Robopsychology Manifesto: Samu in His Prenatal Development

Norbert Bátfai, Renátó Besenczi
Department of Information Technology, University of Debrecen, Hungary
e-mail: batfai.norbert@inf.unideb.hu, besenczi.renato@inf.unideb.hu

***Abstract*—Samu is a disembodied developmental robotic experiment that is intended to resolve the apparent conflict between the embodied and the disembodied approaches to Developmental Robotics, and, as a utopian goal, to become the base of a chatterbot agent which will be able to talk and read in natural language like humans do. In this paper, we outline the design of Samu's mental development process. There are two different levels of development of Samu: one is the level of mental organs, which are small AI algorithms, the other is the chatbot Samu that will use mental organs for solving a given subtask. In this work, we focus on mental organs, and we show that on this level, in certain experiments, Samu is able to acquire knowledge. In addition, we introduce our aspect of robopsychology, which is the development and fine-tuning process of agents which are capable of autonomous mental development. The main goal of this paper is threefold: first, to embed the concept of Samu into Developmental Robotics; second, to present a child development-inspired approach to develop computational mental organs; third, to support the software design of a further prototype of Samu chatbot for testing knowledge sharing between two Samu-type chatbots.**

Keywords: developmental robotics, mental organs, habituation, sensitization, robopsychology

I. INTRODUCTION

The mind-body problem [1] is as old as the history of philosophy. This introduction is based on an informatics interpretation [2] of the mind-body relationship. Suppose that we have a computer which can pass the Turing test [3]. In addition, suppose that this computer does not use quantum mechanics-based models such as the OrchOR [4] model of consciousness to pass the test. In this case, we can obviously say that we have a Turing machine that can pass the Turing test. From an informatics viewpoint, it is trivial to say that the software is the mind and the hardware is the body [5], [6]. But in our special case we can be more precise than this. The body is the universal Turing machine [7, pp. 96] and the mind is our supposed Turing machine that is taken as input to the universal machine. On one hand, this example also demonstrates how near the software is to the hardware since they are both operated as simple Turing machines. On the other hand, it also shows how far the software is from the hardware because the basic operation of the universal Turing machine had been known since Turing's universal simulation theorem [8], but unfortunately nothing has been known about the basic operation of our supposed Turing machine so far.

One key principle of the Developmental Robotics [9], [10] is that a DevRob agent must have a physical body. The supposed example shows that in a given environment, such as the internet, the body can be purely built only in software. The embodied principle can be considered as a thesis about Developmental Robotics. In this sense, a disembodied approach may be considered as an antithesis of that principle.

Samu [11] is a disembodied developmental and family robotic initiative that is intended to solve the apparent contradiction between the embodied and the disembodied approaches to Developmental Robotics, and to become the base for making theoretical investigations and experiments with a chatter bot agent which will be able to talk and read in natural language like humans do. Samu [11] introduces a COP-based (Consciousness Oriented Programming [12]) Q-learning engine to predict the next sentences of a conversation. The idea for the basic architecture of Samu was inspired by the breakthrough paper on deep learning published in Nature in 2015 [13].

During the development of embodied robotic agents, several factors can make the learning process a cumbersome task. The main problem is to achieve a seamless and responsive operation of the body. For example, RoboCup [14] Soccer Humanoid League focuses on the development of artificial soccer. In this situation, learning simple movements, such as kicking the ball or making a move towards the right direction in time, can be considered as a major achievement. In contrast, RoboCup Soccer Simulation League, which is in the research domain of artificial intelligence, provides a nearly realistic soccer experience. The difference between the two is the embodied-disembodied concept. Because of the above mentioned reason, we have decided to choose the disembodied approach for the development of Samu. An other reason is that, in our opinion, the development of the mind and the body-control can be separated. Therefore, any well-defined, complex learning process developed for mind evolution could also be applied to learning proper body usage. So, an agent based on deep Q-learning that is capable of developing its own mind could also be capable of developing its own body in any case where it is applied in a feasible body. By this statement, we directly argue with the fact that the evolution of the mind is impossible without a body. Based on our experiments conducted, we will show that this argument might be valid.

