Relating Cognitive Process Models to Behavioural Models of Agents

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*

Abstract

From an external perspective, cognitive agent behaviour can be described by specifying (temporal) correlations of a certain complexity between stimuli (input states) and (re)actions (output states) of the agent. From an internal perspective the agent's dynamics can be characterized by direct (causal) temporal relations between internal cognitive states of the agent. Internal dynamics and externally observable behaviour of an agent have reciprocal relations with each other. This paper contributes an approach that allows automatic generation of a behavioural specification of an agent from a cognitive process model. Furthermore, by this automated transformation, internal cognitive state properties of an agent can be related by a representation relation to externally observable behavioural patterns.

1. Introduction

The dynamics of a cognitive agent can be considered both from an external and an internal perspective. From the external perspective, behaviour of the agent can be described by temporal relationships of a certain complexity between its input (stimuli) and output (actions) state properties over time, expressed in some (temporal) language, without any reference to internal cognitive state properties of the agent. Within Philosophy of Mind such an external view is considered within the perspective of behaviourism [8, 11]. Behavioural specifications that comprise simple input-output relations can be successfully used for modelling relatively simple types of behaviour (e.g., stimulus-response behaviour [12]). For less simple behavioural types (e.g., adaptive behaviour based on conditioning [2]) a behavioural specification often consists of more complex temporal relations, relating behaviour at a certain point in time to a possibly large number of input states over (past) time.

From the internal perspective the behaviour of the agent can be characterized by a specification of more direct (causal) temporal relations between internal cognitive state properties of the agent. In this paper an automated transformation is presented to obtain for such an internal specification, an externally observable behavioural pattern of an agent. An internal perspective on the dynamics of an agent is taken within functionalism [11]. From this perspective mental (or internal) state properties are described by their *functional* or *causal roles*. *The functional role* of an internal state property is defined by its direct temporal (or causal) relations to input, output and other internal state properties of an agent. These relations are specified in simple, executable formats (i.e., formats suitable for automated analysis). Although functionalism was originally formulated in terms of a Turing machine representation, other executable representations can be used for functional roles as well.

Furthermore, the occurrence of an internal state property at some time point can be (indirectly) related to the occurrence of other (internal and/or externally observable) state properties at the same or at different time points. For example, the belief of an agent about the presence of an object at some place can be related to the observation of the object at that place at some time point in the past and to no observation of the disappearance of the object since then. Within Philosophy of Mind this relation type is called a *representation relation*. If a representation relation is given between an internal state property p and a specification Φ that comprises a set of state properties and temporal (or causal) relations between them and with p, then it is said that p represents Φ, or Φ describes *representational content* of p. Representational content for a property p may be defined both backward and forward in time. In the backward case, the representational content is specified by a history (i.e., a specification that comprises temporal (or causal) relations on past states) that relates to the creation of the agent's state in which p holds. In the example above the observation of the object at some past time point and no observation of the disappearance of the object since then form the representational content of the belief about the presence of the object backwards in time.

In the forward case, the representational content describes possible (conditional) future states, temporally (or causally) related to the agent's state, in which p holds. In the literature on Philosophy of Mind different approaches to defining representational content have been put forward [3]. For example, according to the classical causal/correlation approach [3], the representational content of an internal state property is given by a one-to-one correspondence to an externally observable state property. The application of this approach is limited to simple types of behaviour (e.g., stimulus-response behaviour). In cases when an internal property represents a more complex temporal combination of state properties, other approaches have to be used. For example, the temporal-interactivist approach (cf. [3, 10]) allows defining representational content by referring to multiple externally observable (partially) temporally ordered agent state properties (i.e., an agent's input and output state properties over time). In such a way internal states of an agent can be related explicitly to its externally observable dynamics (e.g., interaction with other agents, the world and itself). Thus, using ideas underlying the temporal-interactivist approach a clear relation can be established between the agent's cognitive process model and its behavioural model. Knowledge of such a relation has many useful implications: the behaviour of an agent may be explained and predicted based on its internal dynamics; the behaviour may be influenced in a desired way by creating/affecting certain internal agent states and relations between them.

This paper considers a rather general specification format both for (internal) cognitive process models and for (external) behavioural specifications. It is shown that this format enables that for every internal state property backward representational content can be identified in an automated manner using the interactivist approach. Furthermore, given a cognitive process model in this format, an external behavioural specification can be automatically generated. A main contribution of this paper is an automated approach to identify such representation relations and behavioural specifications for any given cognitive process model.

