Abstraction Relations Between Internal and Behavioural Agent Models for Collective Decision Making[1]

Alexei Sharpanskykh* and Jan Treur

Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
http://www.few.vu.nl/~{sharp,treur} {sharp, treur}@few.vu.nl

* Alexei Sharpanskykh is the corresponding author
tel.: +31205985887, fax: +31205987653

Abstract
For agent-based modelling of collective phenomena, more and more agent models are employed that go beyond simple reactive behaviour. Such less trivial individual behaviours can be modelled either from an agent-internal perspective, in the form of direct (causal) temporal relations between internal states of the agent, or from an agent-external, behavioural perspective, in the form of more complex input-output relations for the agent. Illustrated by a case study on collective decision making, this paper addresses how the two types of agent models can be related to each other. First an internal agent model for collective decision making is presented, based on neurological principles. It is shown how by an automated systematic transformation from an internal agent model an abstracted behavioural model can be obtained, by abstracting from the internal states. The abstraction approach introduced includes specific methods for abstraction of internal loops, as often occur in neurologically inspired internal agent models, for example to be able to model mutual interaction between cognitive and affective states. As an example of a given behavioural agent model, an existing behavioural agent model for collective decision making incorporating principles on social diffusion is described. It is shown under which conditions and how by an interpretation mapping the obtained abstracted behavioural agent model can be related to this existing behavioural agent model for collective decision making.

Keywords: cognitive agent models, behavioural abstraction of cognitive models, collective decision making, social diffusion models, affective and cognitive aspects in agent models

[1] The material in this paper includes an integration and extension of material from two conference papers:
(1) Sharpanskykh, A., and Treur, J., Abstraction Relations Between Internal and Behavioural Agent Models for Collective Decision Making. In: Pan, J.-S., et al. (eds.), *Proceedings of the Second International Conference on Computational Collective Intelligence, ICCCI'10*. Lecture Notes in Artificial Intelligence, Springer Verlag, 2010;
(2) Sharpanskykh, A., and Treur, J., Behavioural Abstraction of Agent Models Addressing Mutual Interaction of Cognitive and Affective Processes. In: *Proceedings of the Second International Conference on Brain Informatics, BI'10*. Lecture Notes in Artificial Intelligence, Springer Verlag, 2010.

1. Introduction

Agent models used for collective social phenomena with multiple agents traditionally are kept simple, and often are specified by simple reactive rules that determine a direct response (output) based on the agent's current perception (input). This is a way in which complexity of the multi-agent system model can be kept limited. However, in recent years it is more and more acknowledged that in some cases agent models specified in the simple format as input-output associations are too limited. Extending specifications of agent models beyond the format of simple reactive input-output associations essentially can be done in two different manners: (a) by allowing more complex temporal relations between the agent's input and output states over time, or (b) by taking into account internal states within the agents, and internal processes described by temporal (causal) relations between such states. Such less restrictive formats enable the use of agents, for example, with some form of memory, or agents that are able to gradually adapt their responses.

Considering extended formats for specification of agent models used to model collective social phenomena, of the two types (a) and (b) raises a number of (interrelated) questions:

(1) When agent models of type (a) are used in social simulation, do they provide the same results as when agent models of type (b) are used?
(2) How can an agent model of type (a) be related to one of type (b)?
(3) Can agent models of type (a) be transformed into agent models of type (b), by some systematic procedure, and conversely?

For the context of modelling collective social phenomena, a first observation is that interaction between agents or with the environment is in principle assumed to be modelled as taking place via the input and output states of the agents; indeed, this can be considered as defining the notions of input and output. This implies that the internal states in agent models of type (b) do not have a direct impact on the social process; agent models that show the same input-output relations over time will lead to exactly the same results at the collective level, no matter what internal states occur. This suggests that for modelling social phenomena the internal states in such an agent model of type (b) could be hidden or abstracted away by transforming the model in one way or the other into a model of type (a). An interesting challenge here is how this can be done in a precise and systematic manner.

The questions mentioned above are addressed in this paper based on notions such as ontology mappings and extensions thereof, temporal properties expressed in logical, numerical (difference equations) or hybrid (logical and numerical) formats, and logical and numerical relations between such temporal properties. Here the idea to use ontology mappings and extensions of them is adopted from [15], where such techniques are used to address reduction relations between (internal) cognitive/affective and neural agent models. However, as shown below, it turns out that these techniques can be worked out at the social level to relate (more abstract) agent models of type (a) to those of type (b) as well. This will provide a formal basis to address question (2) above. Moreover, based on such a formally defined relation, addressing question (1) it can be established that at the social level the results will be the same; this holds both for specific simulation traces and for the implied temporal properties (patterns) they have in common. It will be discussed how models of type (b) can be abstracted to models of type (a) by a systematic transformation, which has been implemented in Java, thus also providing an answer to question (3).

The approach is illustrated by a case addressing the emergence of group decisions. It is inspired by some basic principles from the neurological literature: body loops and as if body loops and somatic marking as a basis for individual decision making (see [4], [6], [7]), and mirroring of emotions and intentions as a basis for mutual influences between group members (see [10], [11], [12]). Agent models inspired by neurological principles sometimes include internal loops, for example as if body loops, or loops involved in internal adaptation. Therefore special attention is paid on how loops can be abstracted.

The paper is organized as follows. Section 2 presents an internal agent model **IAM** for decision making in a group, based on neurological principles, and modelled both in neural network format and in a hybrid logical/numerical format; cf. [3]. In Section 3 an existing (in numerical format) behavioural agent model **BAM** for group decision making is briefly described, and specified in the same hybrid format. In Section 4 first the internal agent model **IAM** introduced in Section 2 is abstracted to a behavioural model **ABAM**, and next it is illustrated how a specific loop elimination technique can be applied, leading to behavioural agent model **AEBAM** as a variation of **ABAM**; the behavioural agent models **ABAM** and **AEBAM** are shown to display comparable behaviour. Next in Section 5 it is shown how the behavioural agent model **ABAM** can be related to the existing behavioural agent model **BAM**, by means of an interpretation mapping, based on an ontology mappings between the ontologies used for **ABAM** and **BAM**. Section 6 concludes the paper.

2 The Internal Agent Model IAM for Group Decision Making

This case study concerns a neurologically inspired computational modeling approach for the emergence of group decisions, incorporating somatic marking as a basis for individual decision making, see [4], [6], [7] and mirroring of emotions and intentions as a basis for mutual influences between group members, see [10], [11], [12]. The model shows how for many cases, the combination of these two neural mechanisms is sufficient to obtain on the one hand the emergence of common group decisions, and, on the other hand, to achieve that the group members feel OK with these decisions.

Cognitive states of a person, such as sensory or other representations often induce emotions felt within this person, as described by neurologist Damasio [5], [6]. Damasio's *Somatic Marker Hypothesis* (cf. [4], [6], [7]), is a theory on decision making which provides a central role to emotions felt. Within a given context, each represented decision option induces (via an emotional response) a feeling which is used to mark the option. Thus the Somatic Marker Hypothesis provides endorsements or valuations for the different options, and shapes an individual's decision process.

In a social context, the idea of somatic marking can be combined with recent neurological findings on the *mirroring function* of certain neurons (e.g., [10], [11], [12]). Such neurons are active not only when a person prepares for performing a specific action or body change, but also when the person observes somebody else intending or performing this action or body change. This includes expressing emotions in body states, such as facial expressions. The idea is that these neurons and the neural circuits in which they are embedded play an important role in social functioning and in (empathic) understanding of others; (e.g., [10], [11], [12]). They provide a biological basis for many social phenomena; cf. [10]. Indeed, when states of other persons are mirrored by some of the person's own states that at the same time are connected via neural circuits to states that are crucial for the own feelings and actions, then this provides an effective basic (biological) mechanism for how in a social context persons fundamentally affect each other's actions and feelings, and, for example, are able to achieve collective decision making.

Table 1 State ontology used

notation	*description*
SS	sensor state
SRS	sensory representation state
PS	preparation state
ES	effector state
BS	body state
c	observed context information
O	option
c(O)	tendency to choose for option O
b(O)	own bodily response for option O
g(b(O))	other group members' aggregated bodily response for option O
g(c(O))	other group members' aggregated tendency to choose for option O

Given the general principles described above, the mirroring function can be related to decision making in two different ways. In the first place *mirroring of emotions* indicates how emotions felt in different individuals about a certain considered decision option mutually affect each other, and, assuming a context of somatic marking, in this way affect how by individuals decision options are valuated. A second way in which a mirroring function relates to decision making is by applying it to the *mirroring of intentions* or *action tendencies* of individuals for the respective decision options. This may work when by verbal and/or nonverbal behaviour individuals show in how far they tend to choose for a certain option. In the internal agent model **IAM** introduced below both of these (emotion and intention) mirroring effects are incorporated.

An overview of the internal model **IAM** is given in Fig. 1. Here the notations for the state ontology describing the nodes in this network are used as shown in Table 1, and for the parameters as in Table 2.

Moreover, the solid arrows denote internal causal relations whereas the dotted arrows indicate interaction with other group members. The arrow from PS(A, b(O)) to SRS(A, b(O)) indicates an as-if body loop that can be used to modulate (e.g., amplify or suppress) a bodily response (cf. [5]).

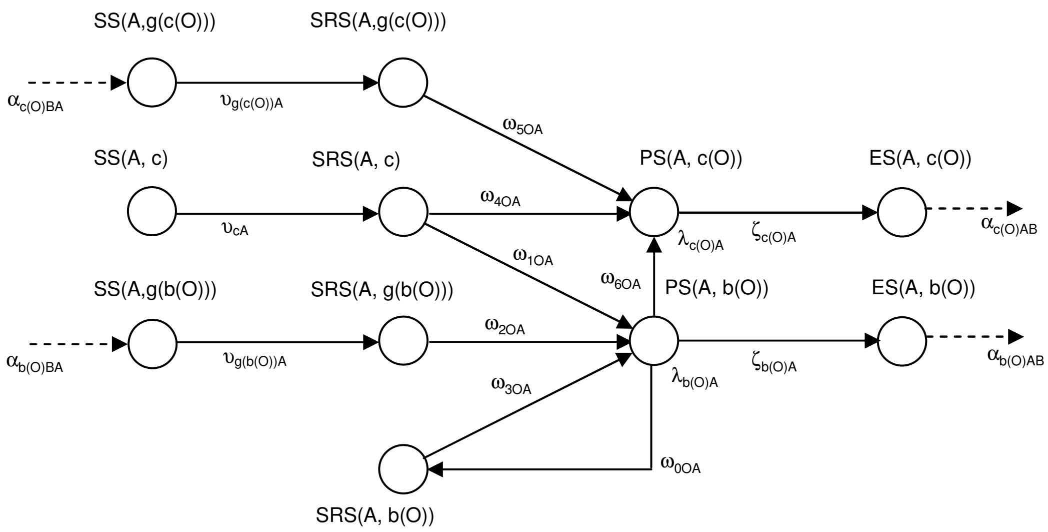

Fig. 1 Overview of the internal agent model **IAM**

Table 2 Parameters for the internal agent model

description	*parameter*	*from*	*to*
strengths of connections within agent A	υ_{SA}	SS(A, S)	SRS(A, S)
	ω_{0OA}	PS(A, b(O))	SRS(A, b(O))
	ω_{1OA}	SRS(A, c)	PS(A, b(O))
	ω_{2OA}	SRS(A, g(b(O)))	
	ω_{3OA}	SRS(A, b(O))	
	ω_{4OA}	SRS(A, c)	PS(A, c(O))
	ω_{5OA}	PS(A, g(b(O)))	
	ω_{6OA}	SRS(A, c(O))	
	ζ_{SA}	PS(A, S)	ES(A, S)
strength for channel for Z from agent B to agent A	α_{ZBA}	sender B	receiver A
change rates for states within agent A	$\lambda_{b(O)A}$	change rate for PS(A, b(O))	
	$\lambda_{c(O)A}$	change rate for PS(A, c(O))	

This model can be described in a detailed manner in different forms. First it is shown how it can be described by a hybrid (cf. [3]) network specification **NS** which can be used in conjunction with a generic mechanism specification **GNP** for propagation of activation over such a network.

