3.1, and should be able to reason on the basis of this model. Therefore, the model for analysis should contain more of less the same concepts as the domain model. However, one important difference is that not all concepts that exist in the domain model can actually be observed by the ambient agent. For example, the 'current attention contribution' of a location is not something that is explicitly observable in the world, nor is the 'attention level' of a person for a location. On the other hand, things like gaze direction, and locations and characteristics of objects can be observed explicitly (e.g., by an eye tracker and a graphical user interface, respectively). As a result, some of the real world concepts from the domain model can directly be incorporated in the analysis model, whereas for some other concepts, the agent should make an estimation: it will form *beliefs* about the state of those concepts. Finally, the analysis model should have information about the human's *desired attention level* for the different locations: where does it want the human to focus at? It is assumed that the agent has tactical domain knowledge that enables it to make such assessments. For example, in the domain of the Tactical Picture Compilation Task, the agent knows the movement information (e.g. speed and direction) of each radar track and uses it to estimate the urgency of identifying the track. This urgency determines the desired attention level at each location. By comparing the desired attention level with the estimated attention level, the analysis model can determine whether there is an *attention discrepancy* at a certain location, i.e., whether the human has too much or too little attention for the location (and, possibly, how big this discrepancy is).

This leads to the following list of concepts that are needed for the attention analysis model:

- gaze direction
- locations of objects
- characteristics of objects
- beliefs on current attention contribution of locations
- beliefs on normalised attention contribution of locations
- beliefs on (old and current) attention level for locations
- desired attention level for locations
- assessments on attention discrepancy for locations

As can be seen, this list is similar to the list of concepts presented in Section 3.1.1, where for concept 4, 5, and 6 a belief is introduced, and two new concepts have been added. Furthermore, also most dynamic relationships are similar to those presented in Section 3.1.1; see Figure 7.

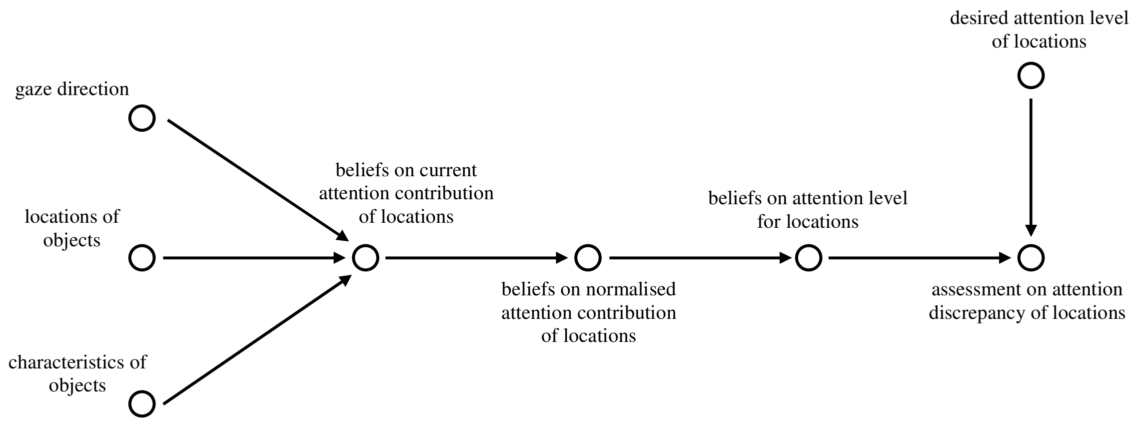

Figure 7. Overview of Analysis Model for Attentional Processes

3.2.2 The Detailed Model

To formalise the concepts introduced in the previous section, again a number of logical atoms are introduced (most of which are similar to Table 1); see Table 3.

Table 3. Formalisation of concepts used for analysis model

concept	formalisation
gaze direction	gaze_at_loc(X:COORD, Y:COORD)
location of objects	is_at_location(O:OBJECT, loc(X:COORD, Y:COORD))
characteristics of objects	has_value_for(O:OBJECT, V:REAL, A:ATTRIBUTE)
belief on current attention contribution of locations	belief(agent, has_current_attention_contribution(loc(X:COORD, Y:COORD), V:REAL))
belief on normalised attention contribution of locations	belief(agent, has_normalised_attention_contribution(loc(X:COORD, Y:COORD), V:REAL))
belief on (old and current) attention level for locations	belief(agent, has_old_attention_level(loc(X:COORD, Y:COORD), V:REAL)) belief(agent, has_attention_level(loc(X:COORD, Y:COORD), V:REAL))
desired attention level of locations	desire(agent, has_attention_level(loc(X:COORD, Y:COORD), V:REAL))
assessment on attention discrepancy of locations	assessment(agent, has_attention_discrepancy(loc(X:COORD, Y:COORD), V:REAL))

Next, to formalise the dynamic relationships for the analysis model, the following LEADSTO relationships are used:

ADR1 (Determining beliefs on attention contributions)

If object O is at location loc(X1,Y1),
and the person's gaze is at location loc(X2,Y2),
and object O has value V1 for brightness,
and object O has value V2 for size,
then the agent believes that the current attention contribution of location loc(X1,Y1) is $(V1*W1_{ag} + V2*W2_{ag}) / (1+ \alpha_{ag} * (X1-X2)^2+ (Y1-Y2)^2)$

is_at_location(O, loc(X1, Y1)) ∧ gaze_at_loc(X2, Y2) ∧
has_value_for(O, V1, brightness) ∧ has_value_for(O, V2, size)
↠ belief(agent, has_current_attention_contribution(loc(X1, Y1), $(V1*W1_{ag} + V2*W2_{ag})/(1+ \alpha_{ag} * ((X1-X2)^2+ (Y1-Y2)^2)))$)

ADR2 (Determining beliefs on normalised attention contributions)

If the agent believes that the current attention contribution of location loc(X,Y) is V,
then the agent believes that the normalised attention contribution of location loc(X,Y) is $V * A_{ag}$ divided by the sum of the current attention contributions of all locations

belief(agent, has_current_attention_contribution(loc(X, Y), V)) ∧
belief(agent, has_current_attention_contribution(loc(1, 1), V1)) ∧
belief(agent, has_current_attention_contribution(loc(1, 2), V2)) ∧
belief(agent, has_current_attention_contribution(loc(1, 3), V3)) ∧
belief(agent, has_current_attention_contribution(loc(2, 1), V4)) ∧
belief(agent, has_current_attention_contribution(loc(2, 2), V5)) ∧
belief(agent, has_current_attention_contribution(loc(2, 3), V6)) ∧
belief(agent, has_current_attention_contribution(loc(3, 1), V7)) ∧
belief(agent, has_current_attention_contribution(loc(3, 2), V8)) ∧
belief(agent, has_current_attention_contribution(loc(3, 3), V9))
↠ belief(agent, has_normalised_attention_contribution(loc(X, Y),$V*A_{ag}$ / (V1+V2+V3+V4+V5+V6+V7+V8+V9)))

ADR3 (Determining beliefs on attention levels)

If the agent believes that the normalised attention contribution of location loc(X,Y) is V1,
and the agent believes that the old attention level of location loc(X,Y) is V2,
then the agent believes that the new attention level of location loc(X,Y) is $d_{ag}*V2 + (1-d_{ag})*V1$

belief(agent, has_normalised_attention_contribution(loc(X, Y),V1)) ∧
belief(agent, has_old_attention_level(loc(X, Y), V2))
↠ belief(agent, has_attention_level(loc(X, Y), $d_{ag}*V2 + (1-d_{ag})*V1$))

ADR4 (Determining beliefs on time shift)
If the agent believes that the attention level of location loc(X,Y) is V,
then the agent believes that the old attention level of location loc(X,Y) becomes V

belief(agent, has_attention_level(loc(X, Y), V))
↠ belief(agent, has_old_attention_level(loc(X, Y), V))

Note that dynamic relationship ADR1-ADR4 are very similar to DDR1-DDR4, but that there are two important differences:

1) for some of the concepts, *estimations* (modelled as *beliefs*) are used instead of the real concepts
2) for some of the relationships, the agent uses its own *parameters* (e.g., α_{ag} instead of α, and d_{ag} instead of d), which need not correspond exactly to the actual parameters

The reason for this second point is that an ambient agent may simply not know what the value of the actual parameter is; therefore it may need to make estimations. However, based on experiences, it may adapt such estimations in an intelligent manner (this topic of parameter adaptation is beyond the scope of this chapter).

Furthermore, one additional LEADSTO relationship is used to determine whether there is an attention discrepancy:

ADR5 (Assessing attention)
If the agent believes that the attention level of location loc(X,Y) is V1,
and the agent desires that the attention level of location loc(X,Y) is V2,
and V1 < V2,
then the agent assesses that there is an attention discrepancy at location loc(X,Y) of size V2-V1

belief(agent, has_attention_level(loc(X, Y), V1)) ∧ desire(agent, has_attention_level(loc(X, Y), V2)) ∧ V1 < V2
↠ assessment(agent, has_attention_discrepancy(loc(X, Y), V2-V1))

Note that this is just one example of a (simple) mechanism to assess attention discrepancies. Other, more sophisticated mechanisms would also consider the case that the estimated attention for a location is too high, or would only signal a discrepancy in case the difference between estimated and desired attention is above a certain threshold.

