Berthouze, L., Kaplan, F., Kozima, H., Yano, H., Konczak, J., Metta, G., Nadel, J., Sandini, G., Stojanov, G. and Balkenius, C. (Eds.)
Proceedings of the Fifth International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems
Lund University Cognitive Studies, 123. ISBN 91-974741-4-2

Symbol manipulation by internal simulation of perception and behaviour

Michel van Dartel* **Eric Postma***
*Universiteit Maastricht, P.O.Box 616, 6200 MD, {mf.vandartel,postma}@cs.unimaas.nl

1. Symbol manipulation in situated robots

In traditional cognitive science, cognition amounts to symbol manipulation (Newell and Simon, 1972). Symbol manipulation is the processing of symbolic descriptions to produce an output that benefits an objective. Moreover, embodied models of cognitive development concern situated robots that do not rely on symbolic descriptions or on their manipulation (see, e.g., (Schlesinger, 2003)). In this paper, we introduce an embodied model of cognitive development that does rely on symbol manipulation. It has been claimed that internalised interaction with the external environment constitutes symbol manipulation (Phaf and Wolters, 1997). A mechanism to internalise such interaction was suggested in (Hesslow, 2002) as part of the *simulation hypothesis*. The simulation hypothesis states that conscious thought is based on the ability to simulate perception and behaviour internally. Such internal simulation consists of imagining actions and their consequences without actually performing them.

In (Ziemke et al., 2005) it was demonstrated that situated robots can perform simple maze-following tasks on the basis of internal simulation. In a study involving situated agents engaged in an active categorisation task (van Dartel et al., 2005), we have shown that situated robots with the ability to simulate perception and behavior internally can outperform robots that do not have this ability (van Dartel et al., 2004). Situated robots can thus exploit their ability to simulate perception and behaviour internally. As was stated above, this ability may also constitute symbol manipulation.

Symbol manipulation is often associated with planning (Newell and Simon, 1972), a skill that is generally regarded as high-level cognition (Cooper, 2002). To plan ahead in time, one needs to be able to represent the current state of the task in symbols and extrapolate to future states by manipulation of these symbols. The Tower of London (ToL) task is a typical planning task (Shallice, 1982) and a standard neuropsychological test to assess frontal lobe damage (Kolb and Whishaw, 1983), which impairs planning performance (Baddeley, 1986).

In this paper we investigate whether symbol manipulation tasks can be solved by robots with the ability to simulate perception and behaviour internally. In order to do so, we constructed the situated model of the Tower of London task called STOL and formulated the first research question: *Does the ability to perform symbol manipulation by internal simulation enhance performance on the ToL task in SToL?* If this appears to be the case, we will try to answer the second research question: *How does the symbol manipulation by internal simulation in SToL enhance performance on the ToL task?*

The ToL task will be discussed in more detail in section 2. The STOL model is described in section 3. The experiments conducted with the STOL model and the results found are reported in section 4. An analysis of the internal simulation of the optimised robot will be conducted in section 5. Finally, the results are discussed and concluded upon in section 6.

2. The Tower of London task

The Tower of London (ToL) task is often employed to test a subject's ability to plan ahead (Shallice, 1982), and proves especially useful to test the development of children's problem solving abilities (Krikorian et al., 1994, Bull et al., 2004). In the ToL task, subjects are asked to change a given starting configuration of three balls on three pegs to a pre-defined goal configuration *in the least possible number of moves*. Figure 1 shows four possible states of the ToL task labelled 0 to 3. Each state consists

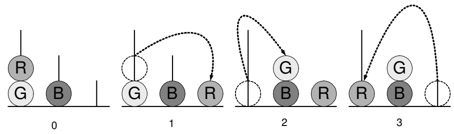

Figure 1: Illustration of the actions required to reach goal state 3 from the initial state 0 through intermediate states 1 and 2 in the ToL task. The arrows represent movements of balls. The shaded circles are labelled according to their colour (R, G, and B, for red, green, and blue, respectively. Dashed circles represent previous ball positions.

of a configuration of the three balls on the three pegs. The balls are labelled according to their colour (R, G, and B, for red, green, and blue, respectively). State 0 is the initial state, states 1 and 2 are intermediate states, and state 3 is the goal state. The figure illustrates how, when starting from the initial state 0, a subject can reach goal state 3 by visiting two intermediate states (1 and 2) without violating the constraints of the ToL task. All possible goal states can be reached from any initial state of the ToL task. However, successful completion of the task and the number of successive moves needed to reach the goal configuration depend on a subject's ability to plan ahead in time. The ToL task is considered a typical high-level planning task, because "successful completion requires the participant to 'look ahead' and solve the problem cognitively before actually moving the balls" ((Bull et al., 2004), p.743). The complexity of a ToL task can be varied by changing the starting state and/or the goal state. A standard set of 12 problems was defined in (Shallice, 1982).