Where can we place Samu in the field of developmental robotics and artificial intelligence? We should note, that in the current development phase of Samu, we are mainly planning Samu's possible operations and application fields, however, for some functions we provide fast prototypes for illustration. So, while the proper positioning can be interpreted on the current phase of development, it may change in the future. We can say that Samu is not a robot in the classic sense, because it has no body with which it can interact with the real world in a physical way. Rather can be considered as an agent that acquires input signals with its sensory input (currently a terminal input) and reinforcement from its supervisor (e.g. its human caregiver or an other Samu agent) and has an influence on the real world with its effector (currently a terminal output). During the process, the learning itself is based on deep Q-learning. Besides, Samu can be considered as an artificial implementation of developmental psychology, because the human learning process and the improvement of infants' mind provide the basic patterns for Samu's mental development.

In this paper, we propose our concept for mental development and learning. In addition, we give an outline of the design of Samu's mental development process as a demonstration. We mainly approach the problem from theoretical side (e.g. child development and computation theory), but in some cases, we give a few working examples with which we can show the current phase of Samu's IT development and mind evolution.

When we refer to Samu as a robot, robotic agent or chat robot, please keep in mind that Samu is a disembodied artificial agent. In addition, we call our experiments by the name of Samu where we test mental organs (e.g. Mental Processing Units, or MPUs) and the chatterbot Samu itself.

A. *Robopsychology on Demand*

Partly based on this present work, [15] has proposed the possibility of the on-demand development of an Asimovian type of robopsychology [16]. We consider our approach to robopsychology as a cross-cutting concept that we would like to weave into the development of Samu. In this paper, we discuss two different levels of development of Samu: the development of computational mental organs and the development of the chat robot Samu. From our viewpoint, the mental organs are small AI algorithms which are intended to learn a well-defined task, e.g. Conway's Game of Life (as described in paragraph II-A1a), and the chatbot Samu which is a software agent that will use mental organs for solving a given subtask. These two levels both use the same COP-based Q-learning engine. The COP-based feature means that the classical Q-learning algorithm is implemented using the prediction of the future as rewarding system. The Q-learning algorithm takes a positive reinforcement if the prediction is correct, otherwise the reinforcement is negative. On the level of mental organs, an organ in question predicts the next state of a small piece of data of its input (for example, in our visual experiments the next pixel of the reality is predicted, see paragraph II-A1a and Fig. 1). On the level of the chatbot Samu, he predicts the next sentence [11]. The reason behind this method is a phenomenon that can be observed even in human conversations: sometimes we finish each other’s sentence. Intuitively, we assume that in some cases we can predict the next sentence in a conversation. This is why we have chosen this method. In the terminology of [15], the software development works on the level of mental organs may be considered as robopsychology activities. For example, fine-tuning the habituation and sensitization process or readjusting the rewarding system to a given task can be considered a robopsychologist's job.

In the following, we also call the mental organs as prenatal development, and the chatbot level is often referred to as postnatal development.

The paper is structured as follows: In the next section, we introduce Samu's possible mental organs, and embed him into the Developmental Robotics research domain. In section III, we show the contact between the mental organs and the chatbot. In section IV, we describe some ideas regarding Samu's postnatal development. We conclude this paper in section V, where we show some future ideas which could be pursued.

II. EMBEDDING SAMU IN DEVELOPMENTAL ROBOTICS

A. *Prenatal Development*

The research of Samu is changing how we think about Samu itself. At the moment, we can also look at Samu as a strongly simplified JIBO [17], [18] type robot that will be simulated and run on the PC of a family. The base of this kind of implementation of Samu will be the project SamuCam [19]. In the following, we introduce the earlier projects that led to this one.

