

Representation for reciprocal agent–environment interaction

Action editor: Vasant Honavar

Tibor Bosse^{a,*}, Catholijn M. Jonker^b, Jan Treur^a

^a *Vrije Universiteit Amsterdam, Department of Artificial Intelligence, De Boelelaan 1081a, NL 1081 HV Amsterdam, The Netherlands*

^b *Delft University of Technology, Department of Mediametrics, Man Machine Interaction Group, Mekelweg 4, 2628 CD Delft, The Netherlands*

Received 15 August 2005; received in revised form 13 August 2008; accepted 25 November 2008

Available online 21 June 2009

Abstract

Declarative modelling approaches in principle assume a notion of representation or representational content for the modelling concepts. The notion of representational content as discussed in literature in cognitive science and philosophy of mind shows complications as soon as agent and environment have an intense reciprocal interaction. In such cases an internal agent state is affected by the way in which internal and external aspects are interwoven during (ongoing) interaction. In this paper it is shown that the classical correlational approach to representational content is not applicable, but the temporal-interactivist approach is. As this approach involves more complex temporal relationships, formalisation was used to define specifications of the representational content more precisely. These specifications have been validated by automatically checking them on traces generated by a simulation model. Moreover, by mathematical proof it was shown how these specifications are entailed by the basic local properties.

© 2009 Elsevier B.V. All rights reserved.

Keywords: Representation; Mental states; Declarative modelling; Philosophy of mind

1. Introduction

Declarative modelling approaches go hand in hand with some assumed notion of representation or representational content for the modelling concepts. Within cognitive and philosophical literature, classical approaches to representational content are based on correlations between an agent's internal state properties and external state properties. For example, the presence of a horse in the field is correlated to an internal state property that plays the role of a percept for this horse. One of the critical evaluations of this approach addresses the limitation that it is static: internal state properties are to be related to single external states, and cannot be related to processes involving multiple states or events over time. Especially in cases where the agent–

environment interaction takes the form of an extensive reciprocal interplay in which both the agent and the environment contribute to the process in a mutual dependency, a classical approach to representational content is insufficient. Some authors even claim that it is a bad idea to aim for a notion of representation in such cases (e.g., Keijzer, 2002; Sun, 2000). Therefore, these cases can be considered a serious challenge to declarative methods.

As an alternative, within Philosophy of Mind, the *interactivist* approach (Bickhard, 1993) is put forward. In Jonker and Treur (2003) it is shown how a temporal-interactivist approach to representational content of an internal state property can be formalised based on sets of agent–environment past and future interaction trajectories or traces.

In this paper it is analysed how some non-classical approaches may be used to define representational content in the case of an extensive agent–environment interplay. In particular, for a case study it will be discussed how the temporal-interactivist approach and second-order approach to representational content can be used. These alternative notions involve more complex temporal

* Corresponding author. Tel.: +31 20 5987750.

E-mail addresses: tbosse@cs.vu.nl (T. Bosse), catholijn@mmi.tudelft.nl (C.M. Jonker), treur@cs.vu.nl (J. Treur).

URLs: <http://www.cs.vu.nl/~tbosse> (T. Bosse), <http://mmi.tudelft.nl/~catholijn> (C.M. Jonker), <http://www.cs.vu.nl/~treur> (J. Treur).

relationships between internal and external states. Formalisation to define specifications of the representational content more precisely was used as a means to handle this complexity. This formalisation provided dynamic properties that can be (and actually have been) formally checked for given traces of the agent–environment interaction.

In Section 2 the modelling approach is briefly introduced. Section 3 introduces the case study and the language used to model this case study. In Section 4 a number of local dynamic properties describing basic mechanisms for the case study are presented; simulations on the basis of these local dynamic properties are discussed in Section 5. Section 6 presents global dynamic properties, describing the process as a whole and larger parts of the process. In Section 7 the interlevel relations between these nonlocal properties and the local properties are discussed. In Section 8 three different approaches to representational content are explored and formalised for the case study. In Section 9 it is shown how these formalisations can be validated against the simulation model, both by mathematical proof and by automated checks. Section 10 is a discussion.

2. Modelling approach

To formally specify dynamic properties that express criteria for representational content from a temporal perspective an expressive language is needed. To this end the *Temporal Trace Language* is used as a tool; cf. (Jonker & Treur, 2002). In this paper for most of the occurring properties both informal or semi-formal and formal representations are given. The formal representations are based on the *Temporal Trace Language (TTL)*, which is briefly defined as follows.

A *state ontology* is a specification (in order-sorted logic) of a vocabulary, i.e., a signature. A state for ontology *Ont* is an assignment of truth-values {true, false} to the set *At(Ont)* of ground atoms expressed in terms of *Ont*. The set of all possible states for state ontology *Ont* is denoted by *STATES(Ont)*. The set of *state properties* *STATPROP(Ont)* for state ontology *Ont* is the set of all propositions over ground atoms from *At(Ont)*. A fixed *time frame* *T* is assumed which is linearly ordered. A *trace* or *trajectory* γ over a state ontology *Ont* and time frame *T* is a mapping $\gamma: T \rightarrow \text{STATES}(\text{Ont})$, i.e., a sequence of states γ_t ($t \in T$) in *STATES(Ont)*. The set of all traces over state ontology *Ont* is denoted by *TRACES(Ont)*. Depending on the application, the time frame *T* may be dense (e.g., the real numbers), or discrete (e.g., the set of integers or natural numbers), or any other form, as long as it has a linear ordering. The set of *dynamic properties* *DYNPROP(Σ)* is the set of temporal statements that can be formulated with respect to traces based on the state ontology *Ont* in the following manner.

Given a trace γ over state ontology *Ont*, the input state of the organism (i.e., state of sensors for external world and body) at time point *t* is denoted by *state* (γ , *t*, input); anal-

ogously, *state*(γ , *t*, output), *state* (γ , *t*, internal) and *state* (γ , *t*, EW) denote the output state, internal state and external state (of the world, including the physical body) for the organism.

These states can be related to state properties via the formally defined satisfaction relation \models , comparable to the Holds-predicate in the Situation Calculus (see Reiter, 2001 for an introduction, and (Pozos Parra, Nayak, & Demolombe, 2005) for an example application): *state*(γ , *t*, output) $\models p$ denotes that state property *p* holds in trace γ at time *t* in the output state of the organism. Based on these statements, dynamic properties can be formulated in a formal manner in a sorted first-order predicate logic with sorts *T* for time points, *Traces* for traces and *F* for state formulae, using quantifiers over time and the usual first-order logical connectives such as \wedge , \vee , \Rightarrow , \forall , \exists .