3. A situated model of the Tower of London task: STOL

Below, we describe STOL in terms of the task, the robot, and the evolutionary algorithm that is used to optimise the robot's performance. In STOL, a robot is optimised to perform the 12 ToL problems defined by (Shallice, 1982).

The robot in STOL is able to perceive the current ToL state and can respond by moving a ball from one peg to another. The model is situated because the robot can observe the consequences of its own actions (Pfeifer and Scheier, 1999), and use these to learn how to reach the goal configuration. Since in (Nolfi and Floreano, 2000) it was shown that embodiment can be simulated, we chose to simulate the embodiment of the robot (cf., (van Dartel et al., 2005, Ziemke et al., 2005)) rather than us a physical robot. The robot consists of a neurocontroller that receives information about the configuration of the balls on the pegs through its sensors. The robot changes the position of a ball according to the output of the neurocontroller and the constraints of the ToL task. The interaction between the robot and the ToL task is realised by encoding ToL states and ToL actions in a straightforward way in the sensors and actuators of the robot.

Figure 2 provides an illustration of the encoding of the ToL state (current configuration of the ToL), goal state (goal configuration of the ToL problem), and expected state (to be described below) into the sensor array. In

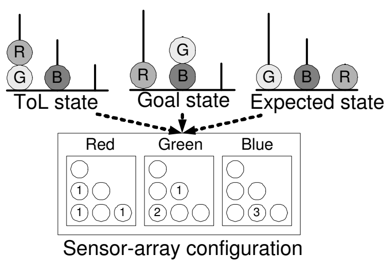

Figure 2: Example of encoding a ToL state, a goal state, and an expected state in activation of the sensor array.

figure 2, the current, goal, and expected states are superimposed resulting in the activation pattern shown in the sensor-array configuration. The actuator array of the robot consists of six actuators. The first three actuators encode the colour of the ball to be moved (red, green, blue). The last three actuators encode the peg to which the ball should be moved (left, middle, right).

The standard neurocontroller of a robot in STOL (henceforth referred to as neurocontroller A) consists of a simple recurrent neural network (RNN). The eighteen input nodes of the RNN sample the sensory activations. The input is mapped onto h hidden nodes and an equal number of context nodes. The six output nodes of the RNN encode the actions in the actuator array of the robot.

In section 1. we argued that a symbol-manipulating robot may require an internal simulation mechanism. Therefore, we test the robot with a neurocontroller with an internal simulation mechanism (neurocontroller B). The internal simulation mechanism consists of an additional output layer and feedback connection to the input layer of the neurocontroller. Figure 3 illustrates the architecture of neurocontroller B. The additional output layer

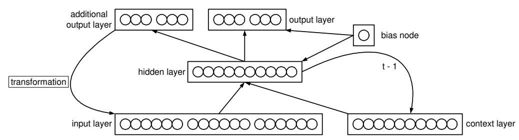

Figure 3: Architecture of neurocontroller B.

employs the same encoding as the plain output layer. However, the output of the additional output layer is not used to reconfigure the ToL, but is internally processed to generate an expected state of the ToL: the ToL state that would result from the action if it was actually performed. This process continues until the plain output of the neurocontroller provides an action that satisfies the constraints of the ToL.

We optimised the performance of the robot on the ToL task using a standard evolutionary algorithm that is similar to the one used in (van Dartel et al., 2005). The fitness function F was defined as $F = ((C+S)\times 1000) - M$, with F the fitness of a robot, C the total number of balls positioned correctly upon termination of each problem, S the number of the 12 ToL problems that were solved, and M the total number of moves that the robot made to solve those problems. In all experiments reported in this paper, the number of generations is $50,000$ and each generation consists of 100 robots. Our performance measures are: (i) $\overline{S}$, the average proportion of solved ToL problems, and (ii) $\overline{M}$, M_{min} (of the solved problems) divided by the average number of moves it used to solve the problems.

4. Experiment and results

In the experiment, STOL was used to determine the performance of the situated robot on Shallice's (1982) test with neurocontrollers A and B. For both neurocontrollers, the number of hidden nodes, h, (which is equal to the number of context nodes) was optimised. Starting from $h = 4$, the value for h was repeatedly increased by 2, until

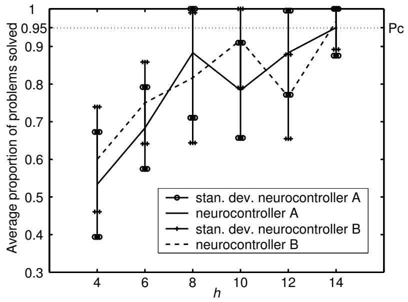

Figure 4: $\overline{S}$ as a function of the number of hidden nodes and context nodes (h) for neurocontroller A (solid line) and neurocontroller B (dashed line).

