

A Recursive BDI-Agent Model for Theory of Mind and its Applications^{*}

TIBOR BOSSE
ZULFIQAR A. MEMON
JAN TREUR

tbosse@few.vu.nl
zamemon@few.vu.nl
treur@few.vu.nl

Vrije Universiteit, Department of Artificial Intelligence, De Boelelaan 1081, NL-1081 HV, Amsterdam, The Netherlands

Abstract. This paper discusses a formal BDI-based agent model for Theory of Mind (ToM). The Model uses BDI concepts to describe the reasoning process of an agent that reasons about the reasoning process of another agent, which is also based on BDI concepts. We discuss three different application areas, and illustrate how the model can be applied to each of them. We explore a case study for each of the application areas and apply our model to it. For each case study, a number of simulation experiments are described, and their results are discussed.

Keywords: *Theory of Mind, Mindreading, BDI-agents, Applications.*

1. Introduction

To function efficiently in social life and within organisations, it is useful if agents can reason about the actual and potential behaviour of the agents around them. To this end, it is very helpful for these agents to have capabilities to predict in which circumstances other agents will show certain appropriate or inappropriate behaviours. If for a considered other agent, generation of actions is assumed to be based on a BDI-model, prediction of such actions will involve reasoning based on a Theory of Mind (Baron-Cohen, 1995; Bogdan, 1997; Malle, Moses, and Baldwin, 2001) involving beliefs, desires and intentions as a basis for the behaviour.

Such a Theory of Mind can be exploited by an agent in two different manners. The first manner is just to predict the behaviour in advance, in order to be prepared that it will occur. For example, if an agent B has done things that are known as absolutely unacceptable for an organisation (or a relationship), then he or she may be able to predict and therefore be prepared on what will happen after a manager (or partner) agent A learns about it. A second manner to exploit reasoning based on a Theory of Mind is to try to affect the occurrence of certain beliefs, desires and intentions at forehand, by manipulating the occurrence of circumstances that are likely to lead to them. For example, the agent B just mentioned can try to hide facts so that the manager (or partner) agent A will never learn about the issue. Such capabilities of anticipatory and manipulatory reasoning based on a Theory of Mind about the behaviour of colleague agents are

^{*} Parts of this article are based on work presented at the AISB 2007 Workshop on Mindful Environments (Bosse, Memon, and Treur, 2007a), the Seventh IEEE/WIC/ACM International Conference on Intelligent Agent Technology (Bosse, Memon, and Treur, 2007b), and the 8th International Conference on Cognitive Modeling (Bosse, Memon, and Treur, 2007c).

considered quite important, not to say essential, to function smoothly in social life.

This type of reasoning has an information acquisition and analysis aspect, and a preparation and action aspect. To describe the latter aspect, for the agent using a Theory of Mind, a model for action preparation based on beliefs, desires and intentions can be used as well. For example, for agent B discussed above, the desire can be generated that agent A will not perform the action to fire (or break up with) him or her, and that agent A will in particular not generate the desire or intention to do so. Based on this desire, the refined desire can be generated that agent A will not learn about the issue. Based on the latter desire, an intention and action can be generated to hide facts for agent A. Notice that agent B reasons on the basis of BDI-models at two different levels, one for B itself, and one as the basis for the Theory of Mind to reason about agent A. In human beings, this level of reasoning can be up to 7 levels, see (Dennett, 1987).

It is this multi-level reasoning architecture that is worked out in this paper in a computational model and applied to three different case studies. The proposed model is an example of recursive modelling and is unique in the sense that it is formalised and executable, as compared to the other models available in the literature. Section 7 presents the detailed comparison with the related models from the literature. To illustrate the proposed model and to make it easily understandable, the paper mainly focuses on the two-level case, i.e., one agent reasoning about the mental states of a second agent. However, in principle the approach can be applied to higher levels of nesting as well (e.g., A desires that B believes that C intends...).

The first case study addressed in this paper illustrates how the model can be used for social manipulation. This case study addresses the scenario of a manager that reasons about the task avoiding behaviour of his employee. The second case study is about animal cognition, and illustrates a scenario of a prey that manipulates the behaviour of a predator. The third case study relates to Virtual Storytelling. Virtual storytelling often uses a fixed, pre-scripted storyline, constraining the characters' autonomy and behaviour. An agent-based approach provides possibilities of a narrative emerging from interaction between a number of autonomous characters that have mindreading capabilities. The approach is applied to generate out of a number of interacting characters a soap storyline, in which some of the characters use mindreading in order to mislead and manipulate the other characters.

The vehicle used to formalise the recursive BDI-agent model is the modelling language LEADSTO (Bosse, Jonker, Meij, and Treur, 2007). In this language, 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 of the form 'conjunction of ground atoms or negations of ground atoms'. In the LEADSTO language the notation $\alpha \rightarrow_{e, f, g, h} \beta$, 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.*

Here, atomic state properties can have a qualitative, logical format, such as an expression $\text{desire}(d)$, expressing that desire d occurs, or a quantitative, numerical format such as an expression $\text{has_value}(x, v)$ which expresses that variable x has value v .

In Section 2, first the general BDI-model is explained. Next, Section 3 describes how the simple BDI model can be extended to a BDI-model of an agent that reasons about another agent's BDI-model. Section 4 illustrates this extended BDI-model by a case study that addresses the scenario of a manager that reasons about the task avoidance behaviour of his employee, and how to prevent that behaviour. Section 5 illustrates the model by another case study, in the domain of animal behaviour: this case study addresses a scenario of a prey that analyses the behaviour of a predator, and prevents being attacked. In Section 6 the model is applied within the domain of Virtual Storytelling, to obtain characters acting in an emergent soap story. For all three case studies, some simulation experiments and their results are presented, and potential applications are discussed. Section 7 discusses related work, and Section 8 concludes the paper with a discussion.

2. The BDI Model

The BDI-model bases the preparation and performing of actions on beliefs, desires and intentions, e.g., (Georgeff and Lansky, 1987; Jonker, Treur, and Wijngaards, 2003; Rao and Georgeff, 1991; 1995). This model shows a long tradition in the literature, going back to Aristotle's analysis of how humans (and animals) can come to actions; cf. (Aristotle, 350 BCa; 350 BCb). He discusses how the occurrence of certain internal (mental) state properties within the living being entail or cause the occurrence of an action in the external world. Such internal state properties are sometimes called by him 'things in the soul', for example, sensation, reason and desire:

'Now there are three things in the soul which control action and truth - sensation, reason, desire.' (Aristotle, 350 BCa), Book VI, Part 2.

Here, sensation indicates the sensing of the environment by the agent, which leads, (in modern terms) to internal representations, called beliefs. Reason indicates the (rational) choice of an action that is reasonable to fulfil the given desire. Based on this, Aristotle introduced the following pattern to explain action (called practical syllogism):

If	A has a desire D
and	A has the belief that action AC is a (or: the best) means to achieve D
then	A will do AC

The BDI-model incorporates such a pattern of reasoning to explain behaviour in a refined form. Instead of a process from desire to action in one step, as an intermediate stage first an intention is generated, and from the intention the action is generated. Thus the process is refined into a two-step process. See Figure 1 for the generic structure of the BDI-model in causal-graph-like style, as often used to visualise LEADSTO specifications. Here the box indicates the borders of the agent, the circles denote state properties, and the arrows indicate dynamic properties expressing that one state property leads to (or causes) another state property. In this model, an action is performed when the subject has the intention to do this action and it has the belief that certain circumstances in the world are fulfilled such that the opportunity to do the action is there. Beliefs are created on the basis of observations. The intention to do a specific type of action is created if there is some desire D, and there is the belief that certain circumstances in the world state are there, that make it possible that performing this action will fulfill

this desire (this is the kind of rationality criterion discussed above; e.g., what is called means-end analysis is covered by this). Whether or not a given action is adequate to fulfill a given desire depends on the current world state; therefore this belief may depend on other beliefs about the world state. Instantiated relations within the general BDI-model as depicted by arrows in graphical format in Figure 1 can be specified in formal LEADSTO format as follows:

$$\begin{array}{ll} \text{desire}(D) \wedge \text{belief}(B1) & \rightarrow \text{intention}(AC) \\ \text{intention}(AC) \wedge \text{belief}(B2) & \rightarrow \text{performs}(AC) \end{array}$$

with appropriate desire D , action AC and beliefs $B1, B2$. Note that the beliefs used here both depend on observations, as shown in Figure 1. Furthermore, \wedge stands for the conjunction operator (and) between the atomic state properties (in the graphical format denoted by an arc connecting two (or more) arrows). Often, dynamic properties in LEADSTO are presented in *semi-formal* format, as follows:

At any point in time
 if desire D is present
 and the belief $B1$ is present
 then the intention for action AC will occur

At any point in time
 if the intention for action AC is present
 and the belief $B2$ is present
 then the action AC will be performed

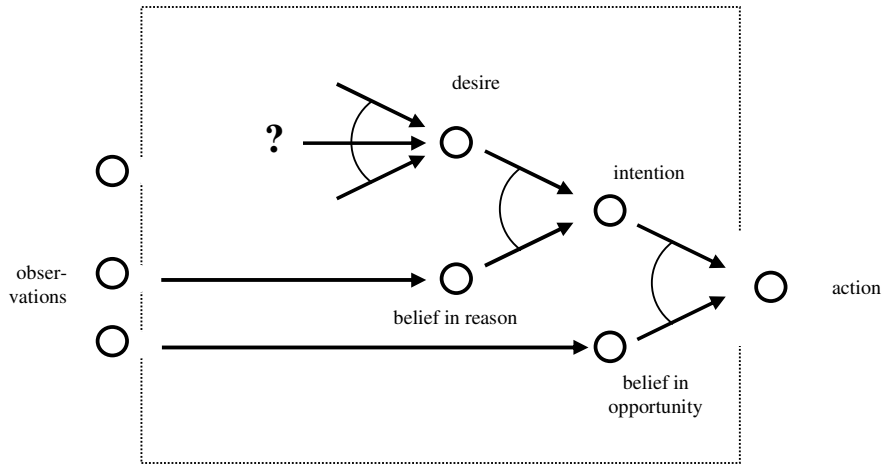


Figure 1. Structure of the general BDI-model

As a generic template, including a reference to the agent A concerned, this can be expressed by:

For any desire D , world state property W , and action AC such that $\text{has_reason_for}(A, D, W, AC)$ holds:

$$\text{desire}(A, D) \wedge \text{belief}(A, W) \rightarrow \text{intention}(A, AC)$$

For any world state property W and action AC such that $\text{is_opportunity_for}(A, W, AC)$ holds:

$$\text{intention}(A, AC) \wedge \text{belief}(A, W) \rightarrow \text{performs}(A, AC)$$

Here $\text{has_reason_for}(A, D, W, AC)$ is a relation that can be used to specify which state property w is considered a reason to intend a certain action AC for desire D .

Similarly $\text{is_opportunity_for}(A, W, AC)$ is a relation that can be used to specify which state property w is considered an opportunity to actually perform an intended action AC .

Assuming that beliefs are available, what remains to be generated in this model are the desires. For desires, there is no generic way (known) in which they are to be generated in the standard model. Often, in applications, generation of desires depends on domain-specific knowledge.

Note that the model as presented here represents the authors' specific interpretation of the BDI framework. For example, for pragmatic reasons, the model assumes rather direct mappings between intentions and actions, and does not consider the potential failure of actions. See the Section 8 for a more elaborated discussion about this issue.

3. The Two-Level BDI-Model

As an instance of the *instrumentalist perspective* and opposed to explanations from a direct physical perspective (the physical stance), in (Dennett, 1987; 1991), the *intentional stance* (or folk-psychological stance) is put forward. In (Dennett, 1991), Dennett explains the advantage of intentional stance explanations for mental phenomena over physical stance explanations:

'Predicting that someone will duck if you throw a brick at him is easy from the folk-psychological stance; it is and will always be intractable if you have to trace the photons from brick to eyeball, the neurotransmitters from optic nerve to motor nerve, and so forth.' (Dennett, 1991), p.42

According to the intentional stance, an agent is assumed to decide to act and communicate based on intentional notions such as beliefs about its environment and its desires and intentions. These decisions, and the intentional notions by which they can be explained and predicted, generally depend on circumstances in the environment, and, in particular, on the information on these circumstances just acquired by interaction (i.e., by observation and communication), but also on information acquired by interaction in the past. To be able to analyse the occurrence of intentional notions in the behaviour of an observed agent, the observable behavioural patterns over time form a basis; cf. (Dennett, 1991).

In the model presented in this paper, the instrumentalist perspective is taken as a point of departure for a Theory of Mind. More specifically, the model describes the reasoning process of an agent B that applies the intentional stance to another agent A by attributing beliefs, desires and intentions. Thus, for agent B a Theory of Mind is obtained using concepts for agent A 's beliefs, desires and intentions.

As a next step, the model is extended with BDI-concepts for agent B 's own beliefs, desires and intentions as well. By doing this, agent B is able to not only have a theory about the mind of agent A , but also to use it within its own BDI-based reasoning processes.

In Section 3.1, the ontological elements needed to describe BDI-concepts at multiple levels are introduced. Based on this ontology, Section 3.2 presents the dynamic properties needed to reason with these concepts.