The paper is organized as follows. Section 2 presents a temporal language for specifying cognitive process models and behavioural models. The theoretical basis for the automated transformation is provided in Section 3. Section 4 presents the transformation algorithm, which was implemented in Java, and evaluates its complexity. The approach is illustrated by examples in Section 5. Section 6 concludes the paper.

2. Modelling language

Both behavioural specifications and cognitive process models are specified using the reified temporal predicate language *RTPL* [7], a many-sorted temporal predicate logic language that allows specification and reasoning about the dynamics of a system. To express state properties ontologies are used. *An ontology* is a signature specified by a tuple $\langle S_1, \ldots, S_n, \ldots, C, f, P, arity \rangle$, where S_i is a sort for $i=1,.., n$, C is a finite set of constant symbols, f is a finite set of function symbols, P is a finite set of predicate symbols, arity is a mapping of function or predicate symbols to a natural number. An interaction ontology InteractOnt is used to describe the externally observable behaviour of an agent. It is the union of input and output ontologies: InteractOnt = InputOnt $\cup$ OutputOnt, used to describe input and output states of an agent correspondingly. Specifically, using InputOnt one can define observations of state properties and communications received (e.g, observed(a) means that an agent has an observation of state property a). The ontology OutputOnt is used to define agent's communications (e.g, communicated(m) means that message m is communicated between agents) and actions (e.g., performing_action(b) represents action b performed by an agent). The internal ontology InternalOnt is used to describe the agent's internal state properties (e.g., sensory representations, beliefs). Ontologies InteractOnt and InternalOnt define disjoint sets of sorts, constants, functions and predicates.

In *RTPL* state properties as represented by formulae within the state language are used as terms (denoting objects). To this end the state language is imported in *RTPL* as follows: For every sort S from the state language the following sorts are introduced in *RTPL*: the sort S^{VARS}, which contains all variable names of sort S, the sort S^{GTERMS}, which contains names of all ground terms constructed using sort S; sorts S^{GTERMS} and S^{VARS} are subsorts of sort S^{TERMS}. Sort STATPROP contains names for all state formulae. To provide names for state formulae φ in *RTPL*, the operator (*) is used (written as φ^*), which maps variable sets, term sets and formula sets of the state language to the elements of sorts S^{GTERMS}, S^{TERMS}, S^{VARS} and STATPROP. It is assumed that the state language and *RTPL* define disjoint sets of expressions. Therefore, further in *RTPL* formulae we shall use the same notations for the elements of the object language and for their names in the *RTPL* without introducing any ambiguity.

The set of function symbols of *RTPL* includes $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$: STATPROP x STATPROP $\rightarrow$ STATPROP; not: STATPROP $\rightarrow$ STATPROP, and $\forall$, $\exists$: S^{VARS} x STATPROP $\rightarrow$ STATPROP, of which the counterparts in the state language are Boolean propositional connectives and quantifiers. Further we shall use $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ in infix notation and $\forall, \exists$ in prefix notation for better readability. To represent dynamics of a system sort TIME (a set of time points) and the ordering relation $>$: TIME x TIME are introduced in *RTPL*. To indicate that some state property holds at some time point the relation at: STATPROP x TIME is introduced. The terms of *RTPL* are constructed by induction in a standard way from variables, constants and function symbols typed with all before-mentioned sorts. The set of *atomic RTPL-formulae* is defined as:

(1) If t is a term of sort TIME, and p is a term of the sort STATPROP, then $at(p, t)$ is an atomic *RTPL* formula.
(2) If τ_1, τ_2 are terms of any *RTPL* sort, then $\tau_1 = \tau_2$ is an *RTPL*-atom.
(3) If t_1, t_2 are terms of sort TIME, then $t_1 > t_2$ is an *RTPL*-atom.

The set of well-formed *RTPL* formulae is defined inductively in a standard way using Boolean connectives and quantifiers over variables of *RTPL* sorts. The language *RTPL* has the semantics of many-sorted predicate logic.

To express properties of behavioural and cognitive process specifications *past* and *past-present* statements are used.

Definition 1 (Past and Past-Present Statement)
A *past statement* for a time point t over state ontology Ont is a temporal statement $\varphi_p(t)$ in the reified temporal predicate logic, such that each time variable s different from t is restricted to the time interval before t: for every time quantifier for a time variable s a restriction of the form $t > s$ is required within the statement.

