Modelling Temporal Aspects of Situation Awareness

Tibor Bosse[1], Robbert-Jan Merk[1,2], and Jan Treur[1]
[1]Vrije Universiteit Amsterdam, Agent Systems Research Group
De Boelelaan 1081, 1081 HV Amsterdam, the Netherlands
[2]National Aerospace Laboratory, Training, Simulation & Operator Performance
Anthony Fokkerweg 2, 1059 CM Amsterdam, The Netherlands
Email: {tbosse, treur}@few.vu.nl merkrj@nlr.nl

Abstract. Computational models of situation awareness can be of interest for different purposes, varying from the study of human cognition in demanding circumstances to the development of human-like virtual opponents in serious gaming applications. This paper presents a novel model of situation awareness, which extends previous work at a number of points. In particular, the model incorporates qualitative time references, offers the possibility to use Allen [1]'s temporal relations, and features an explicit representation of Endsley [4]'s three phases of situation awareness. The behaviour of the model has been tested within a simulation environment for F-16 pilots, and the resulting behaviour has been found satisfactory using a formal verification tool.

Keywords. cognitive modelling, situation awareness, temporal aspects.

1 Introduction

To enhance and maintain capabilities of professionals who have to work in demanding circumstances, often simulation-based training in the form of serious gaming is used. One of the challenges in such applications of serious gaming is to provide a virtual context that has sufficiently realistic characteristics. In cases where other human agents play an important role, in particular these have to be represented as virtual agents with realistic human-like behaviour; e.g., [10]. As humans are not (always) perfect, to be realistic such virtual agents need to be able to display biologically plausible limitations in the form of varying degrees of imperfection. A principled design strategy to develop such agents cannot be based on ad hoc limitations that are incorporated but need input from cognitive, social and neurological sciences to obtain a biologically plausible basis for them. A notion identified in these human-directed disciplines as important for human performance is the notion of *situation awareness* (SA); see also [7].

In [4] situation awareness is defined according to three phases: perception of cues, comprehension and integration of information, and projection onto future events. The extent to which an agent has SA depends on the available cognitive resources, which may be affected, for example, by stress, high work load, fatigue, and time criticality. In demanding circumstances such as in air traffic control or military combat a reduction in SA will often negatively affect performance.

An example application area for virtual agents in simulation-based training is combat flight simulation, a common method used to train fighter pilots, to learn the skills necessary for optimal flight behavior; e.g., [6; 9]. This is the application area addressed here. A computational model for situation awareness is described which has been implemented in an existing training simulator for F-16 pilots. This model reuses some elements of the model described in [5] (for example, the use of an underlying mental model) but it adds a number of other features, for example, the use of qualitative time references, the possibility to use Allen [1]'s temporal relations, and a more explicit representation of Endsley [4]'s three phases.

In the paper, Section 2 describes the modelling approach used, and Section 3 the computational model for SA. In Section 4 it is described how the model has been used in some simulation experiments. Section 5 addresses verification of the model against dynamic properties that are expected. Finally, Section 6 is a discussion.

2 Modelling Approach

To model the different aspects related to the creation of situation awareness from an agent perspective, an expressive modelling language is needed. On the one hand, qualitative aspects have to be addressed, such as observations and beliefs of agents about the world. On the other hand, quantitative aspects have to be addressed. For example, the extent to which a certain belief is active within the agent's working memory can best be described by a real number, and the update of such activation values can best be described by a mathematical formula. Another requirement of the chosen modelling language is its suitability to express on the one hand the basic mechanisms of SA creation (for the purpose of simulation), and on the other hand more global properties of SA (for the purpose of logical analysis and verification). For example, basic mechanisms of SA creation involve functions to derive activations of beliefs, whereas examples of global properties involve the development of such beliefs over longer time periods, like "the activation values of beliefs gradually decay over time".