To model direct temporal dependencies between two state properties, the simpler *leads to* format is used. This is an executable format defined as follows: let α and β be state properties of the form “conjunction of literals” (where a literal is an atom or the negation of an atom), and *e, f, g, h* non-negative real numbers. In the *leads to* language $\alpha \rightarrow_{e, f, g, h} \beta$, means:

If state property α holds for a certain time interval with duration g ,

then after some delay (between e and f) state property β will hold for a certain time interval of length h .

For a precise definition of the *leads to* format in terms of the language TTL, see Jonker, Treur, and Wijngaards (2003). A specification of dynamic properties in *leads to* format has as advantages that it is executable and that it can often easily be depicted graphically. The *leads to* format has shown its value especially when temporal or causal relations in the (continuous) physical world are modelled and simulated in an abstract, non-discrete manner; for example, the intracellular chemistry of *Escherichia coli* (Jonker, Snoep, Treur, Westerhoff, & Wijngaards, 2008).

3. The case study

In this section the case study will be introduced and the internal state properties and their dynamics to model this example are presented.

3.1. Introduction of the case study

The case study addressed involves the processes to unlock a front door that sticks. Between the moment that the door is reached and the moment that the door unlocks the following reciprocal interaction takes place:

- the agent puts rotating pressure on the key,
- the door lock generates resistance in the interplay,

- the agent notices the resistance and increases the rotating pressure,
- the door increases the resistance,
- and so on, without any result.
- finally, after noticing the impasse the agent changes the strategy by at the same time pulling the door and turning the key, which unlocks the door.

This example shows different elements. The first part of the process is described in terms of Sun's subconceptual level, whereas the last part of the process is viewed in terms of the conceptual level (Sun, 2000, 2002). For both parts of the process the notion of representational content will be discussed and formalised.

3.2. State properties

To model the example the following internal state properties are used:

s1	sensory representation for being at the door
s2(r)	sensory representation for resistance r of the lock
p1(p)	preparation for the action to turn the key with rotating pressure p (without pulling the door)
p2	preparation for combined pulling the door and turning the key
c	state for having learnt that turning the key should be combined with pulling the door

The interactions between agent and environment are defined by the following sensor and effector states:

o1	observing being at the door
o2(r)	observing resistance r
a1(p)	action turn the key with rotating pressure p (without pulling the door)
a2	action turn the key while pulling the door

In addition, the following state properties of the world are used:

arriving_at_door	the agent arrives at the door
lock_reaction(r)	the lock reacts with resistance r
door_unlocked	the door is unlocked
d(mr)	resistance threshold mr of the door (indicating that the door will continue to resist until pressure mr or more is used)
max_p(mp)	maximal force on the key that can be exercised by the agent.

4. Local dynamic properties

To model the dynamics of the example, the following local properties (in *leads to* format) are considered. They describe the basic parts of the process.

LP1 (observation of door).

The first local property LP1 expresses that the world state property *arriving_at_door* leads to an observation of being at the door. Formalisation:

$$\text{arriving_at_door} \rightarrow \text{o1}$$

LP2 (observation of resistance).

Local property LP2 expresses that the world state property *lock_reaction* with resistance r leads to an observation of this resistance r.

$$\text{lock_reaction}(r) \rightarrow \text{o2}(r)$$

Note that r is a variable here; the specification should be read as a schema for the set of all instances for r.

LP3 (sensory representation of door).

Local property LP3 expresses that the observation of being at the door leads to a sensory representation for being at the door.

$$\text{o1} \rightarrow \text{s1}$$

LP4 (sensory representation of resistance).

LP4 expresses that the observation of resistance r of the lock leads to a sensory representation for this resistance.

$$\text{o2}(r) \rightarrow \text{s2}(r)$$

LP5 (action preparation initiation).

LP5 expresses that a sensory representation for being at the door leads to a preparation for the action to turn the key with pressure 1.

$$\text{s1} \rightarrow \text{p1}(1)$$

LP6 (pressure adaptation).

LP6 expresses the following: if turning the key with a certain pressure p did not succeed (since the agent received a resistance that equals p), and the agent has not reached its maximal force ($p < mp$), and the agent has not learnt anything yet (not c), then it will increase its pressure.

$$\text{p1}(p) \text{ and } \text{s2}(r) \text{ and } p = r \text{ and } p < mp \text{ and not } c \rightarrow \text{p1}(p + 1)$$

LP7 (birth of learning state).

LP7 expresses that, if turning the key with a certain pressure p did not succeed (since the agent received a resistance that equals p), and the agent has reached the limit of its force ($p \geq mp$), then it will learn that should perform a different action.

$$\text{p1}(p) \text{ and } \text{s2}(r) \text{ and } p = r \text{ and } p \geq mp \rightarrow c$$

LP8 (learning state persistency).

LP8 expresses that the learning state property c persists forever.

$$c \rightarrow c$$

LP9 (alternative action preparation).

LP9 expresses that a sensory representation for resistance r of the lock together with the learning state

property lead to a preparation for combined pulling of the door and turning the key.

$$c \text{ and } s2(r) \rightarrow p2$$

LP10 (action performance).

LP10 expresses that a preparation for the action to turn the key with pressure p (without pulling the door) leads to the actual performance of this action.

$$p1(p) \rightarrow a1(p)$$

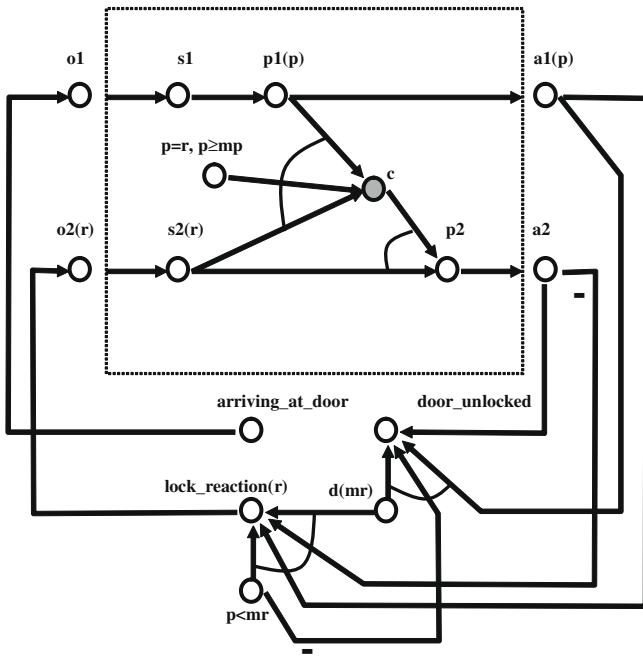


Fig. 1. Overview of the simulation model.