A *past-present statement* (abbreviated as a *pp-statement*) is a statement φ of the form $B \Leftrightarrow H$, where the formula B, called the *body* and denoted by $body(\varphi)$, is a past statement for t, and H, called the *head* and denoted by $head(\varphi)$, is a statement of the form $at(p, t)$ for some state property p.

It is assumed that each output state of an agent specified by an atom $at(\psi, t)$ is generated based on some input and internal agent's dynamics that can be specified by a set of formulae over $\varphi(t) \Rightarrow at(\psi, t)$ with φ a past statement over InputOnt$\cup$InternalOnt. Furthermore, a completion can be made (similar to Clark's completion in logic programming) that combines all statements $[\varphi_1(t) \Rightarrow at(\psi, t)\ ,\ \varphi_2(t) \Rightarrow at(\psi, t)\ ,\ \dots\ ,\ \varphi_n(t) \Rightarrow at(\psi, t)\ ]$ with the same consequent in the specification, into one past-present-statement $\varphi_1(t) \vee \varphi_2(t) \vee \dots \vee \varphi_n(t) \Leftrightarrow at(\psi, t)$. Sometimes this statement is called the *definition* of $at(\psi, t)$. Thus, a specification format is assumed based on past-present statements with *unique heads*: each head occurs only in one statement as a head. Not only output states but also each internal (or mental) state property of an agent $at(\xi, t)$ is assumed to be specified by a past-present statement $\varphi(t) \Leftrightarrow at(\xi, t)$, where $\varphi(t)$ is expressed over InputOnt$\cup$InternalOnt.

Definition 2 (Agent Specifications)
A *cognitive* or *lower level agent specification* is a set of past-present statements based on the ontology InteractOnt$\cup$InternalOnt with unique heads. A *behavioural* or *higher level agent specification* is a set of past-present statements based on the ontology InteractOnt, where the bodies only use InputOnt with unique heads.

Agent specifications are assumed to be stratified [1].

Definition 3 (Stratification of a Specification)
An agent specification Π is *stratified* if there is a partition $\Pi = \Pi_1 \cup \dots \cup \Pi_n$ into disjoint subsets such that the following condition holds: for $i > 1$: if a subformula $at(\varphi, t)$ occurs in a body of a statement in Π_i, then it has a definition within $\cup_{j \leq i} \Pi_j$

The notation $\varphi[at_1, \dots, at_n]$ is used to denote a formula φ with $at_1, \dots, at_n$ as its atomic subformulae. The function STRATUM maps a specification and a natural number (a stratum number) to the set of formulae from the corresponding stratum of the specification.

3. Abstraction and refinement

This Section introduces the theoretical basis for the automated procedure for generation of an agent's behavioral specification from its cognitive process specification described in Section 4.

The rough idea behind the procedure is as follows. Suppose for a certain cognitive state property the pp-specification $B \Leftrightarrow at(p, t)$ is available. Moreover, suppose that in B only two atoms of the form $at(p1, t1)$ and $at(p2, t2)$ occur, whereas as part of the cognitive model also specifications $B1 \Leftrightarrow at(p1, t1)$ and $B2 \Leftrightarrow at(p2, t2)$ are available. Then, within B the atoms can be replaced (by substitution) by the formula B1 and B2. This results in a

$$B[B1/at(p1, t1), B2/at(p2, t2)] \Leftrightarrow at(p, t)$$

which again is a pp-specification. Here for any formula C the expression $C[x/y]$ denotes the formula C transformed by substituting x for y. Such a substitution corresponds to an abstraction step. For the general case the procedure includes a sequence of abstraction steps; the last step produces a behavioural specification that corresponds to a cognitive process model.

To define an abstraction of a lower level agent specification first a step transformation operator is introduced.

Definition 4 (Step Transformation Operator)
The step operator A_1 maps a set of pp-formulae $X1$ into a set of pp-formulae $X2 = X1 \cup X1'$, where $X1'$ is a set of formulae obtained as follows: each atomic subformula at_k of each body $\varphi[at_1, \dots, at_n]$ of a formula from the highest stratum n of $X1$ is substituted by its definition $\varphi_k(t)[at_1, \dots, at_m]$ from a stratum $i \leq n\text{-}1$ of $X1$.