The practical applications of the Turing's simulation theorem (mentioned in the introduction) and similar theorems (see, for example [20] and [21]) provide the basis on which virtual machines currently operate. For example, presently a virtual computer has already been preferred over a physical one for server-side services. From our point of view, it is important that these virtual computers, virtual operating systems are built entirely in software. They are running on simulators or in von Neumann's sense [22] we can say that these systems are based on short codes (aka simulations). Why is this a point of interest? Because the homogeneous structure of the brain [23] suggests that its “suborgans” may be built in its “software”. The last thought of Neumann's unfinished book [22] was broken but it has raised the question of whether the whole body of mathematics is simulated on a primary language of the nervous system. This von Neumannian idea is very impressive. It may even serve as a basis for an answer to Wigner's question about the “unreasonable effectiveness of mathematics in the natural sciences” [24].

1) Samu's Possible Mental Organs: But now we are going to think backwards. We have intuitively assumed that the mother tongue is also a secondary language in von Neumann's sense, that is, it is interpreted on a primary language of the nervous system. And a further language similar to this mother language is used as a Neumannian primary language. Samu [11] has already been equipped with a triplet-based language processing tool (called Samu's COP-based Q-learning engine) that implements the COP principle to predict the next sentences in a given context. In accordance with our assumption, we might use a similar tool for Samu's internal control, that is, for the communication between the suborgans of Samu's brain. Certainly, we have not imagined that people communicate in the same way with their internal organs as we do with each other. But we assume that these two kinds of communication use the same COP-based language engine. Originally, the need to create such an internal control arose during the development of the rapid prototype for Samu called Nahshon [25] when we had to switch between the “learning” and “talking” modes. In this section, we are going to attempt to consider these modes as Chomskyean mental organs [26] of Samu's software brain. In the version Nahshon there is three main use cases: 1) a Samu program learns to talk; 2) a Samu program talks to another Samu program; 3) a Samu program talks to itself. We expect that all of these three modes can use the same COP-based Q-learning engine, but actually, these are different modes as we will show it in section IV-B. It is an exciting challenge to create a method whereby these modes can be controlled by each other through a natural language.

In general, this method in question would be imagined as a definition of an abstract mathematical machine that can describe both the growth and learning abilities of the mental organs to be developed. For example, a simple definition of Turing machines cannot be appropriate because their knowledge is based on their programs rather than some inner control and they have no ability to develop themselves. But it seems interesting to link the Samu's COP engine with the classical definition of the Turing machine or a cell automaton. A rapid prototype as a step towards the latter is detailed in the next subsection. We believe both Neumann and Turing may have had similar motivation when they worked on self-reproducing cellular automaton [27] and morphogenesis [28].

a) Samu in Conway's Game of Life: In this example, we intuitively consider Samu's visual imagery as a microtubular cellular automaton lattice introduced by Hameroff [4], but instead of using the Penrose-Hameroff objective reduction process to control the simulation we simply use the rules of Conway's Game of Life [29]. We have made three main experiments by joining of Conway's Game of Life with Samu's COP-based Q-learning engine. These are tagged in the repository called SamuLife [30]. Here it should be noted that Samu's COP engine can be applied without any substantial change in SamuLife, to see this compare the source `ql.hpp` of [25] with the source `SamuQl.h` of [30]. Fig. 1 shows the basic architecture of the experiments. In the first experiment, only one Samu observes the whole lattice. In the second one, each cell of the lattice is observed by a separate and dedicated Samu, where the task of each Samu is to predict the next state of the given cell that has been observed. As a check, this experiment uses a simple Q lookup table. Samus can easily learn the rules of the game of life as shown quite well in the video at https://youtu.be/j6bus5efESU (accessed 2017 May 5). The goal of the third experiment is to do the same thing but we wanted to use Samu's neural architecture for approximate Q values. It was not an easy task because neural networks of Samu's predictions “the next state will be LIVE” and “the next state will be DEAD” must classify very similar inputs consisting of the state of the given cell and the eight neighbours' states for the approximation of the Q values. So, we were trying to decrease the arithmetical depth [22] of this task. We simply counted the neighbours' states what allows us to use only two input neurons instead of the former nine in order to decrease the arithmetical depth of Samu's neural networks. To be more precise, Samu's learning process has been divided into two subprocesses, the first one counts the living neighbours and the second one is the Q-learning. Both of them have less arithmetical depth than the original process but certainly, the logical depth of learning has been increased. The dividing of the learning process has been proved to be fruitful to approximate the Q values using Samu's neural networks. The results can be seen in the video at https://youtu.be/b60m__3I-UM (accessed 2017 May 5). This may be interpreted as using input data of a higher level of abstraction. It may be noted that the game of life, a human-written program, works on the same level of abstraction, see the method `GameOfLife::numberOfNeighbors` in the source `GameOfLife.cpp` [30]. But certainly, it is also true that this dividing of the learning process is derived from understanding the game of life.