The hybrid predicate-logical Temporal Trace Language (TTL) [2] fulfils all of these desiderata. It integrates qualitative, logical aspects and quantitative, numerical aspects. This integration allows the modeller to exploit both logical and numerical methods for analysis and simulation. Moreover it can be used to express dynamic properties at different levels of aggregation, which makes it well suited both for simulation and logical analysis.

The TTL language 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 physical and/or mental (state) properties that do or do not hold at a certain point in time. These properties are often called *state properties* to distinguish them from dynamic properties that relate different states over 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. Examples of state properties are 'agent A observes world fact W', or 'agent A has belief B with an activation value of 0.6'. Real value assignments to variables are also considered as possible state property descriptions.

To formalise state properties, ontologies are specified in a (many-sorted) first order logical format: an *ontology* is specified as a finite set of sorts, constants within these sorts, and relations and functions over these sorts (sometimes also called signatures). The examples mentioned above then can be formalised by n-ary predicates (or proposition symbols), such as, for example, observation(A, W) or belief(A, B, 0.6). Such predicates are called *state ground atoms* (or *atomic state properties*). For a given ontology Ont, the propositional language signature consisting of all ground atoms based on Ont is denoted by APROP(Ont). One step further, the *state properties* based on a certain ontology Ont are formalised by the propositions that can be made (using conjunction, negation, disjunction, implication) from the ground atoms. Thus, an example of a formalised state property is belief(A, B1, 0.6) & belief(A, B2, 0.4). Moreover, a *state* S is an indication of which atomic state properties are true and which are false, i.e., a mapping S: APROP(Ont) $\rightarrow$ {true, false}. The set of all possible states for ontology Ont is denoted by STATES(Ont).

To describe dynamic properties of complex processes such as the creation of SA, explicit reference is made to *time* and to *traces*. A fixed time frame T is assumed which is linearly ordered. Depending on the application, it may be dense (e.g., the real numbers) or discrete (e.g., the set of integers or natural numbers or a finite initial segment of the natural numbers). Dynamic properties can be formulated that relate a state at one point in time to a state at another point in time. A simple example is the following (informally stated) dynamic property about an agent's observation:

For all traces γ, there is a time point t such that
at t agent A observes world state W

A *trace* γ over an ontology Ont and time frame T is a mapping $\gamma : T \rightarrow$ STATES(Ont), i.e., a sequence of states γ_t ($t \in T$) in STATES(Ont). The temporal trace language TTL is built on atoms

referring to, e.g., traces, time and state properties. For example, 'in trace γ at time t property p holds' is formalised by state(γ, t) $\models$ p. Here $\models$ is a predicate symbol in the language, usually used in infix notation, which is comparable to the Holds-predicate in situation calculus. *Dynamic properties* are expressed by temporal statements built using the usual first-order logical connectives (such as $\neg$, $\wedge$, $\vee$, $\Rightarrow$) and quantification ($\forall$ and $\exists$; for example, over traces, time and state properties). For example, the informally stated dynamic property introduced above is formally expressed as follows:

$\forall\gamma$:TRACES $\exists$t:TIME state(γ, t) $\models$ observation(A, W)

In addition, language abstractions by introducing new predicates as abbreviations for complex expressions are supported.

To be able to perform (pseudo-)experiments, only part of the expressivity of TTL is needed. To this end, the executable LEADSTO language [3] has been defined as a sublanguage of TTL, with the specific purpose to develop simulation models in a declarative manner. In LEADSTO, direct temporal dependencies between two state properties in successive states are modelled by *executable dynamic properties*. LEADSTO subsumes specifications in difference equation format. Based on TTL and LEADSTO, two dedicated pieces of software have recently been developed. The LEADSTO Simulation Environment [3] takes a specification of executable dynamic properties as input, and uses this to generate simulation traces. Second, to automatically analyse the simulation traces or empirical traces, the TTL Checker tool [2] has been developed. This tool takes as input a formula expressed in TTL and a set of traces, and verifies automatically whether the formula holds for the traces. In case the formula does not hold, the checker provides a counter example, i.e., a combination of variable instances for which the check fails.