Definition 5 (Abstraction Operator)
The *abstraction step operator* B_1 maps a stratified set X of pp-formulae with $n>1$ strata to a set of pp-formulae with $n-1$ strata as follows:

$$B_1(X) = A_1(X) \setminus (\mathrm{STRATUM}(X, n) \cup \{ \varphi \in \mathrm{STRATUM}(X, n-1) \mid \exists \psi \in \mathrm{STRATUM}(X, n) \text{ AND } \mathrm{head}(\varphi) \text{ is a subformula of the } \mathrm{body}(\psi)\})$$

For a set of pp-formulae X with one stratum $B_1(X) = X$.
Then, the abstraction operator B is defined for a stratified set X as $B(X) = B_1^{n-1}(X)$.

The following lemma is useful for the proof of Proposition 1.

Lemma 1
For each stratified set of pp-formulae X with $n>1$ strata, the set $B_1(X)$ is stratified using $n-1$ strata, and $B(X)$ is stratified using one stratum.

In the Appendix a proof sketch is given for Lemma 1.
A_1 is a conservative, monotonic operator. According to the Knaster-Tarski Theorem [13] A_1 has a smallest fixed point.

Proposition 1 (Fixed points for A_1 and B_1)
The smallest fixed point of the operator A_1 can be calculated in a finite number of steps, more specifically it is $A_1^{n-1}(X)$, where n denotes the number of strata of X and A_1^{n-1} denotes the $(n-1)$ subsequent applications of the operator A_1. $B_1^{n-1}(X)$ is a fixed point of the operator B_1.

Definition 6 (Refinement)
A set of formulae X in Ont' *refines* a set of formulae Y in $\mathrm{Ont} \subseteq \mathrm{Ont}'$ if and only if
(1) $X \models Y$
(2) For each Ont-model $<I, v>$ of Y, where I is an interpretation function and v is a valuation of variables, exists an expanded Ont'-model $<I', v'>$, which is a model of X.

The following lemma is useful for the proof of Proposition 2.

Lemma 2 (Equivalence of Substituted Formulae)
Let φ' be the formula obtained from a formula φ by a substitution $\{\alpha_1 \setminus \psi_1, \ldots, \alpha_m \setminus \psi_m\}$, where the α_i are the atomic subformula of φ. If $\alpha_i \Leftrightarrow \psi_i$ for all i, then $\varphi' \Leftrightarrow \varphi$.

Proposition 2 (Abstraction and Refinement)
If a set of formula Y is obtained by an abstraction step from a set of formulae X, then X refines Y.
In the Appendix a proof sketch is given. for Proposition 2.

Lemma 3 (Transitivity of Refinement)
If X3 refines X2 and X2 refines X1, then X3 refines X1.

Theorem 1 (Existence of Abstraction)
For the operator B for all X it holds that X refines B(X).
Theorem 1 follows by induction from Proposition 2.

Theorem 2 (Refinement Implies the Same Consequences)
Let X in Ont' be a refinement of Y in Ont and ψ is a formula expressed using Ont. Then

$$X \models \psi \Leftrightarrow Y \models \psi$$

The proof for this Theorem is provided in [14], where a transformation of a higher level agent specification into a lower level agent specification is considered.

Corollary (Abstraction Implies the Same Consequences)
Let Π be a lower level (cognitive) specification and ψ a dynamic interaction property of the agent in its environment expressed using InteractOnt, then

$$\Pi \models \psi \Leftrightarrow B(\Pi) \models \psi$$

This corollary immediately follows from Theorems 1 and 2.

4. Abstraction algorithm

In [1] it is shown that a specification can be stratified iff its dependency graph does not contain any cycles with a negative link. In this paper, the dependency graph of a specification is the directed graph representing the relation refers_to between the at predicate symbols of the specification: p refers_to q iff exists a formula φ in the specification, such that p is a subformula of $\mathrm{head}(\varphi)$ and q is a subformula of $\mathrm{body}(\varphi)$.

The abstraction of a specification that can be stratified is constructed using the following algorithm.