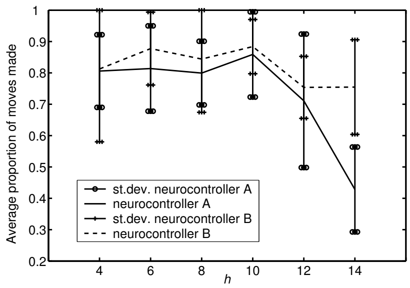

Figure 5: $\overline{M}$ as a function of the number of hidden nodes and context nodes (h) for neurocontroller A (solid line) and neurocontroller B (dashed line).

$\overline{S}$ reached the criterium $p_c \geqq 0.95$. Each individual evolution was replicated 5 times to obtain a good estimate of the optimised robot's mean performance. Figure 4 illustrates the performance of the robot with neurocontroller A and with neurocontroller B expressed in $\overline{S}$, as a function of h. The figure shows that the robot with either neurocontroller achieves the performance criterium $\overline{S} \geq 0.95$ when $h = 14$. Figure 5 illustrates the performance of the robot expressed in $\overline{M}$, as a function of h. Of main relevance is the performance for $h = 14$, because for that value, the robots solve $\overline{S} \geq 0.95$ ToL problems. Strikingly, a significant difference in the average proportion of moves made ($\overline{M}$) by the robot between neurocontroller A and neurocontroller B is observed for $h = 14$.

In summary, the results show that the robot with neurocontroller B uses significantly less moves than with neurocontroller A when the proportion of problems solved is > 0.95. Apparently, when almost all 12 problems of the test are solved by the robot, it can do so more efficiently when equipped with an internal simulation mechanism. Our results provide an answer to the first research question. The ability to perform symbol manipulation by internal simulation enhances performance on the ToL task in STOL.

5. Analysis of internal simulation

To investigate how symbol manipulation by internal simulation in STOL enhances performance on the ToL task we analysed the internal simulation of the optimised robot with neurocontroller B. A typical example of sequences of expected states that are generated by the internal simulation mechanism while solving a ToL problem is shown in table 1. The table shows the sequence of ToL states that are visited by the robot while solving a ToL problem in the left column (top to bottom), and the associated sequences of expected states that are generated by the internal simulation mechanism in the right column (left to right). The arrows in the right column indicate the transitions between the expected states. The number of expected states that are generated before the ToL state is changed by the robot varies per ToL state. The reason is that the neurocontroller does not always produce actions that satisfy the constraints of the ToL, and the neurocontroller generates expected states until such an action is produced. Table 1 gives rise to the three observations discussed below.

ToL state	Expected state							
R G B - (starting state)	R G B -	→	R G B -	→	R G B -	→	B R G - -	
R G B -	R G B -	→	R G B -	→	R G B -			
R - B G	R G B	→	R G B -					
R B G	B R - G	→	B R - G	→	R B G	→	B R - G	
G R B -	G R - B	→	B G R - -	→	B G R - -	→	G R - B	→
	continued		B G R - -	→	B G R - -	→	B G R - -	→
	continued		B G R - -	→	B G R - -	→	B G R - -	
B G R - - (goal state)								

Table 1: The subsequent ToL states that are visited (left column, top to bottom) and the associated expected states that are generated by the optimised robot while solving problem 9 of the task (right column, left to right).

The first observation is that the robot changes the expected state independently from changing the ToL state, i.e., the robot can make a simulated move in the absence of a real move. The second observation is that the expected states occuring during the state preceeding the goal state (in the sixth row of table 1) also occurs as a future ToL state (the goal state), which suggests that moves are sometimes internally simulated before they are performed. However, during the ToL state preceeding the goal state the expected state indicating the move towards the goal state is generated eight times before it is realised. This suggests that the internal simulation mechanism serves the function of building up sufficient activation in the neurocontroller to produce a certain move, rather than to simulate a future move. The third observation is that most of the the expected states either match the current ToL state or never occur as ToL states at all.

On the basis of these three observations, we argue that the success of the robot's behaviour depends on its ability to generate and exploit expected states. The analysis allows us to answer the second research question. The symbol manipulation by internal simulation in STOL enhances performance on the ToL task by generating expected states that either predict future states or build up activation to produce a certain move.