For more details of the LEADSTO language and simulation environment, see [3]. For more details on TTL and the TTL Checker tool, see [2].

3 The Computational Model

In this section the computational model for situation awareness is described in detail.

3.1 Overview

In the model, situation awareness is represented by dynamically generated and updated sets of observations and beliefs. The observations represent the input the agent receives from its environment. Observations are represented in the form

observation(AGENT, WORLD_INFO, REAL)

where the first argument is the owner of the observation, the second argument the world info the observation is about and the third argument the certainty of the owner that the world info really holds. Beliefs are represented in the form

belief(AGENT, WORLD_INFO, REAL, TIME)

where the first argument is the owner of the belief, the second argument the world info the belief is about, the third argument the activation value of that belief and the fourth argument for what time the agent holds that belief.

Generation of beliefs makes use of an underlying mental model. This is a network of observations and beliefs where connections have strengths indicated by a real number in the interval [*0, 1*]; along these connections activation is spread throughout the network. A connection has a source observation or belief and a destination belief. Activation is spread from the source via the connection to the destination belief.

3.2 Updating Activation Values

Updating the beliefs consists of three phases (cf. [4]): *perception*, *comprehension* and *projection*. In *perception*, the certainties of observations are determined and used to derive the activation values of those beliefs directly connected with the observations. In *comprehension*, the activation values of beliefs on the present and the past are updated. In *projection*, the activation values of beliefs for the future are updated. Each of the three phases has a time limit which determines how much time the update process for that phase may take. These three time limits model the phenomenon that humans under time pressure take shortcuts in their reasoning and have degraded SA.

Perception

In the perception stage, the certainty values of the observations are used to determine the activation values of a subset of beliefs on the current situation: those connected directly to observations. This updating process continues until either the time limit of the perception phase is reached or all beliefs have been updated. Currently, it is assumed that an observation is only connected to one belief. The update algorithm for the perception stage:

Select from all the beliefs those whose parent[1] is an observation and put these beliefs in the set S_1.
While the *perception* time limit has not been reached and S_1 is not empty
    Select from S_1 the belief B with the highest activation value.
    Update the activation value of B using the belief activation update formula for perception.
    Remove B from S_1.
End while.

The belief activation update formula for perception is as follows.

$$V_B(t+\Delta t) = V_B(t) + [\alpha_B\,(th(\sigma,\ \tau,\ V_O(t)) - V_B(t)) - \gamma_B\,V_B(t)]\,\Delta t$$

Here the symbols mean:

B	The belief for which the activation value is calculated.
O	The parent observation of B.
α_B	The update speed parameter (how fast recent updates influence the activation value) for B.
γ_B	The decay parameter for B
$V_B(t)$	The previous activation value of B.
$V_O(t)$	The certainty value of observation O.
$th(\sigma,\ \tau,\ V_O(t))$	The threshold function, with parameters σ (steepness), and τ (threshold value).

The continuous logistic threshold function used is as follows:

$$th(\sigma,\tau,V) = \left(\frac{1}{1+e^{-\sigma(V-\tau)}} - \frac{1}{1+e^{\sigma\tau)}}\right)(1+e^{-\sigma\tau)})$$

Here

σ	Steepness parameter
τ	Threshold parameter
V	Input value

[1] The parents of a belief or future belief are defined as those (future) beliefs or observations that are the source in a connection which has the belief as destination belief.