LP11 (alternative action performance).

LP11 expresses that a preparation for combined pulling of the door and turning the key leads to the actual performance of this action.

$$p2 \rightarrow a2$$

LP12 (negative effect of action).

LP12 expresses the following property of the world: if the key is turned with a certain pressure p that is smaller than the maximal resistance of the door ($p < mr$), and the agent is not pulling the door simultaneously, then the lock will react with resistance p .

$$a1(p) \text{ and not } a2 \text{ and } d(mr) \text{ and } p < mr \rightarrow \text{lock_reaction}(p)$$

LP13 (positive effect of action).

LP13 expresses the following property of the world: if the key is turned with a certain pressure p that at least equals the maximal resistance of the door ($p \geq mr$), then the door will unlock.

$$a1(p) \text{ and } d(mr) \text{ and } p \geq mr \rightarrow \text{door_unlocked}$$

LP14 (positive effect of alternative action).

LP14 expresses the following property of the world: if the agent turns the key, and simultaneously pulls the door, then the door will unlock.

$$a2 \rightarrow \text{door_unlocked}$$

In (Fig. 1) an overview of these properties is given in a graphical form. To limit complexity, local property LP6 is not depicted.

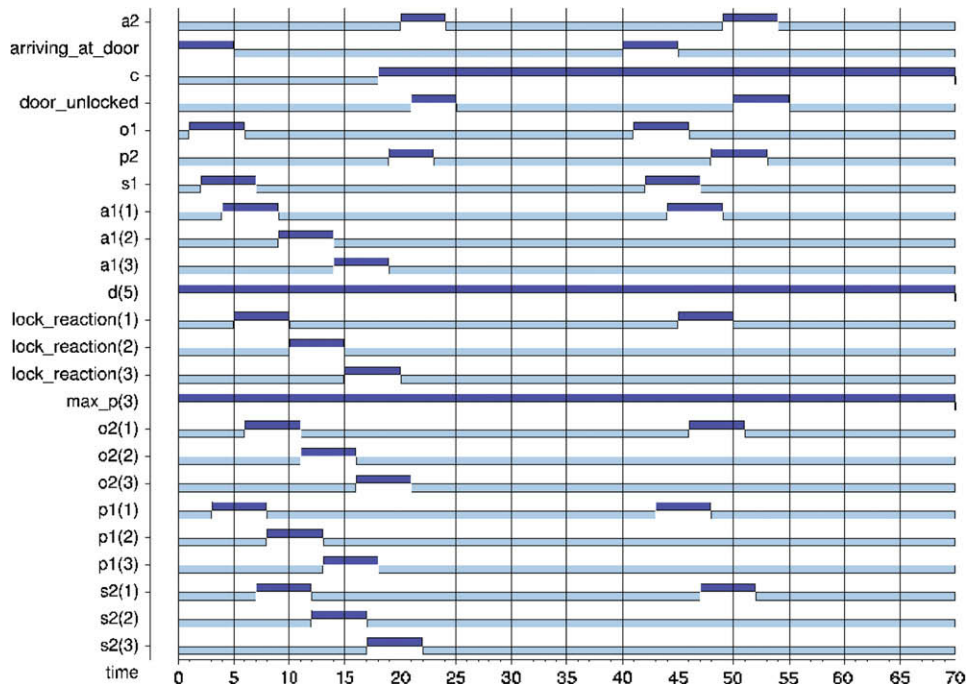


Fig. 2. Example simulation trace.

5. Simulation

A special software environment has been created to enable the simulation of executable models. Based on an input consisting of dynamic properties in *leads to* format, the software environment generates simulation traces. An example of such a trace can be seen in Fig. 2. Time is on the horizontal axis, the state properties are on the vertical axis. A dark box on top of the line indicates that the property is true during that time period, and a lighter box below the line indicates that the property is false. This trace is based on all local properties identified above. In property LP6, the values (0, 0, 1, 5) have been chosen for the timing parameters *e*, *f*, *g*, and *h*. In all other properties, the values (0, 0, 1, 1) have been chosen. As can be seen in Fig. 2, the presence of the agent at the door leads to a corresponding observation result (σ_1), followed by a sensory representation for being at the door. Next, the agent prepares for turning the key (initially with pressure 1), and subsequently performs this action. Since this pressure is insufficient to unlock the door (within this example, the resistant threshold of the door is 5), the door does not open, but a lock reaction (with resistance 1) occurs instead. As a consequence, the agent observes this resistance, and creates a sensory representation of it. At this point, the agent prepares to increase the pressure (see local property LP6), resulting in the action of turning the key with pressure 2. This loop is being activated once more: the agent even tries to turn the key with pressure 3, but then reaches the limit of its force (3 in this example, see LP7) and learns that it should perform a different action. In other words, internal state property *c* becomes true.

Subsequently, the combination of this state property *c* and state property *s2(3)* leads to the preparation for an alternative action: combined pulling of the door and turning the key. As a result of this preparation, the action is actually performed and the door is unlocked. After that, to show that the agent has indeed learned something, the trace continues for a while. At time point 40, the agent again finds itself confronted with a locked door. Again, it starts by trying to turn the key with pressure 1. However, when this approach turns out not to work, this time the agent shows adapted behaviour. It does not try to increase the pressure, but immediately switches to the alternative action instead.

6. Non-local dynamic properties

This section presents dynamic properties for larger parts of the process, i.e., at a nonlocal level. Within these properties, γ is a variable that stands for an arbitrary trace.

GP1 (door eventually unlocked).

Global property GP1 expresses that eventually the door will be unlocked.

$$\forall t: \text{state}(\gamma, t, \text{EW}) \models \text{arriving_at_door} \Rightarrow \\ \exists t' \geq t: \text{state}(\gamma, t', \text{EW}) \models \text{door_unlocked}$$

GP2 (learning occurs).

Global property GP2 expresses that if the maximal resistance of the door is bigger than the maximal rotation force that the agent can exert, then at some point in time learning will occur.

$$\forall t: \text{state}(\gamma, t, \text{EW}) \models d(\text{mr}) \wedge \\ \forall t: \text{state}(\gamma, t, \text{internal}) \models \text{max_p}(\text{mp}) \wedge \text{mr} > \text{mp} \Rightarrow \\ \exists t' \text{state}(\gamma, t', \text{internal}) \models c$$

GP3 ($\text{mr} > \text{mp} \Rightarrow$ door eventually unlocked).