3.2.3 Example Simulation

Also for the analysis model, simulation runs may be performed. An example simulation trace for a situation in which the ambient agent signals a discrepancy is shown in Figure 8 (symbolic concepts) and 9 (numerical concepts). In this trace, a situation is simulated that is similar to the situation shown in Section 3.1. The settings used for the parameters of the domain model are identical to Section 3.1: W1=0.8, W2=0.5, α=0.9, A=0.9, d=0.3, th=0.25. These same settings are used for the parameters of the analysis model: $W1_{ag}$=0.8, $W2_{ag}$=0.5, α_{ag}=0.9, A_{ag}=0.9, d_{ag}=0.3. Also the characteristics and dynamics of the different objects and the gaze were similar to Section 3.1, with one difference: contact39 is now hardly visible. It is very small, and not very bright (both its size and brightness are 0.01). However, since this is a very urgent contact, the agent wants the human to have at least some attention for its location: see desire(agent, has_attention_level(loc(1, 2), 0.2)) in Figure 8.

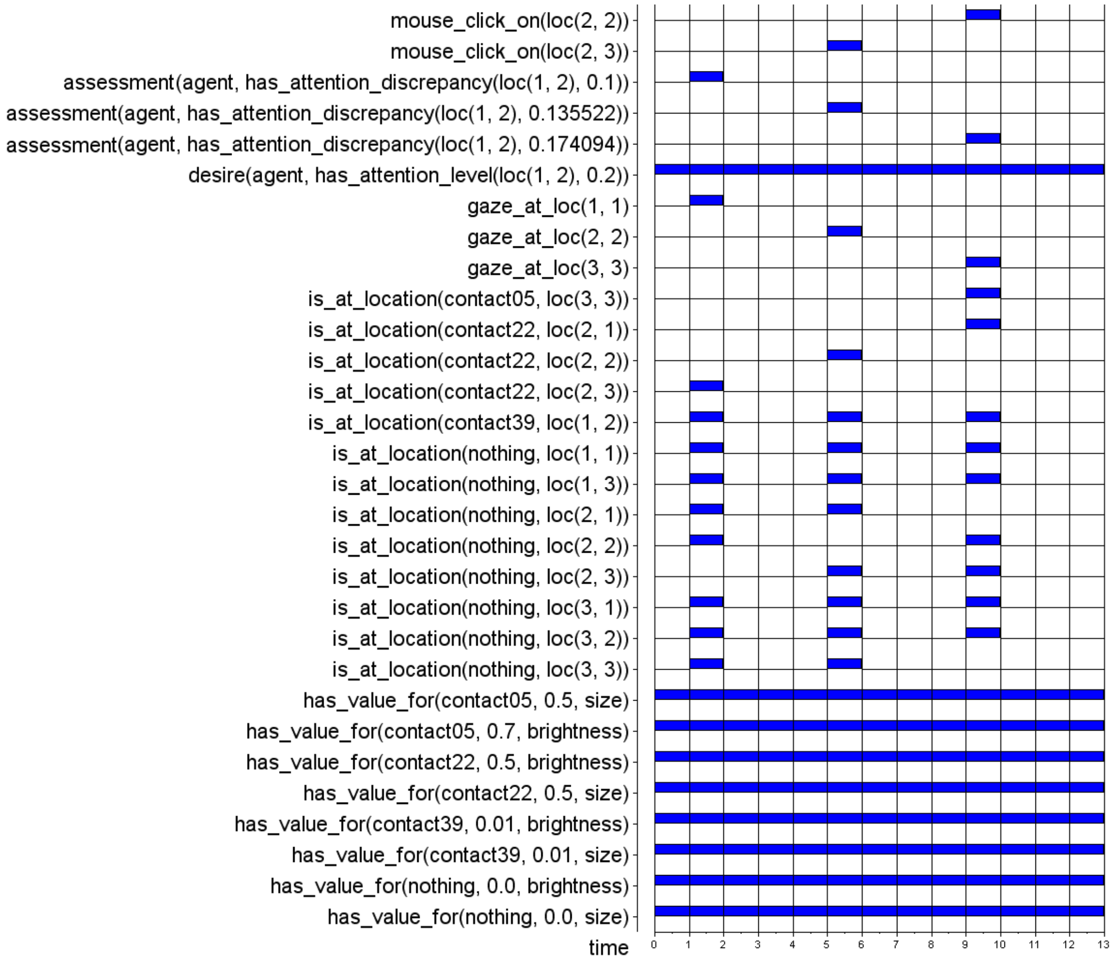

Figure 8. Example simulation trace for analysis model - symbolic concepts.

As shown in Figure 8, the agent (correctly) believes that the human has very little attention for location loc(1,1), loc(1,2), and loc(1,3). The reason for this is that these locations do not contain any salient objects: there is only a - hardly visible - contact at location loc(1,2). This also explains why the user does not click with the mouse on this location (the user decides to click on locations loc(2,2) and loc(2,3) instead). However, since the agent wants the human to have some attention for location loc(1,2), it derives that there is a discrepancy at that location: assessment(agent, has_attention_discrepancy(loc(1, 2), ...)), which increases over time (see Figure 8).

Such discrepancies may be used as input by the support model, in order to determine how the discrepancy should be solved. This is explained in the next section.

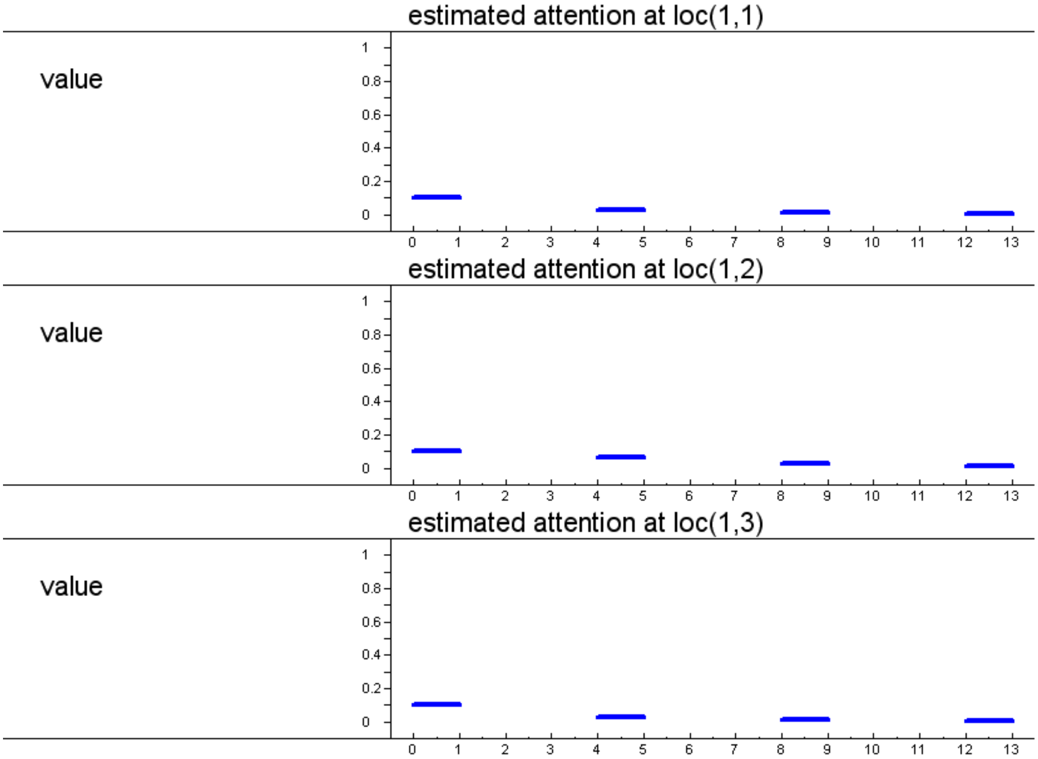

Figure 9. Example simulation trace for analysis model - numerical concepts.

3.3 Model for Support

This section introduces the model for support based on attentional processes. In Section 3.3.1, the main aspects of the model and their relations are introduced. In Section 3.3.2, the detailed model is provided, and Section 3.3.3 presents an example simulation trace.

3.3.1 Analysis of the Main Aspects and Their Relations

In case there is a discrepancy between a person's estimated and desired attention level for a certain location, an ambient agent has several alternatives to provide support. To illustrate this, let us assume that there is an urgent radar contact for which the human has no (or not enough) attention. In that case, one possibility is to adapt one (or more) of the characteristics of the object. For instance, the agent could increase the size and/or brightness of the contact. A second, similar option is to decrease the size and/or brightness of other objects, so that the relative saliency of the contact will increase, or even to make these objects completely invisible. However, such types of support are very intrusive to the activities of the human, and may be disturbing in case (s)he is already very busy with another subtask. Therefore, a third alternative is to leave the human alone, and request another party to concentrate on the urgent object. This other party may be another human, but may also be an intelligent software agent (assuming that there exist agents with sufficient domain knowledge to solve the task in a sufficiently satisfactory manner).

In this section, the focus is on the first, relatively simple alternative: a model is introduced that adapts the characteristics of objects that receive too little attention (in this case, by increasing their brightness). To this end, the model needs several types of information as input. First, obviously, it needs to know whether there is an *attention discrepancy* at a certain location. Second, it needs to know which *objects* are at these locations. And third, it needs to know what the *characteristics* of these objects are. Based on that, it will generate *actions to adapt the characteristics of objects*.