Hybrid Network Specification NS

The network specification **NS** consists of two parts: a network structure specification **NSS** and a network values specification **NVS**.

NSS Network Structure Specification

incoming_connections_to(SS(A,S), SRS(A,S)) with S taking instances c, g(b(O)), g(c(O)) for options O
incoming_connections_to(SRS(A, c), SRS(A, g(b(O))), SRS(A, b(O)), PS(A, b(O))) for options O
incoming_connections_to(SRS(A, c), SRS(A, g(c(O))), PS(A, b(O)), PS(A, c(O))) for options O

incoming_connections_to(PS(A, S), ES(A, S)) with S taking instances b(O), c(O) for options O

NVS Network Values Specification

connection_strength(SS(A,S), SRS(A,S), υ_{SA}) with S taking instances c, b(O), g(b(O)), g(c(O)) for options O
connection_strength(PS(A, b(O)), SRS(A, b(O)), ω_{0OA})
connection_strength(SRS(A, c), PS(A, b(O)), ω_{1OA})
connection_strength(SRS(A,g(b(O))), PS(A, b(O)), ω_{2OA})
connection_strength(SRS(A, b(O)), PS(A, b(O)), ω_{3OA})
connection_strength(SRS(A, c), PS(A, c(O)), ω_{4OA})
connection_strength(SRS(A, g(c(O))), PS(A, c(O)), ω_{5OA})
connection_strength(PS(A, b(O)), PS(A, c(O)), ω_{6OA})
connection_strength(PS(A, S), ES(A, S), ζ_{SA}) with S taking instances b(O), c(O) for options O
change_rate(PS(A, S), λ_{SA}) with S taking instances b(O), c(O) for options O

Note that when adaptivity of connection strengths or other network values, is involved, the values prespecified in **NVS** can be considered initial values. In such a case, **GNP** can be extended by mechanisms to change such values. In hybrid logical/numerical format (cf. [3]) this is expressed as follows. Here → denotes a causal relationship, and the specification is assumed to be universally quantified over the free variables.

GNP Propagation of activation

If nodes N_1, N_2, N_3 are the incoming connections to node N
and the connection strength from N_1 to N is ω_1
and the connection strength from N_2 to N is ω_2
and the connection strength from N_3 to N is ω_3
and the change rate for node N is λ
and node N_1 has activation level V_1
and node N_2 has activation level V_2
and node N_3 has activation level V_3
and node N has activation level V
then node N will have activation level $V + \lambda f(V_1, V_2, V_3, V)\, \Delta t$

incoming_connections_to(N_1, N_2, N_3, N) &
connection_strength(N_1, N, ω_1) & connection_strength(N_2, N, ω_2) & connection_strength(N_3, N, ω_3) &
change_rate(N, λ) &
activation(N_1, V_1) & activation(N_2, V_2) & activation(N_3, V_3) & activation(N, V)
→ activation(N, $V + \lambda f(\omega_1 V_1, \omega_2 V_2, \omega_3 V_3, V)\, \Delta t$)

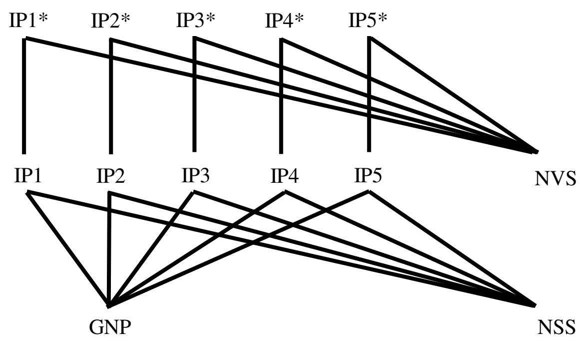

Fig. 2. From network representation to hybrid logical/numerical representation: **NS** ⊢ **IAM**

Here $f(\omega_1 V_1, \omega_2 V_2, \omega_3 V_3, V)$ is a function that provides a value by which V is to be adjusted, given the incoming values $\omega_1 V_1$, $\omega_2 V_2$, $\omega_3 V_3$,, and λ is a change rate for V. As an example, when for f a combination function fc is used, for example, it can be defined as

$$f(\omega_1 V_1, \omega_2 V_2, \omega_3 V_3, V) = fc(\beta, \omega_1 V_1, \omega_2 V_2, \omega_3 V_3) - V$$

with $fc(\beta, V_1, V_2, V_3) = \beta\,(1-(1- V_1)(1- V_2)(1- V_3)) + (1-\beta)\, V_1 V_2 V_3$.

Deriving the internal agent model IAM from network specification NS

The generic specification **GNP** can be instantiated by specific information about the network structure as specified in **NSS**, to obtain a number of properties IP1 to IP6 according to the nodes in the network; this can be viewed as a form of partial knowledge compilation. For example, for the node PS(A, b(O)) the following property can be derived (for some function g that still can be chosen):

IP2 Preparing for a body state

connection_strength(SRS(A, c), PS(A, b(O)), ω_1) &
connection_strength(SRS(A, g(b(O))), PS(A, b(O)), ω_2) &
connection_strength(SRS(A, b(O)), PS(A, b(O)), ω_3) &
change_rate(PS(A, b(O)), λ) &
activation(SRS(A, c), V_1) & activation(SRS(A, g(b(O))), V_2) & activation(SRS(A, b(O)), V_3) &
activation(PS(A, b(O)), V)
$\rightarrow$ activation(PS(A, b(O)), V + λ g($\omega_1 V_1$, $\omega_2 V_2$, $\omega_3 V_3$, V) Δt)

When as a next step also the parameter values as specified in **NVS** are instantiated, indicated as in Table 2, the following property can be obtained

activation(SRS(A, c), V_1) & activation(SRS(A, g(b(O))), V_2) & activation(SRS(A, b(O)), V_3) &
activation(PS(A, b(O)), V)
$\rightarrow$ activation(PS(A, b(O)), V + $\lambda_{b(O)A}$ g($\omega_{1OA}V_1$, $\omega_{2OA}V_2$, $\omega_{3OA}V_3$, V) Δt)

Using a slightly different format of representation (writing N(V) instead of activation(N, V)) this gets the form of property IP2* and similarly the other properties from IP1* to IP5* as shown below and derived from the network specification.

Hybrid Specification of the Internal Agent Model IAM

The following internal dynamic properties in hybrid logical/numerical format (cf. [3]) describe the agent A's internal model **IAM**.

IP1* From sensor states to sensory representations

SS(A, S, V) $\rightarrow$ SRS(A, S, υ_{SA}V)

where S has instances c, g(c(O)) and g(b(O)) for options O.

IP2* Preparing for an emotion expressed in a body state

SRS(A, c, V_1) & SRS(A, g(b(O)), V_2) & SRS(A, b(O), V_3) & PS(A, b(O), V)
$\rightarrow$ PS(A, b(O), V + $\lambda_{b(O)A}$ g($\omega_{1OA}V_1$, $\omega_{2OA}V_2$, $\omega_{3OA}V_3$, V) Δt)

IP3* Preparing for an option choice

SRS(A, c, V_1) & SRS(A, g(c(O)), V_2)) & PS(A, b(O), V_3) & PS(A, c(O), V)
$\rightarrow$ PS(A, c(O), V + $\lambda_{c(O)A}$ h($\omega_{4OA}V_1$, $\omega_{5OA}V_2$, $\omega_{6OA}V_3$, V) Δt)

IP4* From preparation to effector state

PS(A, S, V) $\rightarrow$ ES(A, S, ζ_{SA} V)

where S has instances b(O) and c(O) for options O.

IP5* From preparation to sensory representation of body state

PS(S, V) $\rightarrow$ SRS(S, ω_{0OA}V)

where S has instances b(O) for options O.

Here the functions g(X_1, X_2, X_3, X_4) and h(X_1, X_2, X_3, X_4) are chosen, for example, of the form fc(β, X_1, X_2, X_3) – X_4, where fc(β, X_1, X_2, X_3) =β (1-(1- V_1)(1- V_2)(1- V_3)) + (1-β) V_1 V_2 V_3.

Next the following transfer properties describe the interaction between agents for emotional responses b(O) and choice tendencies c(O) for options O. Thereby the sensed input from multiple agents is aggregated by adding, for example, all influences $\alpha_{b(O)BA}V_B$ on A with V_B the levels of the effector state of agents B ≠ A, to the sum $\Sigma_{B\neq A}$ $\alpha_{b(O)BA}V_B$ and normalising this by dividing it by the maximal value $\Sigma_{B\neq A}$ $\alpha_{b(O)BA}\zeta_{b(O)B}$ for it (when all preparation values would be *1*). This provides a kind of average of the impact of all other agents, weighted by the normalised channel strengths.

ITP Sensing aggregated group members' bodily responses and intentions

$\bigwedge_{B\neq A}$ ES(B, S, V_B) $\rightarrow$ SS(A, g(S), $\Sigma_{B\neq A}$ $\alpha_{SBA}V_B$ / $\Sigma_{B\neq A}$ $\alpha_{SBA}\zeta_{SB}$)

where S has instances b(O), c(O) for options O.

Note that the following logical implications are valid (hierarchically depicted in Fig. 2), since the IP properties were logically derived from the network specification:

GNP & NSS $\Rightarrow$ IP1	GNP & NSS $\Rightarrow$ IP4	IP1 & NVS $\Rightarrow$ IP1*	IP4 & NVS $\Rightarrow$ IP4*
GNP & NSS $\Rightarrow$ IP2	GNP & NSS $\Rightarrow$ IP5	IP2 & NVS $\Rightarrow$ IP2*	IP5 & NVS $\Rightarrow$ IP5*
GNP & NSS $\Rightarrow$ IP3		IP3 & NVS $\Rightarrow$ IP3*	

This can also be summarised as **NS** $\vdash$ **IAM**, where $\vdash$ is a symbol for derivability.

Based on the internal agent model IAT a number of simulation studies have been performed, using MathLab. Some results for two simulation settings with 10 homogeneous agents with the parameters as defined in Table 3 are presented in Figure 3. The initial values for SS(A, g(c(O))), SS(A, c), SS(A, g(b(O)) are set to 0 in both settings.