3.1. Ontology

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

- 1) $\text{belief}(\text{B:AGENT}, \text{depends_on}(\text{IE1:INFO_EL}, \text{IE2:INFO_EL}))$
which expresses that the agent B has the knowledge that state property IE1 depends on state property IE2,
- 2) $\text{belief}(\text{B:AGENT}, \text{leads_to}(\text{IE2:INFO_EL}, \text{IE1:INFO_EL}))$
which represents that the agent has the knowledge that state property IE2 leads to state property IE1
- 3) $\text{belief}(\text{B:AGENT}, \text{leads_to}(\text{at}(\text{has_value}(\text{x}, \text{v}), \text{T}), \text{at}(\text{has_value}(\text{y}, \text{v} + \delta(\text{v}).\text{D}), \text{T} + \text{D})))$
(where $\delta(\text{v})$ is an arithmetical term)
which represents that the agent has the knowledge that variable 'x' having value 'v' at time point T leads to variable 'y' having value 'v + $\delta(\text{v}) \cdot \text{D}$ ' at T + D.

This type of expressions is used to represent the agent's knowledge of a dynamical model of a process. Using the ontology, the functionality of the agent has been specified by generic and domain-specific temporal rules.

Table 1. Generic Ontology for agent's internal states and processes

SORT	Description
ACTION	an action
AGENT	an agent
ACTION_PERFORMANCE	a performed action
WORLD_PROP	a world state property
INFO_RECEIPT	information received
INTERNAL_STATE	an internal state property of an agent
INFO_EL	an information element, possibly complex (e.g., a conjunction of other info elements)
MODEL_RELATION	a meta-statement; represents a <i>dynamic property</i>

SUB-SORT Relationships
$\text{ACTION_PERFORMANCE} \subseteq \text{INFO_EL}$
$\text{WORLD_PROP} \subseteq \text{INFO_EL}$
$\text{INFO_RECEIPT} \subseteq \text{INFO_EL}$
$\text{MODEL_RELATION} \subseteq \text{INFO_EL}$
$\text{INTERNAL_STATE} \subseteq \text{INFO_EL}$

Elements of ACTION_PERFORMANCE Sort	Description
$\text{performs}(\text{A:AGENT}, \text{AC:ACTION})$	Agent A performs Action AC

Elements of WORLD_PROP Sort	Description
$\text{holds_in_world}(\text{IE:INFO_EL})$	IE holds in world

Elements of INFO_RECEIPT Sort	Description
$\text{hears}(\text{A:AGENT}, \text{IE:INFO_EL})$	Agent A hears IE
$\text{observes}(\text{A:AGENT}, \text{IE:INFO_EL})$	Agent A observes IE

Elements of INTERNAL_STATE Sort	Description
belief(A:AGENT, IE:INFO_EL)	Information IE is believed.
desire(A:AGENT, IE:INFO_EL)	Information IE is desired.
intention(A:AGENT, AC:ACTION)	Action AC is intended

Elements of MODEL_RELATION Sort	Description
depends_on(IE1:INFO_EL, IE2:INFO_EL)	an information element IE1 depends on another information element IE2
leads_to(IE1:INFO_EL, IE2:INFO_EL)	an information element IE1 leads to another information element IE2
leads_to(con(IE1:INFO_EL, IE2:INFO_EL), IE3:INFO_EL)	conjunction of two information elements IE1 and IE2 leads to another information element IE3
leads_to(at(has_value(x, v), T), at(has_value(y, v+ $\delta(v)$).D), T+D)) (where $\delta(v)$ is an arithmetical term)	variable x having value v at time T leads to variable y having value v+ $\delta(v)$ *D at time T+D

Table 2 below provides an overview of the predicates used in interactions at the global level. These predicates are using sorts described in Table 1.

Table 2. Predicates used for interaction at global level

Predicate	Description
communication_from_to(IE:INFO_EL, A:AGENT, B:AGENT)	Information IE is communicated by A to B
communicated_from_to(IE:INFO_EL, A:AGENT, B:AGENT)	Information IE was communicated by A to B
has_effect(AC:ACTION, IE:INFO_EL)	Action AC has effect IE
fixed_true(IE:INFO_EL)	IE is a world fact

To this end, a number of meta-representations expressed by meta-predicates using the ontology specified in Table 1 are introduced, e.g.:

belief(B:AGENT, desire(A:AGENT, IE:INFO_EL))

This expresses that agent B believes that agent A has a desire for IE.

desire(B, not(intention(A, AC)))

This expresses that agent B desires that agent A does not intend action AC (note that the Sort declarations have been omitted).

belief(B, depends_on(performs(A, AC), intention(A, AC)))

This expresses that agent B believes that, whether A will perform action AC depends on whether A intends to do AC. Note that the third meta-statement has a more complex structure than the other two, since it represents a statement about a *dynamic property*, rather than a statement about a *state property*. These dependencies can be read from a graph such as depicted in Figures 1 and 2 (right hand side). For example, it is assumed that agent B knows part of this graph in his Theory of Mind, expressed by beliefs such as:

belief(B, depends_on(performs(A, g), intention(A, g)))
belief(B, depends_on(performs(A, g), belief(A, e_n)))
belief(B, depends_on(intention(A, g), desire(A, f)))
belief(B, depends_on(intention(A, g), belief(A, e_{n-1})))
belief(B, depends_on(desire(A, f), belief(A, e₁)))
belief(B, depends_on(desire(A, f), belief(A, e₂)))
belief(B, depends_on(belief(A, e₁), observes(A, e₁)))

These beliefs can also be expressed by the ‘leads_to’ relationship discussed in Table 1, and can be listed as follows:

```
belief(B, leads_to(con(intention(A, g), belief(A, en)), performs(A, g)))
belief(B, leads_to(con(desire(A, f), belief(A, en-1)), intention(A, g)))
belief(B, leads_to(can(belief(A, e1), belief(A, e2)), desire(A, f)))
belief(B, leads_to(observe(A, e1), belief(A, e1)))
```

3.2. Dynamic Properties

Desire refinement in the BDI-model for an agent B attributing motivations to an agent A is formulated (in LEADSTO format) by:

```
desire(B, IE2) ∧ belief(B, depends_on(IE2, IE1)) → desire(B, IE1)
desire(B, IE2) ∧ belief(B, depends_on(IE2, not(IE1))) → desire(B, not(IE1))
desire(B, not(IE2)) ∧ belief(B, depends_on(IE2, IE1)) → desire(B, not(IE1))
desire(B, not(IE2)) ∧ belief(B, depends_on(IE2, not(IE1))) → desire(B, IE1)
```

Similarly, desire refinement for an agent B attributing motivations to an agent A can be formulated in ‘leads_to’ relationship as follows:

```
desire(B, IE2) ∧ belief(B, leads_to(IE1, IE2)) → desire(B, IE1)
desire(B, IE2) ∧ belief(B, leads_to(not(IE1), IE2)) → desire(B, not(IE1))
desire(B, not(IE2)) ∧ belief(B, leads_to(IE1, IE2)) → desire(B, not(IE1))
desire(B, not(IE2)) ∧ belief(B, leads_to(not(IE1), IE2)) → desire(B, IE1)
```

A numerical variant of this is:

```
desire(B, has_value(IE2, v1)) ∧ belief(B, leads_to(has_value(IE1, v2), has_value(IE2, v1))) →
desire(B, has_value(IE1, v2))
```

Moreover the following schemes for intention and action generation are included in the model. For any desire IE1:INFO_EL, world state property IE2:WORLD_PROP, and action AC:ACTION such that has_reason_for(B, IE1, IE2, AC) holds:

```
desire(B, IE1) ∧ belief(B, IE2) → intention(B, AC)
```

For any world state property IE:WORLD_PROP and action AC:ACTION such that is_opportunity_for(B, IE, AC) holds:

```
intention(B, AC) ∧ belief(B, IE) → performs(B, AC)
```

Moreover, some dynamic properties of the world are needed:

```
performs(B, AC) ∧ has_effect(AC, IE) → holds_in_world(IE)
holds_in_world(IE) → observe(A, IE)
(given that A can observe IE)
```

For an overview of the complete two-level BDI-model, see Figure 2. Note that, to illustrate the model, Figure 2 uses the model relation “depends_on”, described in Table 1, which makes this an example instantiation of the generic model, not the generic model itself. Moreover, the variables $a_1, a_2, \dots, a_m, b, d, e \in \{e_1, \dots, e_n\}$, and f are example instances of information elements, which act sometimes as desires, and sometimes as elements that are observed or believed. Similarly, c and g are example instances of actions.

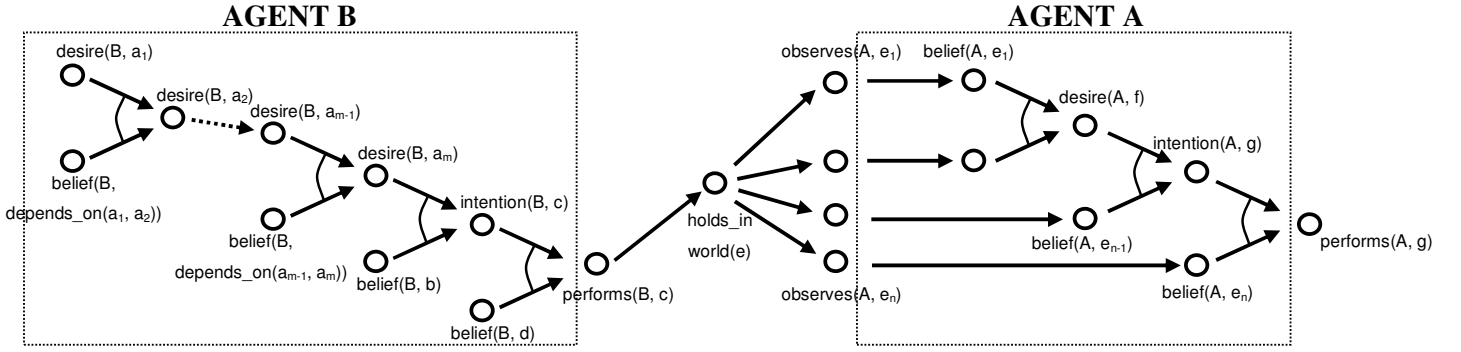


Figure 2. Structure of the Two-Level BDI-model

The model discussed above strictly separates generic knowledge from domain-specific knowledge, which can be filled in within reserved slots (sometimes called *compositionality of knowledge*). As a result, it is very efficient in terms of speed and authoring in a sense that it is reusable in a wide variety of application areas without much implementation effort. These application areas may vary from theoretical to more practical areas in various domains such as cognitive science, life sciences, and social/organisational sciences, and may have both research-driven or engineering-driven goals. For example, the model can be applied in creating virtual characters in games, to make the characters behave more human-like in terms of their reasoning. Incorporating the proposed model in virtual characters will help them to predict the behaviour of other characters or the human player; this allows them to react to a predicted action, instead of only being able to react to an action that has already occurred. In such a domain, the model can be used for rapid prototyping, i.e., to study the behaviour of a system to be designed without actually implementing it. Since the model has been specified in an intuitive, implementation-independent executable format, it is relatively easy to plug it in into any kind of application (e.g., adventure games, training environments, or storytelling applications). On the other hand the model may also be applied from a more theoretical perspective. For instance, within the domain of animal behaviour, it may be used to study the dynamics of mindreading processes in animals by simulating them.

To illustrate the general applicability of the model, in the next three sections, it has been applied to three different case studies, which together more or less cover the wide range of applications sketched above. Section 4 addresses a simple scenario in the context of an organisation, Section 5 addresses a case study in the domain of animal behaviour, and in Section 6 the model is used to create a Virtual Storytelling application.

4. Case Study 1 - Modelling Task Avoidance in Organisations

To illustrate the BDI-model described in Section 2 by a simple example, a specific scenario is addressed (in the domain of an organisation). The scenario is introduced in Section 4.1. Section 4.2 shows how the scenario can be modelled using the presented model, and Section 4.3 discusses simulation experiments that have been performed on the basis of this model.

4.1. Task Avoidance Case

Consider the following scenario within the context of an organisation: a manager observes that a specific employee in the majority of cases functions quite cooperatively, but shows avoidance behaviour in other cases. In these latter cases, the employee starts trying to reject the task if he believes that his agenda already was full-booked for the short term, and he believes that capable colleagues are available with not full-booked agendas. Further observation by the manager reveals the pattern that the employee shows avoidance behaviour, in particular, in cases that a task is only asked shortly before its deadline, without the possibility to anticipate on the possibility of having the task allocated. The manager deliberates about this as follows:

'If I know beforehand the possibility that a last-minute task will occur, I can tell him the possibility in advance, and in addition point out that I need his unique expertise for the task, in order to avoid the behaviour that he tries to avoid the task when it actually comes up.'