In sum, the following concepts are needed for the attention support model:

- assessments on attention discrepancy for locations
- locations of objects
- characteristics of objects
- actions to adapt the characteristics of objects

The dynamic relationships that can be established between these concepts are visualised in Figure 10.

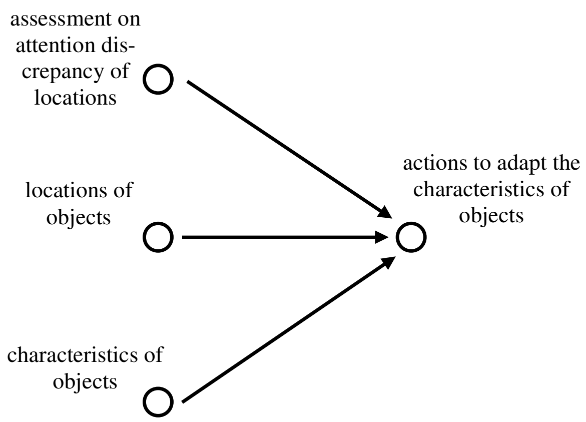

Figure 10. Overview of Support Model for Attentional Processes

Note that in reality, a change in characteristics of objects might also result in a change of the human's gaze direction. However, for simplicity, this relationship is not modelled in this chapter; i.e., the gaze movement is assumed to be independent of the objects.

3.3.2 The Detailed Model

To formalise the concepts introduced in the previous section, the following logical atoms are used (of which the first three are taken literally from Table 3); see Table 4.

Table 4. Formalisation of concepts used for support model

concept	formalisation
assessment on attention discrepancy of locations	assessment(agent, has_attention_discrepancy(loc(X:COORD, Y:COORD), V:REAL))
locations of objects	is_at_location(O:OBJECT, loc(X:COORD, Y:COORD))
characteristics of objects	has_value_for(O:OBJECT, V:REAL, A:ATTRIBUTE)
actions to adapt characteristics of objects	performed(agent, assign_new_value_for(O:OBJECT, V:REAL), A:ATTRIBUTE)

To formalise the dynamic relationship for the support model, the following LEADSTO relationship is used:

SDR1 (Changing brightness)

If the agent believes that there is an attention discrepancy at location loc(X,Y) of size V1,
and object O is at location loc(X,Y),
and object O has value V2 for brightness,
then the agent adapts the brightness of object O to value f(V1,V2)

assessment(agent, has_attention_discrepancy(loc(X,Y), V1)) ∧
is_at_location(O, loc(X, Y)) ∧ has_value_for(O, V2, brightness)
↠ performed(agent, assign_new_value_to(O, f(V1,V2), brightness))

Here, f(V1,V2) is a function of value V1 and V2, which can be filled in per domain and per application. A very simple example for f(V1,V2), which is used in the simulation described in the next section, is the following function: f(V1,V2) = 1. This function just assigns a maximal brightness to objects at locations with too little attention. However, also more sophisticated function can be used, such as the following: f(V1,V2) = V2 + V1 * (1 - V2). This function (which assumes that both V1 and V2 are real values between 0 and 1), increases the value of an attribute (in this case brightness) with a fraction of the distance between the current and the maximum value of the attribute. This fraction is taken proportional to the attention discrepancy of the location involved. Furthermore, similar formulae can be used to assign new values to the other attributes. In addition, the effects of support actions on the world should be modelled. This is not part of the support model itself, but is needed within a simulation, to be able to evaluate whether the support works. This can be done via the following LEADSTO relationships:

SDR2 (Assigning new attribute values)

If the agent adapts attribute a of object O to value V,
then attribute a of object O will have value V

performed(agent, assign_new_value_to(O, V, A))
↠ has_value_for(O, V, A)

SDR3 (Persistence of attribute values)

If attribute A of object O has value V,
and the agent does not adapt attribute a of object O,
then attribute A of object O will keep value v

has_value_for(O, V, A) ∧ not(performed(agent, assign_new_value_to(O, A)))
↠ has_value_for(O, V, A)

Note that relationship SDR3 assumes the existence of predicates of the form performed(agent, assign_new_value_to(O, A)). To achieve this, these predicates can simply be added to the consequent of SDR1.

3.3.3 Example Simulation

To illustrate the functioning of the attention support model, an example simulation trace is shown in Figure 11 (symbolic concepts) and 12 (numerical concepts). Note that this trace is generated on the basis of the support model in combination with the domain model and the analysis model. The trace addresses the same scenario as addressed in Section 3.2. Thus, the settings used for the parameters of the domain model are: W1=0.8, W2=0.5, α=0.9, A=0.9, d=0.3, th=0.25, and the settings used for the parameters of the analysis model are $W1_{ag}$=0.8, $W2_{ag}$=0.5, α_{ag}=0.9, A_{ag}=0.9, d_{ag}=0.3. Also the initial characteristics and dynamics of the different objects and the gaze were similar to Section 3.2. Again contact39 is almost invisible at the start of the simulation. However, a difference is that, this time, the agent will adapt the characteristics of this contact.

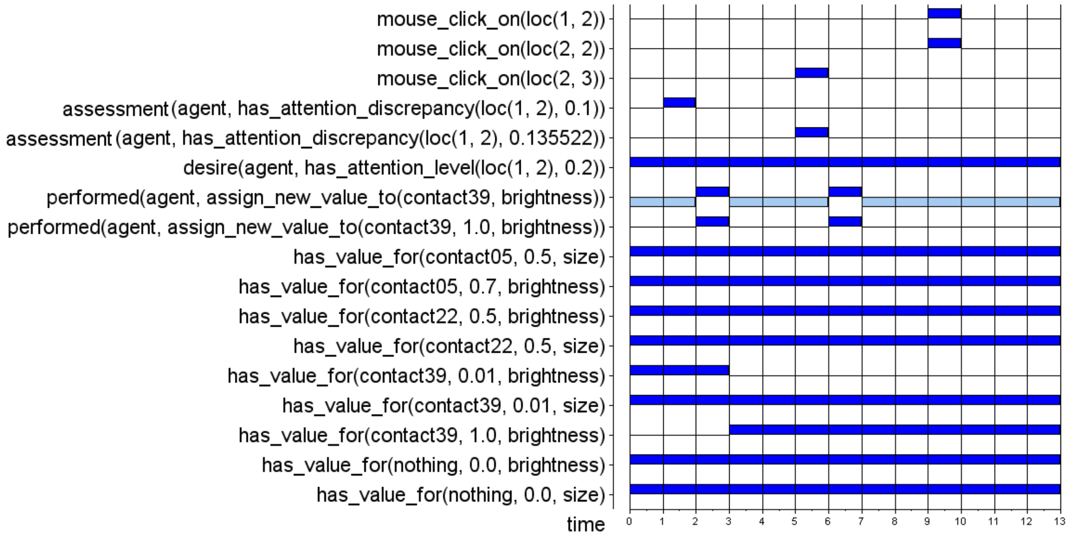

Figure 11. Example simulation trace for support model - symbolic concepts.

Figure 12 shows the agent's estimation of the human's attention for the first three locations. Note that, as in Section 3.2, this estimation is completely correct (as the agent uses the same parameter settings as the real world). As shown in Figure 12, at the start of the simulation, the agent believes that the human has little attention (about 0.1) for location loc(1,2). Since the agent wants the human to have more attention (about 0.2) for this location, it derives that there is a discrepancy at that location of 0.1: see assessment(agent, has_attention_discrepancy(loc(1, 2), 0.1)) at time point 1 of Figure 11. This discrepancy causes the agent to adapt the brightness of contact39 to 1.0, which is effectuated at time point 3. As a result, after a while the human has more attention for location loc(1,2), which makes that (s)he clicks on the location at time point 9. In conclusion, the agent succeeded in attracting the human's attention to the (urgent) location loc(1,2).

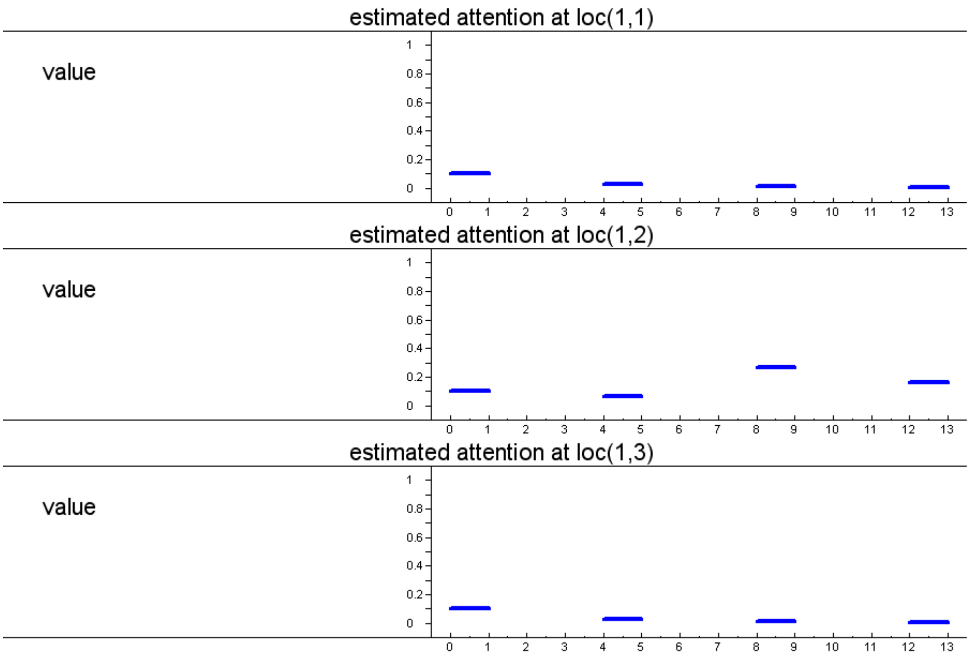

Figure 12. Example simulation trace for support model - numerical concepts.