Table 3 The values of the parameters of model IAM used in two simulation settings

description	*parameter*	*setting 1*	*setting 2*
strengths of connections within agent A	$\upsilon_{g(c(O))A}$	0.65	0.55
	υ_{cA}	0.8	0.8
	$\upsilon_{g(b(O))A}$	0.55	0.75
	ω_{0OA}	0.8	0.9
	ω_{1OA}	0.9	0.8
	ω_{2OA}	0.7	0.6
	ω_{3OA}	0.8	0.7
	ω_{4OA}	0.9	0.8
	ω_{5OA}	0.9	0.4
	ω_{6OA}	0.8	0.7
	$\zeta_{c(O)A}$	0.75	0.45
	$\zeta_{b(O)A}$	0.85	0.55
strength for channel for Z from any agent to any other agent	α_{ZBA}	0.9	0.9
change rates for states within agent A	$\lambda_{b(O)A}$	0.3	0.2
	$\lambda_{c(O)A}$	0.2	0.2
parameter of the combination function fc(β, X_1, X_2, X_3)	β	0.6	0,6

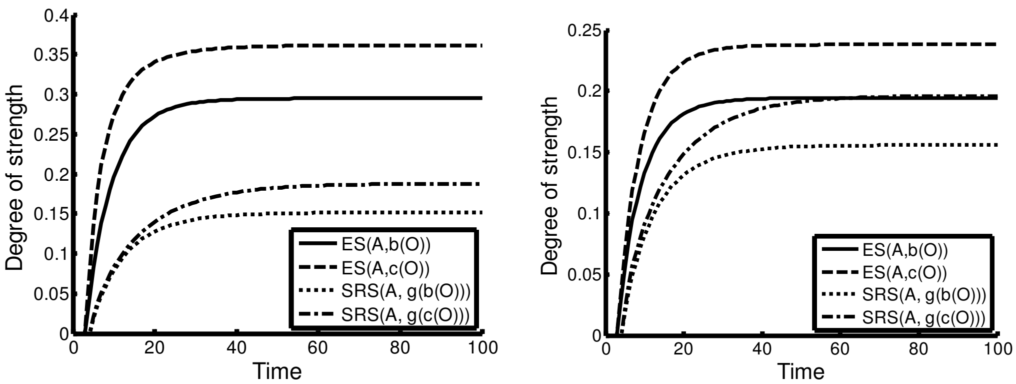

Fig. 3. The dynamics of ES(A, b(O)), ES(A, c(O)), SRS(A, g(b(O))) and SRS(A, g(c(O))) states of an agent A from a multi-agent system with 10 homogeneous agents over time for simulation setting 1 (left) and setting 2 (right) with the parameters from Table 3.

As one can see from Fig. 3, in both simulation settings the dynamics of the multi-agent system stabilizes after some time.

3 A Behavioural Agent-Based Model for Group Decision Making: BAM

In [9], an agent-based model for group decision making is introduced. The model was designed in a manner abstracting from the agents' internal neurological, cognitive or affective processes. It was specified in numerical format by mathematical (difference) equations and implemented in MatLab. In this section, first in Section 3.1 a general agent-based model for social diffusion is briefly introduced and then in Section 3.2 this is used as a basis for an agent-based model for group decision making.

3.1 An agent-based social diffusion model

First a general agent-based social contagion or diffusion model is described for any type of state *S* of a person (for example, an emotion or an intention state). This model is primarily based on ideas about social contagion or diffusion processes, and also loosely inspired by principles that became known from the neurological area. It was obtained as an integration and generalisation of two earlier agent-based models for emotion contagion (absorption and amplification) introduced in [1, 2]. As a first step, the contagion strength for *S* from person *B* to person *A* is defined by:

$$\gamma_{SBA} = \varepsilon_{SB} \cdot \alpha_{SBA} \cdot \delta_{SA} \quad (1)$$

Here ε_{SB} is the personal characteristic *expressiveness* of the sender (person *B*) for *S*, δ_{SA} the personal characteristic *openness* of the receiver (person *A*) for *S*, and α_{SBA} the interaction characteristic *channel strength* for *S* from sender *B* to receiver *A*. The expressiveness describes the strength of expression of given internal states by verbal and/or nonverbal behaviour (e.g., body states). The openness describes how strong stimuli from outside are propagated internally. The channel strength is as before.

To determine the level $q_{SA}(t)$ of an agent *A* for a specific state *S* the following model is used. First, the overall contagion strength γ_{SA} from the group towards agent *A* is calculated:

$$\gamma_{SA} = \sum_{B\neq A} \gamma_{SBA} = (\sum_{B\neq A} \varepsilon_{SB} \cdot \alpha_{SBA}) \cdot \delta_{SA} \quad (2)$$

This value is used to determine the weighed impact $q_{SA}{}^*(t)$ of all the other agents upon state *S* of agent *A*:

$$q_{SA}{}^*(t) = \sum_{B\neq A} \gamma_{SBA} \cdot q_{SB}(t) \,/\, \gamma_{SA} = \sum_{B\neq A} \varepsilon_{SB} \cdot \alpha_{SBA} \cdot q_{SB}(t) \,/\, (\sum_{B\neq A} \varepsilon_{SB} \cdot \alpha_{SBA}) \quad (3)$$

How much this external influence actually changes state *S* of the agent *A* is determined by two additional personal characteristics of the agent, namely the tendency η_{SA} to absorb or to amplify the level of a state and the bias β_{SA} towards positive or negative impact for the value of the state. The model to update the value of $q_{SA}(t)$ over time is then expressed as follows:

$$q_{SA}(t + \Delta t) = q_{SA}(t) + \gamma_{SA}\, \mathbf{c}(q_{SA}{}^*(t), q_{SA}(t))\, \Delta t \quad (4)$$

with

$$\mathbf{c}(X, Y) = \eta_{SA} \cdot [\beta_{SA} \cdot (1 - (1-X) \cdot (1-Y)) + (1 - \beta_{SA}) \cdot XY] + (1 - \eta_{SA}) \cdot X - Y$$

Note that for $\mathbf{c}(X, Y)$ any function can be taken that combines the values of *X* and *Y* and compares the result with Y. For the example function $\mathbf{c}(X, Y)$ adopted as a kind of smallest common multiple from the two existing emotion contagion models, the new value of the state is the old value, plus the change of the value based on the contagion. This change is defined as the multiplication of the contagion strength times a factor for the amplification of information plus a factor for the absorption of information. The absorption part (after $1 - \eta_{SA}$) simply considers the difference between the incoming contagion and the current level for *S*. The amplification part (after η_{SA}) depends on the tendency or bias of the agent towards more positive (part of equation multiplied by β_{SA}) or negative (part of equation multiplied by $1 - \beta_{SA}$) level for *S*. Table 4 summarizes the most important parameters and state variables within the model (note that the last two parameters will be explained in Section 3.2 below).

Table 4. Parameters and state variables

$q_{SA}(t)$	level for state *S* for agent *A* at time *t*
$e_{SA}(t)$	expressed level for state *S* for agent *A* at time *t*
$S_{g(S)A}(t)$	aggregated input for state *S* for agent *A* at time *t*
ε_{SA}	extent to which agent *A* expresses state *S*
δ_{SA}	extent to which agent *A* is open to state *S*
η_{SA}	tendency of agent *A* to absorb or amplify state *S*
β_{SA}	positive or negative bias of agent *A* on state *S*
α_{SBA}	channel strenght for state *S* from sender *B* to receiver *A*
γ_{SBA}	contagion strength for *S* from sender *B* to receiver *A*
$\omega_{c(O)A}$	weigth for group intention impact on *A* 's intention for *O*
$\omega_{b(O)A}$	weigth for own emotion impact on *A* 's intention for *O*

This generalisation of the existing agent-based contagion models is not exactly a behavioural model, as the states indicated by the values $q_{SA}(t)$ still have to be multiplied by the expression factor ε_{SA} to obtain the behavioural states that are observed by the other agents. When for these states the notation $e_{SA}(t)$ is used, then the model can be reformulated as a model in terms of the behavioural output states $e_{SA}(t)$. Moreover, it is also possible to model the interaction between agents via aggregated input or sensor states $s_{g(S)A}(t)$ for *S*. Then, assuming that time taken by interaction is neglectable compared to the internal processes, the following reformulation can be done. First the following aggregated sensoring state is modelled:

$$s_{g(S)A}(t) = \sum_{B\neq A} \alpha_{SBA} \cdot e_{SB}(t) \,/\, (\sum_{B\neq A} \varepsilon_{SB} \cdot \alpha_{SBA})$$

Note that by (3) in fact $s_{g(S)A}(t) = q_{SA}{}^{*}(t)$. Next the model for $e_{SA}(t)$ can be found :

$$\begin{aligned} e_{SA}(t + \Delta t) &= \varepsilon_{SA}\, q_{SA}(t + \Delta t) \\ &= \varepsilon_{SA}\, q_{SA}(t) + \varepsilon_{SA}\, \gamma_{SA}\, \mathbf{c}(q_{SA}{}^{*}(t), q_{SA}(t))\, \Delta t \\ &= e_{SA}(t) + \varepsilon_{SA}\, \gamma_{SA}\, \mathbf{c}(s_{g(S)A}(t), e_{SA}(t)/\varepsilon_{SA})\, \Delta t \end{aligned}$$

Thus in a straightforward manner the following behavioural model is obtained as a generalisation of the existing agent-based emotion contagion models:

$$s_{g(S)A}(t) = \sum_{B \neq A} \alpha_{SBA} \cdot e_{SB}(t) \;/\; (\sum_{B \neq A} \varepsilon_{SB} \cdot \alpha_{SBA}) \qquad (5)$$

$$e_{SA}(t + \Delta t) = e_{SA}(t) + \varepsilon_{SA}\, \gamma_{SA}\, \mathbf{c}(s_{g(S)A}(t), e_{SA}(t)/\varepsilon_{SA})\, \Delta t \qquad (6)$$

3.2 The behavioural agent-based group decision model BAM

To obtain an agent-based social level model for group decision making, the abstract agent-based model for contagion described above for any decision option *O* has been applied to both the emotion states *S* and intention or choice tendency states *S'* for *O*. In addition, an interplay between the two types of states has been modelled. To incorporate such an interaction (loosely inspired by Damasio's principle of somatic marking; cf. [4], [7], the basic model was extended as follows: to update $q_{SA}(t)$ for an intention state *S* relating to an option *O*, both the intention states of others for *O* and the $q_{S'A}(t)$ values for the emotion state *S'* for *O* are taken into account. Note that in this model a fixed set of options was assumed, that all are considered. The emotion and choice tendency states *S* and *S'* for option *O* are denoted by *b(O)* and *c(O)*, respectively:

Expressed level of emotion for option *O* of person *A*: $e_{b(O)A}(t)$
Expressed level of choice tendency or intention for *O* of person *A*: $e_{c(O)A}(t)$

The combination of the own (positive) emotion level and the rest of the group's aggregated choice tendency for option *O* is made by a weighted average of the two:

$$\begin{aligned} q_{c(O)A}{}^{**}(t) &= (\omega_{c(O)A}/\omega_{OA})\, q_{c(O)A}{}^{*}(t) + (\omega_{b(O)A}/\omega_{OA})\, q_{b(O)A}(t) \\ &= (\omega_{c(O)A}/\omega_{OA})\, s_{c(O)A}(t) + (\omega_{b(O)A}/\omega_{OA})\, e_{b(O)A}(t) / \varepsilon_{SA} \\ \gamma_{c(O)A}{}^{*} &= \omega_{OA}\, \gamma_{c(O)A} \end{aligned}$$

where $\omega_{c(O)A}$ and $\omega_{b(O)A}$ are the weights for the contributions of the group choice tendency impact and the own emotion impact on the choice tendency of *A* for *O*, respectively, and $\omega_{OA} = \omega_{c(O)A} + \omega_{b(O)A}$. Then the behavioural agent-based model for interacting emotion and intention (choice tendency) contagion expressed in numerical format becomes:

$$s_{g(b(O))A}(t) = \sum_{B \neq A} \alpha_{b(O)BA} \cdot e_{b(O)B}(t) \;/\; (\sum_{B \neq A} \varepsilon_{b(O)B} \cdot \alpha_{b(O)BA}) \qquad (7)$$

$$e_{b(O)A}(t + \Delta t) = e_{b(O)A}(t) + \varepsilon_{b(O)A}\, \gamma_{b(O)A}\, \mathbf{c}(s_{g(b((O))A}(t), e_{b(O)A}(t)/\varepsilon_{b(O)A})\, \Delta t \qquad (8)$$

with as an example

$$\mathbf{c}(X, Y) = \eta_{b(O)A} \cdot [\beta_{b(O)A} \cdot (1 - (1-X) \cdot (1-Y)) + (1 - \beta_{b(O)A}) \cdot XY] + (1 - \eta_{b(O)A}) \cdot X - Y$$

$$s_{g(c(O))A}(t) = \sum_{B \neq A} \alpha_{c(O)BA} \cdot e_{c(O)B}(t) \;/\; (\sum_{B \neq A} \varepsilon_{c(O)B} \cdot \alpha_{c(O)BA}) \qquad (9)$$

$$e_{c(O)A}(t + \Delta t) = e_{c(O)A}(t) + \varepsilon_{c(O)A}\, \omega_{OA}\, \gamma_{c(O)A}\, \mathbf{d}((\omega_{c(O)A}/\omega_{OA})\, s_{g(c(O))A}(t) + (\omega_{b(O)A}/\omega_{OA})\, e_{b(O)A}(t)/\varepsilon_{b(O)A}, e_{c(O)A}(t)/\varepsilon_{c(O)A})\, \Delta t \qquad (10)$$

with as an example

$$\mathbf{d}(X, Y) = \eta_{c(O)A} \cdot [\beta_{c(O)A} \cdot (1 - (1-X) \cdot (1-Y)) + (1 - \beta_{c(O)A}) \cdot XY] + (1 - \eta_{c(O)A}) \cdot X - Y$$

Hybrid Specification of the Behavioural Agent Model BAM

To be able to relate this model expressed by difference equations to the internal agent model **IAM**, the model is expressed in a hybrid logical/numerical format in a straightforward manner in the following manner, using atoms has_value(x, V) with x a variable name and V a value, thus obtaining the behavioural agent model **BAM**. Here s(g(b((O)), A), s(g(c((O)), A), e(b(O), A) and e(c(O), A) for options O are names of the specific variables involved.