Global property GP3 expresses that if the maximal resistance of the door is bigger than the maximal rotation force that the agent can exert, then at some point in time the door will be unlocked.

$$\forall t: \text{state}(\gamma, t, \text{EW}) \models d(\text{mr}) \wedge \\ \forall t: \text{state}(\gamma, t, \text{internal}) \models \text{max_p}(\text{mp}) \wedge \text{mr} > \text{mp} \Rightarrow \\ \exists t' \text{state}(\gamma, t', \text{EW}) \models \text{door_unlocked}$$

GP4 ($\text{mr} \leq \text{mp} \Rightarrow$ door eventually unlocked).

Global property GP4 expresses that if the maximal resistance of the door is less than or equal to the maximal rotation force that the agent can exert, then at some point in time the door will be unlocked.

$$\forall t: \text{state}(\gamma, t, \text{EW}) \models d(\text{mr}) \wedge \\ \forall t: \text{state}(\gamma, t, \text{internal}) \models \text{max_p}(\text{mp}) \wedge \text{mr} \leq \text{mp} \Rightarrow \\ \exists t' \text{state}(\gamma, t', \text{EW}) \models \text{door_unlocked}$$

GP3 and GP4 are formulated separately because their proofs differ. Next a number of intermediate properties are formulated that form a kind of milestones in the process of opening a door and learning.

M1 (at door \Rightarrow preparation to turn key).

Intermediate property M1 expresses that after the agent stands at the door the agent will prepare for turning the key.

$$\forall t: \text{state}(\gamma, t, \text{EW}) \models \text{arriving_at_door} \Rightarrow \\ \exists t' > t: \text{state}(\gamma, t', \text{internal}) \models p1(1)$$

M2 (lock reaction represented).

Intermediate property M2 expresses that a lock reaction will be represented internally.

$$\forall t: \text{state}(\gamma, t, \text{EW}) \models \text{lock_reaction}(r) \Rightarrow \\ \exists t' > t: \text{state}(\gamma, t', \text{internal}) \models s2(r)$$

M3 (alternative action).

M3 expresses that if lock resistance is internally represented and the agent has learned, then at some later point in time the agent will perform the action *a2*.

$$\forall t: \text{state}(\gamma, t, \text{internal}) \models c \wedge \text{state}(\gamma, t, \text{internal}) \models s2(r) \Rightarrow \\ \exists t' > t: \text{state}(\gamma, t', \text{output}) \models a2$$

M4 (increasing rotation pressure).

M4 expresses that under the condition that agent has not learned *c* yet, the rotation pressure that the agent exerts on the key will always reach the minimum of the maximal resistance of the door and the maximal force that the agent can exert.

$\forall t, \forall mp, \forall mr, \forall sl$
 $\text{not state}(\gamma, t, \text{internal}) \models c \wedge \text{state}(\gamma, t, \text{EW}) \models d(\text{mr}) \wedge$
 $\text{state}(\gamma, t, \text{internal}) \models \text{max_p}(mp) \wedge sl = \text{minimum}(mr,$
 $mp) \wedge$
 $\text{state}(\gamma, t, \text{EW}) \models \text{arriving_at_door} \Rightarrow$
 $\exists t' > t: \text{state}(\gamma, t', \text{internal}) \models p1(sl) \wedge \exists t'' > t':$
 $\text{state}(\gamma, t'', \text{output}) \models a1(sl)$
 Finally, a number of additional properties are needed in order to prove the relations between the properties.

A1 (maximal force).

Additional property A1 expresses that the maximal rotation force that the agent can exert on the key is constant.

$\exists mp \forall t: \text{state}(\gamma, t, \text{internal}) \models \text{max_p}(mp)$

A2 (maximal resistance).

Additional property A2 expresses that the maximal resistance that the door can offer is constant.

$\exists mr \forall t: \text{state}(\gamma, t, \text{EW}) \models d(\text{mr})$

A3 (Closed World Assumption).

The second order property that is commonly known as the Closed World Assumption expresses that at any point in time a state property that is not implied by a specification to be true is false. Let Th be the set of all local properties LP1–LP14.

$\forall P \in \text{At}(\text{ONT}) \forall t: \text{not Th} \vdash \text{state}(\gamma, t) \models P \Rightarrow \text{state}(\gamma, t) \models \text{not P}$

7. Interlevel relations

This section outlines the interlevel connections between dynamic properties at different levels, varying from dynamic properties at the local level of basic parts of the process to dynamic properties at the global level of the overall process. The following interlevel relations between local dynamic properties and non-local dynamic properties can be identified.

GP3 & GP4	\Rightarrow GP1
M2 & M4 & LP7 & LP12	\Rightarrow GP2
M2 & M3 & M4 & LP7 & LP14	\Rightarrow GP3
M4 & LP13	\Rightarrow GP4
LP1 & LP3 & LP5	\Rightarrow M1
LP2 & LP4	\Rightarrow M2
LP8 & LP9 & LP11	\Rightarrow M3
M1 & M2 & LP6 & LP10 & LP12 & A1 & A2 & A3	\Rightarrow M4

The proofs of M1, M2, M3, and GP1 are rather straightforward and left out. A proof sketch of the other properties is provided.

Property M4 can be proved by induction. The induction step is

$\forall t: \text{state}(\gamma, t, \text{output}) \models a1(p) \wedge p < sl \Rightarrow$
 $\exists t1 > t, \exists t2 > t1:$
 $\text{state}(\gamma, t1, \text{internal}) \models p1(p + 1) \wedge \text{state}(\gamma, t2, \text{output}) \models a1(p + 1)$

The induction base is given by properties M1 and LP10, providing $p1(1)$, and $a1(1)$. The induction step is proved along the following lines.

- “not a2” holds at all times during which “not c” holds.

This is proved on the basis of “not c” and A3. A3 states that if a2 cannot be derived from the specification at a certain point in time, then “not a2” holds at that time. So pick any point in time at which “not c” holds and try to prove a2 from all local properties and the additional assumptions A1, A2, and A3. If a2 can be proven, it is due to LP11. The condition of LP11 is p2. The only way to prove p2 is through LP9. The conditions of LP9 are c and s2(r). The condition c is in direct contradiction with “not c”. In the above the temporal elements of the proof were not mentioned. To complete this proof these elements do play a role, for example, c cannot change its truth more than once. It starts out false and remains false until (by application of LP7) it becomes true. Once c is true, it remains true by application of LP8. Therefore, as long as “not c” holds, “not a2” also holds (and even a bit longer).

- a1(p) holds

In proving the induction step, the condition is assumed. Thus a1(p) holds.

- d(mr) holds

Direct from A2.

- p < mr

This is true, since p < sl and sl is the minimum of mp and mr.

- lock_reaction(p) holds.

All conditions of LP12 hold (i.e., a1(p), not a2, d(mr), p < mr), thus LP12 can be applied, which makes sure that lock_reaction(p) holds some time later.