Comprehension

In the comprehension phase, the activation values of all the beliefs on the present situation are updated as long as a certain time limit not has been reached. A belief is updated exactly once during a cycle. To ensure that the update of a belief is based on the most recent information, a belief is only updated if all its parent beliefs were updated; the belief graph is assumed to have no cycles. In each iteration, a belief is selected to be updated. For this belief, all the incoming connections that come from an active belief are used to update the activation value of the selected belief. The selected belief is marked as considered and the next iteration starts. This continues until either the time limit of the comprehension phase is reached or all beliefs on the present and past are updated. In pseudocode:

Select from all the beliefs those whose parent is an observation and put these beliefs in the set $S_{considered}$.
Select from all the beliefs on the present and the past those whose parents are all beliefs and put these beliefs in the set S_{todo}.
While the *comprehension* time limit has not been reached and S_{todo} is not empty
  Select from S_{todo} the belief $B_{selected}$ to be updated using *selected_belief(S_{todo}, $S_{considered}$)*.
  Update the activation value of B_i using the update function for belief activation.
  Remove $B_{selected}$ from S_{todo}, add $B_{selected}$ to $S_{considered}$.
End while.

The function *selected_belief(S_{todo}, $S_{considered}$)* finds the belief $B_{selected}$ for which the following constraints hold:

1) $B_{selected}$ is a member of S_{todo},
2) All the parents of $B_{selected}$ are in $S_{considered}$
3) The weighted sum of activation values of the parents $B_{selected}$ is higher than any such value of all beliefs for which constraints 1 and 2 hold.

The calculation of the activation value of a belief B is done by using only those connections whose source belief has an activation value higher than the minimal activation value Min_{belief}. This models the phenomenon that activation only spreads from beliefs that have some degree of activation. The update function for belief activation is as follows.

$$V_B(t+\Delta t) = V_B(t) + [\alpha_B\ (th(\sigma,\ \tau,\ \Sigma_{C \in g(B)}\ \omega_{C,B}\ V_C(t)) - V_B(t)) - \gamma_B\ V_B(t)]\ \Delta t$$

B	The belief for which the activation value is calculated.
α_B	The update speed parameter (how fast recent updates influence the activation value) for B.
γ_B	The decay parameter for B
$V_B(t)$	The previous activation value of B.
$V_O(t)$	The certainty value of observation O.
$th(\sigma,\ \tau,\ V_O(t))$	The threshold function, with parameters σ (steepness), τ (threshold value for B).
$g(B)$	The set of beliefs C connected to belief B as destination
C	A belief from the set $g(B)$.
$\omega_{C,B}$	The strength of connection from C to B.

Projection

If the time limit has not been reached, the projection stage is processed (until the time limit is reached). In projection, beliefs on the future are updated. The update process is the same as for beliefs on the current situation, only with a different start set of beliefs. In pseudocode, the projection process is described by:

Select all beliefs on the present and past and put these in the set $S_{considered}$.
Select all beliefs on the future and put these in the set S_{todo}.
While the *comprehension* time limit has not been reached and S_{todo} is not empty
  Select from S_{todo} the belief $B_{selected}$ to be updated using *selected_belief(S_{todo}, $S_{considered}$)*.
  Update the activation value of B_i using the update function for belief activation.
  Remove $B_{selected}$ from S_{todo}, add $B_{selected}$ to $S_{considered}$.
End while.

4 Simulation Results

The model has been tested against a case study in the domain of air-to-air combat in a tactical fighter simulator. A simulated fighter airplane was controlled by an agent equipped with the SA model as described in this paper.

The scenario that was used in the simulator is as follows: the agent flies over an enemy Surface-to-Air Missile (SAM) site, a ground-to-air weapon system that can launch radar-guided missiles to take down aircraft. As the agent flies within the weapon range of the SAM site, the site launches a missile, which is then defeated by the agent. The scenario ends when the agent leaves the SAM site's weapon range. While the scenario is simple, it is important that any agents operating in the domain can deal with threats like SAM sites in a believable and realistic way.