Algorithm: BUILD-ABSTRACTION

Input: Lower level specification X
Output: Higher level specification B(X)

```
1   Enforce temporal completion on X.
2   Stratify X:
  2.1 Define the set of formulae of the first stratum (h=1)
      as: $\{\varphi_i: at(a_i, t) \leftrightarrow \psi i_p(at_1,\ldots, at_m) \in X \mid \forall k\ m \geq k \geq 1\ at_k$ is
      expressed using InputOnt$\}$;
      proceed with h=2.
  2.2 The set of formulae for stratum h is identified as
      $\{\varphi_i: at(a_i, t) \leftrightarrow \psi i_p(at_1,\ldots, at_m) \in X \mid \forall k\ m \geq k \geq 1\ \exists l\ l < h\ \exists \psi \in \mathrm{STRATUM}(X, l) \text{ AND } \mathrm{head}(\psi) = at_k \text{ AND } \exists j\ m \geq j \geq 1\ \exists \xi \in \mathrm{STRATUM}(X, h-1) \text{ AND } \mathrm{head}(\xi)=at_j \}$;
      proceed with h=h+1.
  2.3 Until a formula of X exists not allocated to a stratum,
      perform 2.2.
3   Replace each formula of the highest stratum n $\varphi_i$:
    $at(a_i, t) \leftrightarrow \psi i_p(at_1,\ldots, at_m)$ by $\varphi_l\ \delta$ with renaming of
    temporal variables if required, where $\delta = \{at_k \setminus \mathrm{body}(\varphi_k)$ such that $\varphi_k \in X$ and $\mathrm{head}(\varphi_k)=at_k\}$. Further,
    remove all formulae $\{ \varphi \in \mathrm{STRATUM}(X, n-1) \mid \exists \psi \in \mathrm{STRATUM}(X, n) \text{ AND } \mathrm{head}(\varphi)$ is a subformula of the
    $\mathrm{body}(\psi)\})$
```

4 Append the formulae of the stratum n to the stratum $n-1$, which now becomes the highest stratum (i.e, $n=n-1$).

5 Until $n>1$, perform steps 3 and 4.

In Step 3 subformulae of each formula of the highest stratum n of X are replaced by their definitions, provided in lower strata. Then, the formulae of $n-1$ stratum used for the replacement are eliminated from X. As result of such a replacement and elimination, X contains $n-1$ strata (Step 4). Steps 3 and 4 are performed until X contains one stratum only. In such a way, the low level specification is abstracted gradually into a behavioural specification.

To determine the representational content for an internal state from a lower level specification the procedure is applied with the input specification obtained by selecting from a lower level specification only the formulae of the strata with the number $i < k$, where k is the number of the stratum, in which the internal state is defined.

The algorithm has been implemented in Java. Worst case time and representation complexity of the algorithm are satisfactory as will be briefly discussed.

The worst case time complexity of the algorithm is estimated as follows. Time complexity of step 1 is $O(|X|)$. The worst case time complexity for step 2 is $O(|X|^2/2)$. The worst case time complexity for steps 3-5 is calculated as:

$$\begin{aligned}&O(|\text{STRATUM}(X, n)|\cdot |\text{STRATUM}(X, n\text{-}1)|)+\\&O(|\text{STRATUM}(X, n)|\cdot |\text{STRATUM}(X, n\text{-}2)|) + \ldots +\\&O(|\text{STRATUM}(X, n)|\cdot |\text{STRATUM}(X, 1)|)\\&= O(|\text{STRATUM}(X, n)|\cdot|X|).\end{aligned}$$

Thus, the overall time complexity of the algorithm for the worst case is $O(|X|^2)$.

The representation of a higher level specification Φ is more compact than of the corresponding lower level specification Π. First, only InteractOnt is used to specify the formulae of Φ, whereas InteractOnt$\cup$InternalOnt is used to specify the formulae of Π. Furthermore, only a subset of the temporal variables from Π is used in Φ, more specifically, the set of temporal variables from

$$\{\text{body}(\varphi_i) \mid \varphi_i \in \Pi\} \cup \{\text{head}(\varphi_i) \mid \varphi_i \in \Pi \text{ AND } \text{head}(\varphi_i) \text{ is expressed over InteractOnt}\}.$$

However, at step 3 of the algorithm if the number of substitutions of subformulae of a formula by the same definition is $m>1$, than $m-1$ additional time variables are introduced in Φ.

5. Examples

As a first example a specification of cognitive processes of an agent is based on a Belief-Desire-Intention (BDI) model [4] is considered Using a BDI model, actions of an agent can be related to its internal dynamics represented by (temporal) relations between belief, desire and intention states of the agent. In the following a particular internal cognitive model for animal behavior is considered. Initially, the animal observes that it has low energy (e.g., being hungry). The animal is placed in front of a transparent screen, behind which a piece of food is put afterwards. The animal is able to observe the position of the food and of the screen, after which a cup is placed over the food. After some time the screen is raised and the animal chooses to go to the position at which food is present (but invisible). The cognitive model Π_A from the example is formalised by the following pp-properties:

IPA1: *Desire d generation*
At any point in time the (persistent) internal state property d holds *iff* at some time point in the past the agent observed its low energy. Formally:

$$\exists t2\ [\ t1 > t2\ \&\ \text{at}(\text{observed}(\text{own\_low\_energy}), t2)\ ]\ \Leftrightarrow \text{at}(d, t1)$$