Below, this example is formalised, using the BDI-model as introduced in Section 2. First, only the behaviour of the employee is addressed (in which no Theory of Mind is involved); in Section 4.2, the deliberation process of the manager is addressed as well. To this end, the example is made more precise as follows:

The *desire* to avoid a task is created after time t by the employee if the following holds at time t :

- the employee has the belief that a task is requested that has to be finished soon
- the employee has the belief that he did not hear of the possibility that the task may come at any earlier time point

The *intention* to avoid a task is generated after time t if the following holds at time t :

- the desire to avoid the task is available
- the belief that capable colleagues are available (not full booked)

The *action* to avoid the task is generated after time t if the following holds at time t :

- the intention to avoid the task is available
- the belief that the employee's own agenda is full

Using the generic template discussed at the end of Section 2, via the relations

has_reason_for(A, lower_workload, capable_colleagues_available, avoid_task)
is_opportunity_for(A, own_agenda_full, avoid_task)

the following model for agent A is obtained, as shown in Box 1.

$\text{belief}(A, \text{task_may_come}) \wedge \text{belief}(\text{last_minute_request}) \rightarrow \text{desire}(A, \text{lower_workload})$
$\text{desire}(A, \text{lower_workload}) \wedge \text{belief}(A, \text{capable_colleagues_available}) \rightarrow \text{intention}(A, \text{avoid_task})$
$\text{intention}(A, \text{avoid_task}) \wedge \text{belief}(A, \text{own_agenda_full}) \rightarrow \text{performs}(A, \text{avoid_task})$

Box 1. Simulation model for agent A

4.2. A Recursive BDI-Model for Reasoning About Task Avoidance

The model described in Section 3 can be used to describe how the manager agent (from the case described in Section 4.1) can reason and act in an anticipatory manner to avoid the employee's avoidance desire, intention and/or action to occur. The initial desire of B is that A does not perform the action to avoid the task:

```
desire(B, not(performs(A, avoid_task)))
```

Fulfillment of this desire can be obtained in the following three manners:

1) Avoiding A's desire to occur

This can be obtained when the employee hears in advance that possibly a last minute task may occur. This will make the second condition in A's desire generation as described in Section 4.1 fail.

2) Avoiding A's intention to occur (given that the desire occurs)

This can be obtained by refutation of the belief that plays the role of the reason to generate the intention in A's intention generation as described in Section 4.1, e.g., when the employee hears that colleagues do not have the required expertise.

3) Avoiding A's action to occur (given that the intention occurs)

This can be obtained by refutation of the belief that plays the role of opportunity in A's desire action as described in Section 4.1, e.g., when his agenda is not full-booked.

For convenience, the model does not make a selection but addresses all three options to prevent the avoidance action. This means that B generates desires for:

- A hears about the possibility of a last-minute task in advance
hears(A, task_may_come)
- A hears that no colleagues that are capable of performing the task are available
hears(A, not(capable_colleagues_available))
- A hears that his agenda is not full-booked
hears(A, not(own_agenda_full))

To fulfil these desires, intentions are to be generated by B to perform actions such as:

- B tells A about the possibility of a last-minute task in advance
performs(B, tell(A, task_will_come))
- B tells A that no colleagues that are capable of performing the task are available
performs(B, tell(A, not(capable_colleagues_available))
- B tells A that some of the (perhaps less interesting) tasks were taken from A's agenda and were re-allocated to a colleague
performs(B, tell(A, not(own_agenda_full))

Reason for B to choose for these actions is

- the belief of B that telling something will lead to the person hearing it
belief(B, adequate_communication(B, A))

Moreover, these intentions of B can lead to the corresponding actions when the following belief of B in opportunity is there:

- the belief that A is available for B to talk to
belief(B, available_for(A, B))

In addition to the generic BDI-model shown in Sections 2 and 3, the following specific relations are used to model the case study:

```
has_reason_for(B, hears(A, IE), adequate_communication, tell(A, IE))
is_opportunity_for(B, available_for(A, B), tell(A, IE))
has_effect(tell(A, IE), communicated_from_to(IE, B, A))
```

Note that the last minute request itself is an event that not necessarily comes from agent B; it can come from any agent, for example a Director agent. It is modelled as an event in LEADSTO.

4.3. Simulation Experiments

The model as described above has been used to perform a number of simulation experiments, using the LEADSTO software environment (Bosse, Jonker, Meij, and Treur, 2007). This piece of software takes a LEADSTO specification as input, and uses this to generate *traces* (i.e., sequences of states over time). In Figure 3 and 4, examples of resulting simulation traces are shown. In these figures, time is on the horizontal axis; the state properties are on the vertical axis. A box on top of a line indicates that a state property is true. Note that, to enhance readability, only a selection of the relevant atoms is shown. Figure 3 is the resulting simulation trace of the situation explained in Section 4.1 in which *no* Theory of Mind is involved, i.e., only the behaviour of employee is addressed, without social manipulation. The trace depicts that the employee initially receives some inputs (e.g., indicated by the state property

```
hears(employee, capable_colleagues_available)
```

at time point 1).

As a result, the employee has made some beliefs (e.g., the state property

```
belief(employee, capable_colleagues_available)
```

at time point 2), which persists for a longer time. Next, when the employee receives a last minute request at time point 6

```
hears(employee, last_minute_request)
```

he eventually generates desire to avoid the task at time point 8

```
desire(employee, avoid_task)
```

Based on this desire and the input received earlier

hears(employee, capable_colleagues_available)

the employee generates the intention to avoid the task at time point 9:

intention(employee, avoid_task)

Based on this intention and the input received earlier

hears(employee, own_agenda_full)

at time point 1, the employee eventually performs the action of avoiding the task at time point 10.

Figure 4 is the resulting simulation trace of the case study described in Section 4.2, in which the manager agent can reason and act in an anticipatory manner to avoid the employee's avoidance desire, intention and/or action to occur. Figure 4 shows that the manager initially desires that the employee does not perform the action to avoid the task:

desire(manager, not(performs(employee, avoid_task)))

Based on this, he eventually generates number of more detailed desires about what the employee should hear (see, for example, the state property

desire(manager, not(hears(employee, capable_colleagues_available)))

at time point 3). Next, the manager uses these desires to generate some intentions to fulfil these desires (e.g., the state property

intention(manager, tell(employee, not(capable_colleagues_available)))

at time point 4). Eventually, these intentions are performed, and the employee receives some new inputs (e.g., the state property

hears(employee, not(capable_colleagues_available))

at time point 7). As a result, when the employee receives a last minute request at time point 11

hears(employee, last_minute_request)

he does not generate the action to avoid the task.

Note that in the scenario sketched in Figure 4, the manager takes all possible actions (within the given conceptualisation) to fulfil its desires. This is a rather extreme case, since according to the employee's BDI-model, modifying only one of its input will be sufficient to make sure that (s)he does not avoid the task. Other traces can be generated in which the manager takes fewer actions to fulfil its desires.

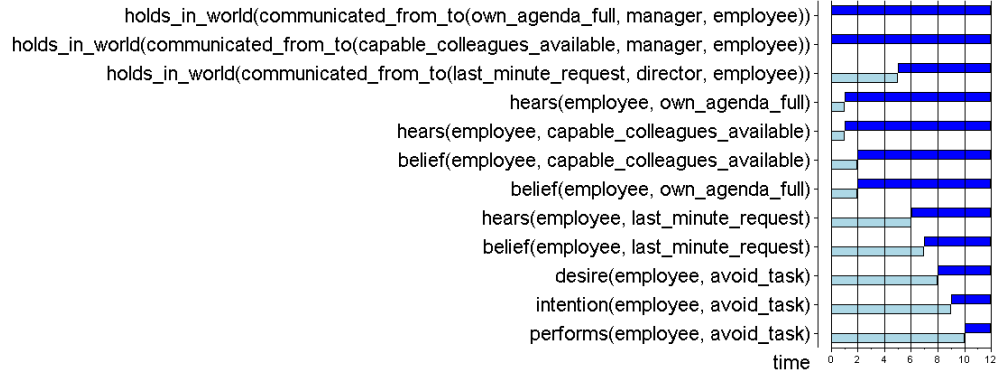


Figure 3. Simulation trace of task avoidance behaviour (without ToM)



Figure 4. Simulation trace where task avoidance behaviour is prevented by social manipulation

5. Case Study 2 - Modelling Animal Behaviour

In addition to human behaviour, the recursive BDI-model can be applied to simulate the mindreading behaviour of animals in certain circumstances, as will be shown in the current section. To this end, a specific scenario (about a predator that wants to attack a prey) is introduced in Section 5.1. Section 5.2 and 5.3 show how the scenario can be modelled formally, and Section 5.3 discusses simulation experiments that have been performed on the basis of this model.

5.1. Animal Behaviour Case

In (Bogdan, 1997), Bogdan introduces the notion of a *goal setting for interpretation* (i.e., a situation in which an organisms needs to interpret the behaviour of another organism in order to satisfy its private goals), which he illustrates as follows:

‘To illustrate, suppose that organism A (interpreter) has a private goal (say resting). It interferes with the goal of another organism S (subject), which is to eat A. Those A-type organisms will be selected who manage to form the social or S-regarding goal of avoiding the nasty type S by countering their inimical behavior, say by threat or deception. The latter goal in turn selects for interpretation, specifically, for interpretation goals such as desire identification and behavior prediction. Those A-type organisms are selected who form and reach such interpretation goals. The environment that selected for such accomplishments is a goal setting of a certain kind, say of behavior manipulation by behavior prediction and desire identification. There could be as many kinds of goal settings for interpretation as there are interpretation goals and tasks to achieve them, and hence as many skills.’ (Bogdan, 1997), p. 111.

Based on this description, a scenario is considered that involves a predator (agent A) and a prey (agent B). Assume that, under certain circumstances, the predator tries to kill the prey, and the prey tries to avoid this by manipulation. First, only the behaviour of the predator is addressed (in which no Theory of Mind is involved). However, in the next section, the cognitive process of the prey involving Theory of Mind is addressed as well. Using the BDI-model as introduced in Section 2, the example is made more precise as follows. The *desire* to eat the prey is created after time t by the predator if the following holds at time t:

- the predator has the belief that the prey is alone
(i.e., not surrounded by other animals)

The *intention* to kill the prey is generated after time t if the following holds at time t:

- the predator has the desire to eat the prey
- the predator has the belief that the prey is weak
(i.e., that it does not show strong, aggressive behaviour)

The *action* to kill the prey is generated after time t if the following holds at time t:

- the predator has the intention to kill the prey
- the predator has the belief that the prey is slow
(i.e., that it does not run very fast, so that it can be caught)

Using the generic template discussed, via the relations

```
has_reason_for(predator, eat_prej, not(prej_shows_aggressive_behaviour), kill_prej)
is_opportunity_for(predator, not(prej_runs_fast), kill_prej)
```

the following model for agent predator is obtained, as shown in Box 2.:

```
belief(predator, not(prej_surrounded_by_other_animals)) → desire(predator, eat_prej)

desire(predator, eat_prej) ∧ belief(predator, not(prej_shows_aggressive_behaviour)) →
  intention(predator, kill_prej)

intention(predator, kill_prej) ∧ belief(predator, not(prej_runs_fast)) →
  performs(predator, kill_prej)
```

Box 2. Simulation model for agent predator

5.2. A Recursive BDI-Model for Animal Behaviour

The model described in Section 3, can be used to describe how the prey agent (from the case described above) acts in an anticipatory manner to avoid the predator's desire, intention and/or action to occur. In (Bosse, Memon, and Treur, 2007c) a precursor model based on 'depends on' relations was used to analyse mindreading behaviour of animals. The initial desire of the prey is that the predator does not perform the action to kill it:

`desire(pre, not(performs(predator, kill(pre))))`

Fulfilment of this desire can be obtained in the following three manners:

1) Avoiding the predator's desire to occur

This can be obtained when the predator observes that the prey is surrounded by other animals. This will make the condition in the predator's desire generation as described earlier fail.

2) Avoiding the predator's intention to occur (given that the desire occurs)

This can be obtained by refutation of the belief that plays the role of the reason to generate the intention in the predator's intention generation as described earlier, i.e., the belief that the prey is weak (and does not show aggressive behaviour).

3) Avoiding the predator's action to occur (given that the intention occurs)

This can be obtained by refutation of the belief that plays the role of opportunity in the predator's desire action as described, i.e., the belief that the prey is slow (and does not run fast).