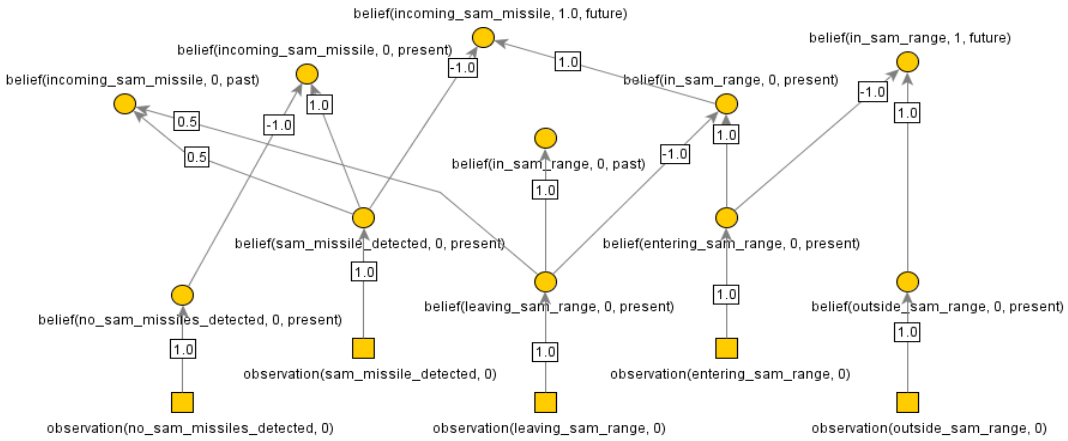

Fig. 1: Belief graph for the case study

Fig. 1 shows the connections and beliefs used for the scenario. Two concepts have all three time labels: if a missile is incoming and whether the agent is within the SAM site's weapon range. The observations and connected beliefs are necessary to derive the six higher-level beliefs.

Two trials have been executed, one in which the time limits are set so that all beliefs are updated continuously (the 'Full SA' trial), and one in which due to the time limits settings only half of all beliefs are updated (the 'Reduced SA' trial). Fig. 2 and Fig. 3 show the progression of activation values for the various beliefs updated in the perception, comprehension and projection phases (time in seconds on horizontal axis, activation value on vertical axis).

In the trial with full SA, the beliefs of the perception phase accurately represent the current changes in the world: for example entering the SAM range ($t \approx 35$) and leaving it ($t \approx 135$). In the comprehension phase, the agent remembers that it just has been in the SAM range after leaving it ($t \approx 135$-170). As the agent leaves the SAM range, it assumes that somewhere in the future it will run into another (see future belief of in_sam_range in the projection phase).

In the trial with reduced SA, the activation values of the agent's beliefs are different. As only a limited amount of beliefs are updated, some beliefs that should have a high activation value do not have this. For example, the present belief of leaving_sam_range near the end of the trial has a low activation value while it should be high, and (as a result) the same holds for the past belief of in_sam_range.