3.4 Discussion

To support a human in the execution of a complex and demanding task, an ambient agent needs to maintain a domain model of the state of attention of this person. This section introduced such a domain model (in Section 3.1), as well as an analysis model to reason about this domain model (Section 3.2), and a support model to determine actions based on this analysis (Section 3.3). Note that, although the example presented here addresses a naval operator who executes the Tactical Picture Compilation Task, in principle the model is generic, and can be applied to various similar domains (varying from persons that inspect surveillance cameras to air traffic controllers).

The presented model can be extended in various manners. Some possibilities are mentioned below:

- Taking more attributes into account than only brightness and size.
- Addressing not only *bottom-up attention* (i.e., attention influenced by characteristics of the object, such as brightness and size), but also *top-down attention* (i.e., attention influenced by characteristics of the person, such as expectations and goals).
- Modelling not only attention for visible objects, but also attention for invisible objects (e.g., an object that has just disappeared from the screen, or an object that is present on a window that is hidden behind another window).
- Considering different and more complex types of support. The presented model only supports the human by modifying the attributes (in particular the brightness) of urgent objects. Other models could, for example, take over the identification task for those objects in case the human is too busy.
- Considering sophisticated techniques for adaptation of parameter values.

4 Assessment of Driver Behaviour

The second example scenario that illustrates the benefit of human knowledge in ambient intelligence applications is about driving behaviour. Circumstances may occur in which a person's internal state is affected in such a way that driving is no longer safe. For example, when a person has taken drugs, either prescribed by a medical professional, or by own initiative, the driving behaviour may be impaired. For the case of alcohol, specific tests are possible to estimate the alcohol level in the blood, but for many other drugs such tests are not available. Moreover, a bad driver state may have other causes, such as highly emotional events, or being sleepy. Therefore assessment of the driver's state by monitoring the driving behaviour itself and analysing the monitoring results is a wider applicable option. In this section, an ambient agent is presented that assesses a person's driving behaviour, and in case of a negative assessment lets cruise control slow down and stop the car (see Bosse, Hoogendoorn, Klein, and Treur, 2008). The approach was inspired by a system that is currently under development by Toyota. This ambient system that in the near future will be incorporated as a safety support system in Toyota's prestigious Lexus line, uses sensors that can detect the driver's steering operations, and sensors that can detect the driver's eye movements.

The presented ambient agent makes use of three types of sensor information. First, the driver's steering behaviour is measured, to assess, among others, whether (s)he is making too many unnecessary movements with the steering wheel. Second, a sensor in the steering wheel measures the driver's perspiration. This can be used to assess the percentage of alcohol in the driver's sweat, which correlates linearly with the alcohol percentage in the blood. Third, a camera that is mounted on the front mirror measures the driver's eye movements, which can be used to assess, for instance, whether (s)he is staring too much in the distance, or whether (s)he is blinking much with the eyes.

4.1 Domain Model

This section introduces the domain model for driver behaviour. In Section 4.1.1, the main aspects of the model and their relations are introduced. In Section 4.1.2, the detailed model is provided, and Section 4.1.3 presents an example simulation trace.

4.1.1 Analysis of the Main Aspects and Their Relations

In the literature, various factors are reported that have a (negative) impact on driver behaviour. These factors include high levels of drugs or alcohol, fatigue, specific emotional states, the presence of other persons in the car, loud music, and the use of mobile phones. The system described in this section is designed for measuring three of these factors. However, it can be extended to account for other factors as well.

First, the *level of drugs* in the blood is considered. This is dependent on the amount of *drug intake* over the last hours. Second, and similarly, the *level of alcohol* in the blood (and also in the sweat) is dependent on the amount of *alcohol intake* over the last hours. The third factor is the person's state of *fatigue*, which depends on whether long periods of *non-stop driving* have been performed recently. All of these factors may lead to a state that negatively influences the driving behaviour of the person. In particular, in case a person has taken certain drugs or alcohol, his or her *steering behaviour* may be affected: (s)he will make more sudden movements in the steering wheel, instead of keeping it straight. In addition, the person's eye movements may become abnormal. A person may start to *stare* (i.e., fixating the eyes on one location, without making any saccades) in case (s)he has consumed a lot of alcohol and/or is very tired. In case of the latter, the person's eyes will also *blink* more often than usually. This happens because the state of fatigue makes that the *speed* increases by which the eyes become *dry*, and also because the *basic eye dryness* (the dryness just after a blink) increases. When a person's eyes become too dry (i.e., they are drier than a certain desired *norm*), a person blinks (i.e., rapidly opens and closes the eyelid) in order to spread tears across and remove irritants[i].

In sum, the following concepts are needed to describe the dynamics of a person's driving behaviour:

- drug intake
- alcohol intake
- non-stop driving
- drug level (in blood)
- alcohol level (in blood and sweat)
- fatigue
- steering behaviour
- staring
- eye blinks
- eye dryness
- basic eye dryness

- eye drying speed
- eye dryness norm

These dynamic relationships that can be identified on the basis of these concepts are visualised in Figure 13.

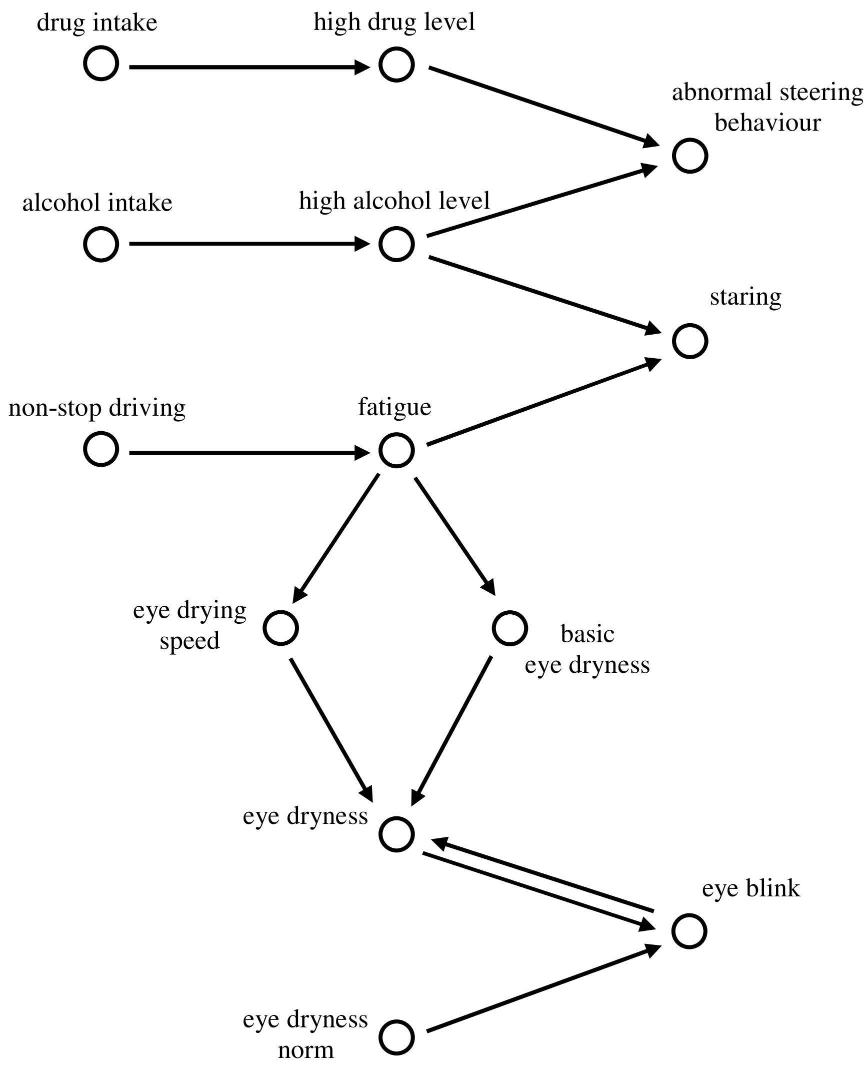

Figure 13. Overview of Domain Model for Driver Behaviour

4.1.2 The Detailed Model

To formalise the concepts introduced in the previous section, for each of them a logical atom is introduced; see Table 5. The predicate affects(P:PRODUCT, S:STATE) is not needed per se, but has been introduced in order to create some more generic relationships.

Table 5. Formalisation of concepts used for domain model

concept	formalisation
drug intake	consumed(take_drugs)
alcohol intake	consumed(take_alcohol)
non-stop driving	consumed(non_stop_driving)
drug level	has_level(drug_level, L:LEVEL)
alcohol level	has_level(alcohol_level, L:LEVEL)
fatigue	has_level(fatigue, L:LEVEL)
abnormal steering behaviour	performed(abnormal_steering_behaviour)
staring	performed(staring)
eye blinks	performed(eye_blink)
eye dryness	eye_dryness(I:REAL)
basic eye dryness	basic_eye_dryness(I:REAL)
eye drying speed	eye_drying_speed(I:REAL)
eye dryness norm	eye_dryness_norm(I:REAL)
relations between factors and driver states	affects(P:PRODUCT, S:STATE)

Note that some atoms make use of sorts. The specific sorts that are used in the presented model, and the elements that they contain, are shown in Table 6. Note that, instead of using the qualitative sort LEVEL, also a quantitative sort can be used (e.g., describing alcohol concentration by a real number).

