

Simulation of Conditioning Mechanisms in Agents

Tibor Bosse¹, Catholijn M. Jonker², and Jan Treur¹

¹*Vrije Universiteit Amsterdam, Department of Artificial Intelligence
{tbosse, treur}@cs.vu.nl, http://www.cs.vu.nl/~{tbosse, treur}*

²*Radboud Universiteit Nijmegen, Nijmegen Institute for Cognition and Information
C.Jonker@nici.ru.nl, http://www.nici.ru.nl/~catholj*

Abstract — In order to create adaptive Agent Systems with abilities matching those of their biological counterparts, a natural approach is to incorporate classical conditioning mechanisms into such systems. However, existing models for classical conditioning are usually based on differential equations. Since the design of Agent Systems is traditionally based on qualitative conceptual languages, these differential equations are often not directly appropriate to serve as an input for Agent System design. To deal with this problem, this paper introduces a formal approach to describe and analyse the dynamics of a conditioning process at a neural level. The approach is based on logical specification and analysis of dynamic properties of conditioning. The approach is illustrated for the case of *Aplysia*.

I. INTRODUCTION

Intelligent Agents often operate in dynamic and uncertain environments. Therefore, an important challenge for Agent-Oriented Software Engineering (AOSE) is to incorporate learning mechanisms into Agent Systems. A basic learning mechanism that can be found in many organisms is *classical conditioning* [9]. Thus, in order to create Intelligent Agents Systems with abilities matching those of their biological counterparts, a natural approach is to build classical conditioning into such systems, e.g., [1].

However, in the literature classical conditioning is usually described and analysed informally. If formalisation is used, this is often based on mathematical models using differential equations, e.g., Dynamical Systems Theory [10]. In contrast, Agent-Based Systems traditionally make use of logical, conceptual languages, such as Golog [11] or 3APL [3]. Most of these languages are good for expressing qualitative relations, but less suitable to work with complex differential equations. Therefore, using mathematical models as a direct input for the design of Agent Systems is not trivial.

To bridge the gap between the quantitative nature of existing conditioning models and the conceptual, logical type of languages typically used to design Agent Systems, this paper introduces a logical approach for the analysis and formalisation of conditioning processes that combines qualitative and quantitative concepts, cf. [7]. Using this approach, the dynamics of conditioning can be analysed from two perspectives: the perspective of *externally* observable behaviour and the perspective of *internal* mechanisms to

realise the behaviour. From the external perspective the dynamics of the observed behaviour can be analysed, i.e., how during a history of learning experiences the behaviour is changing. From an internal perspective the dynamics of the actual underlying neural mechanisms that play a role can be investigated and the behaviour they generate determined. One of the contributions of this paper is to relate the dynamics of models for these (internal) neural mechanisms to the dynamics of the externally observable behaviour.

As an alternative to neural models, other kinds of models of internal mechanisms can be designed and analysed. Such internal models have been developed within different areas, varying from symbolic to connectionist and dynamical systems models. According to [5], behaviour can be described from three different perspectives:

- 1) biochemical
- 2) physiological/neural
- 3) behavioural

In general, models that belong to category 1) tend to get extremely complex and therefore not easy to handle. Instead, models that belong to category 3) can be manageable, but these models are usually not executable, and therefore not suited for simulation. To solve both problems, this paper introduces a high-level modelling approach in which a neural description of conditioning still yields a manageable model. If the actual underlying neural mechanisms are taken as a point of departure to analyse conditioning, the sea hare *Aplysia* is an appropriate species to study, since its neural mechanisms have been well-investigated; cf. [4]. In this paper it will be shown how our modelling approach can be used to simulate *Aplysia*'s neural mechanisms underlying conditioning. The results of such a simulation can be used to create requirements in AOSE.

An overview of the paper is as follows. In Section 2 the high-level modelling approach is briefly introduced. Section 3 introduces the case study and the state properties for this case study. In Section 4 the (executable) local dynamic properties describing basic mechanisms for the case study are presented; simulations on the basis of these local dynamic properties are discussed in Section 5. In Section 6 the interlevel relations between dynamic properties of the externally observable behaviour and the local properties describing the internal mechanisms are discussed. Section 7 concludes the paper with a discussion.