For convenience, the model does not make a selection but addresses all three options to prevent the killing action. This means that the prey generates *desires* for:

- The predator observes that the prey is surrounded by other animals
`observes(predator, prey_surrounded_by_other_animals)`
- The predator observes that the prey shows aggressive behaviour
`observes(predator, prey_shows_aggressive_behaviour)`
- The predator observes that the prey runs fast
`observes(predator, prey_runs_fast)`

To fulfil these desires, intentions are to be generated by the prey to actions such as:

- call for help of other animals: `call_for_help`
- show aggressive behaviour: `show_aggressive_behaviour`
- run fast: `run_fast`

Reasons for the prey to choose for these intentions are beliefs in, respectively:

- The predator is paying attention to the prey's gaze (so that it will notice it when the prey calls for help of other animals)
`predator_is_noticing_preys_gaze`
- The predator is paying attention to the prey's gesture (so that it will notice it when the prey shows aggressive behaviour)
`predator_is_noticing_preys_gesture`

- The predator is at a reasonable distance away (so that it is able to run away without being caught)
predator_is_reasonable_distance_away

Moreover, the intentions of the prey can lead to the corresponding actions when the following beliefs of the prey in opportunities are there:

- Other animals are around (so that it is possible to call for their help)
other_animals_around
- The predator is about to attack (so that it is possible to show aggressive behaviour)
predator_about_to_attack
- No obstacle is blocking the escape route of the prey (so that it is possible to run away)
no_obstacle

In addition to the generic BDI-model shown before, the following specific relations were used to model the case study, as shown in Box 3.

```

belief(pre(dependes_on(perform(predator, kill(pre)), intention(predator, kill(pre))))
belief(pre(dependes_on(perform(predator, kill(pre)), not(belief(predator, prey_runs_fast))))
belief(pre(dependes_on(intention(predator, kill(pre)), desire(predator, eat(pre))))
belief(pre(dependes_on(intention(predator, kill(pre)), not(belief(predator,
    prey_shows_aggressive_behaviour))))
belief(pre(dependes_on(desire(predator, eat(pre)), not(belief(predator,
    prey_surrounded_by_other_animals))))
belief(pre(dependes_on(belief(predator, prey_surrounded_by_other_animals),
    observes(predator, prey_surrounded_by_other_animals)))
belief(pre(dependes_on(belief(predator, prey_shows_aggressive_behaviour),
    observes(predator, prey_shows_aggressive_behaviour)))
belief(pre(dependes_on(belief(predator, prey_runs_fast), observes(predator,
    prey_runs_fast)))

has_reason_for(pre, observes(predator, prey_surrounded_by_other_animals),
    predator_is_noticing_preys_gaze, call_for_help)
has_reason_for(pre, observes(predator, prey_shows_aggressive_behaviour),
    predator_is_noticing_preys_gesture, show_aggressive_behaviour)
has_reason_for(pre, observes(predator, prey_runs_fast),
    predator_is_reasonable_distance_away, run_fast)

is_opportunity_for(pre, other_animals_around, call_for_help)
is_opportunity_for(pre, predator_about_to_attack,
    show_aggressive_behaviour)
is_opportunity_for(pre, no_obstacle, run_fast)

has_effect(call_for_help, prey_surrounded_by_other_animals)
has_effect(show_aggressive_behaviour, prey_shows_aggressive_behaviour)
has_effect(run_fast, prey_runs_fast)

```

Box 3. Domain specific relations for the animal behaviour case study

5.3. Simulation Model

By combining the relations in Section 5.2, with the generic LEADSTO rules provided in Section 3, a complete executable LEADSTO specification for the recursive BDI-model has been created, see Box 4 and 5.

```

desire(pre, not(perform(predator, kill(pre)))) ^
belief(pre, depends_on(perform(predator, kill(pre)), intention(predator, kill(pre)))) →
  desire(pre, not(intention(predator, kill(pre))))

desire(pre, not(perform(predator, kill(pre)))) ^
belief(pre, depends_on(perform(predator, kill(pre)), not(belief(predator, pre_runs_fast))))
→ desire(pre, belief(predator, pre_runs_fast))

desire(pre, not(intention(predator, kill(pre)))) ^
belief(pre, depends_on(intention(predator, kill(pre)), desire(predator, eat(pre)))) →
  desire(pre, not(desire(predator, eat(pre))))

desire(pre, not(intention(predator, kill(pre)))) ^
belief(pre, depends_on(intention(predator, kill(pre)),
not(belief(predator, pre_shows_aggressive_behaviour)))) →
  desire(pre, belief(predator, pre_shows_aggressive_behaviour))

desire(pre, not(desire(predator, eat(pre)))) ^
belief(pre, depends_on(desire(predator, eat(pre)),
not(belief(predator, pre_surrounded_by_other_animals)))) →
  desire(pre, belief(predator, pre_surrounded_by_other_animals))

desire(pre, belief(predator, pre_runs_fast)) ^
belief(pre, depends_on(belief(predator, pre_runs_fast),
observes(predator, pre_runs_fast))) →
  desire(pre, observes(predator, pre_runs_fast))

desire(pre, belief(predator, pre_shows_aggressive_behaviour)) ^
belief(pre, depends_on(belief(predator, pre_shows_aggressive_behaviour),
observes(predator, pre_shows_aggressive_behaviour))) →
  desire(pre, observes(predator, pre_shows_aggressive_behaviour))

desire(pre, belief(predator, pre_surrounded_by_other_animals)) ^
belief(pre, depends_on(belief(predator, pre_surrounded_by_other_animals),
observes(predator, pre_surrounded_by_other_animals))) →
  desire(pre, observes(predator, pre_surrounded_by_other_animals))

desire(pre, observes(predator(pre_surrounded_by_other_animals))) ^
belief(pre, predator_is_noticing_preys_gaze) → intention(pre, call_for_help)

desire(pre, observes(predator(pre_shows_aggressive_behaviour))) ^
belief(pre, predator_is_noticing_preys_gesture) →
  intention(pre, show_aggressive_behaviour)

desire(pre, observes(predator(pre_runs_fast))) ^
belief(pre, predator_is_reasonable_distance_away) → intention(pre, run_fast)

```

Box 4. Simulation model for the animal behaviour case study

```

intention(pre, call_for_help) ^ belief(pre, other_animals_around) →
    performs(pre, call_for_help)

intention(pre, show_aggressive_behaviour) ^ belief(pre, predator_about_to_attack) →
    performs(pre, show_aggressive_behaviour)

intention(pre, run_fast) ^ belief(pre, no_obstacle) → performs(pre, run_fast)

performs(pre, call_for_help) ^
    has_effect(call_for_help, prey_surrounded_by_other_animals) →
        holds_in_world(pre, prey_surrounded_by_other_animals)

performs(pre, show_aggressive_behaviour) ^
    has_effect(show_aggressive_behaviour, prey_shows_aggressive_behaviour) →
        holds_in_world(pre, prey_shows_aggressive_behaviour)

performs(pre, run_fast) ^ has_effect(run_fast, prey_runs_fast) →
    holds_in_world(pre, prey_runs_fast)

holds_in_world(pre, prey_surrounded_by_other_animals) →
    observes(predator, prey_surrounded_by_other_animals)

holds_in_world(pre, prey_shows_aggressive_behaviour) →
    observes(predator, prey_shows_aggressive_behaviour)

holds_in_world(pre, prey_runs_fast) → observes(predator, prey_runs_fast)

```

Box 5. Simulation model for the animal behaviour case study (continued)

5.4. Simulation Experiments

Also for this case study, the LEADSTO software environment (Bosse, Jonker, Meij, and Treur, 2007) has been used to perform a number of simulation experiments. In Figure 5 and 6, examples of resulting simulation traces are shown. In these figures, time is on the horizontal axis; the state properties are on the vertical axis. A box on top of a line indicates that a state property is true. Note that, for presentation purposes, only a selection of the relevant atoms is shown.

Figure 5 is the resulting simulation trace of the situation in which *no* Theory of Mind is involved, i.e., only the behaviour of the predator is addressed, without manipulation by the prey. The trace depicts that the predator initially receives some inputs (e.g., indicated by the state property

```
observes(predator, not(pre, prey_surrounded_by_other_animals))
```

at time point 1.

As a result, the predator has made some beliefs (e.g., the state property

```
belief(predator, not(pre, prey_surrounded_by_other_animals))
```

at time point 2), which persists for a longer time. Due to this belief, it generates the desire to eat the prey at time point 3

```
desire(predator, eat(pre))
```

Based on this desire and the belief

belief(predator, not(pre_y_shows_aggressive_behaviour))

the predator generates the intention to kill the prey at time point 4:

intention(predator, kill(pre_y))

Based on this intention and the belief

belief(predator, not(pre_y_runs_fast))

the predator eventually performs the action of killing the prey at time point 5.

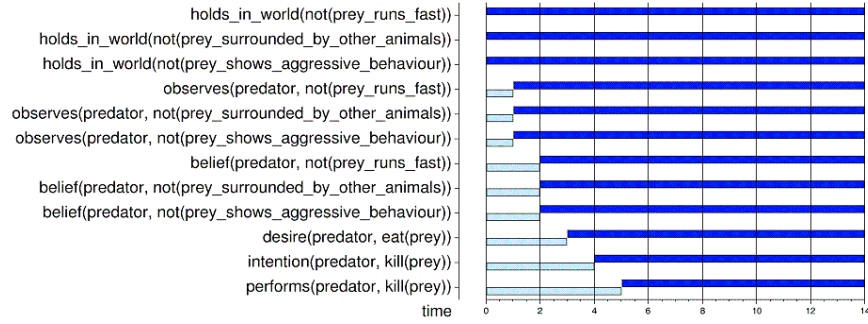


Figure 6 is the resulting simulation trace of the extended case study, in which the prey agent can act in an anticipatory manner to avoid the predator's desire to eat the prey, and intention and/or action to kill it. Figure 6 shows, among others, that the prey initially desires that the predator does not perform the action to kill it:

desire(pre_y, not(performs(predator, kill(pre_y))))

Based on this, the prey eventually generates a number of more detailed desires about what the predator should observe (see, for example, the state property

desire(pre_y, observes(predator, pre_y_shows_aggressive_behaviour))

at time point 3). Next, the prey uses these desires to generate some intentions to fulfill these desires (e.g., the state property

intention(pre_y, show_aggressive_behaviour)

at time point 4). Eventually, when the opportunities are there, these intentions are performed, and the predator observes some new inputs (e.g., the state property

observes(predator, pre_y_shows_aggressive_behaviour)

at time point 8). As a result, the predator eventually does not generate the action to kill the prey.

Note that in the scenario sketched in Figure 6, the prey takes all possible actions (within the given conceptualisation) to fulfill its desires.

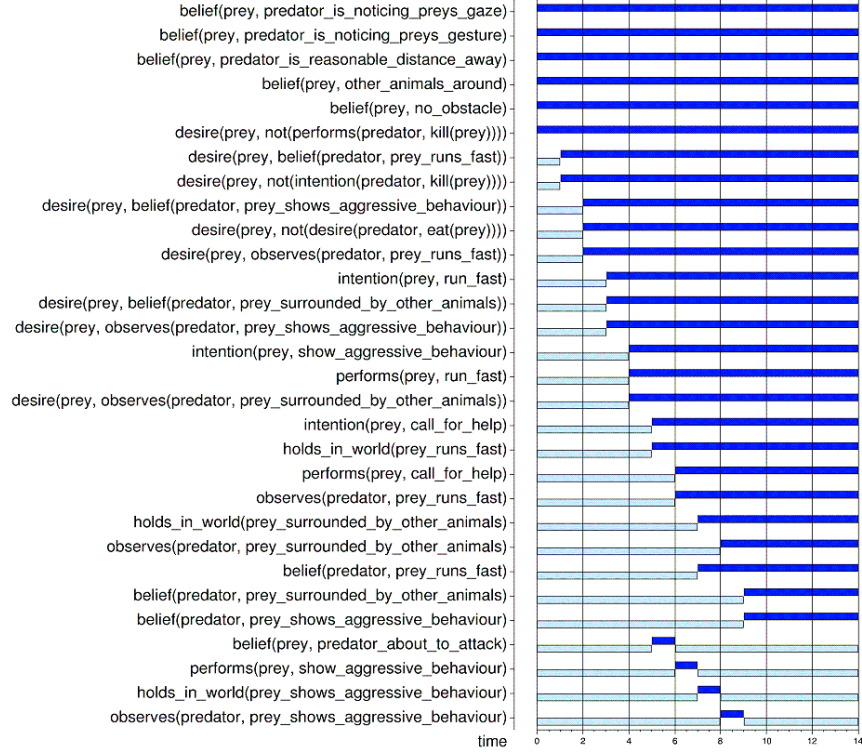


Figure 6: Simulation trace of the prey's manipulation of the predator's behaviour.

6. Case Study 3 - Modelling Virtual Characters

Virtual storytelling addresses automated generation of stories with characters showing more or less human-like behaviour. Traditionally the approaches to develop virtual stories involve stories with a fixed, pre-scripted storyline constraining the characters in their autonomy and behaviour, but recently more interest is shown in emergent narrative, i.e., storylines generated by interaction between a number of characters with certain personalities that (inter)act as autonomous agents.

When a story is generated by autonomous characters, then the characters should be able to behave in more human-like manners to get realistic emergent storyline, and need more complex personalities with human-like properties such as emotions and theories of mind. To accomplish this, more sophisticated computational cognitive models are to be incorporated within such virtual characters.

We will explore the possibilities to equip these characters involved in virtual stories in particular with mindreading capabilities, using the single level BDI-model presented in Section 3. By offering virtual agents such capabilities, they will be able to select behaviours that are useful in social functioning in a more human-like manner, with more possibilities of choice, thus enhancing the emergent narrative effect.