Table 6. Sorts used

sort	description of use	elements
LEVEL	extent to which a certain factor (e.g., drugs, alcohol, or fatigue) is present in a person's body	{low, high}
REAL	the set of real numbers	the set of real numbers
PRODUCT	different products that can be consumed	{drugs, alcohol, non_stop_driving}
STATE	different aspects of the driver's physical state	{drug_level, alcohol_level, fatigue}

To formalise the dynamic relationships between these concepts, the following relationships are used, expressed in LEADSTO. We both give a semi-formal description in natural language as a formal specification in a logical notation.

DDR1 (Consuming a substance)
If a person consumes a certain product (e.g., drugs),
and this product affects a certain aspect of the driver's state (e.g., drug level in his blood),
then this aspect will be high (e.g., (s)he will have a high concentration of drugs in the blood)

consumed(P) & affects(P,S)
↠ has_level(S, high)

DDR2 (Not consuming a substance)
If a person does not consume a certain product,
and this product affects a certain aspect of the driver's state,
then this aspect will be low

not(consumed(P)) & affects(P,S)
↠ has_level(S, low)

Note that relationship DDR1 and DDR2 can also be replaced by mathematical formulae, stating, for example, how the consumption of alcohol influences the concentration of alcohol in the blood, and how this concentration decreases over time. To this end, more detailed medical knowledge (e.g., about drug and alcohol uptake by the blood) could be exploited.

DDR3 (From high drug level to abnormal steering)
If a person has a high drug level,
then (s)he will perform abnormal steering behaviour

has_level(drug_level, high)
↠ performed(abnormal_steering_behaviour)

DDR4 (From high alcohol level to abnormal steering and staring)
If a person has a high alcohol level,
then (s)he will perform abnormal steering behaviour
and (s)he will stare

has_level(alcohol_level, high)
↠ performed(abnormal_steering_behaviour) ∧ performed(staring)

DDR5 (From fatigue to staring)
If a person has a high state of fatigue,
then (s)he will stare

has_level(fatigue, high)
↠ performed(staring)

Finally, a number of relationships are used to model the processes related to eye blinking, as shown below. Here, d1 and d2 are constants that represent some basic eye dryness in case of, respectively, a high or low state of fatigue. Moreover, s1 and s2 represent the speed by which the eyes become drier in case of, respectively, a high or low state of fatigue.

DDR6 (Effect of fatique on eye dryness)
If a person has a high state of fatigue,
then the person's basic eye dryness will be d1,
and the person's eye drying speed will be s1,

has_level(fatigue, high)
↠ basic_eye_dryness(d1) & eye_drying_speed(s1)

DDR7 (Basic eye dryness)
If a person has a low state of fatigue,
then the person's basic eye dryness will be d2,
and the person's eye drying speed will be s2,

has_level(fatigue, low)
↠ basic_eye_dryness(d2) & eye_drying_speed(s2)

DDR8 (Eye blinking)
If a person's eye dryness is X,
and the person's eye dryness norm is Y,
and X > Y,
then the person will perform an eye blink

eye_dryness(X) &
eye_dryness_norm(Y) &
X > Y
↠ performed(eye_blink)

DDR9 (Effect of blinking on dryness)
If a person performs an eye blink,
and the person's basic eye dryness is d,
then the person's eye dryness will be d

performed(eye_blink) &
basic_eye_dryness(D)
↠ eye_dryness(D)

DDR10 (Effect of not blinking on dryness)

If a person does not perform an eye blink,
and the person's eye dryness is X,
and the person's eye drying speed is S,
then the person's eye dryness will be X+S*(1-X)

not(performed(eye_blink)) &
eye_dryness(X) &
eye_drying_speed(S)
→» eye_dryness(X+S*(1-X))

4.1.3 Example Simulation

This section presents an example simulation trace that was generated on the basis of the domain model for driver behaviour. This trace is displayed in Figure 14. The settings for this trace were: d1=0.1, d2=0.2, s1=0.05. s2=0.15, and eye_dryness_norm=0.5. The upper graph denotes the symbolic predicates, and the lower to graphs denote the numerical concepts (i.e., eye dryness). Note that predicates of type affects(…) have been left out, for readability.

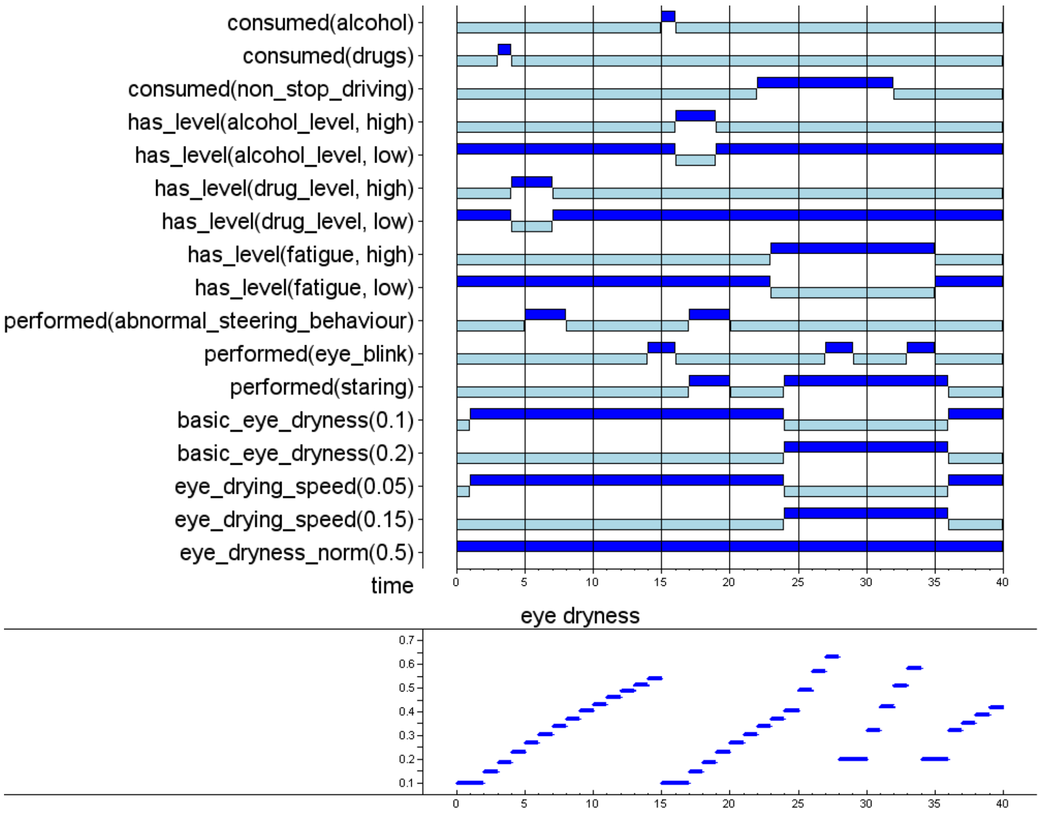

Figure 14. Example simulation trace for domain model.

As shown in Figure 14, the person involved initially has a low concentration of drugs and alcohol in the blood, and a low state of fatigue. However, at time point 3 (s)he uses some drugs, which leads to an increased drug level between time point 4 and 7 (it is assumed that this lasts 3 time steps), and abnormal steering behaviour. After that, everything is back to normal, until time point 15. At that moment, the person consumes some alcohol, which leads to an increased alcohol level for a while. As a result of this, the person -again- steers abnormally, and now also stares in the distance. However, until that moment, the person has not been very tired (i.e., a low state of fatigue), which resulted in a basic eye dryness of 0.1 and an eye drying rate of 0.05. As a consequence, the person's eyes have been drying very slowly, and (s)he has blinked only once (at time point 14, when the eyes were drier than the norm of 0.5). However, between time point 22 and 32, the person enters a long period of non-stop driving. As a result, his state of fatigue becomes high, and his values for basic eye dryness and eye drying rate increase. This causes the eyes to dry rather quickly, so that the person blinks twice in a relatively short period. After time point 36, everything is back to normal again.

4.2 Model for Analysis

This section introduces the model for analysis of driver behaviour. In Section 4.2.1, the main aspects of the model and their relations are introduced. In Section 4.2.2, the detailed model is provided, and Section 4.2.3 presents an example simulation trace.

4.2.1 Analysis of the Main Aspects and Their Relations

To be able to analyse the dynamics of a driver's behaviour, an ambient agent should, in principle, be equipped with the domain model introduced in Section 4.1, and should be able to reason on the basis of this model. Therefore, the model for analysis should contain more of less the same concepts as the domain model. Below, an analysis model to assess the driver's state of fatigue is worked out in detail. This means that the parts of the domain model that address fatigue are re-used here; to keep the model simple, the parts about drug and alcohol intake are not addressed.