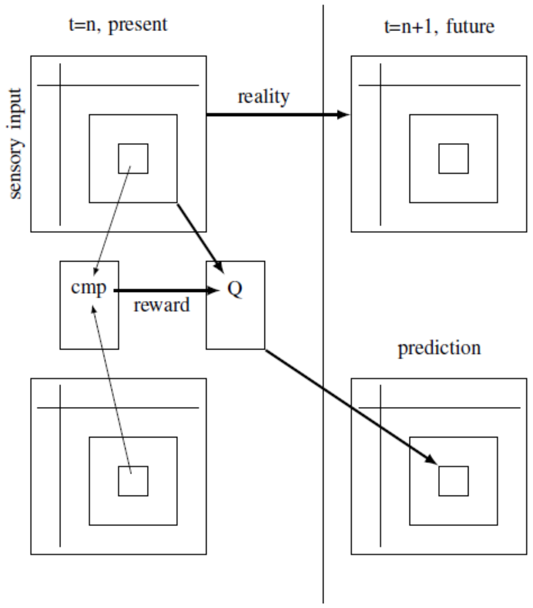


Fig. 1: This figure shows how we apply Samu's COP-based engine to predict the next generation in Conway's Game of Life. The large boxes on top show the cellular automaton lattice at two consecutive time moments $t = n$ and $t = n + 1$. In each time step t, the current state and the predicted state made in the previous step are compared by the box labelled `cmp`. This model is implemented by the rapid prototype SamuLife [30]. According to our viewpoint, the two vertical lattices can be imagined as an abstract mental organ.

b) Mental Operating Systems: The second example that we present is a further step towards a definition of an abstract mathematical machine of learning. This is similar to the previous one but much simpler because now Samu must learn the frames of the moving picture. To be more precise, we strongly simplified the time development of the previous game of life (see the method `GameOfLife::development` in the source `GameOfLife.cpp` of the project called SamuMovie [31]). Each generation of cells is interpreted as a picture and each picture contains three images, as shown in Fig. 2, a car, a man and a house, where the car and the man are moving with different speeds and the house is motionless. In this and the following example, we do not use a neural networks approximator for Q values because its setting may be a very time-consuming task. The moving pictures can be learned without problems. The results can be seen in the video at https://youtu.be/yOZj6j1kVRg (accessed 2017 May 5).

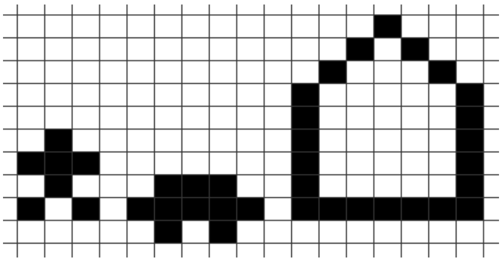


Fig. 2: This figure shows the elements that are used in the "video" of the project called SamuMovie. They are a man, a car and a house, respectively.