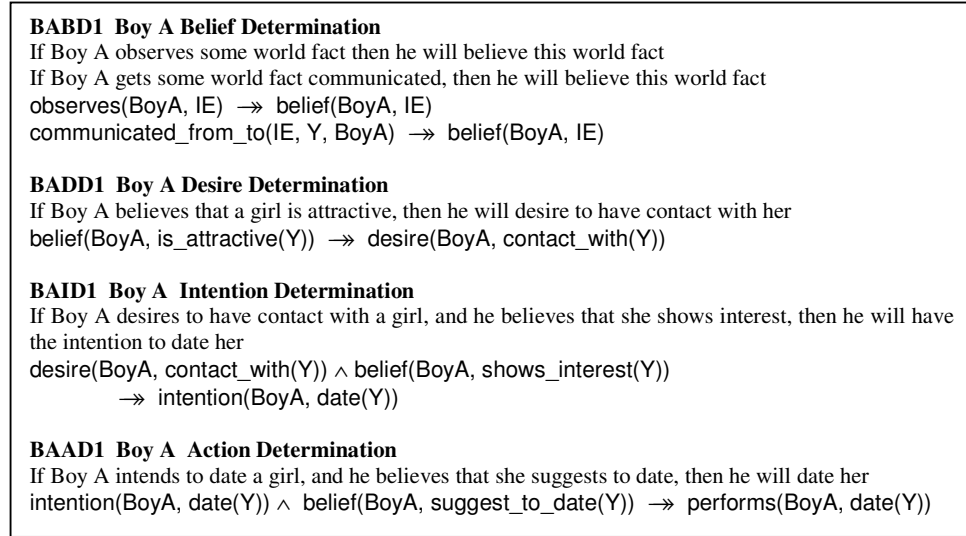
Below, Section 6.1 presents models for virtual characters based on the single level BDI-model described in Section 2. Based on this model, Section 6.2 introduces a model for a mindreading character, using the multi-level BDI-model described in Section 3. Section 6.3 discusses simulation experiments that have been performed on the basis of this model, and how these can be used to create

virtual stories. Finally, in Section 6.4 it is shown how such virtual stories can be analysed using automated tools.

6.1. BDI-Model for Virtual Characters

In this section, we will apply the model described in Section 3, to obtain behavioural descriptions of characters acting in an emergent soap story. The example soap story addressed concentrates on four characters: Boy A, Boy B, Girl A, Girl B. As Boy B is attracted to Girl A, who however is dating Boy A, he exploits mindreading capabilities to come to more favourable circumstances, involving Girl B as well.

The BDI-model was used to model autonomous characters. The first character shown as an illustration is Boy A. This is a character that tries to date any attractive girl, as defined by the following BDI-model, see Box 6.



Box 6. BDI-model for Boy A

Using the visualisation template provided in Figure 1, this model can be depicted as shown in Figure 7.

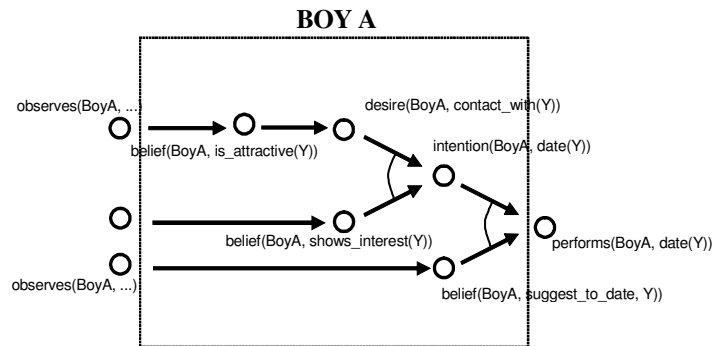


Figure 7. BDI-model for Boy A

Note that this BDI-model defines an *internal* view on the agent's cognitive processes. In addition, an *external* behaviour view generated by these processes can be defined, by formulating a more complex expression in terms of inputs and outputs only. For the behaviour of Boy A, this external view is defined by:

BA1 Boy A Behaviour External View

If Boy A observes an attractive girl that shows interest in him and suggests to date, then he will date her.
 $\text{observes}(\text{BoyA}, \text{is_attractive}(\text{Y}) \wedge \text{shows_interest}(\text{Y})) \wedge$
 $\text{communicated_from_to}(\text{suggest_to_date}, \text{Y}, \text{BoyA}) \rightarrow \text{performs}(\text{BoyA}, \text{date}(\text{Y}))$

Often characters for virtual stories or plays are defined by an external behavioural view only. However, since an internal view provides the possibility in the story to provide (in addition to the character's behaviour) more insight in the internal world of the character (e.g., the motivations behind the behaviour), also (BDI-based) internal models for the characters have been created.

The second character, girl A, is looking for a partner of a good family, who does not date other girls and is rich (a good job), as expressed in the following BDI-model, see Box 7. For simplicity, formalisations are sometimes left.

GABD1 Girl A Belief Determination

If Girl A observes some world fact then she will believe this world fact
 If Girl A gets some world fact communicated, then she will believe this world fact

GADD1 Girl A Desire Determination

If Girl A believes that a certain boy has a good family and does not date other girls, then she will desire to be with him

GAID1 Girl A Intention Determination

If Girl A desires to be with a boy, and he is rich, then she will have the intention to date him

GAAD1 Girl A Action Determination

If Girl A intends to date a boy, and he proposes to date, then she will date him

Box 7. BDI-model for Girl A

The external behaviour view generated by this model is:

GA1 Girl A Behaviour External View

If Girl A observes a boy that is from a good family, rich, does not date other girls, and he suggests to date, then she will date him.
 $\text{observes}(\text{GirlA}, \text{good_family}(\text{X}) \wedge \text{is_rich}(\text{X})) \wedge \text{not_dating}(\text{X}) \wedge$
 $\text{communicated_from_to}(\text{suggest_to_date}, \text{X}, \text{GirlA})$
 $\rightarrow \text{performs}(\text{GirlA}, \text{date}(\text{X}))$

The third character discussed is a type of poor girl who has no good family: Girl B, defined in Box 8.

GBBD1 Girl B Belief Determination

If Girl B observes some world fact then she will believe this world fact
 If Girl B gets some world fact communicated, then she will believe this world fact

GBDD1 Girl B Desire Determination

If Girl B believes that she has no good family and no good job, then she will desire to get money.

GBID1 Girl B Intention Determination

If Girl B desires to get money, and she believes somebody offers her money to date a boy, then she will have the intention to show interest in that boy and suggest to date

GBAD1 Girl B Action Determination

If Girl B intends to date a boy, and he is around, then she will show interest in him and suggest him to date

Box 8. BDI-model for Girl B

The external behaviour view generated by this model is:

GB1 Girl B Behaviour External View

If Girl B is offered money to date any boy, then she will show interest in this boy and suggest him to date.
observes(no_good_famliy \wedge no_good_job \wedge BoyA_is_around) \wedge
communicated_from_to(money_offer_for_dating(X2), X1, GirlB)
→ performs(GirlB, show_interest_in(X2)) \wedge communication_from_to(suggest_to_date, GirlB, X2)

The fourth character used in the example story presented, Boy B, also has been specified by a BDI-model. However, this BDI-model is more complex, as it uses mindreading capabilities in the desire determination process, and is discussed in the next section.

6.2. Model of a Mindreading Character

To represent a Theory of Mind on agent A within an agent B, a nested expression is used, e.g.:

belief(B, leads_to(con(intention(A, AC) , belief(A, IE)), performs(A, AC)))

This expresses (where con indicates a conjunction, see Table 1):

Agent B believes that, when A has the intention to perform action X and the belief that B2 holds, then A will perform action X.

In this manner, temporal relations depicted in a graph as in Figure 1 can be represented. Boy B's Theory of Mind for Girl A, resp. Girl B, Boy A is expressed in Box 9:

BBToMGA Boy B Theory of Mind for Girl A

belief(BoyB, leads_to(observes(GirlA, IE), belief(GirlA, IE)))
belief(BoyB, leads_to(communicated_from_to(IE, Y, GirlA), belief(GirlA, IE)))
belief(BoyB, leads_to(con(belief(GirlA, good_familiy(Y)),
belief(GirlA, not_dating(Y))), desire(GirlA, be_with(Y))))
belief(BoyB, leads_to(con(desire(GirlA, be_with(Y)),
belief(GirlA, rich(Y))), intention(GirlA, date(Y))))
belief(BoyB, leads_to(con(intention(GirlA, date(Y)),
belief(GirlA, suggests_to_date(Y))), performs(GirlA, date(Y))))

BBToMGB Boy B Theory of Mind for Girl B

belief(BoyB, leads_to(observes(GirlB, IE), belief(GirlB, IE)))
belief(BoyB, leads_to(communicated_from_to(IE, Y, GirlB), belief(GirlB, IE)))
belief(BoyB, leads_to(con(belief(GirlB, poor_family,
belief(GirlB, no_good_job))), desire(GirlB, get_money)))
belief(BoyB, leads_to(con(desire(GirlB, get_money),
belief(GirlB, money_offered_for_dating(Y))),
intention(GirlB, show_interest_in_and_suggest_to_date(Y))))
belief(BoyB, leads_to(con(intention(GirlB, show_interest_in_and_suggest_to_date(Y)),
belief(GirlB, is_around(Y))), con(performs(GirlB, show_interest_in(Y)),
communication_from_to(suggest_to_date, GirlB, Y))))

BBToMBA Boy B Theory of Mind for Boy A

belief(BoyB, leads_to(observes(BoyA, IE), belief(BoyA, IE)))
belief(BoyB, leads_to(communicated_from_to(IE, Y, BoyA), belief(BoyA, IE)))
belief(BoyB, leads_to(belief(BoyA, is_attractive(Y)),
desire(BoyA, contact_with(Y))))
belief(BoyB, leads_to(con(desire(BoyA, contact_with(Y)),
belief(BoyA, shows_interest(Y))), intention(BoyA, date(Y))))
belief(BoyB, leads_to(con(intention(BoyA, date(Y)),
belief(BoyA, suggests_to_date(Y))), performs(BoyA, date(Y))))

Box 9. Two-level BDI-model for Boy B

As can be seen from the specifications above, in this case Boy B has a complete and correct theory about the mind of the other characters (i.e., the relations exactly match the models provided in Section 6.1). Note that this is not necessarily the case; interesting stories can be generated in cases where a character thinks to know other characters very well, but turns out to be wrong. In such situations, it may be useful for the character to update its Theory of Mind.

6.2.1. Generating Desires

When an agent B has a Theory of Mind of another character, it can take this into account in generating its own desires. For example, if agent B desires some action of agent A to take place, as a consequence it can generate a desire that agent A has the intention to perform this action. From the latter desire agent B can generate the desire that agent A has a desire that is fulfilled by the action, and so on. The following generic specifications enable such desire generation processes in agent B's BDI-model (here IE1 and IE2 may be negations of other statements):

BBToMX

If agent B believes that IE1 leads to IE2 and desires IE2, then it will desire IE1.
 If agent B believes that IE1 leads to IE2 and it desires not IE2, then it will desire not IE1.
 If agent B desires the conjunction of IE1 and IE2 and it believes that IE1 and IE2 are not fixed, then it will desire IE1 and it will desire IE2.
 If agent B desires the conjunction of IE1 and IE2 and it believes that IE1 is fixed, then it will desire IE2.
 If agent B desires the conjunction of IE1 and IE2 and it believes that IE2 is fixed, then it will desire IE1.
 If agent B believes that IE1 is fixed and that IE1 leads to IE2, then it will believe that IE2 is fixed.
 $\text{desire}(B, IE2) \wedge \text{belief}(B, \text{leads_to}(IE1, IE2, D)) \rightarrow \text{desire}(B, IE1)$
 $\text{desire}(B, \text{not}(IE2)) \wedge \text{belief}(B, \text{leads_to}(IE1, IE2, D)) \rightarrow \text{desire}(B, \text{not}(IE1))$
 $\text{desire}(B, \text{con}(IE1, IE2)) \wedge \text{not belief}(B, \text{fixed_true}(IE1)) \wedge$
 $\text{not belief}(B, \text{fixed_true}(IE2)) \rightarrow \text{desire}(B, IE1 \wedge \text{desire}(B, IE2))$
 $\text{desire}(B, \text{con}(IE1, IE2)) \wedge \text{belief}(B, \text{fixed_true}(IE1)) \rightarrow \text{desire}(B, IE2)$
 $\text{desire}(B, \text{con}(IE1, IE2)) \wedge \text{belief}(B, \text{fixed_true}(IE2)) \rightarrow \text{desire}(B, IE1)$
 $\text{belief}(B, \text{fixed_true}(IE1)) \wedge \text{belief}(B, \text{leads_to}(IE1, IE2, D)) \rightarrow \text{belief}(B, \text{fixed_true}(IE2))$

Here a belief on fixed_true(IE) expresses that the agent considers the indicated state property IE true and unchangeable.

6.2.2. Performing Actions

Within the model of agent B, the mindreading is used to generate specific desires. For intention and action generation based on these desires, instantiations of the schemes shown in Section 3 are included as well. For the example character Boy B:

BBID1 Boy B Intention Determination

If Boy B desires that girl B gets money for dating Boy A, and he believes he has money, then he will have the intention to offer Girl B money to date Boy A.

BBAD1 Boy B Action Determination

If Boy B intends to offer Girl B money to date Boy A, and he believes she is around, then he will offer her money to date Boy A.

Moreover, as before, Boy B has specifications for beliefs.

BBBD1 Boy B Belief Determination

If Boy B observes some world fact then he will believe this world fact
 If Boy B gets some world fact communicated, then he will believe this world fact

All of the specifications presented above describe a rather complex internal view on a mindreading agent. In contrast, the external view on character Boy B's behaviour, implied by the internal view, is quite simple: he just has to offer Girl B money to date Boy A.