The input that the ambient agent can use to make estimations about the state of fatigue is the amount of *non-stop driving*. The agent can *observe* this, because it knows when the engine is turned on and off. Based on that information, it can derive *beliefs* about the driver's state of *fatigue*. In combination with the agent's *desires* about the driver's state of *fatigue*, this can be used to determine whether there is a *discrepancy* between the driver's believed and desired state of fatigue. However, this is not the only useful information the agent can exploit. Because of the camera on the front mirror, it can also *observe* the driver's *eye blinks*. Therefore, it is useful for the agent to also make predictions (modelled as *beliefs*) about when the driver will perform these *eye blinks*, so that it can compare this with the observed eye blinks in order to update its knowledge in case its beliefs (that were generated only on the basis of the information about non-stop driving) are not entirely correct (note that the exact mechanisms to make such updates are beyond the scope of this chapter). In order to make predictions about eye blinks, the agent may use several of the relationships of the domain model (modelled in the form of *beliefs*). These beliefs may involve *eye dryness*, *basic eye dryness*, *eye drying speed*, and *eye dryness norm*. This leads to the following list of concepts that are needed for the attention analysis model:

- observations on non-stop driving
- observations on eye blinks
- beliefs on fatigue
- beliefs on eye blinks
- beliefs on eye dryness
- beliefs on basic eye dryness
- beliefs on eye drying speed
- beliefs on eye dryness norm
- desires on fatigue
- beliefs on fatigue discrepancy

These dynamic relationships that are formulated based on these concepts are visualised in Figure 15.

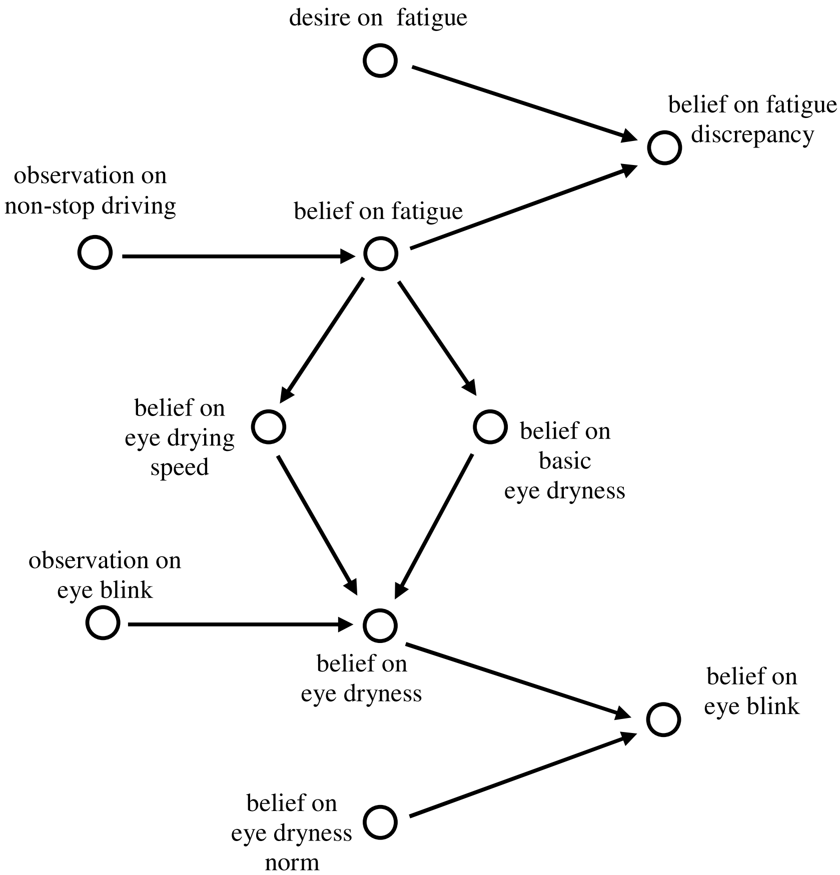

Figure 15. Overview of Analysis Model for Driver Behaviour

4.2.2 The Detailed Model

To formalise the concepts introduced in the previous section, again a number of logical atoms are introduced; see Table 7.

Table 7. Formalisation of concepts used for analysis model

concept	**formalisation**
observations on non-stop driving	observed(agent, consumed(non_stop_driving))
observations on eye blinks	observed(agent, performed(eye_blink))
beliefs on fatigue	belief(agent, has_level(fatigue, L:LEVEL))
beliefs on eye blinks	belief(agent, performed(eye_blink))
beliefs on eye dryness	belief(agent, eye_dryness(I:REAL))
beliefs on basic eye dryness	belief(agent, basic_eye_dryness(I:REAL))
beliefs on eye drying speed	belief(agent, eye_drying_speed(I:REAL))
beliefs on eye dryness norm	belief(agent, eye_dryness_norm(I:REAL))
desires on fatigue	desire(agent, has_level(fatigue, L:LEVEL))
beliefs on fatigue discrepancy	belief(agent, has_discrepancy(fatigue, L:LEVEL))
beliefs on relations between factors and driver states	belief(agent, affects(P:PRODUCT, S:STATE))

Next, to formalise the dynamic relationships for the analysis model, the following relationships in LEADSTO are used:

ADR1 (Generating beliefs for consuming a substance)
if the agent observes that a person consumes a certain product (e.g., non-stop driving),
and the agent believes that this product affects a certain aspect of the driver's state (e.g., fatigue),
then the agent will believe that this aspect will be high

observed(agent, consumed(P)) ∧ belief(agent, affects(P,S))
↠ belief(agent, has_level(S, high))

ADR2 (Generating beliefs for not consuming a substance)
if the agent observes that a person does not consume a certain product,
and the agent believes that this product affects a certain aspect of the driver's state,
then the agent will believe that this aspect will be low

not(observed(agent, consumed(P))) ∧ belief(agent, affects(P,S))
↠ belief(agent, has_level(S, low))

ADR3 (Generating beliefs for effect of high fatique on eye dryness)
if the agent believes that a person has a high state of fatigue,
then the agent believes that the person's basic eye dryness will be d1,
and the agent will believe that the person's eye drying speed will be s1,

belief(agent, has_level(fatigue, high))
↠ belief(agent, basic_eye_dryness(d1)) ∧ belief(agent, eye_drying_speed(s1))

ADR4 (Generating beliefs for effect of low fatique on eye dryness)
if the agent believes that a person has a low state of fatigue,
then the agent believes that the person's basic eye dryness will be d2,
and the agent will believe that the person's eye drying speed will be s2,

belief(agent, has_level(fatigue, low))
↠ belief(agent, basic_eye_dryness(d2)) ∧ belief(agent, eye_drying_speed(s2))

ADR5 (Generating beliefs for eye blinking)
if the agent believes that a person's eye dryness is X,
and the agent believes that the person's eye dryness norm is Y,
and X > Y,
then the agent will believe (predicts) that the person will perform an eye blink

belief(agent, eye_dryness(X)) ∧ belief(agent, eye_dryness_norm(Y)) ∧ x > y
↠ belief(agent, performed(eye_blink))

ADR6 (Generating beliefs for eye dryness)
if the agent observes that a person performs an eye blink,
and the agent believes that the person's basic eye dryness is d,
then the agent will believe that the person's eye dryness will be d

observed(agent, performed(eye_blink)) ∧ belief(agent, basic_eye_dryness(D))
↠ belief(agent, eye_dryness(D))

ADR7 (Generating beliefs for effect of not blinking on dryness)
if the agent observes that a person does not perform an eye blink,
and the agent believes that the person's eye dryness is X,
and the agent believes that the person's eye drying speed is S,
then the agent will believe that the person's eye dryness will be X+S*(1-X)

not(observed(agent, performed(eye_blink))) ∧ belief(agent, eye_dryness(X)) ∧ belief(agent, eye_drying_speed(S))
↠ belief(agent, eye_dryness(X+S*(1-X)))

ADR8 (Assessing fatigue)
if the agent desires that the driver has a low state of fatigue,
and the agent believes that the driver has a high state of fatigue,
then the agent will believe that the there is a (high) discrepancy between the driver's desired and believed state of fatigue

desire(agent, has_level(fatigue, low)) ∧ belief(agent, has_level(fatigue, high))
↠ belief(agent, has_discrepancy(fatigue, high))

4.2.3 Example Simulation

Also for the analysis model, simulation runs may be performed. An example simulation trace for a situation in which the ambient agent signals a discrepancy is shown in Figure 16.

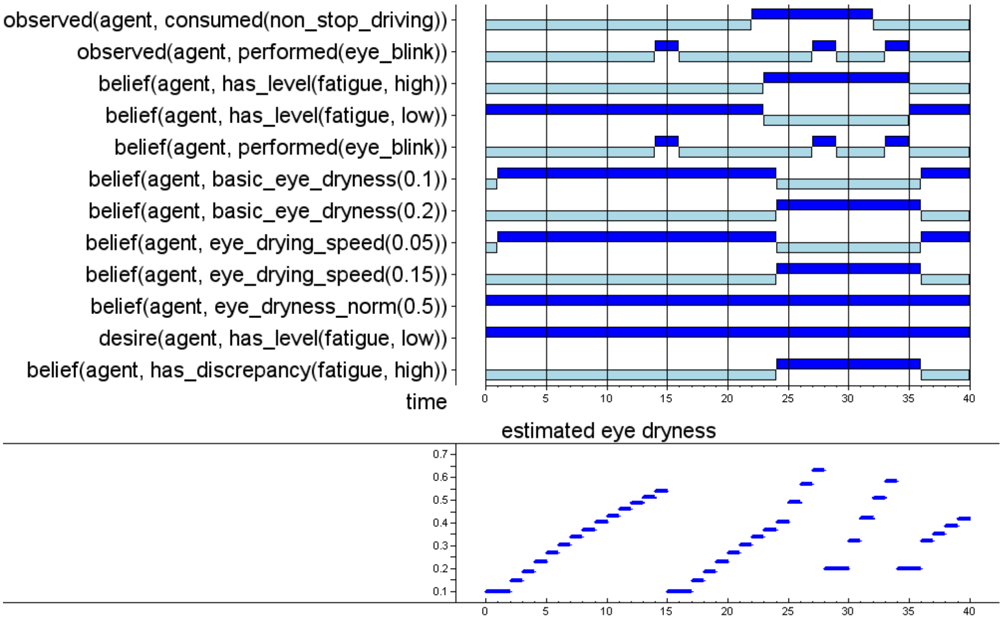

Figure 16. Example simulation trace for analysis model.