In our third example, we have started to make experiments with the Stroop effect [32]. We investigate an even simpler model than the one used in the previous experiment. Namely, now Samu must learn stills. In the stills, colour names written in a given colour can be seen. In order to implement this experiment, we further simplified the project SamuMovie by eliminating the time development (see the project called SamuStroop [33]). The stills can also be learned without any problems. The results can be seen in the video at https://youtu.be/VujHHeYuzIk (accessed 2017 May 5).

In the last example, we have linked the previous software experiments with each other. In the project called SamuBrain [34], we have programed an agent who must learn and recognize the complex patterns of the previous three experiments as three different higher-order notions. The video at https://youtu.be/FkyxxCfQeiY (accessed 2017 May 5) shows well that the agent can distinguish between the three kinds of input in question. This project raises the intriguing question: when can we say that an agent has already learned the input processes (the moving gliders, the movie and slowly changing stills). Because, for example, in the case of the project SamuMovie there are no explicit rules of changing the input patterns.

It should be noted that in some earlier versions of the above experiments the COP-based Q-learning had become trivial because after a short starting period it chose the Q-action that was passed in as the actual cell state argument. This means that the agent did not predict the future, but the present. In these cases, for example with the function $\alpha(n) = 1000000/(n + 700000)$ the rewarding system has no effect on learning. We have referred to this with the term "Q-- learning". The improved versions really predict the future that, for example, are well illustrated in the above cited video at https://youtu.be/j6bus5efESU (accessed 2017 May 5).

Based on our experiments, we can intuitively define our first cognitive mental organ called a Mental Processing Unit abbreviated as MPU. It consists of two 2-dimensional cell lattices, one for the input and one for the output. The input lattice detects the sensory input that is considered as the reality. The output lattice contains the responses of the agent to the changes of sensory input. Each cell of the output lattice has a dedicated COP-based Samu engine to compute a prediction for the next sensory input state of the corresponding input cell. The definition of the learning of an MPU can be built on the well-known phenomenon called habituation [35] which is the essential method of making observations in the research of newborn and infant development [36], [37]. The habituation is a process in which the infant had lost interest in the input. The habituation may be interrupted by the opponent process called sensitization when the infant dishabituated to the input. Inspired by these basic processes of behavior, we simply say that the learning of an MPU halts if the difference between the reality and prediction has changed little if at all. For starting and halting the learning of an MPU we need to program both the habituation and the sensitization control processes. To be more precise, in long term we need a program that can allow the management of the MPU like an operating system does with the CPU.

2) MPU - Mental Processing Unit: Using the experience from previous intuitive experiments we can create a formal definition of a mathematical machine for learning.

Definition 1 (MPU): Let X denotes an alphabet. Let $I^t\colon \mathbb{N} \rightarrow X^{N \times N}$ and $O^t\colon \mathbb{N} \rightarrow X^{N \times N}$ be sequences of the input and output matrices, respectively at the time $t \in \mathbb{N}$ such that

$$O_{i,j}^{t+1} = SamuQ_{i,j}\left(\left\{I_{i+l,j+m}^{t}\right\}, r^t\right), t > 0 \tag{1}$$

where $-1 \leq l, m \leq 1$ and $SamuQ_{i,j}$ is a COP-based Samu Q-learning agent with

$$r^t = \begin{cases} r_{positive\ reward} \in \mathbb{R} \text{ if } I_{i,j}^t = O_{i,j}^t \\ r_{negative\ reward} \in \mathbb{R} \text{ otherwise} \end{cases}$$