BB1 Boy B Behaviour External View

At some point in time Boy B will offer money to Girl B to date Boy A.

`observes(has_money \wedge GirlB_is_around) \rightarrow`

`communication_from_to(money_offer_for_dating(BoyA), BoyB, GirlB)`

One might be tempted to just ignore the complex internal specification, and define the character Boy B by the simple external view specification instead. For the events in the story this will make no difference. However, as also indicated earlier, leaving out internal models of characters would provide a very inflexible, brittle solution, in which the actions performed by the characters are not explainable by referring to their inner world and motivations.

6.3. Simulation Results

The BDI-models of the four autonomous characters described in the previous sections have been used to generate emergent virtual storylines. To this end, first the models have been used as input for the LEADSTO simulation software environment (Bosse, Jonker, Meij, and Treur, 2007) in order to generate simulation traces. An example of (a part of) such a trace is shown in Figure 8. Note that the generic names Boy A, Girl A, Boy B and Girl B are replaced by story names Al, Ann, Bob and Bo, respectively. Figure 8 shows a story of Bob, who desires to date with Ann. However, since Ann is already dating with Al, Bob invents a plan (based on reasoning about the minds of the different characters involved) to create more favourable circumstances: he offers money to another girl, named Bo, and asks her to seduce Al, in order to let Ann lose interest in Al.

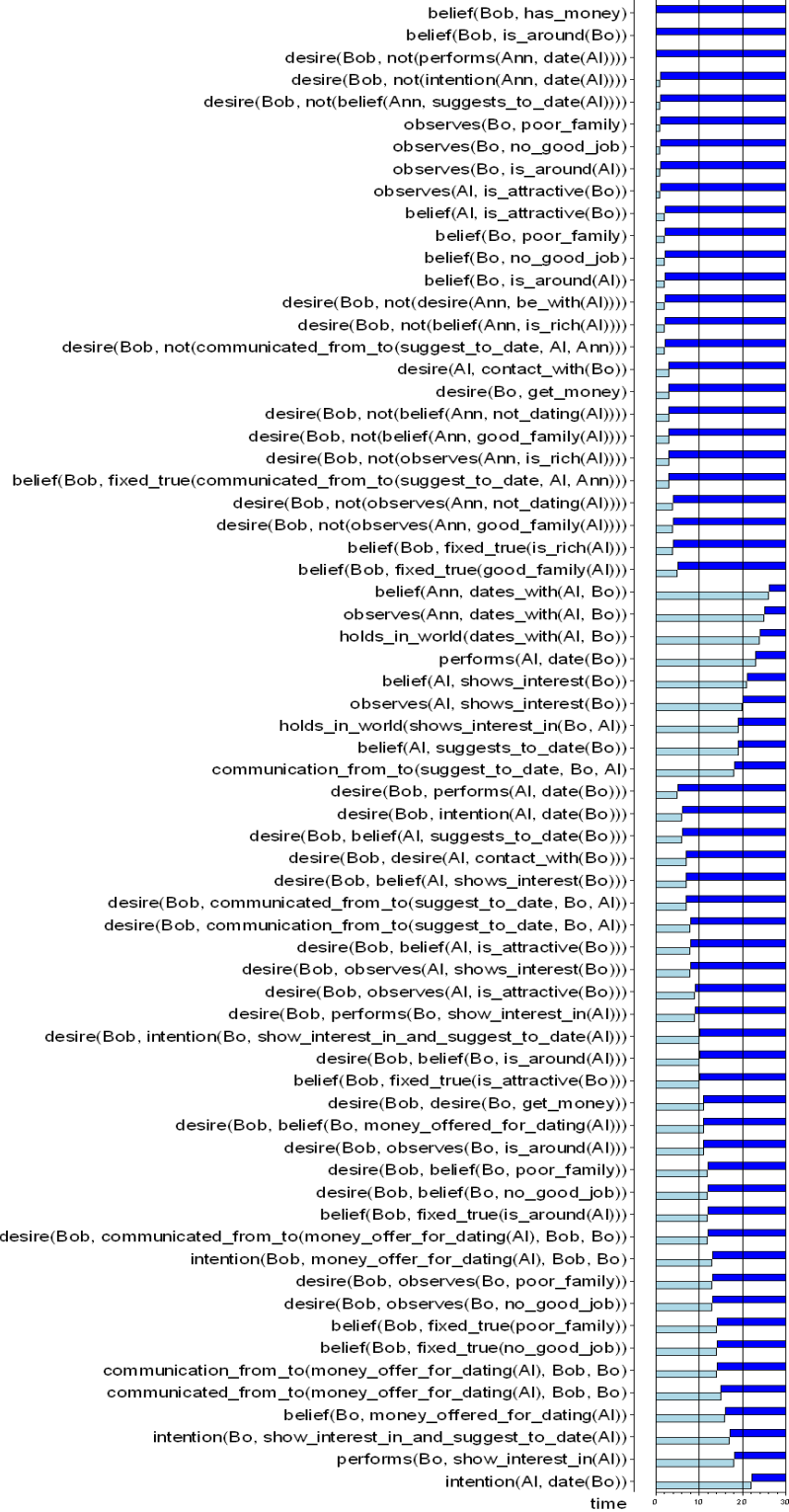


Figure 8. Example simulation trace

After generating such simulation traces, for each of the atomic state properties that occurs in the model, a mapping has been created to a text fragment. For example, the state property

belief(A, shows_interest(B)),

corresponds to the text fragment

“A believes that B shows interest in him”.

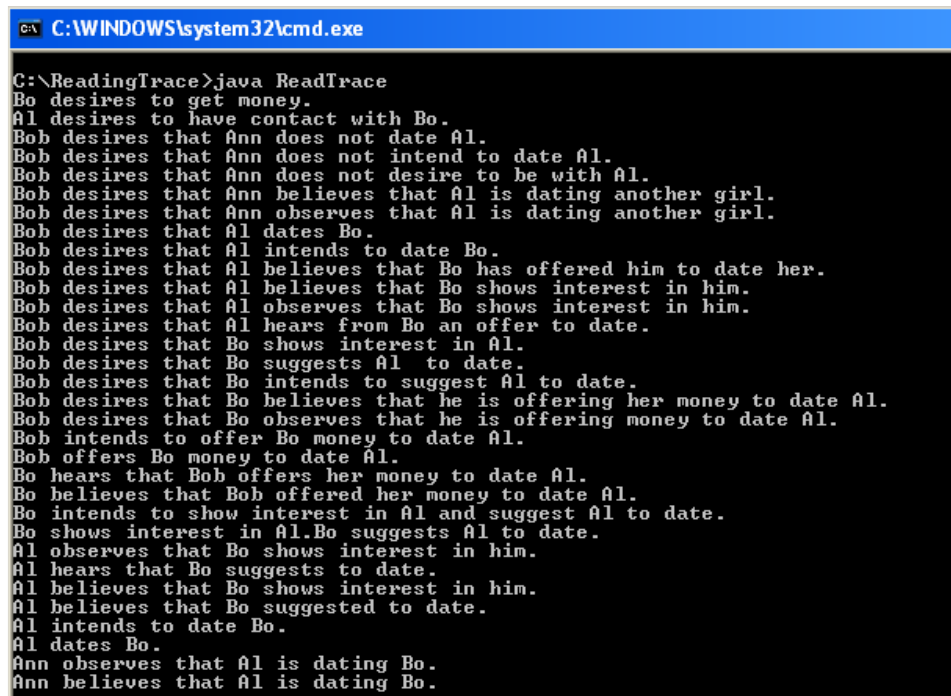
Similarly, the state property

`desire(A, contact_with(B))`

corresponds to

“A desires to have contact with B”.

Using these mappings and a specific conversion program that has been written, the LEADSTO simulation traces are automatically translated into virtual storylines in textual format. An example of fragments of such a generated storyline is shown in Figure 9.



```
C:\WINDOWS\system32\cmd.exe
C:\>ReadingTrace>java ReadTrace
Bo desires to get money.
A1 desires to have contact with Bo.
Bob desires that Ann does not date A1.
Bob desires that Ann does not intend to date A1.
Bob desires that Ann does not desire to be with A1.
Bob desires that Ann believes that A1 is dating another girl.
Bob desires that Ann observes that A1 is dating another girl.
Bob desires that A1 dates Bo.
Bob desires that A1 intends to date Bo.
Bob desires that A1 believes that Bo has offered him to date her.
Bob desires that A1 believes that Bo shows interest in him.
Bob desires that A1 observes that Bo shows interest in him.
Bob desires that A1 hears from Bo an offer to date.
Bob desires that Bo shows interest in A1.
Bob desires that Bo suggests A1 to date.
Bob desires that Bo intends to suggest A1 to date.
Bob desires that Bo believes that he is offering her money to date A1.
Bob desires that Bo observes that he is offering money to date A1.
Bob intends to offer Bo money to date A1.
Bob offers Bo money to date A1.
Bo hears that Bob offers her money to date A1.
Bo believes that Bob offered her money to date A1.
Bo intends to show interest in A1 and suggest A1 to date.
Bo shows interest in A1. Bo suggests A1 to date.
A1 observes that Bo shows interest in him.
A1 hears that Bo suggests to date.
A1 believes that Bo shows interest in him.
A1 believes that Bo suggested to date.
A1 intends to date Bo.
A1 dates Bo.
Ann observes that A1 is dating Bo.
Ann believes that A1 is dating Bo.
```

Figure 9. Fragments of a Generated Storyline

6.4. Formal Analysis of Dynamics Properties

For the above model, it is easy to produce various simulations based on different settings, initial conditions and external events offered. Moreover, it is possible to incorporate nondeterministic behaviours by temporal rules that involve probabilistic effects, cf. (Bosse, Jonker, Meij, and Treur, 2007). Thus large sets of traces can be generated. When such a set is given, it is more convenient to check them on interesting (emergent) properties automatically, than going through them by hand. Furthermore, it may also be useful when insight is provided how dynamic properties of the multiagent system as a whole depend on dynamic properties of the agents within the system, and further on, how these relate to properties of specific components within the agents. This section shows how this can be achieved.

In order to analyse whether the resulting storylines satisfy interesting (expected or unexpected) properties, a number of *dynamic properties* have been specified for different aggregation levels of the multi-agent system behind the emergent story, cf. (Jonker and Treur, 2002). The main property considered for the story as a whole is: will at the end Girl A date Boy B? This property is formalised as Global Property GP1:

GP1 At some point in time Girl A will date Boy B.
 $\text{true} \rightarrow \text{performs}(\text{GirlA}, \text{date}(\text{BoyB}))$

Whether or not this property is fulfilled depends on properties of the agents' behaviours. Furthermore, these properties of the agents' behaviours depend on their internal components, in this case components for belief, desire, intention, and action determination. In Figure 10, it is shown how the property GP1 at the highest level relates to properties of the agents, and how properties of the agents relate to properties of their components. In this picture, a connection between a property and a set of properties at a lower level indicates that the lower level properties together (logically) entail the higher level property. The properties are described in more detail below.

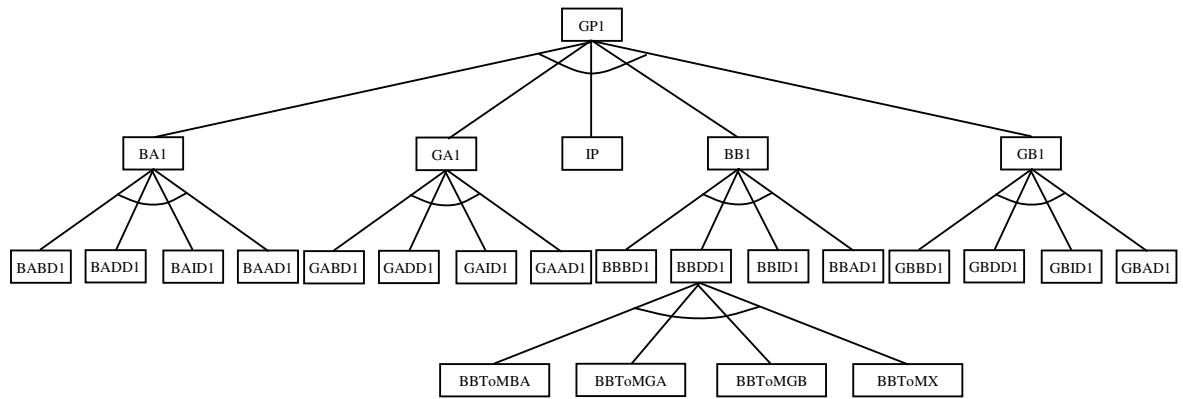


Figure 10. Interlevel relations between dynamic properties in the virtual story

For example, the property GP1 of the system as a whole can be logically related to properties of the agents by the following *interlevel relation*:

$$\text{BA1} \ \& \ \text{GA1} \ \& \ \text{BB1} \ \& \ \text{GB1} \ \& \ \text{IP} \Rightarrow \text{GP1}$$

Here IP stands for simple interaction properties expressing that generated output will reach the input of the relevant agent. As indicated in Figure 10, each property of an agent is logically related to properties of the components within the agent. The interlevel relations for resp. Boy A, Girl A, and Girl B are:

$$\text{BABD1} \ \& \ \text{BADD1} \ \& \ \text{BAID1} \ \& \ \text{BAAD1} \Rightarrow \text{BA1}$$

$$\text{GABD1} \ \& \ \text{GADD1} \ \& \ \text{GAID1} \ \& \ \text{GAAD1} \Rightarrow \text{GA1}$$

$$\text{GBBD1} \ \& \ \text{GBDD1} \ \& \ \text{GBID1} \ \& \ \text{GBAD1} \Rightarrow \text{GB1}$$

The first interlevel relation, for example, expresses that the behavioural property for Boy A holds when belief, desire, intention, and action determination have the right properties: BABD1, BADD1, BAID1, BAAD1; see Section 6.1. For the other two cases (Girl A, Girl B) it is similar. The interlevel relation within Boy B,

$$\text{BBBD1} \& \text{BBDD1} \& \text{BBID1} \& \text{BBAD1} \Rightarrow \text{GA1}$$

involves different elements. The second property is defined as follows (for the other three, see Section 6.2).