As shown in the figure, this scenario is quite similar to the scenario shown in Figure 14 (with all information related to drug and alcohol intake left out). The driver starts with a low state of fatigue, which means that (s)he performs eye blinks at a low frequency. However, between time point 22 and 32, the person drives a long distance without taking a break, which is observed by the agent. Based on this, the agent believes that (s)he is in a high state of fatigue. Since this is in conflict with the agent's desire (namely that the driver is in a low state of fatigue), the agent derives that there is a fatigue discrepancy (which can be input for the support model introduced in the next section).

Furthermore, notice that, based on its belief that the driver is in a high state of fatigue, the agent also predicts that the driver's eyes will become dry very quickly, and that (s)he will blink at time point 27 and 33. Since the agent's analysis model is 100% correct, these predictions correspond exactly to reality. However, in cases that the analysis model is not that perfect, the predicted eye blinks will not exactly overlap with the observed eye blinks. In such cases, the parameters used by the model will have to be updated.

Note that the model presented in this section can only be used to assess fatigue based on observations about non-stop driving (and eye blinks). Thus, it covers the bottom part of Figure 15. However, the upper part of that picture has not been worked out here. To this end, an analysis model to assess the driver's drug and alcohol concentration should be developed. One additional difficulty in that case would be that some relevant concepts that exist in the domain model cannot directly be observed by the ambient agent. For example, the amount of alcohol and drug intake cannot easily be measured by an ambient agent. On the other hand, it is possible to measure information about whether the driver shows abnormal steering behaviour or is staring (the concepts on the upper-right hand side of Figure 15). These types of information can be measured, respectively, by the movements of the steering wheel and the camera on the front mirror.

Thus, it is still possible to obtain information about states in these domain models, but these state are only present at the right-hand side of the model. As a result, no simple forward reasoning pattern (of the format "if I believe X and I believe that X leads to Y, then I also believe Y") can be applied. Instead, other reasoning methods should be applied. An example of such a method is *abduction*. In a nutshell, this method has the format "if I believe Y and I believe that X leads to Y, then I also believe that X could have been the case". By applying such methods, on could derive, for example, that the driver has been drinking on the basis of an abnormal steering pattern that is observed (e.g., in case the steering wheel moves more than 3 times from left to right and back within five seconds).

In case different analysis models are developed, which each analyse one particular aspect of the driver's state (e.g., drug concentration, alcohol concentration, and state of fatigue), the results of analyses can also be combined, in order to provide more evidence for a certain hypothesis (e.g., in case the driver has a high blinking frequency and has not stopped the car for a while, (s)he is very likely to have a high state of fatigue). To this end, it may be useful to develop a multi-agent system, consisting of different (monitoring) agents that each analyse a specific factor separately, and another agent that combines the conclusions of the monitoring agents. Such an approach is taken in (Bosse, Hoogendoorn, Klein, and Treur, 2008). An overview of the multi-agent system used in that paper is shown in Figure 17.

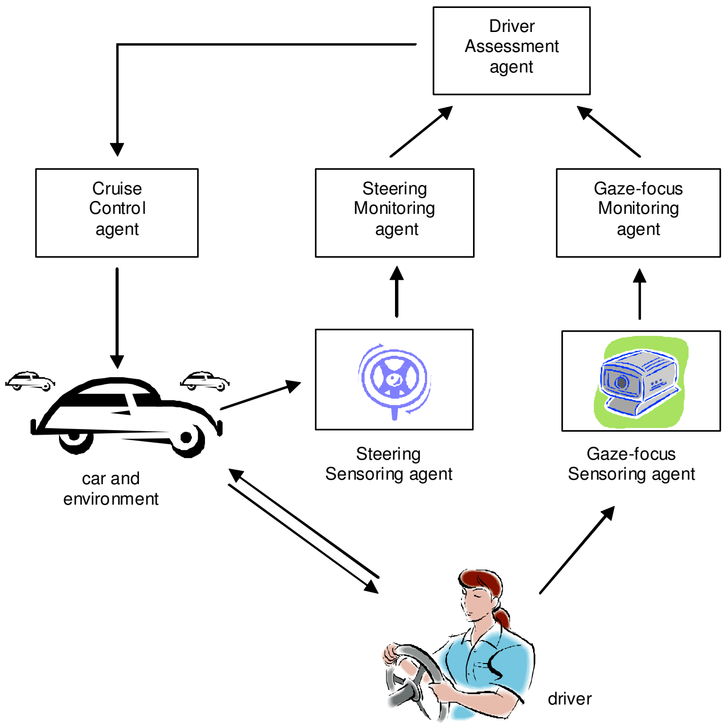

Figure 17. Multi-agent system for ambient driver support (Bosse, Hoogendoorn, Klein, and Treur, 2008).

Here, the Steering Monitoring agent receives the sensor input from the Steering Sensoring agent, and uses this to determine whether there is an abnormal steering pattern. Similarly, the

Gaze-focus Monitoring agent uses sensor input from the Gaze-focus Sensoring agent to determine whether there is an abnormal gaze-focus pattern. Next, the Driver Assessment agent combines the information from the two monitoring agents to determine whether the driver is in an impaired condition. If this is the case, the Driver Assessment agent sends a signal to the Cruise Control agent, who can support the driver, e.g., by slowing down the car. Development of such a system composed of multiple cooperating agents is left as an exercise for the reader.

4.3 Model for Support

This section introduces the model for support of a car driver. In Section 4.3.1, the main aspects of the model and their relations are introduced. In Section 4.3.2, the detailed model is provided, and Section 4.3.3 presents an example simulation trace.

4.3.1 Analysis of the Main Aspects and Their Relations

In case an ambient agent in a car signals that the driver is in an impaired condition, it has several alternatives to provide support. Some possible support actions that are mentioned in the literature are:

- keep a safe distance with the car in front
- slow down the car and safely pull it over
- wake up a sleepy driver (e.g., by making a loud sound)
- give advice to the driver (e.g., tell him or her to take a break)

The choice for which action to perform may depend on the specific state the driver is in. For example, if the agent believes that the driver is only slightly tired, it may advice him or her to take a break. However, when it estimates that the driver is practically asleep, a more urgent measure is necessary.

In sum, the following concepts are needed for the driver support model:

- beliefs about discrepancies between the driver's believed and desired drug level
- beliefs about discrepancies between the driver's believed and desired alcohol
- beliefs about discrepancies between the driver's believed and desired state of fatigue
- actions to keep distance with the car in front
- actions to slow down the car
- actions to wake up a sleepy driver
- actions to give advice to the driver

For the dynamic relationships between these concepts, multiple options are possible that connect a specific belief (or a combination of beliefs) to a particular action. In the next section, an example specification is proposed. The relationships used in that example are visualised in Figure 18.

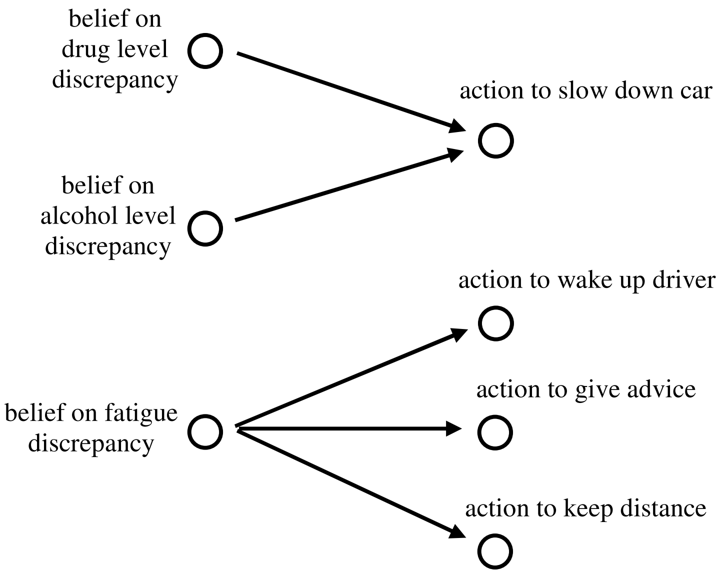

Figure 18. Overview of Support Model for Driver Behaviour

4.3.2 The Detailed Model

To formalise the concepts introduced in the previous section, the following logical atoms are used; see Table 8.