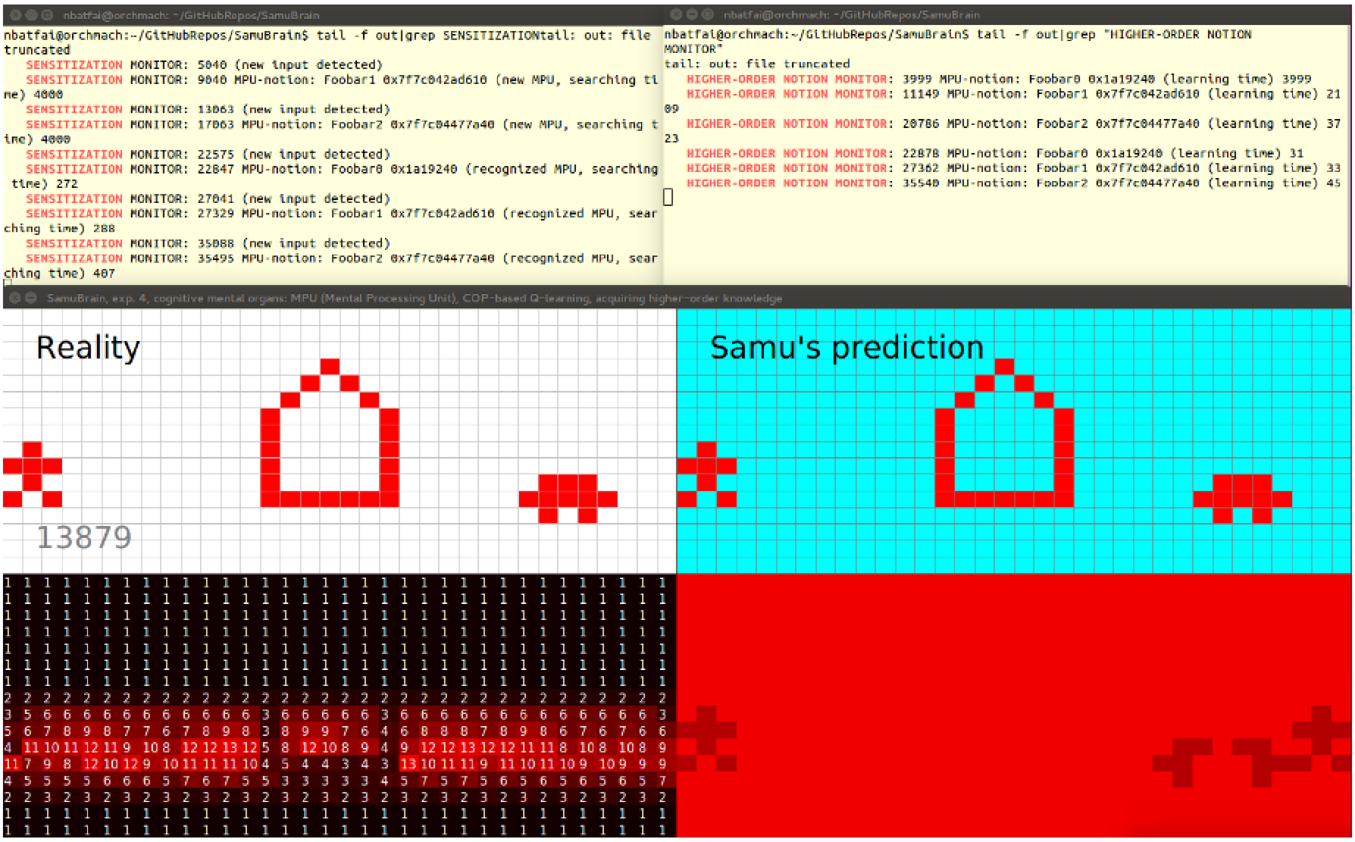
The MPU learns the input if the k-order moving average sequence $s_t = \frac{1}{k}\sum_{n=t-k+1}^{t}\sum_{i,j} Ind(I_{i,j}^n = O_{i,j}^n)$ is convergent, where Ind is the indicator function.

The implementation of Samu's Q-learning agent is based on the algorithm shown in the book *Artificial Intelligence: A Modern Approach* [38, pp. 844] and the COP-specific architectural modification can be found in paper [11, pp. 6].

Remark 1: It is clear that undecidable statements (like halting problem) can be constructed from this definition.