II. MODELLING APPROACH

To formally specify dynamic properties that express criteria for representational content from a temporal perspective an expressive language is needed. Dynamics will be described in the next section as evolution of *states* over time. The notion of state as used here is characterised on the basis of an ontology defining a set of state properties that do or do not hold at a certain point in time. Dynamic properties can be formulated that relate a state at one point in time to one or more states at other points in time. A simple example is the following dynamic property specification:

at any point in time t_1 if the agent observes rain at t_1 , then there exists a point in time t_2 after t_1 such that at t_2 the agent has internal state property s

Here, for example, s can be viewed as a sensory representation of the rain. To express such dynamic properties, and other, more sophisticated ones, the temporal trace language TTL is used, cf. [7]. Within this language, explicit references can be made to time points and traces. Here *trace* or *trajectory* over an ontology Ont is a time-indexed sequence of states over Ont . The sorted predicate logic temporal trace language TTL is built on atoms referring to, e.g., traces, time and state properties. For example, ‘in the internal state of agent A in trace γ at time t property s holds’ is formalised by $state(\gamma, t, internal(A)) \models s$. Here \models is a predicate symbol in the language, usually used in infix notation, which is comparable to the *Holds*-predicate in situation calculus. Dynamic properties are expressed by temporal statements built using the usual logical connectives and quantification (for example, over traces, time and state properties).

To be able to perform some (pseudo-)experiments, a simpler temporal language has been used to specify simulation models in a declarative manner. This language (the *leads to* language) enables to model direct temporal dependencies between two state properties in successive states. This executable format is defined as follows. Let α and β be state properties of the form ‘conjunction of atoms or negations of atoms’, and e, f, g, h non-negative real numbers. Then the notation $\alpha \rightarrow_{e, f, g, h} \beta$, means:

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

For a precise definition of the *leads to* format in terms of the language TTL, see [7]. A specification of dynamic properties in *leads to* format has as advantages that it is executable and that it can often easily be depicted graphically. The *leads to* format has shown its value especially when temporal or causal relations in the (continuous) physical world are modelled and simulated in an abstract, non-discrete manner; for example, the intracellular chemistry of *E. coli* [6].

III. THE APLYSLIA CASE STUDY

In this section the *Aplysia* case study will be described, both from an external and an internal perspective. In addition, the state properties used to model the example are presented.

A. External Perspective

Aplysia is a sea hare that is often used to do experiments. It is able to learn; for example, it performs classical conditioning in the following manner. This (a bit simplified) description is mainly based on [4], pp. 155-156. First the (learning) behaviour viewed from an external perspective is addressed. Initially, *before learning*, the following behaviour is shown:

- a tail shock leads to a response (contraction)
- a light touch on its siphon is insufficient to trigger such a response

Now suppose the following experimental protocol is undertaken. In each trial the subject is touched lightly on its siphon and then, shocked on its tail (as a consequence it responds). It turns out that after a number of trials (three in the current example) the behaviour has changed. Thus, *after learning*, *Aplysia* shows the following behaviour:

- the animal also responds (contracts) on a siphon touch.

Note that, to characterise behaviour, there is a difference between the *learned* behaviour (which is simply an *adapted* stimulus-response behaviour) and the *learning* behaviour, which is a form of *adaptive* behaviour, no stimulus-response behaviour. To specify such behaviours the following sensor and effector states are used: *tail_shock*, *siphon_touch*, *contraction*. In terms of these state properties the following global dynamic properties can be specified in *leads to* format:

GP1 *tail_shock* $\rightarrow_{e, f, g, h}$ *contraction* (always)

GP2 *siphon_touch* $\rightarrow_{e, f, g, h}$ *contraction* (after learning)

However the learning behaviour itself is not expressible in *leads to* format, but it is in TTL format:

GP3 at any point in time t ,
if a siphon touch occurs
and at three different earlier time points t_1, t_2, t_3 ,
a siphon touch occurred, directly followed by a tail shock
then it will contract

Formally:

$\forall \gamma \forall t \text{ state}(\gamma, t) \models \text{siphon_touch} \ \& \ \exists t_1, t_2, t_3, u_1, u_2, u_3 \ t_1 < u_1 < t_2 < u_2 < t_3 < u_3 < t \ \& \ \text{state}(\gamma, t_1) \models \text{siphon_touch} \ \& \ \text{state}(\gamma, u_1) \models \text{tail_shock} \ \& \ \text{state}(\gamma, t_2) \models \text{siphon_touch} \ \& \ \text{state}(\gamma, u_2) \models \text{tail_shock} \ \& \ \text{state}(\gamma, t_3) \models \text{siphon_touch} \ \& \ \text{state}(\gamma, u_3) \models \text{tail_shock} \ \Rightarrow \ \exists t' \geq t \ \text{state}(\gamma, t') \models \text{contraction}$

As can be seen, the temporal complexity of the learning behaviour specification is much higher than that of the learned behaviour.