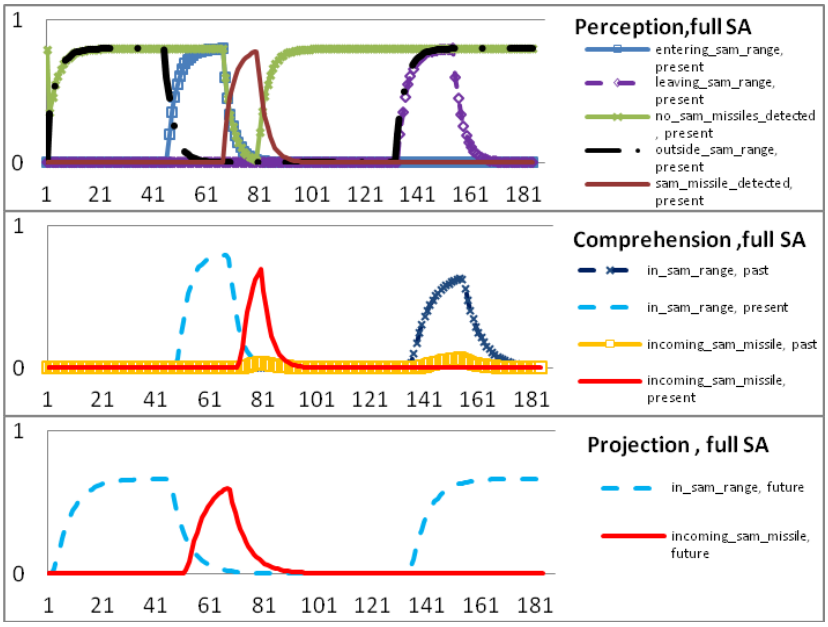

Fig. 2: Activation values with full Situation Awareness.

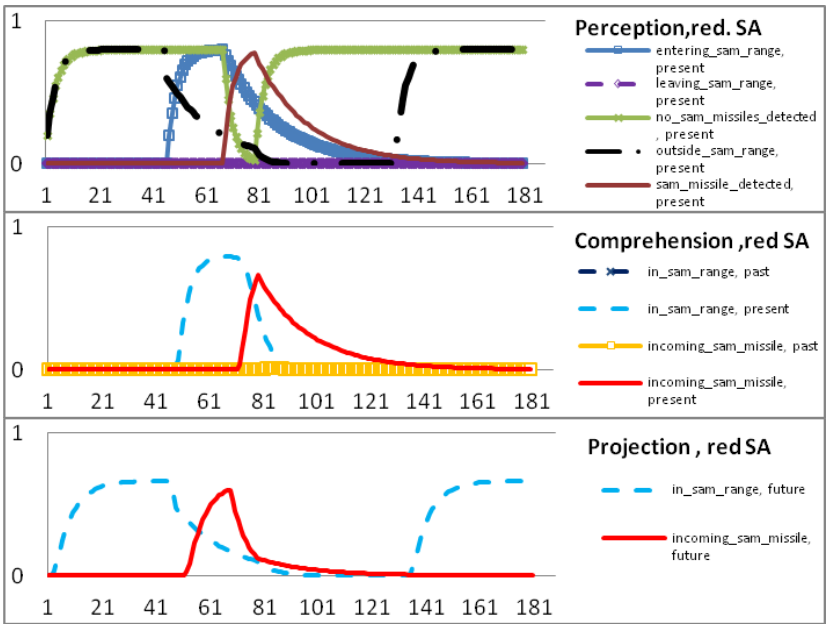

Fig. 3: Activation values with reduced Situation Awareness

5 Verification of Dynamic Properties

This section addresses analysis of the simulation model by verification of dynamic properties. Following [2], the dynamics of a simulation model can be studied by specifying certain dynamic statements that are (or are not) expected to hold in terms of temporal logical expressions, and

automatically verifying these statements against simulation traces. The purpose of this type of verification is to check whether the simulation model behaves as it should. A typical example of a property that may be checked is whether no unexpected situations occur, such as a variable running out of its bounds (e.g., some activation value > 1). By running a large number of simulations and verifying such properties against the resulting simulation traces, the modeler can easily locate the sources of errors in the simulation.

For the SA model presented in this paper, various dynamic properties have been formalized in the language TTL [2], and have been checked against the simulation traces described in Section 4. Below, a number of them are introduced, both in semi-formal and in informal notation (note that they are all defined for a particular trace and time interval between tb and te):

P1 - Monotonic Decrease of Activation Values
For all time points t1 and t2 between tb and te in trace γ
if at t1 the activation value of the agent's belief of world info w at interval i is x1
and at t2 the activation value of the agent's belief of world info w at interval i is x2
and $t1 < t2$
then $x1 \geq x2$.