BP1 Generating a body state
has_value(s(g(b(O)), A), V_1) & has_value(e(b(O), A), V)
→ has_value(e(b(O), A), V + $\varepsilon_{b(O)A}\, \gamma_{b(O)A}\, \mathbf{c}(V_1, V/\varepsilon_{b(O)A})\, \Delta t$)

BP2 Generating an option choice intention
has_value(s(g(c(O)), A), V_1) & has_value(e(b(O), A), V_2) & has_value(e(c(O), A), V)
→ has_value(e(c(O), A), V + $\varepsilon_{c(O)A}\, \omega_{OA}\, \gamma_{c(O)A}\, \mathbf{d}((\omega_{c(O)A}/\omega_{OA})\, V_1 + (\omega_{b(O)A}/\omega_{OA})\, V_2/\varepsilon_{b(O)A}, V/\varepsilon_{c(O)A})\, \Delta t$)

BTP Sensing aggregated group members' bodily responses and intentions
$\bigwedge_{B \neq A}$ has_value(e(S, B), V_B) → has_value(s(g(S), A), $\Sigma_{B \neq A}\, \alpha_{SBA} V_B / \Sigma_{B \neq A}\, \alpha_{SBA} \varepsilon_{SB}$)

In Section 5 the behavioural agent model **BAM** is related to the internal agent model **IAM** described in Section 2. This relation goes via the abstracted (from **IAM**) behavioural agent model **ABAM** introduced in Section 4.

4 Abstracting Internal Model IAM to Behavioural Agent Model ABAM

In this section two methods for behavioural model abstraction are described. First, in section 4.1 a method for abstraction of cognitive specifications by elimination of sensory representation and preparation atoms is presented. Then, in section 4.2 an abstraction technique is described which addresses elimination of internal loops based on equilibria. Both abstraction techniques proposed are compared and evaluated in Section 4.3.

4.1 Abstraction by elimination of sensory representation and preparation atoms

First, in this section, from the model **IAM** by a systematic transformation, an abstracted behavioural agent model **ABAM** is obtained. In Section 5 the two behavioural agent models **ABAM** and **BAM** will be related. In [14] an automated abstraction transformation is described from a non-cyclic, stratified internal agent model to a behavioural agent model. As in the current situation the internal agent model is not assumed to be noncyclic, this existing transformation cannot be applied. In particular, for the internal agent model considered as a case in Section 2 the properties IP2* and IP3* are cyclic by themselves (recursive). Moreover, the as-if body loop described by properties IP2* and IP5* is another cycle. Therefore, the transformation introduced here exploits a different approach. The two main steps in this transformation are: elimination of sensory representation atoms, and elimination of preparation atoms (see also Fig. 4).

1. Elimination of sensory representation atoms

It is assumed that sensory representation atoms may be affected by sensor atoms, or by preparation atoms. These two cases are addressed as follows

a) Replacing sensory representation atoms by sensor atoms

- Based on a property SS(A, S, V) $\rightarrow$ SRS(A, S, υV) (such as IP1*), replace atoms SRS(A, S, V) in an antecedent (for example, in IP2* and IP3*) by SS(A, S, V/υ).

b) Replacing sensory representation atoms by preparation atoms

- Based on an as-if body loop property PS(A, S, V) $\rightarrow$ SRS(A, S, ωV) (such as IP5*), replace atoms SRS(A, S, V) in an antecedent (for example, in IP2*) by PS(A, b(O), V/ω).

Note that this transformation step is similar to the principle exploited in [14]. It may introduce new occurrences of preparation atoms; therefore it should preceed the step to eliminate preparation atoms. In the case study this transformation step provides the following transformed properties (replacing IP1*, IP2*, IP3*, and IP5*; see also Fig. 4):

IP2 Preparing for a body state**

SS(A, c, V_1/υ_{cA}) & SS(A, g(b(O)), $V_2/\upsilon_{g(b(O))A}$) & PS(A, b(O), V_3/ω_{0OA}) & PS(A, b(O), V)
$\rightarrow$ PS(A, b(O), V + $\lambda_{b(O)A}$ **g**($\omega_{1OA}V_1$, $\omega_{2OA}V_2$, $\omega_{3OA}V_3$, V) Δt)

IP3 Preparing for an option choice**

SS(A, c, V_1/υ_{cA}) & SS(A, g(c(O)), $V_2/\upsilon_{g(c(O))A}$)) & PS(A, b(O), V_3) & PS(A, c(O), V)
$\rightarrow$ PS(A, c(O), V + $\lambda_{c(O)A}$ **h**($\omega_{4OA}V_1$, $\omega_{5OA}V_2$, $\omega_{6OA}V_3$, V) Δt)

2. Elimination of preparation atoms

Preparation atoms in principle occur both in antecedents and consequents. This makes it impossible to apply the principle exploited in [14]. However, preparation states often have a direct relationship to effector states. This is exploited in the second tranasformation step.

- Based on a property PS(A, S, V) $\rightarrow$ ES(A, S, ζV) (such as in IP4*), replace each atom PS(A, S, V) in an antecedent or consequent by ES(A, S, ζV).

In the case study this transformation step provides the following transformed properties (replacing IP2**, IP3**, and IP4*; see also Fig. 4):

IP2 Preparing for a body state**

SS(A, c, V_1/υ_{cA}) & SS(A, g(b(O)), $V_2/\upsilon_{g(b(O))A}$) & ES(A, b(O), $\zeta_{b(O)A}$ V_3/ω_{0OA}) & ES(A, b(O), $\zeta_{b(O)A}$ V)
$\rightarrow$ ES(A, b(O), $\zeta_{b(O)A}$ V + $\zeta_{b(O)A}$ $\lambda_{b(O)A}$ **g**($\omega_{1OA}V_1$, $\omega_{2OA}V_2$, $\omega_{3OA}V_3$, V) Δt)

IP3 Preparing for an option choice**

SS(A, c, V_1/υ_{cA}) & SS(A, g(c(O)), $V_2/\upsilon_{g(c(O))A}$)) & ES(A, b(O), $\zeta_{b(O)A}$ V_3) & ES(A, c(O), $\zeta_{c(O)A}$ V)
$\rightarrow$ ES(A, c(O), $\zeta_{c(O)A}$ V + $\zeta_{c(O)A}$ $\lambda_{c(O)A}$ **h**($\omega_{4OA}V_1$, $\omega_{5OA}V_2$, $\omega_{6OA}V_3$, V) Δt)

By renaming V_1/υ_{cA} to V_1, $V_2/\upsilon_{g(b(O))A}$ to V_2 , $\zeta_{b(O)A}\, V_3/\omega_{0OA}$ to V_3, $\zeta_{b(O)A}\, V$ to V (in IP2**), resp. $V_2/\upsilon_{g(c(O))A}$ to V_2, $\zeta_{b(O)A}\, V_3$ to V_3, and $\zeta_{c(O)A}\, V$ to V (in IP3**), the following is obtained.

IP2* Preparing for a body state**

SS(A, c, V_1) & SS(A, g(b(O)), V_2) & ES(A, b(O), V_3) & ES(A, b(O), V)
$\rightarrow$ ES(A, b(O), $V + \zeta_{b(O)A}\, \lambda_{b(O)A}\, \mathbf{g}(\omega_{1OA}\upsilon_{cA}\, V_1,\ \omega_{2OA}\upsilon_{g(b(O))A}\, V_2,\ \omega_{3OA}\omega_{0OA}\, V_3/\zeta_{b(O)A},\ V/\zeta_{b(O)A})\, \Delta t$)

IP3* Preparing for an option choice**

SS(A, c, V_1) & SS(A, g(c(O)), V_2) & ES(A, b(O), V_3) & ES(A, c(O), V)
$\rightarrow$ ES(A, c(O), $V + \zeta_{c(O)A}\, \lambda_{c(O)A}\, \mathbf{h}(\omega_{4OA}\upsilon_{cA}\, V_1,\ \omega_{5OA}\upsilon_{g(c(O))A}\, V_2,\ \omega_{6OA} V_3/\zeta_{b(O)A},\ V/\zeta_{c(O)A})\, \Delta t$)

Based on these properties derived from the internal agent model **IAM** the hybrid specification of the abstracted behavioural model **ABAM** can be defined.

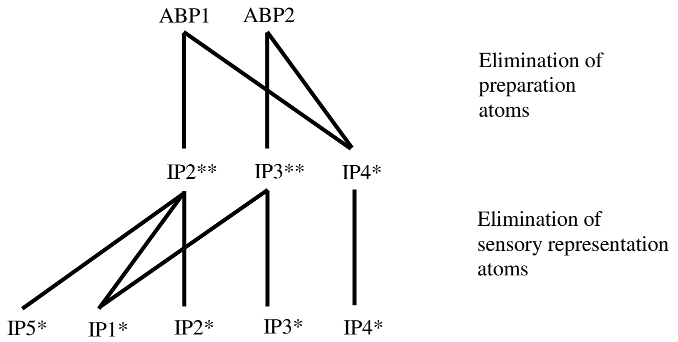

Fig. 4. Abstraction steps from internal model to abstracted behavioural model: **IAM** ⊢ **ABAM**

Hybrid Specification of the Abstracted Behavioural Agent Model ABAM

Note that in IP2*** V_2 and V have the same value, so a slight further simplification can be made by replacing V_3 by V. After renaming of the variables according to

ABP1			ABP2		
V_1	$\rightarrow$	W_0	V_1	$\rightarrow$	W_0
V_2	$\rightarrow$	W_1	V_2	$\rightarrow$	W_1
V_3	$\rightarrow$	W	V_3	$\rightarrow$	W_2
V	$\rightarrow$	W	V	$\rightarrow$	W

the following abstracted behavioural model **ABAM** for agent A is obtained:

ABP1 Generating a body state

SS(A, c, W_0) & SS(A, g(b(O)), W_1) & ES(A, b(O), W)
$\rightarrow$ ES(A, b(O), $W + \zeta_{b(O)A}\, \lambda_{b(O)A}\, \mathbf{g}(\omega_{1OA}\upsilon_{cA}\, W_0,\ \omega_{2OA}\upsilon_{g(b(O))A}\, W_1,\ \omega_{3OA}\omega_{0OA}\, W/\zeta_{b(O)A},\ W/\zeta_{b(O)A})\, \Delta t$)

ABP2 Generating an option choice intention

SS(A, c, W_0) & SS(A, g(c(O)), W_1) & ES(A, b(O), W_2) & ES(A, c(O), W)
$\rightarrow$ ES(A, c(O), $W + \zeta_{c(O)A}\, \lambda_{c(O)A}\, \mathbf{h}(\omega_{4OA}\upsilon_{cA}\, W_0,\ \omega_{5OA}\upsilon_{g(c(O))A}\, W_1,\ \omega_{6OA}\, W_2/\zeta_{b(O)A},\ W/\zeta_{c(O)A})\, \Delta t$)

ITP Sensing aggregated group members' bodily responses and intentions

$\wedge_{B\neq A}$ ES(B, S, V_B) $\rightarrow$ SS(A, g(S), $\Sigma_{B\neq A}\ \alpha_{SBA}V_B / \Sigma_{B\neq A}\ \alpha_{SBA}\zeta_{SB}$)

where S has instances b(O), c(O) for options O.