concept	**formalisation**
beliefs of discrepancies of the driver's drug level	belief(agent, has_discrepancy(drug_level, L:LEVEL))
beliefs of discrepancies of the driver's alcohol level	belief(agent, has_discrepancy(alcohol, L:LEVEL))
beliefs of discrepancies of the driver's state of fatigue	belief(agent, has_discrepancy(fatigue, L:LEVEL))
actions to keep distance with the car in front	performed(agent, keep_distance)
actions to slow down the car	performed(agent, slow_down_car)
actions to wake up a sleepy driver	performed(agent, wake_up_driver)
actions to give advice	performed(agent, give_advice)

Table 8. Formalisation of concepts used for support model

To formalise dynamic relationships that connect beliefs to support actions, the following relationships in LEADSTO are used:

SDR1 (Generating a wake up action)
if the agent believes that there is a high discrepancy with respect to the driver's state of fatigue,
then the agent will try to wake up the driver

belief(agent, has_discrepancy(fatigue, high))
↠ performed(agent, wake_up_driver)

SDR2 (Generating advice to keep distance)
if the agent believes that there is a low discrepancy with respect to the driver's state of fatigue,
then the agent will give the driver advice to take a break

belief(agent, has_discrepancy(fatigue, low))
↠ performed(agent, give_advice) ∧ performed(agent, keep_distance)

SDR3 (For alcohol slowing down the car)
if the agent believes that there is a high discrepancy with respect to the driver's alcohol level,
then the agent will try to slow down the car
belief(agent, has_discrepancy(alcohol_level, high))
↠ performed(agent, slow_down_car)

SDR4 (For drugs slowing down the car)
if the agent believes that there is a high discrepancy with respect to the driver's drug level,
then the agent will try to slow down the car
belief(agent, has_discrepancy(drug_level, high))
↠ performed(agent, slow_down_car)

Note that this knowledge base takes a relatively simple approach, based on a 1:1 mapping between an estimated driver state and a set of support actions. The reader is encouraged to develop other, more sophisticated methods for support. For example, the agent could first generate a number of support options, and then select the one with the highest feasibility.

4.3.3 Example Simulation

Simulation of the model presented in Section 4.3.2 is straightforward, and therefore not worked out here.

4.4 Discussion

In order to assess whether a driver is capable of driving a car, an ambient agent needs to maintain a domain model of the state of alertness this person. This section introduced such a domain model (in Section 4.1), which involved various risk factors such as alcohol intake, drug intake, and periods of non-stop driving. In addition, an analysis model was introduced to reason about this domain model (Section 4.2), as well as a support model to determine actions based on the analysis (Section 4.3).

The presented model can be extended in various manners. Some possibilities are mentioned below:

- Taking more factors into account than alcohol, drugs, and fatigue. For example, emotions, other passengers, music, or mobile phones.
- Modelling the concentration of alcohol and drugs in more detail, using a continuous, numerical approach.
- Developing more complex models for analysis (e.g., a model to analyse steering patterns by taking speed and direction of steering movements into account).
- Designing a multi-agent system (instead of a single agent system) for analysis and support of driver behaviour.
- Considering different and more complex types of support (e.g., a mechanism to compare different options and select the one with the highest feasibility).
- Considering sophisticated techniques for adaptation of parameter values.

5 Conclusion

The scientific area that addresses Ambient Intelligence applications in which knowledge from the human-directed sciences is incorporated, has a high potential to provide nontrivial Ambient Intelligence applications based on human-like understanding. Such understanding can result in better informed actions and will feel more natural for humans. The resulting human-environment systems are coupled not only by their mutual interaction, but also in a reflective manner in the sense that both the human and the ambient system have and/or develop a model of the interactive processes of the human and the environment.

In this chapter, a structured way to realize this reflectiveness in Ambient Intelligence applications has been described. This approach comprises the usage of a *domain* model of the process as a basis for an *analysis* model and a *support* model within an agent model of an Ambient Intelligence application. The approach has been illustrated in two different examples.

It is possible to also account for automated adaptation of an Ambient Intelligence application. This could be modelled by integrating the domain model in yet another way within an agent model, i.e. a fourth (sub)model in Figure 3:

- *adaptation model*
  To tune parameters in the domain model better to the specific characteristics of the human by reasoning based on the domain model.

By (human and technological) learning, adaptation and development processes for both the human and the environment the human and technological awareness can grow over time.

Reflective coupled human-environment systems can have a positive impact at different aggregation levels, from individual via an organisation within society to the society as a whole:

- *Individual level*
  - more effective functioning
  - stimulating healthy functioning and preventing health problems to occur
  - support of learning and development
- *Organisation level*
  - efficient functioning organisation by wellfunctioning members
  - learning and adaptation of the organisation
- *Society level*:
  - limiting costs for illness and inability to work
  - efficient management of environment

Some more specific examples of today's societal challenges, to which reflective coupled human-environment systems can contribute, are elderly care, health management, crime and security. Reflective coupled human-environment systems provide a solid foundation for human-like Ambient Intelligence applications with significant benefits for individuals, organisations, and the society as a whole. For some further examples of such Ambient Intelligence applications, see (Azizi, Klein and Treur, 2010; Both, Hoogendoorn, Klein, and Treur, 2009) on ambient support of depressed persons.

References

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.

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

Aziz, A.A., Klein, M.C.A., and Treur, J., (2010). An Integrative Ambient Agent Model for Unipolar Depression Relapse Prevention. *Journal of Ambient Intelligence and Smart Environments*, vol. 2, pp. 5-20.

Baron-Cohen, S. (1995). *Mindblindness*. MIT Press

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 AI Tools*, vol. 16, issue 3, pp. 435-464.

Bosse, T., Hoogendoorn, M., Klein, M.C.A., and Treur, J. (2008). A Component-Based Agent Model for Assessment of Driving Behaviour. In: Sandnes, F.E., Burgess, M., and Rong, C. (eds.), *Proceedings of the Fifth International Conference on Ubiquitous Intelligence and Computing, UIC'08*. Lecture Notes in Computer Science, vol. 5061, Springer Verlag, 2008, pp. 229-243.

Bosse, T., Lambalgen, R. van, Maanen, P.P. van, and Treur, J., (2011). A System to Support Attention Allocation: Development and Application. *Web Intelligence and Agent Systems Journal*, 2011, in press.

Bosse, T., Maanen, P.P. van, and Treur, J. (2006). A Cognitive Model for Visual Attention and its Application. In: Nishida, T., Klusch, M., Sycara, K., Yokoo, M., Liu, J., Wah, B., Cheung, W., and Cheung, Y.-M. (eds.), *Proceedings of the Sixth International Conference on Intelligent Agent Technology, IAT'06*. IEEE Computer Society Press, pp. 255-262.

Bosse, T., Maanen, P.-P. van, Treur, J., Simulation and Formal Analysis of Visual Attention. *Web Intelligence and Agent Systems Journal*, vol. 7, 2009, pp. 89-105.

Both, F., Hoogendoorn, M., Klein, M.C.A., and Treur, J., (2009). Design and Analysis of an Ambient Intelligent System Supporting Depression Therapy. In: Luis Azevedo and Ana Rita Londral (Eds), *Proceedings of the Second International Conference on Health Informatics, HEALTHINF'09*. INSTICC Press, pp. 142-148.

Caffier, P.P., Erdmann, U., and Ullsperger, P. (2003), Experimental evaluation of eye-blink parameters as a drowsiness measure. *European Journal of Applied Physiology*, vol. 89, no. 3-4, pp. 319-325, 2003.

Dennett, D.C. (1987). *The Intentional Stance*. MIT Press. Cambridge Mass.

Gärdenfors, P. (2003), *How Homo Became Sapiens: On The Evolution Of Thinking*. Oxford University Press, 2003.

Goldman, A.I. (2006). *Simulating Minds: The Philosophy, Psychology and Neuroscience of Mind Reading*. Oxford University Press.

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.

Heuvelink, A., Both, F. (2007). BoA: A cognitive tactical picture compilation agent. In: *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2007*. IEEE Computer Society Press, 2007, pp. 175-181.

Itti, L. and Koch, C., Computational Modeling of Visual Attention, *Nature Reviews Neuroscience*, Vol. 2, No. 3, 2001, pp. 194-203.

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

Treur, J., (2008). On Human Aspects in Ambient Intelligence. 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, pp. 262-267.

Wickens, C.D. (2002). Situation awareness and workload in aviation. *Current Directions in Psychological Science 11(4),* pp 128-133.

KEYTERMS

Human Awareness: awareness by the human about the human and environmental processes and their interaction

Technological Awareness: awareness by the environment about the human and environmental processes and their interaction

Situation Awareness: the perception of environmental elements within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future

Tactical Picture Compilation: the task in a military context to identify and classify all entities in the environment by reasoning with the available information, e.g. to determine the threat of a entity.

Visual Attention Manipulation: the task of tracking and steering the visual attention of a human related to the requirements of the task

LEADSTO: a specification language that can be used to formally describe simulation models in a declarative manner, combining numerical and logical statements.

Domain Model: a model of the interaction between the environment and the human in some scenario subject to support via ambient intelligence

Analysis Model: a model that allows for analysis of the human's states and processes by reasoning based on observations (possibly using specific sensors) and a domain model

Support Model: a model that allows for generating support for the human by reasoning based on the domain model

Driver Behaviour: N/A (general term)

ENDNOTES

[i] For a more detailed description of the relation between eye movements and fatigue, see e.g., (Caffier, Erdmann, and Ullsperger, 2003).