$P1(\gamma{:}TRACE, tb, te{:}TIME, w{:}WORLD\_INFO, i{:}INT) \equiv$
$\forall t1,t2{:}TIME\ x1,x2{:}REAL$
$state(\gamma, t1) \models belief(w, x1, i) \ \&\ state(\gamma, t2) \models belief(w, x2, i)\ \&$
$tb \leq t1 \leq te\ \&\ tb \leq t2 \leq te\ \&\ t1 < t2$
$\Rightarrow x1 \geq x2$[2]

Property P1 can be used to check whether the activation values of certain beliefs decrease (or increase) monotonically over a certain time period. For instance, after certain world information is observed, the corresponding present belief (according to graphs as in Fig. 1) should first increase for a number of time steps, and should then decrease monotonically until the same observation is made again. As a concrete example, within the 'Full SA' trace, the activation value of present belief sam_missile_detected should decrease monotonically from time point 83 until the end of the simulation (see also Fig. 2). This indeed turned out to be the case: property P1(full_SA_trace, 83, 200, sam_missile_detected, present) succeeded. For the other beliefs, similar checks were performed, both for the Full SA and for the Reduced SA trace. The results indicated that the activation values of all beliefs indeed decrease over the expected time periods.

P2 - Activation Values between Boundaries
For all time points t between tb and te in trace γ
if at t the activation value of the agent's belief of world info w at interval i is x
then $min < x < max$.

$P2(\gamma{:}TRACE, tb, te{:}TIME, w{:}WORLD\_INFO, i{:}INT, max, min{:}REAL) \equiv$
$\forall t{:}TIME\ x{:}REAL$
$state(\gamma, t) \models belief(w, x, i)\ \&\ tb \leq t \leq te$
$\Rightarrow min \leq x \leq max$

This property can be used to check whether the activation values stay between certain boundaries. For the current model, they should never become lower than 0 or higher than 1, which indeed turned out to be the case for the generated traces (i.e., property P2(trace, 0, 200, w, i, 0, 1) succeeded for all traces, w, and i). A similar property (not shown here) was checked for the strength of all connections in the model; these also turned out to stay between 0 and 1.

P3 - Generation of Present Beliefs
For all time points t1 between tb and te in trace γ
if at t1 the agent observes world info w (with certainty 1.0)
then within d time points the agent will have a present belief of w with an activation value x of at least p.

[2] Note that a stronger variant of this (and similar) properties can be created by replacing $\geq$ by $>$.

P3(γ:TRACE, tb, te:TIME, w:WORLD_INFO, d:TIME, p:REAL) ≡
∀t1:TIME
state(γ, t1) ⊨ observation(w, 1.0) & tb ≤ t1 ≤ te
⇒ ∃t2:TIME ∃x:REAL [state(γ, t2) ⊨ belief(w, x, present) & t1 ≤ t2 ≤ t1+d & x ≥ p]

By checking property P3, one can check whether the generation of present beliefs is performed correctly. Also, the property can be used to check whether the activation value of the belief reaches a specified threshold p, and whether this happens within a specified time period d. For example, for the Full SA trace, this property pointed out that any observation by an agent is converted within 13 time steps into a corresponding present belief with at least activation value 0.78: property P3(trace, 0, 200, w, 13, 0.78) succeeded for all w. For the Reduced SA trace, this property did not succeed. The reason for this is, among others, that the observation of leaving_sam_range is not converted into a corresponding belief in this trace. This illustrates that, with reduced SA, the agent generally has less resources to convert observations into beliefs.

6 Discussion

In recent years, the concept of situation awareness has received an increasing attention within Artificial Intelligence, Cognitive Science, and Human-Computer Interaction. Computational models of SA can be of interest for different purposes, varying from the study of human cognition in demanding circumstances to the development of human-like virtual opponents in serious gaming applications. In this paper, a novel SA model has been presented, which has been inspired by [5]. This model has been based upon the three key stages within situation awareness as defined by Endsley [4]: the perception of cues, the comprehension and integration of information, and the projection of information into future events.