- s2(p) holds

Based on lock_reaction(p), M2 can be applied, thus some time later s2(p) will hold.

- p1(p + 1) holds at some time point t1 later than the chosen time t.

By application of LP6 some time later (call this time point t_1) $p_1(p+1)$ will hold. Note that the conditions of LP6 are met: $p < mp$ holds, since $p < sl$, and sl the minimum of mp and mr .

- $a_1(p+1)$ holds at some time t_2 later than t_1 .

This is proved by applying LP10 with $p+1$. This proves that the induction step holds.

Now assuming that the antecedent of M4 holds, implies that subsequently (over time) LP1, LP3, LP5 and LP10 can be applied. In that manner, from arriving at the door, an observation of that fact is derived, leading to an internal representation thereof (s_1), leading to an internal state in which $p_1(1)$ holds, leading to an output state in which $a_1(1)$ holds. Therefore, all circumstances hold for the induction step to be applicable. Application of the induction step leads to the conclusion that at some point in time $p_1(sl)$ holds in the internal state and some time later again $a_1(sl)$ holds in the output state. Thus proving the conclusion of M4 under the assumption that the antecedent of M4 holds. This concludes the proof by induction of M4.

Property GP2 can be proved as follows. Since $mr > mp$, $sl = mp$. Applying M4 gives us $\exists t'$: $state(\gamma, t', output) \models a_1(mp)$. By application of LP12, we get some time later $lock_reaction(mp)$, application of M2 gives us, some time later again, $s_2(mp)$. Finally, application of LP7 provides us with the learned c .

The proof of Property GP3 follows the following subsequent time points of interest: application of M4 gives a time point t_1 such that $p_1(mp)$ holds, application of M2 give a time t_2 such that $s_2(mp)$ holds, application of LP7 gives a time t_3 such that c holds, application of M3 gives a time t_4 such that a_2 holds, application of LP14 gives a time t_5 such that $door_unlocked$ holds.

The proof of property GP4 is rather short, by application of M4 at a certain time t_1 $a_1(mr)$ will hold, by application of LP13 a later time t_2 exist at which $door_unlocked$ holds. All proofs can be worked out in more details by using the timing parameters of the local properties involved.

8. Representational content

In the literature on Philosophy of Mind different types of approaches to representational content of an internal state property have been put forward, for example the correlational, interactivist, relational specification and second-order representation approach (cf. Kim, 1996, pp. 191–193, 200–202; Bickhard, 1993). These approaches have in common that the occurrence of the internal state property at a specific point in time is related to the occurrence of other state properties, at the same or at different time points. The “other state properties” can be of three types:

- external world state properties*, independent of the agent,

Table 1
Correlational approach.

Internal state property	Content (backward)	Content (forward)
s_1	arriving_at_door	lock_reaction(1)
$s_2(r)$	lock_reaction(r)	Impossible
$p_1(1)$	arriving_at_door	lock_reaction(1)
$p_1(2)$	Impossible	lock_reaction(2)holds
p_2	Impossible	door_unlocked
c	Impossible	Impossible

- the agent’s sensor state and effector state properties, i.e., the agent’s *interaction state properties* (interactivist approach),
- internal state properties* of the agent (higher-order representation).

Furthermore, the type of relationships can be (1) purely functional *one-to-one correspondences*, (e.g., the correlational approach), or (2) they can involve more *complex relationships* with a number of states at different points in time in the past or future, (e.g., the interactivist approach). So, six types of approaches to representational content are distinguished, that can be indicated by codings such as A1, A2, and so on. Below, examples are given.

8.1. Correlational approach

According to the Correlational approach, the representational content of a certain internal state is given by a one-to-one correlation to another (in principle external) state property: type A1. Such an external state property may exist backward as well as forward in time. Hence, for the current example, the representational content for internal state property s_1 can be defined as world state property *arriving_at_door*, by looking *backward* in time. Intuitively, this is a correct definition, since for all possible situations where the agent has s_1 , it was indeed physically present at the door, and conversely. Likewise, the representational content for internal state property p_2 can be defined as action property a_2 , by looking *forward* in time, or, rather, as world state property *door_unlocked*. However, for many other internal state properties the representational content cannot be defined adequately according to the correlational approach. In these cases, reference should not be made to one single state in the past or in the future, but to a temporal sequence of inputs or output state properties, which is not considered to adequately fit in the correlational approach. An overview for the content of all internal state properties according to the correlational approach (if any), is given in Table 1. These relationships can easily be specified in the language TTL.

8.2. Temporal-interactivist approach

The temporal-interactivist approach (Bickhard, 1993; Jonker & Treur, 2003) relates the occurrence of internal state properties to sets of past and future interaction traces:

type B. This can be done in the form of functional one-to-one correspondences (type B1), or by involving more complex relationships over time (type B2). In this paper the focus is on the more advanced case, i.e., the B2 type. As an example, consider the internal state property c . The representational content of c is defined in a semantic manner by the pair of sets of past interaction traces and future interaction traces (here InteractionOnt denotes the input and output state ontology and IntOnt the internal state ontology; $\gamma_{\leq t}^{\text{InteractionOnt}}$ denotes the trace γ up to t , with states restricted to the interaction states):

Table 2
Temporal-interactivist approach (semantic description).

I.s.p.	Content (backward)	Content (forward)
s1	o1	a1(1)
s2(r)	o2(r)	if c (defined by o1, ..., o2(3)), then a2
p1(1)	o1	a1(1)
p1(2)	o1, a1(1), o2(1)	a1(2)
p1(3)	o1, a1(1), o2(1), a1(2), o2(2)	a1(3)
p2	o1, a1(1), o2(1), a1(2), o2(2), a1(3), o2(3)	a2
c	o1, a1(1), o2(1), a1(2), o2(2), a1(3), o2(3)	if o2(r), then a2

Table 3
Temporal-interactivist approach (syntactic description, backward).