IPA2: *Intention i generation:*
At any point in time the (persistent) internal state property i holds *iff* at some time point in the past the internal state property d was true, and the internal state property b was true. Formally:

$$\exists t6\ t5 > t6\ \&\ \text{at}(d, t6)\ \&\ \text{at}(b, t6) \Leftrightarrow \text{at}(i, t5)$$

IPA3: *Action goto p2 generation:*
At any point in time the agent goes to p2 *iff* at some time point in the past the internal state property i was true and the agent observed the absence of the screen. Formally:

$$\exists t8\ t7>t8\ \&\ \text{at}(\text{observed}(\text{no\_screen}), t8)\ \&\ \text{at}(i,t8) \Leftrightarrow \text{at}(\text{performing\_action}(\text{goto\_p2}),t7)$$

IPA4: *Belief b generation:*

At any point in time internal state property b holds *iff* at some time point in the past the agent observed that food is present at position p2, and since then did not observe the absence of food. Formally:

$$\exists t10\ t9 > t10\ \&\ \text{at}(\text{observed}(\text{food\_p2}), t10)\ \&\ \forall t11\ t11>t10\ \&\ t11<t9\ \text{not}(\text{at}(\text{observed}(\text{no\_food\_p2}), t11)) \Leftrightarrow \text{at}(b,t9)$$

By performing Steps 1 and 2 of the procedure described in Section 4 the specification of the cognitive model given above is automatically stratified as follows: stratum 1: {IPA1, IPA4}; stratum 2: {IPA2}; stratum 3: {IPA3}. Further, in Step 3 in the property IPA3 of the highest stratum internal state property i is replaced by its definition from IPA2:

$$\exists t8\ t7>t8\ \&\ \text{at}(\text{observed}(\text{no\_screen}), t8)\ \&\ \exists t6'\ t8 > t6'\ \&\ \text{at}(d, t6)\ \&\ \text{at}(b, t6') \Leftrightarrow \text{at}(\text{performing\_action}(\text{goto\_p2}),t7)$$

Further, the formula IPA2 is removed, and the property resulted from the replacement is added to the stratum 2, which now becomes the highest stratum. In the following execution of Step 3 both state properties b and d are replaced by their definitions and the following property is obtained and added to stratum 1:

$$\exists t8\ t7>t8\ \&\ \text{at}(\text{observed}(\text{no\_screen}), t8)\ \&$$

∃t6 t8 >t6 & ∃t2 [t6 > t2 & at(observed(own_low_energy), t2)
& ∃t10 t6 >t10 & at(observed(food_p2), t10)
& ∀t11 t11>t10 & t11<t6 not(at(observed(no_food_p2), t11))
⇔ at(performing_action(goto_p2),t7)

Both properties IPA1 and IPA4 are removed, and the obtained property becomes the only property in the specification, thus it defines the behavioural specification for Π_A.

The representational content for the intention state i from Π_A is generated as:

∃t6 t5 >t6 & ∃t2 t6 > t2 & at(observed(own_low_energy), t2) &
∃t10 t6 >t10 & at(observed(food_p2), t10)
& ∀t11 t11>t10 & t11<t6 & not(at(observed(no_food_p2), t11))
⇔ at(i, t5)

In the second example a model based on the theory of consciousness by Antonio Damasio [6] is considered. In particular, the notions of 'emotion', 'feeling', and 'core consciousness' or 'feeling a feeling' are addressed. Damasio [6] describes an emotion as neural object (or internal emotional state) as an (unconscious) neural reaction to a certain stimulus, realised by a complex ensemble of neural activations in the brain. As the neural activations involved often are preparations for (body) actions, as a consequence of an internal emotional state, the body will be modified into an externally observable state. Next, a feeling is described as the (still unconscious) sensing of this body state. Finally, core consciousness or feeling a feeling is what emerges when the organism detects that its representation of its own body state (the proto-self) has been changed by the occurrence of the stimulus: it becomes (consciously) aware of the feeling. In Figure 1 a cognitive model for this process is depicted. Here s0 is an internal representation of the situation that no stimulus is sensed, and no changed body state, s1 is an internal representation of the sensed stimulus without a sensed changed body state yet, and s2 is an indication for both sensed stimulus and changed body state (which is the core consciousness state).