B. Internal Perspective

Roughly spoken the internal neural mechanism for *Aplysia*’s conditioning can be depicted as in Figure 1; adjusted from [4].

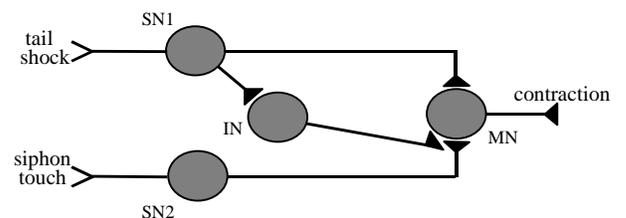


Fig. 1. Neural mechanisms

A tail shock activates a sensory neuron SN1. Activation of this neuron SN1 activates the motoneuron MN; activation of MN makes the sea hare move. A siphon touch activates the sensory neuron SN2. Activation of this sensory neuron SN2 normally does not have sufficient impact on MN to activate MN. After learning, activation of SN2 has sufficient impact to activate MN. In addition, activation of SN1 also leads to activation of the intermediary neuron IN. If both SN2 and IN are activated simultaneously, this changes the synapse between SN2 and MN: it makes that in this synapse more neurotransmitter is produced if SN2 is activated. After a number of times this leads to the situation that also activation of SN2 yields activation of MN.

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

- SN1 sensory neuron 1 is activated
- SN2 sensory neuron 2 is activated
- IN intermediary neuron IN is activated
- MN motoneuron MN is activated
- S(r) the synapse between SN2 and MN is able to produce an amount r of neurotransmitter

The dynamics of these internal state properties involve temporal *leads to* relationships, which are analysed in more detail in the next section.

IV. LOCAL DYNAMIC PROPERTIES

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

- LP1 tail_shock $\rightarrow_{e,f,g,h}$ SN1
- LP2 siphon_touch $\rightarrow_{e,f,g,h}$ SN2
- LP3 SN1 $\rightarrow_{e,f,g,h}$ IN \wedge MN
- LP4 $S(r) \wedge SN2 \wedge IN \wedge r < 4 \rightarrow_{e,f,g,h} S(r+1)$
- LP5 $S(4) \wedge SN2 \rightarrow_{e,f,g,h}$ MN
- LP6 MN $\rightarrow_{e,f,g,h}$ contraction
- LP7 $S(r) \wedge \text{not } S(r+1) \wedge r < 4 \rightarrow_{e,f,g,h} S(r)$
- LP8 $S(4) \rightarrow_{e,f,g,h} S(4)$
- LP9 start $\rightarrow_{e,f,g,h} S(1)$

In Figure 2 an overview of these properties is given in a graphical form. Here, the circles denote state properties and the arrows denote dynamic properties.

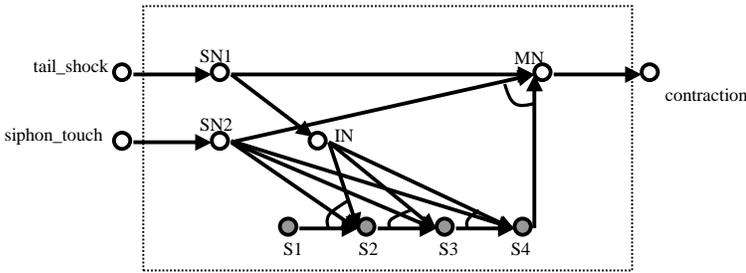


Fig. 2. Overview of the basic dynamics of the simulation model

Note that this model is based on a number of simplifications. For example, it is assumed that after exactly 4 steps the strength of the synapse between SN2 and MN is

maximal, and that there is no extinction. However, since our modelling approach supports the use of quantitative concepts (such as real numbers and mathematical operations), it is easy to incorporate such features in the model. A rather straightforward way to do this would be to replace LP4 through LP8 by the following local properties:

- LP4 $S(r) \wedge SN2 \wedge IN \rightarrow_{e,f,g,h} S(\beta^*(K-r) + (r^*\epsilon))$
- LP5 $S(r) \wedge SN2 \wedge r > t \rightarrow_{e,f,g,h}$ MN
- LP7 $S(r) \wedge \text{not } SN2 \rightarrow_{e,f,g,h} S(r^*\epsilon)$
- LP8 $S(r) \wedge \text{not } IN \rightarrow_{e,f,g,h} S(r^*\epsilon)$

Here, β indicates the learning rate, K indicates the maximal strength of the synapse between SN2 and MN (e.g., 4), ϵ indicates the extinction rate, and t indicates the minimum threshold of S needed to have SN2 influence MN. For all values, real numbers can be used.

V. SIMULATION

A special software environment has been created to enable the simulation of executable models. Based on an input consisting of dynamic properties in *leads to* format, the software environment generates simulation traces. An example of such a trace can be seen in Figure 3. Here, time is on the horizontal axis, the state properties are on the vertical axis. A dark box on top of the line indicates that the property is true during that time period, and a lighter box indicates that the property is false. This trace is based on all local properties identified above. In property LP1 and LP2 the values (0,0,1,3) have been chosen for the timing parameters e , f , g , and h . In all other properties, the values (0,0,1,1) have been chosen.

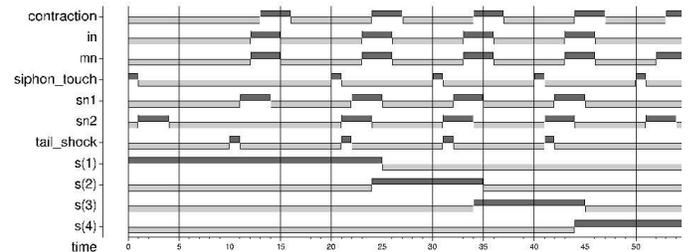


Fig. 3. Example simulation trace

As can be seen in Figure 3, at the beginning of the trace the organism has not performed any conditioning. The initial siphon touch it receives does lead to the activation of sensory neuron SN2, but the synapse between SN2 and motoneuron MN does not produce much neurotransmitter yet (indicated by internal state property S(1)). Thus, the activation of SN2 does not yield an activation of MN, and consequently no external action follows. In contrast, it is shown that a shock of the organism's tail does initially lead to the external action of contraction. This can be seen in Figure 3 between time point 10 (when the tail shock occurs) and time point 13 (when the animal contracts). After that, the actual learning phase starts. This phase consists of a sequence of three trials where a siphon touch is immediately followed by a tail shock. As a result, the sensory neuron SN2 is activated at the same time as the

intermediary neuron IN, which causes the synapse to change so that it can produce an increased amount of neurotransmitter each time SN2 is activated. Such a change in the synapse is indicated by a transition from one internal state property to another (first from S(1) to S(2), then to S(3), and finally to S(4)). As soon as internal state property S(4) holds (see time point 44), the conditioning process has been performed successfully. From that moment, *Aplysia*'s behaviour has changed: it also contracts on a siphon touch.

As a side remark, notice that the amount of trials (three) is kept low to keep the example simple. However, similar experiments have been performed with a case of 1000 learning steps. Since the abstract way of modelling used for the simulation is not computationally expensive, also these simulations took no more than 90 seconds. In addition, our simulation approach has possibilities to incorporate real numbers in state properties, and to perform complex mathematical operations with these numbers. This makes it more expressive than more traditional forms of temporal logic.

VI. INTERLEVEL RELATIONS

As mentioned in the Introduction, an important challenge is to verify whether the internal (neural) mechanisms of adaptive agents (such as *Aplysia*) entail the expected behaviour from an external perspective. Or, in terms of the present case study, can it be proven that the local dynamic properties specified in Section 4 together imply the global dynamic property GP3 (expressing the learning behaviour, see Section 3A)?