I.s.p.	Content (backward)
s1	is_followed_by(γ , o1, input, s1, internal) & is_preceded_by(γ , s1, internal, o1, input)
s2(r)	is_followed_by(γ , o2(r), input, s2(r), internal) & is_preceded_by(γ , s2(r), internal, o2(r), input)
p1(1)	is_followed_by(γ , o1, input, p1(1), internal) & is_preceded_by(γ , p1(1), internal, o1, input)
p1(2)	$\forall t1, t2, t3 [t1 \leq t2 \leq t3 \ \& \ \text{state}(\gamma, t1, \text{input}) \models o1$ & $\text{interplay_up_to}(\gamma, t2, t3, 1) \ \& \ \text{not} \ [\exists t11, t12, t17$ $[t11 \leq t12 \leq t17 \leq t3$ & $\text{state}(\gamma, t11, \text{input}) \models o1 \ \& \ \text{interplay_up_to}(\gamma, t12, t17, 3)]]$ $\Rightarrow \exists t4 \geq t3 \ \text{state}(\gamma, t4, \text{internal}) \models p1(2)]$ & $\forall t4 [\text{state}(\gamma, t4, \text{internal}) \models p1(2) \Rightarrow \exists t1, t2, t3 t1 \leq t2 \leq t3 \leq t4$ & $\text{state}(\gamma, t1, \text{input}) \models o1 \ \& \ \text{interplay_up_to}(\gamma, t2, t3, 1)]$
p1(3)	$\forall t1, t2, t5 [t1 \leq t2 \leq t5 \ \& \ \text{state}(\gamma, t1, \text{input}) \models o1$ & $\text{interplay_up_to}(\gamma, t2, t5, 2) \Rightarrow \exists t6 \geq t5 \ \text{state}(\gamma, t6, \text{internal}) \models$ $p1(3)]$ & $\forall t6 [\text{state}(\gamma, t6, \text{internal}) \models p1(3) \Rightarrow \exists t1, t2, t5 t1 \leq t2 \leq t5 \leq t6$ & $\text{state}(\gamma, t1, \text{input}) \models o1 \ \& \ \text{interplay_up_to}(\gamma, t2, t5, 2)]$
p2	$\forall t1, t2, t7 [t1 \leq t2 \leq t7 \ \& \ \text{state}(\gamma, t1, \text{input}) \models o1 \ \&$ $\text{interplay_up_to}(\gamma, t2, t7, 3) \Rightarrow \exists t8 \geq t7 \ \text{state}(\gamma, t8, \text{internal}) \models p2]$ & $\forall t8 [\text{state}(\gamma, t8, \text{internal}) \models p2 \Rightarrow \exists t1, t2, t7 t1 \leq t2 \leq t7 \leq t8$ & $\text{state}(\gamma, t1, \text{input}) \models o1 \ \& \ \text{interplay_up_to}(\gamma, t2, t7, 3)]$
c	$\forall t1, t2, t7 [t1 \leq t2 \leq t7 \ \& \ \text{state}(\gamma, t1, \text{input}) \models o1 \ \&$ $\text{interplay_up_to}(\gamma, t2, t7, 3) \Rightarrow \exists t8 \geq t7 \ \text{state}(\gamma, t8, \text{internal}) \models c]$ & $\forall t8 [\text{state}(\gamma, t8, \text{internal}) \models c \Rightarrow \exists t1, t2, t7 t1 \leq t2 \leq t7 \leq t8$ & $\text{state}(\gamma, t1, \text{input}) \models o1 \ \& \ \text{interplay_up_to}(\gamma, t2, t7, 3)]$

$$\text{PITRACES}(c) = \{ \gamma_{\leq t}^{\text{InteractionOnt}} \mid t \in T, \text{state}(\gamma, t, \text{IntOnt}) \models c \}$$

$$\text{FITRACES}(c) = \{ \gamma_{\geq t}^{\text{InteractionOnt}} \mid t \in T, \text{state}(\gamma, t, \text{IntOnt}) \models c \}$$

Here, the first set, $\text{PITRACES}(c)$, contains all past interaction traces for which sequence of time points exists such that at these time points first o1 occurs, next a1(1), next o2(1), next a1(2), next o2(2), next a1(3), and next o2(3). For this example, a learning phase of 3 trials has been chosen. The second set, $\text{FITRACES}(c)$, contains all future interaction traces for which no o2(r) occurs, or o2(r) occurs and after this a2 occurs.

An overview for the representational content of all internal state properties according to the temporal-interactivist approach is given, in an informal notation, in Table 2.

Note that these relationships are defined at a semantic level, and are thus of type B2a. Different interaction state properties, separated by commas, should be read as the temporal sequence of these states. Again, a learning phase of 3 trials has been chosen. In order to obtain a description at a syntactic level, the relationships given in Table 2 are characterised by formulae in a specific language, TTL in our case. Thus, the representational content of a certain internal state is then defined by specifying a formal temporal relation of the internal state property to sensor and action states in the past and future. A number of such formal temporal relations are given in Table 3. Because of space limitations, only the backward content is shown.

Within (Table 3), the following abstractions are used:

$$\text{is_followed_by}(\gamma, X, I1, Y, I2) \equiv$$

$$\forall t1: \text{state}(\gamma, t1, I1) \models X \Rightarrow \exists t2 \geq t1: \text{state}(\gamma, t2, I2) \models Y$$

This expresses that X is always followed by Y.

$$\text{is_preceded_by}(\gamma, Y, I1, X, I2) \equiv$$

$$\forall t1: \text{state}(\gamma, t2, I1) \models Y \Rightarrow \exists t1 \leq t2: \text{state}(\gamma, t1, I2) \models X$$

This expresses that Y is always preceded by X. These abstractions can be used like $\text{is_preceded_by}(\gamma, s1, \text{internal}, o1, \text{input})$, $\text{is_followed_by}(\gamma, o2(1), \text{input}, s2(1), \text{internal})$, et cetera. The next abstraction describes that the interplay between agent and environment in which the agent increases pressure and the environment increases resistance is performed up to a certain level of pressure.

$$\text{interplay_up_to}(\gamma, t1, t2, 1) \equiv t1 \leq t2 \ \&$$

$$\text{state}(\gamma, t1, \text{output}) \models a1(1) \ \& \ \text{state}(\gamma, t2, \text{input}) \models$$

$$o2(1)$$

$$\text{interplay_up_to}(\gamma, t1, t4, 2) \equiv \exists t2, t3 [t1 \leq t2 \leq t3 \leq t4]$$

$$\text{interplay_up_to}(\gamma, t1, t2, 1) \ \&$$

$$\text{state}(\gamma, t3, \text{output}) \models a1(2) \ \& \ \text{state}(\gamma, t4, \text{input}) \models$$

$$o2(2)$$

$$\text{interplay_up_to}(\gamma, t1, t6, 3) \equiv \exists t4, t5 [t1 \leq t4 \leq t5 \leq t6]$$

$$\text{interplay_up_to}(\gamma, t1, t4, 2) \ \&$$

$$\text{state}(\gamma, t5, \text{output}) \models a1(3) \ \& \ \text{state}(\gamma, t6, \text{input}) \models$$

$$o2(3)$$

8.3. Second-order representation

In approaches to representational content of type C, internal state properties are related to other internal state properties. For example, in Sun's dual approach to cognition (Sun, 2000, 2002), conceptual level state properties are related to subconceptual level state properties:

On this view, high-level conceptual, symbolic representation is rooted, or grounded, in low-level behavior (compartment) from which it obtains its meanings and for which it provides support and explanations. The rootedness/groundedness is guaranteed by the way high-level representation is produced: It is, in the main, extracted out of low-level behavioral structures. (Sun, 2000).