The cognitive model for this example comprises the following properties expressed in past-present format:

LP1: *Generation of the sensory representation for music*
At any point in time the sensory representation for music occurs *iff* at some time point in the past the sensor state for music occurred. Formally:
∃t2 t1 > t2 & at(sr_music, t2) ⇔ at(sr_music, t1)

LP2: *Generation of the preparation*
At any point in time the preparation p occurs *iff* at some time point in the past the sensory representation for music occurred. Formally:

∃t4 t3 > t4 & at(sr_music, t4) ⇔ at(p, t3)

LP3: *Generation of the body state*
At any point in time the body state S occurs *iff* at some time point in the past the preparation p occurred. Formally:

∃t6 t5 > t6 & at(p, t6) ⇔ at(S, t5)

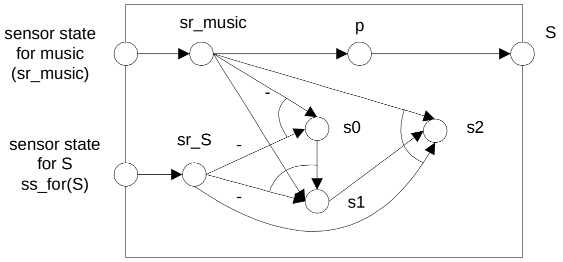

Figure 1. Cognitive model based on the theory of core consciousness by Damasio [6]

LP4: *Generation of the sensor state*
At any point in time the sensor state for S occurs *iff* at some time point in the past the body state S occurred. Formally:

∃t8 t7 > t8 & at(S, t8) ⇔ at(ss_for(S), t7)

LP5: *Generation of the sensory representation for S*
At any point in time the sensory representation for S occurs *iff* at some time point in the past the sensor state vector for S occurred
Formally: ∃t10 t9 > t10 & at(ss_for(S), t10) ⇔ at(sr_S, t9)

LP6: *Generation of s0*
At any point in time s0 occurs *iff* at some time point in the past no sensory representation for music and no sensory representation for S occurred. Formally:

∃t12 t11 > t12 & not(at(sr_music, t12)) & not(at(sr_S, t12)) ⇔ at(s0, t11)

LP7: *Generation of s1*
At any point in time s1 occurs *iff*
at some time point in the past the sensory representation for music and no sensory representation for S and s0 occurred. Formally:

∃t14 t13 > t14 & at(sr_music, t14) & at(s0, t14) & not(at(sr_S, t14)) ⇔ at(s1, t13)

LP8: *Generation of s2*
At any point in time s2 occurs *iff*
at some time point in the past the sensory representation for music and the sensory representation for S and s1 occurred. Formally:
∃t16 t15 > t16 & at(sr_music, t16) & at(s1, t16) & at(sr_S, t16) ⇔ at(s2, t15)

The generated behavioural property is:

∃t6 t5 > t6 & ∃t4 t6 > t4 & at(sr_music, t4) ⇔ at(S, t5)

The generated representation relation for state s2 is:

exists(t16) t15>t16 exists(t2) t16>t2 at(ss_music,t2) &
exists(t10) t16>t10 at(ss_for(S),t10) & exists(t14) t16>t14
exists(t2') t14>t2' at(ss_music,t2') & not(exists(t10')
t14>t10' at(ss_for(S),t10')) & exists(t12) t14>t12
not(exists(t2'') t12>t2'' & at(ss_music,t2'')) &
not(exists(t10'') t12>t10'' at(ss_for(S),t10''))
⇔ at(s2,t15)

6. Discussion

The dynamics of an agent can be specified from an external perspective by a behavioral specification and from an internal perspective by a cognitive process specification. The question arises how an agent's behaviour is related to its cognitive process model. This question can be reformulated as two problems: (1) given a behavioural specification, what cognitive process model(s) realise(s) this specification? (2) given a cognitive process specification, which externally observable behavioural pattern can be generated based on this specification. The first problem has been addressed in [14] by proposing an automated refinement transformation of an agent's behavioural specification into a cognitive process model. Such a model comprises postulated internal states and direct temporal relations between such states. This paper addresses the second problem by proposing an approach for automated generation of an agent's behavioural specification from a cognitive process specification. Using this approach also the representational content (backward in time) of an internal (or mental) state property of an agent can be generated in an automated manner. Furthermore, the proposed approach can be applied to verify if every possible agent behaviour generated by a cognitive process model satisfies some behavioural requirements imposed on the agent (e.g., environmental requirements). More specifically, using the proposed approach the generated behavioural specification provides an exact representation of the agent interaction functionality in its environment, based on which (1) the satisfaction of the environmental requirements can be determined, and (2) it can be determined whether the agent's functionality is minimal in that respect, i.e., it is not more complex than needed to satisfy these requirements.