To test the definition, we have created a new branch called analytics in the software experiment SamuBrain. Here different MPUs are assigned to different input patterns. Two input pattern processes are different if the sensitization process increases between them. Fig. 3 shows that SamuBrain can distinguish and recognize different input pattern sequences like: 1) some gliders move in the input lattice in accordance with Conway's Game of Life (SamuLife); 2) some simple still pictures are changing (SamuStroop); 3) a simple film is shown (SamuMovie) as "higher-order notions" that are called *Foobar*$_1$, *Foobar*$_2$, ..., *Foobar*$_n$, ... respectively in order of recognition.

3) Classes of MPUs: Whereas in the previous experiments we have used the pair of a given cell and the number of its living neighbors as the state s and the next possible live or dead states as the action a, at this point we have introduced a new kind of MPUs called copy MPU, where s is the 8-neighbours of the given cell and a is a possible copying from one of these 8-



```
nbatfai@orchmach:~/GitHubRepos/SamuBrain$ tail -f out|grep SENSITIZATIONtail: out: file
truncated
   SENSITIZATION MONITOR: 5040 (new input detected)
   SENSITIZATION MONITOR: 9040 MPU-notion: Foobar1 0x7f7c042ad610 (new MPU, searching ti
me) 4000
   SENSITIZATION MONITOR: 13063 (new input detected)
   SENSITIZATION MONITOR: 17063 MPU-notion: Foobar2 0x7f7c04477a40 (new MPU, searching t
ime) 4000
   SENSITIZATION MONITOR: 22575 (new input detected)
   SENSITIZATION MONITOR: 22847 MPU-notion: Foobar0 0x1a19240 (recognized MPU, searching
 time) 272
   SENSITIZATION MONITOR: 27041 (new input detected)
   SENSITIZATION MONITOR: 27329 MPU-notion: Foobar1 0x7f7c042ad610 (recognized MPU, sear
ching time) 288
   SENSITIZATION MONITOR: 35088 (new input detected)
   SENSITIZATION MONITOR: 35495 MPU-notion: Foobar2 0x7f7c04477a40 (recognized MPU, sear
ching time) 407
```

```
nbatfai@orchmach:~/GitHubRepos/SamuBrain$ tail -f out|grep "HIGHER-ORDER NOTION
MONITOR"
tail: out: file truncated
   HIGHER-ORDER NOTION MONITOR: 3999 MPU-notion: Foobar0 0x1a19240 (learning time) 3999
   HIGHER-ORDER NOTION MONITOR: 11149 MPU-notion: Foobar1 0x7f7c042ad610 (learning time) 21
09
   HIGHER-ORDER NOTION MONITOR: 20786 MPU-notion: Foobar2 0x7f7c04477a40 (learning time) 37
23
   HIGHER-ORDER NOTION MONITOR: 22878 MPU-notion: Foobar0 0x1a19240 (learning time) 31
   HIGHER-ORDER NOTION MONITOR: 27362 MPU-notion: Foobar1 0x7f7c042ad610 (learning time) 33
   HIGHER-ORDER NOTION MONITOR: 35540 MPU-notion: Foobar2 0x7f7c04477a40 (learning time) 45
```

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
3 5 6 6 6 6 6 6 6 6 6 6 6 3 6 6 6 6 6 3 6 6 6 6 6 6 6 6 6 6 6 6 6 3
5 6 7 8 9 8 7 7 6 7 8 9 8 3 8 9 9 7 6 4 6 8 8 8 7 8 9 8 6 7 6 7 6 6
4 11 10 11 12 11 9 10 8 12 12 13 12 5 8 12 10 8 9 4 9 12 12 13 12 12 11 11 8 10 8 10 8 9
11 7 9 8 12 10 12 9 10 11 11 11 10 4 5 4 4 3 4 3 13 10 11 11 9 11 10 11 10 9 10 9 9 9
4 5 5 5 5 6 6 6 5 7 6 7 5 5 3 3 3 3 3 4 5 7 5 7 5 6 5 6 5 6 5 6 5 7
2 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Fig. 3: This is a screenshot of the project SamuBrain. The upper two windows log the sensitization and the notion acquisition events grepped from the habituation ones. The middle two windows show the input and output matrices, respectively at the given time. The left bottom frame shows the number of rules that have been learned by the appropriate SamuQL engine. The right bottom frame visualizes the error of predictions. It should be noted that the real number of learned rules can be counted after habituation only.