Verification of such interlevel relations can be performed in two manners: 1) by automated checks and 2) by mathematical proof. To achieve the former, additional software has been developed that takes traces and formally specified properties as input and checks whether a property holds for a trace (see [7]). Using automatic checks of this kind, all properties presented in this paper have been checked against traces such as depicted in Figure 3. In particular, dynamic property GP3 has been checked successfully. Usually, the duration of such checks does not take more than a second. However, note that these checks are only an empirical validation, they are no exhaustive proof as, e.g., model checking is. Currently, the possibilities are explored to combine TTL with existing model checking techniques.

To verify the relations between local and global dynamic properties exhaustively, mathematical relations can be (and have been) specified by hand between properties at different levels. An example of such a relation is described by the following implication:

$$\text{LP1 through LP9 \& CWA} \quad \Rightarrow \quad \text{GP3}$$

This relation states that the local properties together imply the global property GP3. Moreover, one additional property is introduced, i.e., the Closed World Assumption (CWA). For space limitations, the full proof of this relationship is left out. However, interested readers can find it at the following URL: <http://www.cs.vu.nl/~tbosse/aplysia/>.

VII. DISCUSSION

To bridge the gap between the quantitative nature of existing conditioning models and the conceptual, logical type of languages typically used to design Agent Systems, this paper introduces an analysis of the dynamics of classical conditioning from a logical perspective. It provides two types of temporal logical formalisations, one at the behavioural level, and one at the neurological level; cf. [5]. The neural processes of the *Aplysia* case study (cf. [4]) have been formalised by identifying executable local dynamic properties for the basic dynamics of *Aplysia*'s neural conditioning mechanism. On the basis of these local properties simulations have been made. Moreover, it is shown how the descriptions at different levels can be logically related to each other, which can be considered as a formalisation of the (inter-level) reduction relations between the two levels. The analysis results can be used to extend existing methodologies for AOSE by including learning mechanisms as observed in nature.

Concerning related work, in [2] another formal model is described of the dynamics of conditioning processes, using a similar modelling approach. However, that paper focuses on human conditioning, based on existing literature such as [8]. Instead, the current paper focuses on the specific case of *Aplysia*, of which the neural mechanisms are much simpler and therefore better understood. As a consequence, the model presented in the current paper is at a neural level, whereas the model of [2] is at a functional level. Another difference is that their model concentrates more on the temporal aspects of the conditioning.

REFERENCES

- [1] Balkenius, C. and Morén, J. Dynamics of a classical conditioning model. *Autonomous Robots*, 7, 1999, 41-56.
- [2] Bosse, T., Jonker, C.M., Los, S.A., Torre, L. van der, and Treur, J. Formalisation and Analysis of the Temporal Dynamics of Conditioning. In: J.P. Mueller and F. Zambonelli (eds.), *Proceedings of the Sixth International Workshop on Agent-Oriented Software Engineering, AOSE'05*. To appear, 2005.
- [3] Dastani, M., Dignum, F., and Meyer, J-J.Ch. *3APL: A Programming Language for Cognitive Agents*. ERCIM News, European Research Consortium for Informatics and Mathematics, Special issue on Cognitive Systems, No. 53, 2003.
- [4] Gleitman, H. *Psychology*, W.W. Norton & Company, New York, 1999.
- [5] Hawkins, R.D., and Kandel, E.R. Is There a Cell-Biological Alphabet for Simple Forms of Learning? *Psychological Review*, vol. 91, 1984, pp. 375-391.
- [6] Jonker, C.M., Snoep, J.L., Treur, J., Westerhoff, H.V., and Wijngaards, W.C.A. BDI-Modelling of Intracellular Dynamics. In: A.B. Williams and K. Decker (eds.), *Proc. of the First International Workshop on Bioinformatics and Multi-Agent Systems, BIXMAS'02*, 2002, pp. 15-23.
- [7] Jonker, C.M., Treur, J., and Wijngaards, W.C.A. A Temporal Modelling Environment for Internally Grounded Beliefs, Desires and Intentions. *Cognitive Systems Research Journal*, vol. 4, 2003, pp. 191-210.
- [8] Machado, A. Learning the temporal Dynamics of Behaviour. *Psychological Review*, vol. 104, 1997, pp.241-265.
- [9] Pavlov, I.P. *Conditioned reflexes*. Oxford: Oxford University Press, 1927.
- [10] Port, R.F. and van Gelder, T.J. *Mind as Motion: Explorations in the Dynamics of Cognition*. MIT Press, Cambridge, Mass., 1995.
- [11] Reiter, R. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.