In this paper the dynamics of an agent have been considered at two levels of abstraction. However, in general both behavioural specifications and cognitive process models can be considered at different levels of abstraction (i.e., with different degrees of detail). To perform abstraction of such specifications the same principles can be applied as described in this paper.

The proposed approach can be used for intelligent agents that support humans in different contexts (e.g., support of elder people in their houses [9]). More specifically, by establishing representation relations such agents may identify the activation of human's internal states by monitoring his/her externally observable behaviour.

7. References

[1] K.R. Apt, H.A. Blair, A. Walker, Towards a Theory of Declarative Knowledge. Foundations of Deductive Databases and Logic Programming, 1988: 89-148 (1988)

[2] C. Balkenius and J. Moren, Dynamics of a classical conditioning model. Autonomous Robots, 7, 41-56 (1999)

[3] M.H. Bickhard, Representational Content in Humans and Machines. Journal of Experimental and Theoretical Artificial Intelligence, 5, 285-333 (1993)

[4] M. E. Bratman. Intentions, Plans, and Practical Reason. Harvard University Press: Cambridge, MA (1987)

[5] E.M.Clarke, O.Grumberg, D.A.Peled, Model Checking, MIT Press, Cambridge Massachusetts (1999)

[6] A. Damasio, 2000. The Feeling of What Happens: Body, Emotion and the Making of Consciousness. MIT Press.

[7] A. Galton, Operators vs Arguments: The Ins and Outs of Reification. Synthese, vol. 150, 415-441 (2006)

[8] J. Heil, Philosophy of Mind. Routledge (2000)

[9] K. Haigh, J. Phelps, An Open Agent Architecture for Assisting Elder Independence. In Proceedings of Autonomous Agents and Multi Agent Systems. ACM Press, 578-586 (2002)

[10] C.M.Jonker, J.Treur, A Temporal-Interactivist Perspective on the Dynamics of Mental States. Cognitive Systems Research Journal, vol. 4, 137-155 (2003)

[11] J.Kim, Philosophy of Mind. Westview Press (1996)

[12] B.F. Skinner, The generic nature of the concepts of stimulus and response. Journal of General Psychology, 12, 40-65 (1935)

[13] A. Tarski, A Lattice-Theoretical Fixpoint Theorem and its Applications. Pacific Journal of Mathematics, 5, 285-309 (1955)

[14] A. Sharpanskykh 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)

Appendix

Proof sketch for Lemma 1

By Definitions 4 and 5: $B_1(X) = X \cup X' \setminus (\text{STRATUM}(X, n) \cup \{ \varphi \in \text{STRATUM}(X, n\text{-}1) \mid \exists \psi \in \text{STRATUM}(X, n)$ AND $\text{head}(\varphi)$ is a subformula of the $\text{body}(\psi)\})$. Each formula ξ in X' is defined in a stratum $k \leq n\text{-}1$. Thus, $B_1(X)$ is stratified using n-1 strata. Then, $B(X)= B_1^{n-1}(X)$ is stratified using one stratum.

Proof sketch for Proposition 2

To prove that X refines $B_1(X)$ the satisfaction of two criteria should be shown:

(1) $X \models B_1(X)$.

(2) For each model $<I, v>$ of $B_1(X)$ an expanded model $<I', v'>$ exists which is a model of X.

The first criterion is proven as follows. From Definition 5 it follows that $B_1(X) \subseteq A_1(X)$. Then, to prove that $X \models B_1(X)$ it is sufficient to show that $X \models A_1(X)$. From Definition 4 it holds $A_1(X) = X \cup X'$. Thus, if $X \models X'$, then $X \models B_1(X)$. According to Definition 6, each formula φ' in set X' is obtained from a formula φ in X by a substitution of formulae that are equivalent in X. By Lemma 2 it holds $\varphi' \Leftrightarrow \varphi$. Thus, each formula in X' is equivalent to some formula from X; therefore, $X \models X'$. From this follows that $X \models B_1(X)$.

To prove the second refinement criterion it is required to identify for each model M of $B_1(X)$ an expanded model M' in which all formulae of X are true. To construct the interpretation I is expanded with the definitions of the domains $D_1, D_2, \ldots, D_n$ corresponding to the sorts of Ont, the interpretations of the constant symbols, functions and relations from Ont.