The current model extends the model by [5], among others, by incorporating qualitative time references, the possibility to use Allen [1]'s temporal relations, and a more explicit representation of Endsley [4]'s three phases. Moreover, the behaviour of the model has been tested by implementing it within an agent that acts in a simulation environment for F-16 pilots. The runs of the simulator have been logged in the form of *traces*, and using a formal verification tool [2], a number of relevant dynamic properties have been checked against these traces. Although this is obviously not an exhaustive proof for the correctness of the model, it provides a first indication that the model behaves as intended.

Future work will address further testing of the model, using different and more complex scenarios. In addition, a more elaborated experiment is currently being set up, in which multiple domain experts will play realistic missions against the model, and will be asked to evaluate various aspects of the agent's perceived behaviour by means of a questionnaire.

Regarding related work, other models have been proposed for the design of intelligent agents with SA, see e.g. [7; 12]. However, these models are limited as they do not represent all necessary aspects and stages of SA as have been distinguished in this paper. A more detailed model of SA can be found in [8], but it does not make use of a general method to integrate observations into higher level beliefs and is therefore difficult to apply in new situations. A computational model proposed for SA that takes Endsley's model as a point of departure is described in [11]. The first two phases are covered, but the temporal projection phase of Endsley's model is not modelled in [11].

References

1. Allen, J.F.: An interval-based representation of temporal knowledge, Proceedings of the Seventh International Joint Conference on Artificial Intelligence (IJCAI), pp. 221-226 (1981)

2. Bosse, T., Jonker, C.M., van der Meij, L. Sharpanskykh, A., and Treur, J.: Specification and Verification of Dynamics in Agent Models. International Journal of Cooperative Information Systems, vol. 18, pp. 167-193 (2009)
3. Bosse, T. Jonker, C.M., Meij, L. van der, and Treur, J.: A Language and Environment for Analysis of Dynamics by Simulation. International Journal of Artificial Intelligence Tools, volume 16, issue 3, pp. 435-464 (2007)
4. Endsley, M.R.: Toward a theory of Situation Awareness in dynamic systems. Human Factors 37(1), 32-64 (1995)
5. Hoogendoorn, M., van Lambalgen, R.M., and Treur, J.: Modeling Situation Awareness in Human-Like Agents using Mental Models. In: Walsh, T. (ed.), Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, IJCAI'11, pp. 1697-1704 (2011)
6. Jacobs, J.W. and Dempsey, J.V.: Simulation and gaming: Fidelity, feedback, and motivation. In: J. V. Dempsey & G. C. Sales (Eds.), Interactive instruction and feedback (pp. 197-227). Englewood Cliffs, NJ: Educational Technology (1993)
7. Jones, R.M., Laird, J.E., Nielsen, P.E., Coulter, K.J., Kenny, P. & Koss, F.V.: Automated intelligent pilots for combat flight simulation. AI Magazine 20(1), 27-42 (1999)
8. Juarez-Espinoza, O. and Gonzalez, C.: Situation awareness of commanders: a cognitive model. Paper presented at the conference on Behavior representation in Modeling and Simulation (BRIMS), Arlington, VA (2004)
9. Salas, E. and Cannon-Bowers, J.A.: The science of training: a decade of progress. Annual Review of Psychology 52, 471-499 (2001)
10. Silverman, B.G., Cornwell, J, Johns, M., and O'Brien, K.: Human behavior models for agents in simulators and games: part I: enabling science with PMFserv. Presence: Teleoperators and Virtual Environments 15(2), 139-162 (2006)
11. So, R. and Sonenberg, L., Situation Awareness in Intelligent Agents: Foundations for a Theory of Proactive Agent Behavior. In: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04), pp. 86-92 (2004)
12. Wickens, C.D., McCarley, J.S., Alexander, A.L., Thomas, L.C., Ambinder, M., and Zheng, S.: Attention-Situation Awareness (A-SA) model of pilot error. In: D.C. Foyle, & B.L. Hooey (Eds.) Human Performance Modeling in Aviation. CRC Press, Taylor and Francis Group, NW (2008)