BBDD1 Boy B Desire Determination

If Boy B desires that Girl A does not date Boy A, then he will desire that she observes that Boy A dates another girl.

If Boy B desires that Girl A observes that Boy A dates another girl, then he will desire that Girl B shows interest in him and suggests him to date.

If Boy B desires that Girl B shows interest in Boy A and suggests him to date, then he will desire that girl B is offered money for dating Boy A.

$$\begin{aligned} & \text{desire}(\text{BoyB}, \text{not}(\text{dates}(\text{GirlA}, \text{BoyA}))) \rightarrow \text{desire}(\text{BoyB}, \text{observes}(\text{GirlA}, \text{not}(\text{not_dating}(\text{BoyA}))) \\ & \text{desire}(\text{BoyB}, \text{observes}(\text{GirlA}, \text{dates}(\text{BoyA}, X))) \\ & \rightarrow \text{desire}(\text{BoyB}, \text{observes}(\text{BoyA}, \text{shows_interest}(\text{GirlB})) \wedge \\ & \quad \text{communicated_from_to}(\text{suggest_to_date}, \text{GirlB}, \text{BoyA})) \\ & \text{desire}(\text{BoyB}, \text{observes}(\text{BoyA}, \text{shows_interest}(\text{GirlB})) \wedge \\ & \quad \text{communicated_from_to}(\text{suggest_to_date}, \text{GirlB}, \text{BoyA})) \\ & \rightarrow \text{desire}(\text{BoyB}, \text{communicated_from_to}(\text{money_offer_for_dating}(\text{BoyA}), \text{BoyB}, \text{GirlB})) \end{aligned}$$

This property BBDD1 has another interlevel relation to the properties BBToMBA, BBToMGA, BBToMGB, BBToMX defined in Section 6:

$$\text{BBToMBA} \& \text{BBToMGA} \& \text{BBToMGB} \& \text{BBToMX} \Rightarrow \text{BBDD1}$$

Using a dedicated Checker Tool (Bosse et al., 2006) for checking logical expressions against traces, the dynamic properties as specified above have been automatically verified against the generated storyline shown in Section 6.3. In this case, they all turned out to hold. However, there may be certain situations in which expected properties (such as “At some point in time, Boy B will date Girl A”) do not hold. In such situations, the Checker Tool and the interlevel relationships between dynamic properties may be used for *diagnosis* of the story. For example, suppose for a given story at some point in time it has been detected (using the Checker Tool) that the dynamic property GP1 does not hold, i.e., Boy B does never date Girl A. Given the AND-tree structure in Figure 10, at least one of the children nodes of GP1 will not hold, which means that either BA1, GA1, BB1, or GB1 will not hold. Suppose by further checking it is found that GB1 does not hold. Then the diagnostic process can be continued by focusing on this property. It follows that either GBBD1, GBDD1, GBID1, or GBAD1 does not hold. Checking these three properties will pinpoint the cause of the error. Notice that this diagnostic process is economic in the sense that the whole subtree under, e.g., BB1 is not examined since there is no reason for that, as BB1 holds.

Note that, although the verification of the individual dynamic properties (i.e., the nodes in Figure 10) against traces can be done automatically by means of the Checker Tool (Bosse et al., 2006), construction of the interlevel relations themselves (i.e., the edges in Figure 10) is a manual process. This process can be compared to the construction of proofs as often performed in practice by mathematicians (and thus does not involve any computer techniques such as automated theorem proving).

7. Relation to Other Work

The multi-level BDI-Agent Model for Theory of Mind presented in this article is an example of recursive modelling; see also (Boella and Torre, 2004; Castelfranchi, 1998; Gmytrasiewicz and Durfee, 1995; Marsella, Pynadath, and Read, 2004). In the field of Agent Theory, the idea of recursive modelling has been described in (Gmytrasiewicz and Durfee, 1995) as follows:

‘Recursive modeling method views a multiagent situation from the perspective of an agent that is individually trying to decide what physical and/or communicative actions it should take right now. [...] In order to solve its own decision-making situation, the agent needs an idea of what the other agents are likely to do. The fact that other agents could also be modeling others, including the original agent, leads to a recursive nesting of models.’ (Gmytrasiewicz and Durfee, 1995), p.125.

In (Marsella, Pynadath, and Read, 2004), PsychSim - an implemented multi-agent based simulation tool for modelling interactions and influence among groups or individuals - has been described in the context of childhood bullying and aggression, which provides interesting insight into the role that Theory of Mind plays in human behaviour. In this work, an agent’s Theory of Mind about other agents is crucial in the following sense:

‘For example, a bully motivated by the approval of his classmates would use his mental model of them to predict whether they would enjoy his act of aggression and laugh along with him. Similarly the bully would use his mental model of the teacher to predict whether he will be punished or not’ (Marsella, Pynadath, and Read, 2004), p. 247.

In PsychSim, agents maintain models of each other’s beliefs, goals, policies, etc., and are able to reason about it. This is a form of recursive agent modelling specifically organised to model psychological factors that play a role in influencing human communication and human social interaction in general.

The work by (Castelfranchi, 1998) considers many topics, like foundation of sociality (cooperation, competition, groups, organisation, etc), levels of coordination and cooperation, emergent pre-cognitive structures and constraints. Specifically it addresses influencing other agents and trying to change their behaviour based on a Theory of Mind of the agent:

‘The explicit representation of the agents mind in terms of beliefs, intentions, etc., allows for reasoning about them, and – even more importantly – it allows for the explicit influencing of others, trying to change their behavior (via changing their goals/beliefs).[...] The agents should have some decision function (that implicitly or explicitly presupposes some goal/desire/preference). The influencing agent should give them some hints for this decision, in order to change their behavior’ (Castelfranchi, 1998), p. 178.

However, in that work no formalisation is presented. In contrast, the model presented here has been formally specified.

Also compared to the earlier work described in (Jonker, Treur, and Vries, 2002), a main difference is that in the current paper the agent model is executable and therefore can easily be used for simulation. Moreover, it not only addresses reasoning about the other agent’s beliefs, desires and intentions, but also integrates this with reasoning about the agent’s own beliefs, desires and intentions, and actions in order to perform social manipulation. This part was not formalised in (Jonker, Treur, and Vries, 2002).

From a theoretical angle, much literature is available in foundations of approaches as the one presented here. For example in literature such as (Halpern, Fagin, Moses, and Vardi, 1995), (Jonker, Treur, and Wijngaards, 2003), (Laaksohiet, 2005), a modal logic perspective is used to obtain formal semantics for languages that allow the modeller to express that an agent has reflective knowledge about what another agent knows. However, most of such modal logic approaches do not address the dynamics of the agents’ processes in an executable manner. An exception is (Barringer et al., 1996), where executable temporal logic is used as a basis; however, there the reflective aspect is not incorporated.

The papers mentioned above mainly address Theory of Mind in the sense that an agent reasons about the epistemic (e.g., beliefs) and motivational (e.g.,

desires and intentions) states of another agent. However, according to Gärdenfors (2001), humans can also have a Theory of Mind that involves other mental states, like emotional and attentional states. Also in these areas, computational models have recently been developed. For example, literature that addresses Theory of Mind models of attentional states includes (Asteriadis, Tzouveli, Karpouzis, and Kollias, 2009) and (Bosse, Lambalgen, Maanen, and Treur, 2009). Similarly, Theory of Mind models of emotional states can be found in (Bosse and Lange, 2008), (Bosse, Memon, and Treur, 2008) and (Goldman, 2006). In addition, some papers propose models that involve reasoning about the interaction between multiple mental states; for instance, the work in (Memon and Treur, 2010) addresses a Theory of Mind model for the interaction between beliefs and emotions.

Concerning related work for the Animal Behaviour case study, there is a large body of literature on Theory of Mind in non-human primates, e.g., (Barrett and Henzi, 2005; Heyes, 1998) in particular in chimpanzees (Matsuzawa, Tomonaga, and Tanaka, 2006) and macaques (Sinha, 2003). This literature illustrates that non-human primates use Theories of Mind about other primates while interacting socially with them in specific types of behaviour like imitation, social relationships, deception, and role-taking. Moreover, recent literature suggests that dogs use a certain kind of Theory of Mind as well, e.g., (Horowitz, 2002; Virányi, Topál, Miklósi, and Csányi, 2006). However, none of these papers contains a computational model of Theory of Mind in non-human primates. In contrast, the given case study presents such a model, and illustrates how it can be applied to simulate the behaviour of a prey animal that tries to manipulate the attacking behaviour of a predator.

Concerning related work for the virtual storytelling case study, there is a lot of literature in the area of *interactive drama*, e.g., (Laaksolahti, 2005; Mateas and Stern, 2003), where a user (player) enters in a virtual world, interacts with computer controlled characters, and through his/her interaction influences both the characters and the overall development of the story. Compared to these approaches, our work is slightly different in the sense that the user does not have any interaction with the system. Rather, the storyline is being generated by the interaction between the autonomous virtual characters which are equipped with computational cognitive models, in particular, with mindreading capabilities.

Table 3. Summary of comparison of the proposed model with the literature

	epistemic states	motivational states	attention	emotion	levels of nesting	executable	formal basis
proposed model	X	X			n	X	X
(Asteriadis et al., 2009)			X		2	X	
(Barringer et al. 1996)	X	X			n		X
(Bosse, Lambalgen, Maanen, Treur, 2009)			X		2	X	X
(Bosse and Lange, 2008)	X	X		X	2	X	X
(Bosse, Memon, Treur, 2008)				X	2	X	X
(Castelfranchi, 1998)	X	X			-		
(Gmytrasiewicz and Durfee, 1995)	X	X			5		X
(Goldman, 2006)				X	-		
(Jonker, Treur, and Vries, 2002)	X	X			2		X
(Marsella, Pynadath, and Read, 2004)	X	X			n	X	
(Memon and Treur, 2010)	X			X	2	X	X

As a summary, Table 3 compares the proposed model in the current paper with the closest related work from the literature discussed above in terms of the mental and/or affective states it covers, levels of nesting it employs and whether the model is executable and/or has a formal basis. The different articles from the literature are on the vertical axis (cited by their respective number used in the References section) and all the features are listed on the horizontal axis. A cross indicates that a certain feature is employed within a certain article of the literature and/or in the current paper.

8. Discussion

In order to function efficiently in social life, it is very helpful for an agent to have capabilities to predict in which circumstances the agents in its environment will show certain behaviours. To this end, such an agent will have to perform reasoning based on a Theory of Mind (Baron-Cohen, 1995). This paper presents a model for reasoning based on a Theory of Mind, which makes use of BDI-concepts at different levels. First, the model uses BDI-concepts *within* the Theory of Mind (i.e., it makes use of beliefs, desires and intentions to describe the reasoning process of another agent). Second, it uses BDI-concepts for reasoning *about* the Theory of Mind (i.e., it makes use of beliefs, desires and intentions to describe an agent's meta-reasoning about the reasoning process of another agent). At this second level, meta-statements are involved, such as 'B believes that A desires d' or 'B desires that A does not intend a'. These meta-statements are about the states occurring within the other agent. In addition, meta-statements are involved about the dynamics occurring within the other agents. An example of such a (more complex) meta-statement is 'B believes that, if A performs a, then earlier he or she intended a'.

The multi-level BDI-based model as presented can be exploited both for *social anticipation* (i.e., in order to be prepared for the behaviour of another agent) and for *social manipulation* (i.e., in order to affect the behaviour of another agent at forehand). The model has been formalised using the high-level modelling language LEADSTO, which describes dynamics in terms of direct temporal dependencies between state properties in successive states.

To illustrate the general applicability of the model, it has been applied to three different case studies, which together cover a wide range of applications, including cognitive areas (i.e., Section 4 and 5), as well as engineering areas (i.e., Section 6). The first case study shows how the model can be used to simulate social manipulation within an organisation. This case study addresses the scenario of a manager that reasons about the task avoiding behaviour of his employee. Such models can be used, on the one hand, to get more insight into the social interactions within a given organisation, and - if the model has enough detail - to make predictions about these. On the other hand, they can be used from an engineering perspective, for example to create intelligent virtual agents (e.g., in training environments, games, or virtual stories, as shown in the third case study) with a Theory of Mind.