Note that as all steps made are logical derivations it holds **IAM** ⊢ **ABAM**. In particular the following logical implications are valid (depicted hierarchically in Fig. 4):

IP1* & IP5* & IP2* $\Rightarrow$ IP2**	IP4* & IP2** $\Rightarrow$ ABP1
IP1* & IP3* $\Rightarrow$ IP3**	IP4* & IP3** $\Rightarrow$ ABP2

Assumptions underlying the transformation

The transformation as described is based on the following of assumptions:

- Sensory representation states are affected (only) by sensor states and/or preparation states
- Preparation atoms have a direct relationship with effector atoms; there are no other ways to generate effector states than via preparation states
- The time delays for the interaction from the effector state of one agent to the sensor state of the same or another agent are small so that they can be neglected compared to the internal time delays
- The internal time delays from sensor state to sensory representation state and from preparation state to effector state within an agent are small so that they can be neglected compared to the internal time delays from sensory representation to preparation states

The transformation can be applied to any internal agent model satisfying these assumptions.

The proposed abstraction procedure has been implemented in Java. The automated procedure requires as input a text file with a specification of an internal agent model and generates a text file with the corresponding abstracted behavioural model as output. The computational complexity of the procedure is O(|M|*|N| + |L|*|S|), where M is the set of srs atoms in the IAM specification, N is the set of the srs state generation properties in the specification, L is the set of the preparation atoms and srs atoms in the loops in the specification, and S is the set of the effector state generation properties in the specification.

Using the automated procedure the hybrid specification of ABAM has been obtained. With this specification simulation has been performed with the values of parameters as described in Table 3. The obtained curves for ES(A, c(O)) and ES(A, b(O)) are the same as the curves depicted in Fig. 3 for the IAM model. This outcome confirms that both the ABAM and IAM models generate the same behavioural traces and that the abstraction transformation is correct.

4.2 Abstraction by elimination of loops based on equilibria

In more complex internal models internal cycles may occur, for example, adaptive models based on some internal feedback or reinforcement principle, or models in which cognitive states affect affective states and these affective states in turn affect the same cognitive states. A specific example of this is the use of an as if body loop, as also occurs in the internal agent model **IAM** between SRS(A, b(O)) and PS(A, b(O)); see Figure 1. By the abstraction method described in Section 4.1 above this loop was eliminated for this case by replacing SRS(A, b(O)) by a previous state of PS(A, b(O)). However, loops can occur in internal agent models in different ways, and different methods can be developed to eliminate them. In this section an approach is described that addresses abstraction of such cyclic internal structures based on equilibria. The idea is that when such loops are processed with high speed, they will reach an equilibrium fast, and for the rest of the internal agent model, the equilibrium value can be used instead of the intermediate values. After the elimination of cyclic structures from an internal specification in this way, standard techniques (such as [14], [13]) can be applied to obtain an abstracted behavioural specification.

Assumptions underlying the loop elimination approach

1. Internal dynamics develop an order of magnitude *faster* than the dynamics of the world external to the agent.
2. Loops are *internal* in the sense that they do not involve the agent's output states.
3. Different loops have *limited mutual interaction*; in particular, loops may contain internal loops; loops may interact in couples; interacting couples of loops may interact with each other by forming noncyclic interaction chains.
4. For static input information any internal loop reaches an *equilibrium state* for this input information.
5. It can be *specified* how the value for this equilibrium state of a given loop depends on the input values for the loop.
6. In the agent model the loop can be *replaced* by the equilibrium specification of 4.

The idea is that when these assumptions are fulfilled, for each received input, before new input information arrives, the agent computes its internal equilibrium states, and based on that determines its behaviour.

Loop elimination setup

To address the loop elimination process, the following representation of a loop is assumed

$$\text{has\_value}(u, V_1)\ \&\ \text{has\_value}(p, V_2) \rightarrow_{d,d,1,1} \text{has\_value}(p, V_2 + f(V_1, V_2)d) \qquad (1)$$

Here u is the name of an input variable, p of the loop variable, t is a variable of sort TIME, and $f(V_1, V_2)$ is a function combining the input value with the current value for p, d is the duration of the time step.

Note that an equilibrium state for a given input value V_1 in (1) is a value V_2 for p such that $f(V_1, V_2) = 0$. A specification of how V_2 depends on V_1 is a function g such that $f(V_1, g(V_1)) = 0$. Note that the latter expression is an implicit function definition, and under mild conditions (e.g., $\partial f(V_1, V_2)/\partial V_2 \neq 0$, or strict monotonicity of the function $V_2 \rightarrow f(V_1, V_2)$) the Implicit Function Theorem within calculus guarantees the existence (mathematically) of such a function g. However, knowing such an existence in the mathematical sense is not sufficient to obtain a procedure to calculate the value of g for any given input value V_1. When such a specification of g is obtained, the loop representation shown above can be transformed into:

$$\text{has\_value}(u, V1) \rightarrow_{D,D,1,1} \text{has\_value}(p, g(V1))$$

where D is chosen as a timing parameter for the process of approximating the equilibrium value up to some accuracy level.

To obtain a procedure to compute g based on a given function f, two options are available. The first option is, for a given input V_1 by numerical approximation of the solution V_2 of the equation $f(V_1, V_2) = 0$. This method can always be applied and is not difficult to implement using very efficient standard procedures in numerical analysis, taking only a few steps to come to high precision. The second option, elaborated further below is by symbolically solving the equation $f(V_1, V_2) = 0$ depending on V_1 in order to obtain an explicit algebraic expression for the function g. This option can be used successfully when the symbolic expression for the function f is not too complex; however, it is still possible to have it nonlinear.

In various agent models involving such loops a threshold function is used to keep the combined values within a certain interval, for example [$0, 1$]. A threshold function can be defined, for example, in three ways:

(1) as a piecewise constant function, jumping from 0 to 1 at some threshold value
(2) by a logistic function with format $1/(1+exp(-\sigma(V_1+ V_2-\tau))$, or
(3) by a function $\beta(1-(1- V_1)(1- V_2)) + (1-\beta) V_1 V_2$.

The first option provides a discontinuous function, which is not desirable for analysis. The third format is used here, since it provides a continuous function, can be used for explicit symbolic manipulation, and is effective as a way of keeping the values between bounds. Note that this function can be written as a linear function of $\mathbf{V_2}$ with coefficients in V_1 as follows:

$$f(V_1, V_2) = \beta(1-(1- V_1)(1- \mathbf{V_2})) + (1-\beta) V_1 \mathbf{V_2} - \mathbf{V_2} = - [(1- \beta)(1- V_1) +\beta V_1]\, \mathbf{V_2} + \beta V_1$$

From this form it follows that

$$\partial f(V_1, \mathbf{V_2}) /\partial \mathbf{V_2} = \partial -[[(1- \beta)(1- V_1) +\beta V_1]\, \mathbf{V_2} + \beta V_1]/\partial \mathbf{V_2} = - [(1- \beta)(1- V_1) +\beta V_1] \leq 0$$

This is only 0 for extreme cases: $\beta = 0$ and $V_1 = 1$ or $\beta = 1$ and $V_1 = 0$. So, for the general case $V_2 \rightarrow f(V_1, V_2)$ is strictly monotonically decreasing, which shows that it fulfills the conditions of the Implicit Function Theorem, thus guaranteeing the existence of a function g as desired.

Obtaining the equilibrium specification: single loop case

Using the above expression, the equation $f(V_1, V_2) = 0$ can be easily solved symbolically: $\mathbf{V_2} = \beta V_1 / [(1-\beta)(1- V_1) +\beta V_1]$. This provides an explicit symbolic definition of the function g: $g(V_1) = \mathbf{V_2} = \beta V_1 / [(1- \beta)(1- V_1) +\beta V_1]$. For each β with $0<\beta <1$ this g is a strictly monotonically increasing function with $g(0) = 0$ and $g(1) = 1$. A few cases for specific values of the parameter β are as follows: (i) $\beta = 0$, $g(V_1) = 0$; (ii) $\beta = 0.5$, $g(V_1) = V_1$; (iii) $\beta =1$, $g(V_1) = 1$.

Obtaining the equilibrium specification: interacting loops case

Interaction between two loops occurs when the outcome of one loop is used as (part of) input in another loop; it may occur in two forms: monodirectional or bidirectional. In the monodirectional case the previously described method can be used in a straightforward manner one-by-one for each of the loops, first for the loop providing input for the other loop.

The bidirectional case requires more elaboration. First it is assumed that the input from the other loop is combined with the externally provided input as follows: $v_1 = \lambda_1(u_1)p_2 + \mu_1(u_1)$ and $v_2 = \lambda_2(u_2)p_1 + \mu_2(u_2)$ where u_i denotes the external input (what was indicated above by V_2) for a loop i, p_i the state of the loop (what was indicated above by V_2), and λ_i and μ_i are functions of the external input u_i. Special cases are:

(1) $\lambda_1(u_1) = w_1$ and $\mu_1(u_1) = w_2 u_1$, in which case they are combined according to a weighted sum,
(2) $\lambda_1(u_1) = u_1$ and $\mu_1(u_1) = 0$, in which case p_2 acts as a modifier of the external input u_1; e.g., an estimated degree of reliability of the incoming information
(3) $\lambda_1(u_1) = - [(1- \beta)(1-u_1) +\beta u_1]$ and $\mu_1(u_1) =\beta u_1$ which provides the combination function used in $f(V1, V2)$ above.