Two possibilities arise: either the other internal state properties are not considered to be representational (this seems to be Sun's position), or they are themselves considered representations of something else. In the latter case, which is explored here, the conceptual level state properties become second-order representations: representations of representations. In the main example of this paper, the internal state property *c* can be considered to be at the conceptual level, whereas the other, *s* and *p* properties are considered subconceptual. Then, in the spirit of Sun (2000), the representational content of *c* can be defined in terms of the other internal state properties as shown below. However, keep in mind that this approach only makes sense if the low-level internal state properties are considered to be representational already.

Backward: *c* will occur if in the past once *s*1 occurred, then *p*1(1), then *s*2(1), then *p*1(2), then *s*2(2), then *p*1(3), then *s*2(3), and conversely. Formally:

$$\begin{aligned} & \forall t1, t2, t3, t4, t5, t6, t7 [t1 \leq t2 \leq t3 \leq t4 \leq t5 \leq t6 \leq t7 \\ & \& \text{state}(\gamma, t1, \text{internal}) \models s1 \\ & \& \text{state}(\gamma, t2, \text{internal}) \models p1(1) \& \text{state}(\gamma, t3, \text{internal}) \models \\ & s2(1) \\ & \& \text{state}(\gamma, t4, \text{internal}) \models p1(2) \& \text{state}(\gamma, t5, \text{internal}) \models \\ & s2(2) \\ & \& \text{state}(\gamma, t6, \text{internal}) \models p1(3) \& \text{state}(\gamma, t7, \text{internal}) \models \\ & s2(3) \\ & \Rightarrow \exists t8 \geq t7 \text{state}(\gamma, t8, \text{internal}) \models c] \& \\ & \forall t8 [\text{state}(\gamma, t8, \text{internal}) \models c \Rightarrow \\ & \exists t1, t2, t3, t4, t5, t6, t7 t1 \leq t2 \leq t3 \leq t4 \leq t5 \leq \\ & t6 \leq t7 \leq t8 \\ & \& \text{state}(\gamma, t1, \text{internal}) \models s1 \\ & \& \text{state}(\gamma, t2, \text{internal}) \models p1(1) \& \text{state}(\gamma, t3, \text{internal}) \models \\ & s2(1) \\ & \& \text{state}(\gamma, t4, \text{internal}) \models p1(2) \& \text{state}(\gamma, t5, \text{internal}) \models \\ & s2(2) \\ & \& \text{state}(\gamma, t6, \text{internal}) \models p1(3) \& \text{state}(\gamma, t7, \text{internal}) \models \\ & s2(3)] \end{aligned}$$

Forward: if *c* occurs, then in the future, if *s*2(*r*) occurs, then *p*2 will occur. Formally:

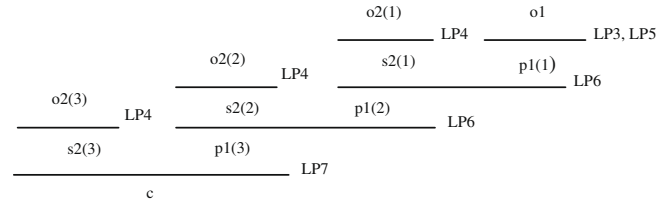
$$\forall t1 [\text{state}(\gamma, t1, \text{internal}) \models c \Rightarrow$$


Fig. 3. Proof tree.

$$\begin{aligned} & \forall t2 \geq t1 [\text{state}(\gamma, t2, \text{internal}) \models s2(r) \Rightarrow \\ & \exists t3 \geq t2 \text{state}(\gamma, t3, \text{internal}) \models p2]] \end{aligned}$$

9. Validation

A large variety of techniques exist for (automated) verification of relevant properties of complex systems (for examples see Pokorny & Ramakrishnan, 2005; Vasconcelos, 2005; Walton, 2005) and the references in these papers. In the current research, the specifications of representational content have been validated in two ways: (1) by relating them to the local dynamic properties by mathematical proof, and (2) by automatically checking them for the simulated traces.

An example of the former is as follows. Consider the formula that presents the backward representational content for internal state property *c* in Table 3. Consider first the direction from observations to *c*. Given *o*1, *o*2(1), *o*2(2), and *o*2(3) at the different subsequent time points the proof obligation is *c*. Given *o*1, by applying (in this order) LP3, LP5 we obtain *p*1(1) which we need to derive from the given *o*2(1) using LP4, *s*2(1) and by application of LP6 on *p*1(1) and *s*2(1) we obtain *p*1(2). Given *o*2(2), by application of LP4 we obtain *s*2(2) and on the basis of *p*1(2) LP6 is again applicable resolving into *p*1(3). Given *o*2(3), apply LP4 to obtain *s*2(3), and using *p*1(3) LP7 is applicable and *c* is obtained. These dependencies are graphically represented in Fig. 3. The reverse direction again depends on property A3 and all local properties.

In addition to the software described in Section 5, other software has been developed that takes traces and formally specified properties as input and checks whether a property holds for a trace. Using automatic checks of this kind, many of the properties presented in this paper have been checked against a number of generated traces as depicted in Fig. 2. In particular, the global properties GP1, GP2, GP3, and GP4, and the intermediate properties M1, M2, M3, and M4 have been checked, and all turned out to hold for the given traces. Furthermore, all properties for representational content denoted in Table 3 have been checked. The duration of these checks varied from one second to a couple of minutes, depending on the complexity of the formula (in particular, the amount of time points). Success of these checks would validate our choice for the representational content (according to the temporal-interactivist approach) of the internal state properties *s*1, *s*2(*r*), *p*1(1), *p*1(2), *p*1(3), *p*2, and *c*. However, note that these checks

are only an empirical validation, they are no exhaustive proof as, e.g., model checking is. Currently, the possibilities are explored to combine TTL with existing model checking techniques.