The second case study is about animal cognition, and illustrates a scenario of a prey that manipulates the behaviour of a predator. At least, this case study has indicated that it is possible to apply computational models for Theory of Mind to animal behaviour. Moreover, the model indeed shows the anticipatory behaviour of higher animals as described in literature such as (Bogdan, 1997). In this sense the model has been validated positively. However, notice that this is a relative

validation, only with respect to the literature that forms the basis of the model. In cases that the available knowledge about the functioning of such animals is improving, the model can be improved accordingly. In this sense the approach anticipates further development.

The third case study demonstrates how the model can be used to obtain characters acting in an emergent soap story, for the purpose of Virtual Storytelling applications. The example soap story addressed concentrated on four characters: Boy A, Boy B, Girl A, and Girl B. In the presented scenario, Boy B exploited mindreading capabilities in order to gain the interest of Girl A. One of the main advantages of generating a storyline based on autonomous virtual agents equipped with cognitive and/or psychological models (in particular, a model for Theory of Mind) is that the characters may behave in a more realistic manner, since they are able to reason about (and possibly manipulate) the behaviour of other characters. In the presented stories, this type of meta-reasoning was shown by Boy B, who was able to predict how the other characters would react in certain circumstances, and used this information to manipulate them. Another advantage of having more autonomous virtual agents is that different variations of the story can be generated based on different settings, initial conditions and external events offered. For instance, in the soap story presented in this paper, the character of Girl B can be changed in the sense that instead of producing the behaviour according to the intention of Boy B (i.e., showing interest in and suggesting to date Boy A), Girl B tells everything about Boy B's plans to Girl A (e.g., the fact that Boy B has offered her money). To create such more complex cognitive models, this paper proposes a BDI-based approach. However, it is not claimed that this is the only possible way to do this.

Note that the presented model makes a number of simplifying assumptions. In particular, it does not explicitly address situations in which the reasoning process does not follow the expected pattern. This has a limitation in a sense that, if for example, an opportunity does not occur in the real world, then the current model does not have any other ways to let the agent fulfill its desire. Another limitation is that, as soon as the agent performs an action, then the desire should be satisfied and it should not hold the time afterwards; however, in the current model, the desire still remains intact. A third case that is not addressed is the situation in which an action fails. In that case, a rational agent would consider other actions that satisfy its desire, i.e., would make some kind of re-planning. Although it is not difficult to extend the current BDI model with such mechanisms, this has been left out of the scope of the current paper, mainly because it would unnecessarily increase the complexity of the Theory of Mind model. Nevertheless, in future work, the possibilities will be explored to extend the model in this direction.

Furthermore, in the presented model, a given Agent B not only has a Theory of Mind about an Agent A, but it also uses this theory in its own reasoning process in order to do social manipulation, i.e., to change the behaviour of Agent A. The model was designed in such a way that the Agent A does not know beforehand that Agent B is trying to manipulate him by changing some of the beliefs. Thus, the situation was not considered that Agent A tries to not to be manipulated by Agent B. In future research it will be addressed how such elements can be included in the model. For instance, in the task avoidance case study, the employee may have models of other employees to infer who else is available for a task. In addition, a number of other extensions will be addressed, such as more experiments with a deeper nesting of Theory of Mind concepts (e.g.,

A believes that B desires that C performs an action), and probabilistic and adaptive Theory of Mind models. Another interesting challenge will be to explore to what extent the model can be used to develop applications within Ambient Intelligence (Aarts, Collier, Loenen, and Ruyter, 2003). According to this view (which envisions a world in which humans are surrounded by pervasive computing technology that is responsive to their state), an Ambient Agent can be seen as an entity that has a Theory of Mind of a human. For example, such an agent may have knowledge about a person's state of emotion, stress, or workload, and provide adaptive support based on that knowledge. Some initial studies, in which the presented model has been used for this purpose (Bosse, Hoogendoorn, Klein, and Treur, 2008), have indicated that this is a promising direction.

9. References

- Aarts, E.; Collier, R.; van Loenen, E.; Ruyter, B. de (eds.) (2003). *Ambient Intelligence. Proceedings of the First European Symposium, EUSAI 2003*. LNCS, vol. 2875. Springer Verlag, 2003, pp. 432.
- Aristotle (350 BCa). *Nicomachean Ethics* (translated by W.D. Ross)
- Aristotle (350 BCb). *De Motu Animalium* On the Motion of Animals (translated by A. S. L. Farquharson)
- Asteriadis, S., Tzouveli, P., Karpouzis, K., and Kollias, S. (2009). Estimation of behavioral user state based on eye gaze and head pose – application in an e-learning environment. *Multimedia Tools and Applications*, 41, 469-493.
- Baron-Cohen, S. (1995). *Mindblindness*. MIT Press.
- Barrett, L. and Henzi, P. (2005). The social nature of primate cognition. *Proceedings of The Royal Society of Biological Sciences*, vol. 272, pp. 1865-1875.
- Barringer, H., M. Fisher, D. Gabbay, R. Owens, & M. Reynolds (1996). *The Imperative Future: Principles of Executable Temporal Logic*, Research Studies Press Ltd. and John Wiley & Sons.
- Boella, G. and van der Torre, L. (2004). Groups as agents with mental attitudes. In: Jennings, N.R., Sierra, C., Sonenberg, L., and Tambe, M. (eds.), *Proceedings of the third international joint conference on Autonomous Agents and Multi Agent Systems, AAMAS'04*, pp. 964-971.
- Bogdan, R.J. (1997). *Interpreting Minds*. MIT Press.
- Bosse, T., Hoogendoorn, M., Klein, M.C.A., and Treur, J. (2008). An Agent-Based Generic Model for Human-Like Ambience. In: M. Mühlhäuser, A. Ferscha, and E. Aitenbichler (eds.), *Constructing Ambient Intelligence: AmI-07 Workshops Proceedings*. Communications in Computer and Information Science (CCIS), vol. 11, Springer Verlag, 2008, pp. 93-103.
- Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A., and Treur, J. (2006). Specification and Verification of Dynamics in Agent Models. *International Journal of Cooperative Information Systems*, vol. 18, 2009, pp. 167 - 193. Preliminary version in: Proc. of the 6th Int. Conf. on Intelligent Agent Technology, IAT'06. IEEE Computer Society Press, 2006, pp. 247-254.

- Bosse, T., Jonker, C.M., Meij, L. van der, and Treur, J. (2007). A Language and Environment for Analysis of Dynamics by Simulation. *Int. Journal of Artificial Intelligence Tools*, vol. 16, 2007, pp. 435-464. Earlier version in: Eymann, T. et al. (eds.), Proc. of the 3rd German Conf. on Multi-Agent System Technologies, MATES'05. Springer LNAI, vol. 3550, pp. 165-178.
- Bosse, T., Lambalgen, R. van, Maanen, P.P. van, and Treur, J. (2009). Attention Manipulation for Naval Tactical Picture Compilation. In: Baeza-Yates, R., Lang, J., Mitra, S., Parsons, S., and Pasi, G. (eds.), *Proceedings of the 9th IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT'09*. IEEE Computer Society Press, 2009, pp. 450-457.
- Bosse, T. and Lange, F.P.J. de (2008). Development of Virtual Agents with a Theory of Emotion Regulation. In: Jain, L., Gini, M., Faltings, B.B., Terano, T., Zhang, C., Cercone, N., and Cao, L. (eds.), *Proceedings of the Eighth IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT'08*. IEEE Computer Society Press, 2008, pp. 461-468.
- Bosse, T., Memon, Z.A., and Treur, J. (2008). Adaptive Estimation of Emotion Generation for an Ambient Agent Model. In: *Proceedings of the Second European Conference on Ambient Intelligence, Aml'08*. Lecture Notes in Computer Science, vol. 5355. Springer Verlag, 2008, pp. 141-156.
- Bosse, T., Memon, Z.A., and Treur, J. (2007a). A Two-Level BDI-Agent Model for Theory of Mind and its Use in Social Manipulation. In: *Proceedings of the AISB 2007 Workshop on Mindful Environments*, pp 335-342.
- Bosse, T., Memon, Z.A., and Treur, J. (2007b). Emergent Storylines Based on Autonomous Characters with Mindreading Capabilities. In: Lin, T.Y., Bradshaw, J.M., Klusch, M., Zhang, C., Broder, A., and Ho, H. (eds.), *Proceedings of the Seventh IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT'07*. IEEE Computer Society Press, 2007, pp. 207-214.
- Bosse, T., Memon, Z.A., and Treur, J. (2007c). Modelling Animal Behaviour Based on Interpretation of Another Animal's Behaviour. In: Lewis, R., Polk, T, and Laird, J. (eds.), *Proceedings of the 8th International Conference on Cognitive Modeling, ICCM'07*. Taylor and Francis, 2007, pp. 193-198.
- Castelfranchi, C. (1998). *Modelling Social Action for AI Agents*. Artificial Intelligence, vol. 103, 1998, pp. 157-182
- Dennett, D.C. (1987). *The Intentional Stance*. MIT Press. Cambridge Mass.
- Dennett, D.C. (1991). Real Patterns. *The Journal of Philosophy*, vol. 88, pp. 27-51.
- Gärdenfors, P. (2001). Slicing the Theory of Mind. In: *Danish yearbook for philosophy*, 36, pp. 7-34. Museum Tusculanum Press.
- Georgeff, M. P. and Lansky, A. L. (1987). Reactive Reasoning and Planning. In: Forbus, K. and Shrobe, H. (eds.), *Proceedings of the Sixth National Conference on Artificial Intelligence, AAAI'87*. Menlo Park, California. American Association for Artificial Intelligence, 1987, pp. 677-682.
- Gmytrasiewicz, P. J. and Durfee, E. H. (1995). A rigorous, operational formalization of recursive modeling. In: Lesser, V. (ed.), *Proceedings of the First International Conference on Multiagent Systems*, pp. 125-132, 1995.

- Goldman, A.I. (2006). *Simulating Minds: the Philosophy, Psychology and Neuroscience of Mindreading*. Oxford University Press.
- Halpern, J.Y., Fagin, R., Moses, Y., and Vardi, M. Y. (1995). *Reasoning About Knowledge*. MIT Press, 1995.
- Heyes, C.M. (1998). Theory of mind in nonhuman primates. *Behavioural and Brain Sciences*, vol. 21, pp. 101-134.
- Horowitz, A. (2002). The behaviors of theories of mind, and a case study of dogs at play. PhD. Thesis, University of California, 2002.
- Jonker, C.M. and Treur, J. (2002). Compositional Verification of Multi-Agent Systems: a Formal Analysis of Pro-activeness and Reactiveness. *Intern. J. of Cooperative Information Systems*, vol. 11, 2002, pp. 51-92.
- Jonker, C.M., Treur, J., and Vries, W. de (2002). Temporal Analysis of the Dynamics of Beliefs, Desires, and Intentions. *Cognitive Science Quarterly*, vol. 2, 2002, pp.471-494.
- Jonker, C.M., Treur, J., and Wijngaards, W.C.A. (2003). A Temporal Modelling Environment for Internally Grounded Beliefs, Desires and Intentions. *Cognitive Systems Research Journal*, vol. 4(3), 2003, pp. 191-210.
- Laakolahti, J. (2005). Methods for evaluating a dramatic game. In: Kaleidoscope Workshop on Narrative Learning Environments, 9-10 June 2005, Lisboa, Portugal
- Malle, B.F., Moses, L.J., and Baldwin, D.A. (2001). *Intentions and Intentionality: Foundations of Social Cognition*. MIT Press.
- Marsella, S.C., Pynadath, D.V., and Read, S.J. (2004). PsychSim: Agent-based modeling of social interaction and influence. In: Lovett, M., Schunn, C.D., Lebiere, C., and Munro, P. (eds.), *Proceedings of the International Conference on Cognitive Modeling, ICCM 2004*, pp. 243-248 Pittsburg, Pennsylvania, USA.
- Matsuzawa, T., Tomonaga, M., and Tanaka, M. (2006). *Cognitive Development in Chimpanzees*. Springer Verlag, Tokyo, 2006.
- Mateas, M. and Stern, A. (2003). Integrating plot, character and natural language processing in the interactive drama Façade, 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE '03), Darmstadt, Germany, March 24-26, 2003
- Memon, Z.A. and Treur, J. (2010). A Cognitive Agent Model for Simulation-Based Mindreading and Empathic Understanding. *International Journal of Human-Computer Studies*, to appear.
- Rao, A.S. and Georgeff, M.P. (1991). Modelling Rational Agents within a BDI-architecture. In: Allen, J., Fikes, R. and Sandewall, E. (eds.), *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, (KR'91)*. Morgan Kaufmann, pp. 473-484.
- Rao, A.S. and Georgeff, M.P. (1995). BDI-agents: from theory to practice. In: Lesser, V. (ed.), *Proceedings of the International Conference on Multiagent Systems*, pp. 312 – 319.

Sinha, A. (2003). A beautiful mind: Attribution and intentionality in wild bonnet macaques. *Current Science*, vol. 85. no. 7, 2003.

Virányi, Zs., Topál, J., Miklósi, Á, and Csányi, V. (2006). A nonverbal test of knowledge attribution: a comparative study of dogs and children. *Animal Cognition*, vol. 9, no. 1, pp. 13-26.