To solve the two coupled equations for this case a simplified notation is used: $v_1 = \lambda_1 p_2 + \mu_1$ and $v_2 = \lambda_2 p_1 + \mu_2$.

$$[(1- \beta_1)(1-(\lambda_1 \mathbf{p_2} + \mu_1)) +\beta_1(\lambda_1 \mathbf{p_2} + \mu_1)]\, \mathbf{p_1} = \beta_1(\lambda_1 \mathbf{p_2} + \mu_1)$$
$$[(1- \beta_2)(1-(\lambda_2 \mathbf{p_1} + \mu_2)) +\beta_2(\lambda_2 \mathbf{p_1} + \mu_2)]\, \mathbf{p_2} = \beta_2(\lambda_2 \mathbf{p_1} + \mu_2)$$

These equations can be rewritten as follows:

$$(2\beta_1-1)\lambda_1 \mathbf{p_1} \mathbf{p_2} +[(1- \beta_1)(1- \mu_1) +\beta_1\mu_1]\, \mathbf{p_1} = \beta_1(\lambda_1 \mathbf{p_2} + \mu_1)$$
$$(2\beta_2-1)\lambda_2 \mathbf{p_1} \mathbf{p_2} +[(1- \beta_2)(1- \mu_2) +\beta_2\mu_2]\, \mathbf{p_2} = \beta_2(\lambda_2 \mathbf{p_1} + \mu_2)$$

Multiplying the first equation by $(2\beta_2-1)\lambda_2$ and the second by $(2\beta_1-1)\lambda_1$ and subtracting them from each other provides one equation that can be rewritten into a form that provides an explicit expression of $\mathbf{p_2}$ in terms of $\mathbf{p_1}$:

$$p_2 = [\,(2\beta_2\text{-}1)\lambda_2\,[\,(1\text{-}\beta_1)(1\text{-}\mu_1) + \beta_1\mu_1 + (2\beta_1\text{-}1)\lambda_1\,\beta_2\lambda_2\,]\,p_1 + (2\beta_1\text{-}1)\lambda_1\,\beta_2\,\mu_2 - (2\beta_2\text{-}1)\lambda_2\,\beta_1\mu_1)\,]\,/\,[\,(2\beta_1\text{-}1)\lambda_1\,[\,(1\text{-}\beta_2)(1\text{-}\mu_2) + \beta_2\mu_2 + (2\beta_2\text{-}1)\lambda_2\,\beta_1\lambda_1]]$$

Filling the expression for p_2 in the second equation provides one equation in p_1:

$$[(1\text{-}\beta_2)(1\text{-}(\lambda_2 p_1 + \mu_2)) + \beta_2(\lambda_2 p_1 + \mu_2)\,]\;[\,(2\beta_2\text{-}1)\lambda_2\,[\,(1\text{-}\beta_1)(1\text{-}\mu_1) + \beta_1\mu_1 + (2\beta_1\text{-}1)\lambda_1\,\beta_2\lambda_2\,]\,p_1 + (2\beta_1\text{-}1)\lambda_1\,\beta_2\,\mu_2 - (2\beta_2\text{-}1)\lambda_2\,\beta_1\mu_1)\,]\,/\,[\,(2\beta_1\text{-}1)\lambda_1\,[\,(1\text{-}\beta_2)(1\text{-}\mu_2) + \beta_2\mu_2 + (2\beta_2\text{-}1)\lambda_2\,\beta_1\lambda_1]] = \beta_2(\lambda_2 p_1 + \mu_2)$$

By solving this equation an explicit symbolic expression is obtained for p_1 and for p_2.

The two loops in the model **IAM** interact monodirectionally, thus both these loops can be eliminated as shown above for the single loop case. As the result of the loop abstraction process, an intermediate specification **IS** is obtained, which can be further abstracted to the behavioural level using standard abstraction techniques for non-cyclic specifications (such as the one from [14]).

First, the as-if body loop specified by the properties IP5* and IP2* is eliminated. To this end, the equation for the equilibrium state for emotion is obtained as:

$$fc(\omega_{1OA}V_1,\ \omega_{2OA}V_2,\ \omega_{0OA}\ \omega_{3OA}V_3) - V_3 = 0$$

Here V_1 is the sensory representation state value for option c; V_2 is the sensory representation state value for g(b(O)), and V_3 is the value for the preparation state for b(O); ω_{0OA}, ω_{1OA}, ω_{2OA}, ω_{3OA} are parameters defined in section 2.
By solving this equation, the expression for V_3 is found:

$$V_3 = fa(V_1, V_2) = \beta\,((1\text{-}\omega_{1OA}V_1)(1\text{-}\omega_{2OA}V_2) - 1)/(\beta\,\omega_{0OA}\,\omega_{3OA}(1\text{-}\omega_{1OA}V_1)(1\text{-}\omega_{2OA}V_2) + (1\text{-}\beta)\omega_{0OA}\,\omega_{3OA}\omega_{1OA}\omega_{2OA}\,V_1V_2\text{-}1)$$

Thus, the first property in **IS** is the following:

ISP1 Preparing for an emotion expressed in a body state
SRS(A, c, V_1) & SRS(A, g(b(O)), V_2) → PS(A, b(O), fa(V_1, V_2))

Then the second loop, involving properties IP3* and IP2* in the model **IAM**, is eliminated by determining the equilibrium for the intention state c(O):

$$V = fc(\omega_{4OA}V_1,\ \omega_{5OA}V_2,\ \omega_{6OA}V_3)$$

Here V_1 is the sensory representation state value for option c; V_2 is the sensory representation state value for g(c(O)), and V_3 is the value for the preparation state for b(O); ω_{4OA}, ω_{5OA}, ω_{6OA} are parameters defined in section 2.

Thus, the second property in **IS** is the following:

ISP2 Preparing for an option choice
SRS(A, c, V_1) & SRS(A, g(c(O)), V_2)) & PS(A, b(O), V_3) → PS(A, c(O), fc($\omega_{4OA}V_1$, $\omega_{5OA}V_2$, $\omega_{6OA}V_3$))

The properties IP1* and IP4* from IAM are not involved in any loops, therefore adopted in IS without any modifications:

ISP3 From sensor states to sensory representations
SS(A, S, V) → SRS(A, S, υ_{SA}V)
where S has instances c, g(c(O)) and g(b(O)) for options O.

ISP4 From preparation to effector state
PS(A, S, V) → ES(A, S, ζ_{SA} V)
where S has instances b(O) and c(O) for options O.

Then, by applying the abstraction algorithm from [14] the following behavioural specification **AEBAM** is obtained from **IS**:

ALBP1 Generating a body state
SS(A, c, V_1) & SS(A, g(b(O)), V_2) → ES(A, b(O), ζ_{SA} fa(υ_{OA} V_1, $\upsilon_{g(b(O))A}$ V_2))

ALBP2 Generating an option choice intention
SS(A, c, V_1) & SS(A, g(c(O)), V_2)) & SS(A, g(b(O)), V_3)

$\rightarrow$ ES(A, c(O)), ζ_{SA} fc(υ_{OA} $\omega_{4OA}V_1$, $\upsilon_{g(c(O)A}$ $\omega_{5OA}V_2$, ω_{6OA} fa($\upsilon_{OA}V_1$, $\upsilon_{g(b(O)A}V_3$)))

ITP Sensing aggregated group members' bodily responses and intentions

$\bigwedge_{B \neq A}$ ES(B, S, V_B) $\rightarrow$ SS(A, g(S), $\Sigma_{B\neq A}$ $\alpha_{SBA}V_B$ / $\Sigma_{B\neq A}$ $\alpha_{SBA}\zeta_{SB}$)

where S has instances b(O), c(O) for options O.

In the following section the abstraction methods from this and the previous section 4.1, and the behavioural models **ABM** and **ALBM** produced by these methods are compared.

4.3 Comparison of the abstraction approaches

The abstracted models **ABAM** and **AEBAM** are compared by evaluating to which extent the behaviour produced by each of them approximates the behaviour generated by the internal model **IAM**. To this end, the root mean squared error measure is used:

$$err = \sqrt{\frac{\sum_{i=1}^{N}(f_i - y_i)^2}{N}}$$

where N is the number of time points, f_i is the value of an output of an abstracted model at time point t_i, and y_i is the value of the corresponding output of the model **IAM**.

Note that the error estimate for the model **AEBAM** depends on the choice of the timing parameter D, used to compensate for the difference in speeds of internal and external dynamics. According to the assumption 1 for **AEBAM**, internal dynamics develop an order of magnitude faster than the dynamics of the world external to the agent. Thus, for the model evaluation two values of D were chosen: D=10 and D=30. The root mean squared error estimates for both abstracted models are provided in Table 5; the dynamics of the outputs of the models being compared are depicted in Fig.5.

Table 5 The root mean squared errors for the model outputs ES(b(O)) and ES(c(O)) in settings 1 and 2

Model	**ES(b(O))**	**ES(c(O))**
ABAM	0.017	0.026
AEBAM (D=10/D=30)	0.033 / 0.033	0.026 / 0.025

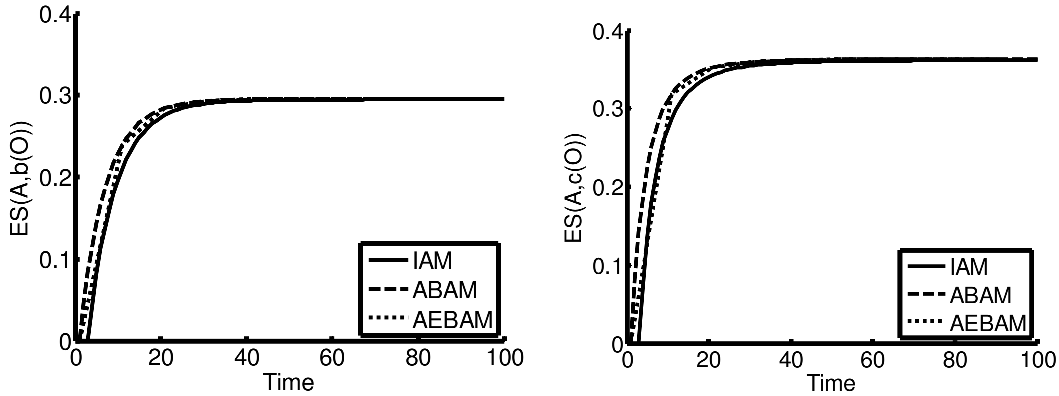

Fig. 5. Dynamics of the output state ES(A, b(O)) (left) and of the output state ES(A, c(O)) (right) generated by the models IAM, ABAM and AEBAM

As one can see from the results, the **ABAM** model provides a slightly better approximation of the agent's behaviour than the **AEBAM** model. Partially this can be explained by the fact that **AEBAM** abstracts away the details of the dynamics of cyclic processes by replacing them with equilibrium states. This is particularly evident from the results for ES(b(O)), which is directly influenced by an as if body loop. However, the abstraction method used for **AEBAM** has a wider applicability than the one used for **ABAM**. In particular, the latter can handle models with simple single loops only, whereas the former can also be used to abstract models with interacting loops.

As one can see from Table 5, there is almost no difference between the outcomes for D=10 and for D=30. However, for D < 10 the error estimates are higher. Thus, a factor of 10 difference assumed between the speeds of internal and external processes is sufficient to ensure the required approximation precision.

Both abstracted models have comparable computational complexity: at every time point to calculate each output state of the model a corresponding single rule is used, defining a mapping of input states to the output state.

5 Relating the Behavioural Agent Models BAM and ABAM

In this section the given behavioural agent model **BAM** described in Section 3 is related to the behavioural agent model **ABAM** obtained from the internal agent model **IAM** by the abstraction process described in Section 4.

5.1 Ontology mappings and interpretation mappings

In this section the notion of interpretation mapping induced by an ontology mapping is briefly introduced (e.g., [8], pp. 201-263; [15]).