6. Discussion and Conclusions

To evaluate the performance of the robot in STOL reported in section 4., we compared the performance of the robot with that of human subjects. In figure 6 we plotted the performances of the robot (for $h = 14$) with neurocontroller A and with neurocontroller B together with those of human subjects on Shallice's test reported by (Owen et al., 1990) in figure. The figure shows that the performance of the robot with neurocontroller B (right) better matches the performance of human subjects on the ToL task than the performance of the robot with neurocontroller A (left).

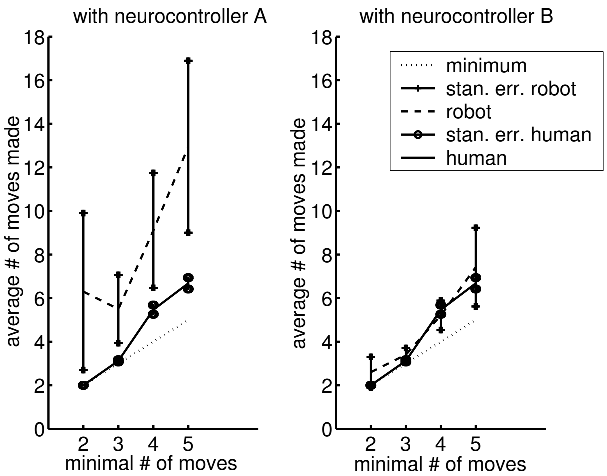

Figure 6: Average number of moves made by the robot with neurocontroller A (left), the robot with neurocontroller B (right), and human subjects (left and right) over the minimal number of moves in which a problem could be solved (Redrawn from Owen *et al.* (1990)).

On the basis of the results reported in section 4., the analysis reported in section 5., and the comparison shown in figure 6 we conclude that the performance of infants on symbol manipulation tasks may rely on their ability to simulate perception and behaviour internally. Therefore, embodied models of development that involve symbol manipulation should incorporate a mechanism for the internal simulation of perception and behaviour.

References

Baddeley, A. D. (1986). *Working memory*. Oxford University Press, Oxford, NY.

Bull, R., Espy, K. A., and Senn, T. E. (2004). A comparison of performance on the towers of london and hanoi in young children. *J Child Psychol Psychiatry*, 45(4):743754.

Cooper, R. (2002). *Modelling High-Level Cognitive Processes*. Lawrence Erlbaum Associates, Mahwah, NJ.

Hesslow, G. (2002). Conscious thought as simulation of behaviour and perception. *Trends Cog Sci*, 6:242–247.

Kolb, B. and Whishaw, I. (1983). Performance of schizophrenic patients on tests sensitive to left or right frontal temporal or parietal function in neurological patients. *J Nerv Ment Dis*, 171:435443.

Krikorian, R., Bartok, J., and Gay, N. (1994). The tower of london procedure: A standard method and developmental data. *J Cli Exp Neuropsychol*, 16:840–850.

Newell, A. and Simon, H. (1972). *Human Problem Solving*. Prentice-Hall, Englewood Cliffs, NJ.

Nolfi, S. and Floreano, D. (2000). *Evolutionary robotics*. MIT Press, Cambridge, MA.

Owen, A. M., Downes, J. J., J.Sahakian, B., Polkey, C. E., and Robbins, T. W. (1990). Planning and spatial working memory following frontal lobe lesions in man. *Neuropsychol*, 28(10):1021–1034.

Pfeifer, R. and Scheier, C. (1999). *Understanding Intelligence*. MIT Press, Cambridge, MA.

Phaf, R. H. and Wolters, G. (1997). A constructivist and connectionist view on conscious and unconscious processes. *Phil Psychol*, 10:287–307.

Schlesinger, M. (2003). A lesson from robotics: Modeling infants as autonomous agents. *Adaptive Beh*, 11:97–107.

Shallice, T. (1982). Specific impairments of planning. *Phil Trans Royal Soc London*, B 298:199–209.

van Dartel, M. F., Postma, E. O., and van den Herik, H. J. (2004). Categorisation through internal simulation of perception and behaviour. In Schomaker, L., Taatgen, N., and Verbrugge, R., (Eds.), *Proc BNAIC04*, pages 187–194. Groningen, The Netherlands.

van Dartel, M. F., Sprinkhuizen-Kuyper, I. G., Postma, E. O., and van den Herik, H. J. (to appear, 2005). Reactive agents and perceptual ambiguity. *Adaptive Beh*.

Ziemke, T., Jirenhed, D., and Hesslow, G. (to appear, 2005). Internal simulation of perception. a minimal neurorobotic model. *Neurocomp*.