Although they are not exhaustive, even the empirical checks mentioned above have already proved their value. Initially, one of these checks did not succeed. It turned out that the backward representational content defined for $p1(2)$ was not correctly chosen. At that time, it was defined as follows:

$$\frac{\forall t1, t2, t3 [t1 \leq t2 \leq t3 \ \& \ \text{state}(\gamma, t1, \text{input}) \models o1 \ \& \ \text{interplay_up_to}(\gamma, t2, t3, 1) \Rightarrow \exists t4 \geq t3 \ \text{state}(\gamma, t4, \text{internal}) \models p1(2)] \ \& \ \forall t4 [\text{state}(\gamma, t4, \text{internal}) \models p1(2) \Rightarrow \exists t1, t2, t3 \ t1 \leq t2 \leq t3 \leq t4 \ \& \ \text{state}(\gamma, t1, \text{input}) \models o1 \ \& \ \text{interplay_up_to}(\gamma, t2, t3, 1)]}{}$$

According to the above notation, the sequential occurrence of the state properties $o1$, $a1(1)$, and $o2(1)$ always implies that state property $p1(2)$ will occur. However, a close examination of Fig. 2 reveals that this is not always the case. Whenever the agent has learned, it will not increase its pressure on the key anymore. As a result, the extra condition not c had to be added to the representational content. All the other checks concerning the properties of Table 3 did succeed immediately.

10. Discussion

The classical correlational approach to representational content requires a one-to-one correspondence between an internal state property of an agent and one external world state property. For embodied agents that have an extensive reciprocal interaction with their environment, this classical correlational approach does not suffice. In particular, an internal state in such an agent does not depend on just one state property of the external world, but is affected both by external aspects of the world and by internal aspects of the agent itself and the way in which these aspects are interwoven during the (ongoing) interaction process.

Given this problem, it is under debate among several authors whether adequate alternative notions of representational content exist for such an embodied agent's internal states. Some authors claim that for at least part of the internal states it makes no sense to consider them as conceptual or as having representational content (e.g., Clark, 1997; Keijzer, 2002; Reiter, 2001). Other authors claim that some notions of representational content can be defined, but these strongly deviate from the classical correlational approach (e.g., Bickhard, 1993; Jonker & Treur, 2003; Kim, 1996).

Given the above considerations, the case of an intensive agent–environment interaction is a challenge for declarative approaches in the sense that internal states depending on such an interaction have no simple-to-define representa-

tional content. The formally defined and validated representation relations presented in this paper show how it is still possible to obtain a declarative perspective also for such a case. It is shown how formal methods allow to address the temporal structure entailed by suitable representation relations in these cases in a manageable declarative form.

More specifically, in this paper, for some notions of representational content it was explored in a case study how they work out, and, especially, how the temporal structure can be handled by formalisation. The processes of the case study have been formalised by identifying executable local dynamic properties for the basic dynamics. On the basis of these local properties a simulation model has been made. The formalised specifications of the representational content of the internal state properties have been validated by automatically checking them on the traces generated by the simulation model. Moreover, by mathematical proof it was shown how these specifications are entailed by the basic local properties. This shows that the internal state properties indeed fulfil the representational content specification.

The use of the Temporal Trace Language TTL has a number of practical advantages. In the first place, it offers a welldefined language to formulate relevant dynamic relations in practical domains, with first order logic expressivity and semantics. Furthermore, it has the possibility of explicit reference to *time points* and *time durations* that enables modelling of the dynamics of continuous real-time phenomena, such as sensory and neural activity patterns in relation to mental properties. These features go beyond the expressive power available in standard linear or branching time temporal logics, such as LTL and CTL.

Moreover, the possibility to quantify over traces allows for specification of *more complex adaptive behaviours*. As within most temporal logics, reactivity and pro-activeness properties are specified. In addition, in TTL also properties expressing different types of adaptive behaviour can be expressed. For example a property such as “exercise improves skill”, which is a relative property in the sense that it involves the comparison of two alternatives for the history. Another property of this type is trust monotony: “the better the experiences with something or someone, the higher the trust”. This type of relative property can be expressed in our language, whereas in standard forms of temporal logic different alternative histories cannot be compared. For an excellent review of standard temporal logics (see Vardi, 2001).

Note that, in addition to simulated traces, the TTL checking software is also able to take other (e.g., empirical) traces as input, enabling the validation of the representational content of internal states in real-world situations.

References

- Bickhard, M. H. (1993). Representational content in humans and machines. *Journal of Experimental and Theoretical Artificial Intelligence*, 5, 285–333.

- Clark, A. (1997). *Being there: Putting brain, body and world together again*. MIT Press.
- Jonker, C. M., Snoep, J. L., Treur, J., Westerhoff, H. V., & Wijngaards, W. C. A. (2008). BDI-modelling of complex intracellular dynamics. *Journal of Theoretical Biology*, 251, 1–23.
- Jonker, C. M., & Treur, J. (2002). Compositional verification of multi-agent systems: A formal analysis of pro-activeness and reactiveness. *International Journal of Cooperative Information Systems*, 11, 51–92.
- Jonker, C. M., & Treur, J. (2003). A temporal-interactivist perspective on the dynamics of mental states. *Cognitive Systems Research Journal*, 4, 137–155.
- Jonker, C. M., Treur, J., & Wijngaards, W. C. A. (2003). A temporal modelling environment for internally grounded beliefs, desires and intentions. *Cognitive Systems Research Journal*, 4(3), 191–210.
- Keijzer, F. (2002). Representation in dynamical and embodied cognition. *Cognitive Systems Research Journal*, 3, 275–288.
- Kim, J. (1996). *Philosophy of mind*. Westview Press.
- Pokorny, L. R., & Ramakrishnan, C. R. (2005). Modeling and verification of distributed autonomous agents using logic programming. In *Proceedings of the second international workshop on declarative agent languages and technologies, DALT'04. Lecture notes in artificial intelligence*. Springer Verlag.
- Pozos Parra, P., Nayak, A., & Demolombe, R. (2005). Theories of intentions in the framework of situation calculus. In *Proceedings of the second international workshop on declarative agent languages and technologies, DALT'04. Lecture notes in artificial intelligence*. Springer Verlag.
- Reiter, R. (2001). *Knowledge in action: Logical foundations for specifying and implementing dynamical systems*. MIT Press.
- Sun, R. (2000). Symbol grounding: A new look at an old idea. *Philosophical Psychology*, 13(2), 149–172.
- Sun, R. (2002). *Duality of the mind*. Lawrence Erlbaum.
- Vardi, M. Y. (2001). Branching vs. linear time: Final showdown. In *Proceedings of TACAS 2001 – Tools and algorithms for the construction and analysis of systems. Genova, Italy, April 2–6, 2001. Lecture notes in computer science* (Vol. 2031, pp. 1–22). New York, NY: Springer Verlag.
- Vasconcelos, W. W. (2005). Norm verification and analysis of electronic institutions. In *Proceedings of the second international workshop on declarative agent languages and technologies, DALT'04. Lecture notes in artificial intelligence*. Springer Verlag.
- Walton, C. (2005). Model checking agent dialogues. In *Proceedings of the second international workshop on declarative agent languages and technologies, DALT'04. Lecture notes in artificial intelligence*. Berlin: Springer Verlag.