Basic ontology mapping π

For example, suppose in some ontology state properties a_2 and b_2 are given:

a_2 it is cold and dry b_2 it is warm

Moreover, suppose that in another ontology, state properties a_1 and b_1 are given:

a_1 the molecules have a low level of movement and few molecules are present
b_1 the molecules have a high level of movement

Then the state properties a_2 and b_2 can be related to state properties a_1 and b_1 respectively by an ontology mapping π as follows:

$\pi(a_2) = a_1$ $\pi(b_2) = b_1$

A given basic ontology mapping π can be extended to a mapping of (more complex) state properties in a straightforward, compositional manner, based on rules as:

$\pi(A \,\&\, B) = \pi(A) \,\&\, \pi(B)$ $\pi(A \vee B) = \pi(A) \vee \pi(B)$
$\pi(A \Rightarrow B) = \pi(A) \Rightarrow \pi(B)$ $\pi(\neg A) = \neg \pi(A)$

Interpretation mapping π^* for dynamic properties

Using compositionality a basic ontology mapping used above can be extended to an interpretation mapping for temporal expressions. As an example, when $\pi(a_2) = a_1$, $\pi(b_2) = b_1$, then this induces a mapping π^* from dynamic property $a_2 \rightarrow b_2$ to $a_1 \rightarrow b_1$ as follows:

$\pi^*(a_2 \rightarrow b_2) = \pi^*(a_2) \rightarrow \pi^*(b_2) = \pi(a_2) \rightarrow \pi(b_2) = a_1 \rightarrow b_1$

In a similar manner by compositionality a mapping for more complex temporal predicate logical relationships A and B can be defined, using

$\pi^*(A \,\&\, B) = \pi^*(A) \,\&\, \pi^*(B)$ $\pi^*(A \vee B) = \pi^*(A) \vee \pi^*(B)$
$\pi^*(A \Rightarrow B) = \pi^*(A) \Rightarrow \pi^*(B)$ $\pi^*(\neg A) = \neg \pi^*(A)$
$\pi^*(\forall T\, A) = \forall T\, \pi^*(A)$ $\pi^*(\exists T\, A) = \exists T\, \pi^*(A)$

5.2 Mapping the given behavioural model BAM onto the abstracted ABAM

First, consider how a basic ontology mapping π is defined. The atoms used in the behavioural model **BAM** are mapped by π as follows:

π(has_value(e(S, A), V)) = ES(A, S, V) where instances for S are b(O), c(O) for options O
π(has_value(s(S, A), V)) = SS(A, S, V) where instances for S are g(b((O)), g(c((O)) for options O

Next by compositionality the interpretation mapping π^* is defined for the specification of the behavioural model **BAM** as follows:

Mapping BP1 Generating a body state

π^*(BP1) = π^*(has_value(s(g(b(O)), A), V_1) & has_value(e(b(O), A), V)
→ has_value(e(b(O), A), V + $\varepsilon_{b(O)A}\, \gamma_{b(O)A}$ **c**(V_1, V/$\varepsilon_{b(O)A}$) Δt))
= π^*(has_value(s(g(b(O)), A), V_1) & has_value(e(b(O), A), V))
→ π^*(has_value(e(b(O), A), V + $\varepsilon_{b(O)A}\, \gamma_{b(O)A}$ **c**(V_1, V/$\varepsilon_{b(O)A}$) Δt))
= π^*(has_value(s(g(b(O)), A), V_1)) & π^*(has_value(e(b(O), A), V))
→ π^*(has_value(e(b(O), A), V + $\varepsilon_{b(O)A}\, \gamma_{b(O)A}$ **c**(V_1, V/$\varepsilon_{b(O)A}$) Δt))

$= \pi(\text{has\_value}(s(g(b(O)), A), V_1)) \ \& \ \pi(\text{has\_value}(e(b(O), A), V))$
$\quad \rightarrow \pi(\text{has\_value}(e(b(O), A), V + \varepsilon_{b(O)A}\, \gamma_{b(O)A}\, \mathbf{c}(V_1, V/\varepsilon_{b(O)A})\ \Delta t))$
$= SS(A, g(b(O)), V_1) \ \& \ ES(A, b(O), V) \rightarrow ES(A, b(O), V + \varepsilon_{b(O)A}\, \gamma_{b(O)A}\, \mathbf{c}(V_1, V/\varepsilon_{b(O)A})\ \Delta t)$

Mapping BP2 Generating an option choice intention

$\pi^*(BP2) = \pi^*(\text{has\_value}(s(g(c(O)), A), V_1) \ \& \ \text{has\_value}(e(b(O), A), V_2) \ \& \ \text{has\_value}(e(c(O), A), V)$
$\quad \rightarrow \text{has\_value}(e(c(O), A), V + \varepsilon_{c(O)A}\, \omega_{OA}\, \gamma_{c(O)A}\, \mathbf{d}((\omega_{c(O)A}/\omega_{OA})\, V_1 + (\omega_{b(O)A}/\omega_{OA})\, V_2/\varepsilon_{b(O)A}, V/\varepsilon_{c(O)A})\ \Delta t))$
$= \pi(\text{has\_value}(s(g(c(O)), A), V_1)) \ \& \ \pi(\text{has\_value}(e(b(O), A), V_2)) \ \& \ \pi(\text{has\_value}(e(c(O), A), V))$
$\quad \rightarrow \pi(\text{has\_value}(e(c(O), A), V + \varepsilon_{c(O)A}\, \omega_{OA}\, \gamma_{c(O)A}\, \mathbf{d}((\omega_{c(O)A}/\omega_{OA})\, V_1 + (\omega_{b(O)A}/\omega_{OA})\, V_2/\varepsilon_{b(O)A}, V/\varepsilon_{c(O)A})\ \Delta t))$
$= SS(A, g(c(O)), V_1) \ \& \ ES(A, b(O), V_2) \ \& \ ES(A, c(O), V)$
$\quad \rightarrow ES(A, c(O), V + \varepsilon_{c(O)A}\, \omega_{OA}\, \gamma_{c(O)A}\, \mathbf{d}((\omega_{c(O)A}/\omega_{OA})\, V_1 + (\omega_{b(O)A}/\omega_{OA})\, V_2/\varepsilon_{b(O)A}, V/\varepsilon_{c(O)A})\ \Delta t)$

Mapping BTP Sensing aggregated group members' bodily responses and intentions

$\pi^*(BTP) = \pi^*(\wedge_{B\neq A}\ \text{has\_value}(e(S, B), V_B) \rightarrow \text{has\_value}(s(g(S), A), \Sigma_{B\neq A}\ \alpha_{SBA}V_B / \Sigma_{B\neq A}\ \alpha_{SBA}\varepsilon_{SB}))$
$= \wedge_{B\neq A}\ \pi(\text{has\_value}(e(S, B), V_B)) \rightarrow \pi(\text{has\_value}(s(g(S), A), \Sigma_{B\neq A}\ \alpha_{SBA}V_B / \Sigma_{B\neq A}\ \alpha_{SBA}\varepsilon_{SB}))$
$= \wedge_{B\neq A}\ ES(B, S, V_B) \rightarrow SS(A, g(S), \Sigma_{B\neq A}\ \alpha_{SBA}V_B / \Sigma_{B\neq A}\ \alpha_{SBA}\varepsilon_{SB})$

So to explore under which conditions the mapped behavioural model BAM is the abstracted model ABAM, it can be found out when the following identities (after unifying the variables V_i, V and W_i, W for values) hold.

$$\pi^*(BP1) = ABP1 \qquad \pi^*(BP2) = ABP2 \qquad \pi^*(BTP) = ITP$$

However, the modelling scope of **ABAM** is wider than the one of **BAM**. In particular, in **ABAM** an as-if body loop is incorporated that has been left out of consideration for **BAM**. Moreover, in the behavioural model **BAM** the options O are taken from a fixed set, given at forhand and automatically considered, whereas in **ABAM** they are generated on the basis of the context c. Therefore, the modelling scope of **ABAM** is first tuned to the one of **BAM**, to get a comparable modelling scope for both models **IAM** and **ABAM**. The latter condition is achieved by taking the activation level W_0 of the sensor state for the context c and the strengths of the connections between the sensor state for context c and preparations relating to option O can be set at *1* (so $\upsilon_{cA} = \omega_{1OA} = \omega_{4OA} = 1$); thus the first argument of g and h becomes 1. The former condition is achieved by leaving out of **ABAM** the dependency on the sensed body state, i.e., by making the third argument of g zero (so $\omega_{0OA} = 0$). Given these extra assumptions and the mapped specifications found above, when the antecedents where unified according to $V_i \leftrightarrow W_i$, $V \leftrightarrow W$ the identities are equivalent to the following identities in V, V_i

$$\varepsilon_{b(O)A}\, \gamma_{b(O)A}\, \mathbf{c}(V_1, V/\varepsilon_{b(O)A}) = \zeta_{b(O)A}\, \lambda_{b(O)A}\, \mathbf{g}(1, \omega_{2OA}\upsilon_{g(b(O))A}\, V_1, 0, V/\zeta_{b(O)A})$$
$$\varepsilon_{c(O)A}\, \omega_{OA}\, \gamma_{c(O)A}\, \mathbf{d}((\omega_{c(O)A}/\omega_{OA})\, V_1 + (\omega_{b(O)A}/\omega_{OA})\, V_2/\varepsilon_{b(O)A}, V/\varepsilon_{c(O)A}) = \zeta_{c(O)A}\, \lambda_{c(O)A}\, \mathbf{h}(1, \omega_{5OA}\upsilon_{g(c(O))A}\, V_1, \omega_{6OA}\upsilon_{b(O)A}\, V_2/\zeta_{b(O)A}, V/\zeta_{c(O)A})$$
$$\Sigma_{B\neq A}\ \alpha_{SBA}V_B / \Sigma_{B\neq A}\ \alpha_{SBA}\varepsilon_{SB} = \Sigma_{B\neq A}\ \alpha_{SBA}V_B / \Sigma_{B\neq A}\ \alpha_{SBA}\zeta_{SB}$$

The last identity is equivalent to

$$\varepsilon_{SB} = \zeta_{SB}$$

for all S and B with $\alpha_{SBA} > 0$ for some A. When $\varepsilon_{SB} = \zeta_{SB}$ is assumed for all S and B (no fully isolated group members), the first two identities can be rewritten into:

$$\varepsilon_{b(O)A}\, \gamma_{b(O)A}\, \mathbf{c}(V_1, V/\varepsilon_{b(O)A}) = \varepsilon_{b(O)A}\, \lambda_{b(O)A}\, \mathbf{g}(1, \omega_{2OA}\upsilon_{g(b(O))A}\, V_1, 0, V/\varepsilon_{b(O)A})$$
$$\varepsilon_{c(O)A}\, \omega_{OA}\, \gamma_{c(O)A}\, \mathbf{d}((\omega_{c(O)A}/\omega_{OA})\, V_1 + (\omega_{b(O)A}/\omega_{OA})\, V_2/\varepsilon_{b(O)A}, V/\varepsilon_{c(O)A}) = \varepsilon_{c(O)A}\, \lambda_{c(O)A}\, \mathbf{h}(1, \omega_{5OA}\upsilon_{b(O)A}\, V_2/\varepsilon_{b(O)A}, \omega_{6OA}\upsilon_{g(c(O))A}\, V_1, V/\varepsilon_{c(O)A})$$

Replacing $V/\varepsilon_{b(O)A}$ by V, resp. $V/\varepsilon_{c(O)A}$ by V, this can be transformed into

$$\gamma_{b(O)A}\, \mathbf{c}(V_1, V) = \lambda_{b(O)A}\, \mathbf{g}(1, \omega_{2OA}\upsilon_{g(b(O))A}\, V_1, 0, V)$$
$$\omega_{OA}\, \gamma_{c(O)A}\, \mathbf{d}((\omega_{c(O)A}/\omega_{OA})\, V_1 + (\omega_{b(O)A}/\omega_{OA})\, V_2, V) = \lambda_{c(O)A}\, \mathbf{h}(1, \omega_{5OA}\upsilon_{g(c(O))A}V_1, \omega_{6OA}\upsilon_{b(O)A}V_2, V)$$

There may be multiple ways in which this can be satisfied for all values of V_1, U_2, U. At least one possibility is the following. Assume for all agents A

$$\lambda_{b(O)A} = \gamma_{b(O)A} \qquad \upsilon_{b(O)A} = 1$$
$$\lambda_{c(O)A} = \omega_{OA}\, \gamma_{c(O)A} \qquad \upsilon_{g(S)A} = 1$$

for S is $b(O)$ or $c(O)$. Then the identities simplify to

$$\mathbf{c}(V_1, U) = \mathbf{g}(1, \omega_{2OA}\, V_1, 0, V)$$
$$\mathbf{d}((\omega_{c(O)A}/\omega_{OA})\, V_1 + (\omega_{b(O)A}/\omega_{OA})\, V_2, V) = \mathbf{h}(1, \omega_{5OA}V_1, \omega_{6OA}V_2, V)$$

Furthermore, taking

$$\omega_{2OA} = 1 \qquad \omega_{5OA} = \omega_{c(O)A}/\omega_{OA} \qquad \omega_{6OA} = \omega_{b(O)A}/\omega_{OA}$$

the following identities result (replacing $\omega_{5OA}V_1$ by V_1 and $\omega_{6OA}V_2$ by V_2)

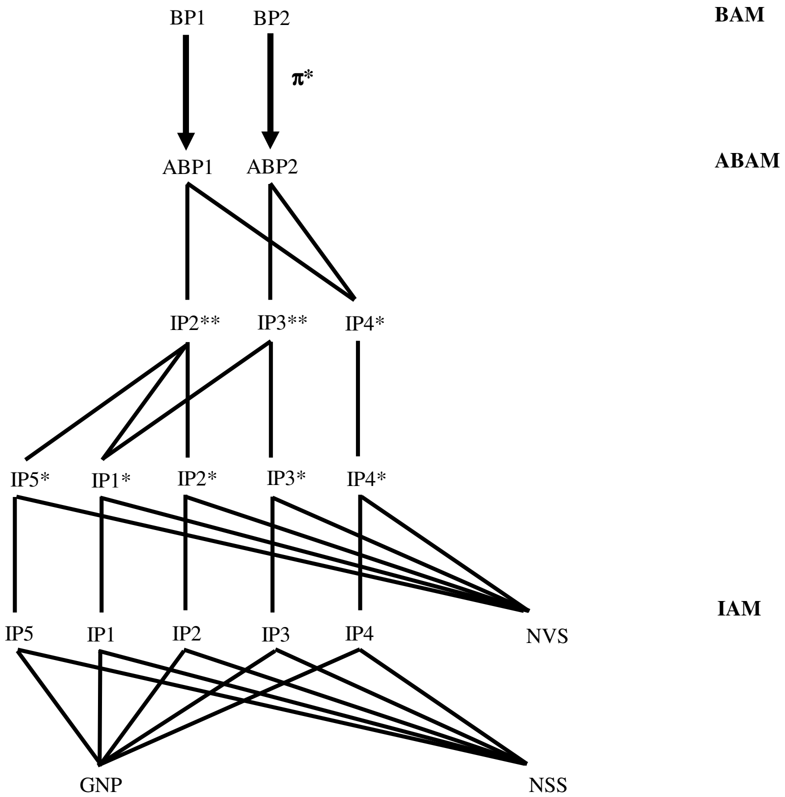

Fig. 7. Logical relations from network specification via internal agent model and abstracted behavioural model to behavioural agent model: $\mathbf{NS} \vdash \mathbf{IAM} \vdash \mathbf{ABAM} = \pi(\mathbf{BAM})$

$$\mathbf{c}(V_1, V) = \mathbf{g}(1, V_1, 0, V) \qquad \mathbf{d}(V_1 + V_2, V) = \mathbf{h}(1, V_1, V_2, V)$$

There are many possibilities to fulfill these identities. For any given functions $\mathbf{c}(X, Y)$, $\mathbf{d}(X, Y)$ in the model **BAM** the functions $\mathbf{g}$, $\mathbf{h}$ in the model **IAM** defined by

$$\mathbf{g}(W, X, Y, Z) = \mathbf{c}(W - 1 + X + Y, Z) \qquad \mathbf{h}(W, X, Y, Z) = \mathbf{d}(W - 1 + X + Y, Z)$$

fulfill the identities $\mathbf{g}(1, X, 0, Z) = \mathbf{c}(X, Z)$ and $\mathbf{h}(1, X, Y, Z) = \mathbf{d}(X+Y, Z)$. It turns out that for given functions $\mathbf{c}(X, Y)$, $\mathbf{d}(X, Y)$ in the model **BAM** functions $\mathbf{g}$, $\mathbf{h}$ in the model **IAM** exist so that the interpretation mapping $\boldsymbol{\pi}$ maps the behavioural model **BAM** onto the model **ABAM**, which is a behavioural abstraction of the internal agent model **IAM** (see also Fig. 5):

$$\pi^*(\mathrm{BP1}) = \mathrm{ABP1} \qquad \pi^*(\mathrm{BP2}) = \mathrm{ABP2} \qquad \pi^*(\mathrm{BTP}) = \mathrm{ITP}$$

As an example direction, when for $\mathbf{c}(X, Y)$ a threshold function th is used, for example, defined as $\mathbf{c}(X, Y) = th(\sigma, \tau, X+Y) - Y$ with $th(\sigma, \tau, V) = 1/(1 + e^{-\sigma(V - \tau)})$, then for $\tau' = \tau + 1$ the function $\mathbf{g}(W, X, Y, Z) = th(\sigma, \tau', W+X+Y+Z) - Z$ fulfils $\mathbf{g}(1, X, 0, Z) = \mathbf{c}(X, Z)$. Another example of a function $\mathbf{g}(V, W, X, Y)$ that fulfills the identity when $\mathbf{c}(X, Z) = 1 - (1 - X)(1 - Z) - Z$ is $\mathbf{g}(W, X, Y, Z) = W\,[1 - (1 - W)(1 - X)(1 - Z)] - Z$. As the properties specifying **ABAM** were derived from the properties specifying **IAM** (e.g., see Figs. 2 and 3), it holds $\mathbf{IAM} \vdash \mathbf{ABAM}$, and as a compositional interpretation mapping $\boldsymbol{\pi}$ preserves derivation relations, the following relationships holds for any temporal pattern expressed as a hybrid logical/numerical property A in the ontology of **BAM**:

$$\mathbf{BAM} \vdash A \;\Rightarrow\; \pi(\mathbf{BAM}) \vdash \pi(A) \;\Rightarrow\; \mathbf{ABAM} \vdash \pi(A) \;\Rightarrow\; \mathbf{IAM} \vdash \pi(A) \;\Rightarrow\; \mathbf{NS} \vdash \pi(A)$$

Such a property A may specify certain (common) patterns in behaviour; the above relationships show that the internal agent model **IAM**, and the network specification **NS** share the common behavioural patterns of the behavioural model **BAM**. An example of such a property A expresses a pattern that under certain conditions after some point in time there is one option O for which both b(O) and c(O) have the highest value for each of the agents (joint decision).

6 Discussion

This paper addressed how internal agent models and behavioural agent models for collective desion making can be related to each other. The relationships presented were expressed for specifications of the agent models in a hybrid logical/numerical format. Two agent models for collective decision making were first presented. First an internal agent model **IAM** derived from neurological principles modelled in a network specification **NS** was introduced with **NS** ⊢ **IAM**, where ⊢ is a symbol for derivability. Next, an existing behavioural agent model **BAM**, incorporating principles on social contagion or diffusion, was described, adopted from [9].

It was shown how the internal agent model **IAM** can be systematically transformed into an abstracted behavioural model **ABAM** (resp., **AEBAM**), where the internal states are abstracted away, and such that **IAM** ⊢ **ABAM**. The abstraction approach also addresses elimination of internal loops, as often occur in neurologically inspired agent models, for example, to model mutual interaction between cognitive and affective states, or internal adaptation techniques (modelling neurological plasticity). In contrast to **ABAM**, in **AEBAM** loops are eliminated based on equilibria.

This generic transformation has been implemented in Java. Moreover, it was shown that under certain conditions the obtained agent model **ABAM** can be related to the behavioural agent model **BAM** by an interpretation mapping **π**, i.e., such that **π(BAM) = ABAM**. In this way hybrid logical/numerical relations where obtained between the different agent models according to:

NS ⊢ IAM ⊢ ABAM = π(BAM)

These relationships imply that, for example, collective behaviour patterns shown in multi-agent systems based on the behavioural agent model **BAM** are shared (in the form of patterns corresponding via **π**) for multi-agent systems based on the models **ABAM**, **IAM** and **NS**.

References

1. Bosse, T., Duell, R., Memon, Z.A., Treur, J., and Wal, C.N. van der, (2009a). A Multi-Agent Model for Mutual Absorption of Emotions. In: Otamendi, J., Bargiela, A., Montes, J.L., Pedrera, L.M.D. (eds.), *Proceedings of the 23th European Conference on Modelling and Simulation, ECMS'09*. European Council on Modeling and Simulation, 2009, pp. 212-218.
2. Bosse, T., Duell, R., Memon, Z.A., Treur, J., and Wal, C.N. van der (2009b). A Multi-Agent Model for Emotion Contagion Spirals Integrated within a Supporting Ambient Agent Model. In: Yang, J.-J.; Yokoo, M.; Ito, T.; Jin, Z.; Scerri, P. (eds.), *Proceedings of the 12th International Conference on Principles of Practice in Multi-Agent Systems, PRIMA'09*. Lecture Notes in Artificial Intelligence, vol. 5925. Springer Verlag, 2009, pp. 48–67.
3. Bosse, T., Jonker, C.M., Meij, L. van der, and Treur, J., A Language and Environment for Analysis of Dynamics by Simulation. *International Journal of Artificial Intelligence Tools*, vol. 16, 2007, pp. 435-464.
4. Damasio, A., *Descartes' Error: Emotion, Reason and the Human Brain*, Papermac, London. (1994)
5. Damasio, A., *The Feeling of What Happens. Body and Emotion in the Making of Consciousness*. New York: Harcourt Brace. (1999)
6. Damasio, A., *Looking for Spinoza: Joy, Sorrow, and the Feeling Brain*. Vintage books, London. (2003)
7. Damasio, A., The Somatic Marker Hypothesis and the Possible Functions of the Prefrontal Cortex. *Philosophical Transactions of the Royal Society: Biological Sciences*, vol. 351, pp. 1413-1420. (1996)
8. Hodges, W. (1993). *Model theory*. Cambridge University Press.
9. Hoogendoorn, M., Treur, J., Wal, C. N. van der, and Wissen, A. van (2010). Modelling the Emergence of Group Decisions Based on Mirroring and Somatic Marking. In: *Proceedings of the Second International Conference on Brain Informatics, BI'10*. Lecture Notes in Artificial Intelligence, Springer Verlag, 2010, to appear..
10. Iacoboni M., *Mirroring People*. Farrar, Straus & Giroux, New York (2008).
11. Pineda, J.A. (ed.). *Mirror Neuron Systems: the Role of Mirroring Processes in Social Cognition*. Humana Press Inc. (2009)
12. Rizzolatti, G., and Sinigaglia, C., *Mirrors in the Brain: How Our Minds Share Actions and Emotions*. Oxford University Pres, (2008)
13. Sharpanskykh, A. and J. Treur, Verifying Interlevel Relations within Multi-Agent Systems. In: Proceedings of the 17th European Conference on Artificial Intelligence, IOS Press 247-254 (2006)
14. Sharpanskykh, A., and Treur, J., Relating Cognitive Process Models to Behavioural Models of Agents. In: Jain, L., Gini, M., Faltings, B.B., Terano, T., Zhang, C., Cercone, N., Cao, L. (eds.), *Proceedings of the 8th IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT'08*. IEEE Computer Society Press, 2008, pp. 330-335.
15. Treur, J., On the Use of Reduction Relations to Relate Different Types of Agent Models. *Web Intelligence and Agent Systems Journal*, 2010, to appear.