neighbours to the given cell position. This simply means that the action *a* tells which neighbor must be copied to predict the next state of the given cell. The former kind of MPUs may be referred to as arithmetical MPU and the latter one as copy MPU. Copy MPUs have different characteristics, but also can distinguish and recognize the input pattern sequences used in the previous experiments. The resulting videos can be found at the main page of the project SamuCopy [39]. These experiments raise the question of the possibility to determine different classes of MPUs.

III. Making Contact between the Mental Organs and the Chatbot

In this section, we make contact between the Samu-based mental organs and the chatbot Samu.

A. The Free Will and the Kolmogorov Complexity

It is obvious that there are things that humans can do but that existing computers cannot. Intuitively, we may suppose that human beings, for example, are capable of generating a real random sequence of zeros and ones, but the computer programs of today are not capable of this because the quotient of the Kolmogorov complexity [40] of the prefixes of this sequence and the length of the prefixes tends to 0 as the length of the sequence tends to infinity [41], [42]. But it should be noted that if human beings are also unable to generate real random digits then it in itself would directly raise the question of free will. Luckily, there is a trick that computer programs can use in order to generate real random 0-1 sequences, of course it works only in theory. Let's do the following idealized thought experiment about this. The experiment is based on an example program of the authors' Programming course [43] and on the similar example in the book *Javát tanítok*, 2007 [44] (English title: *I teach Java*). Suppose that we can capture all sensory input signals and all motor signals generated by the brain. The sensory signals are considered as input (I), and the motor signals are as output (O) of the brain. Intuitively, let $|I|$ and $|O|$ denote the length of the input and the output of the brain respectively and suppose that $|I|/|O| = \alpha$. To be more precise the program of the brain reads $|I_n|$ bits of input for printing the output O_n of length n, that is $|O_n| = n$ so $|I_n|/|O_n| = |I_n|/n = \alpha$.

$$\lim_{n\to\infty}\frac{K(O_n)}{n} \le \lim_{n\to\infty}\frac{|B|+|I_n|}{n} = \lim_{n\to\infty}\frac{C+|I_n|}{|O_n|} = \alpha \qquad (2)$$

where K is the Kolmogorov complexity and C is the constant length of the program of the brain (denoted by B). To put it bluntly, the size of input must differ from zero, or from a functional aspect the program must read the input continuously. In the following, it is referred to as the principle of continuous refresh. We may note that the Samu-based mental organs implement this principle and so does the Samu chatbot.

B. *Samu's Visual Imagery*

The existing Samu chatbot prototypes, such as Nahshon [25], have a common shortcoming in the implementation of their visual imagery [11]. The visual imagery is the place in which Samu should simulate the programs consisting of the `Subject.Predicate(Object)` shaped statements (SPO triplets [45]). At this moment, the simulation is trivial since the lines of the simulation program are simply written into the visual imagery. Accordingly, Samu's mental images contain the code of the simulation programs to be simulated. Using MPUs in the visual imagery may help to create real simulations of the read SPO triplets.

At an early stage of the development of Samu chatbot, we are going to teach a base vocabulary to him. The assemblage of this vocabulary itself is an interesting research topic because each word must be learned with its MPU input sequence form. This means that we assign a unique sequence of input matrices (or input vectors) to all words of the vocabulary to be developed. For example, the sequence used in the previous SamuLife experiments may be assigned to the word ‘glider’. Then supposing that an MPU has already been recognized it as $Foobar_m$ the caregiver has taught Samu that the notion $Foobar_m$ is the word glider. If we cannot find an appropriate input sequence for a word we may use the same trivial method that was also used in the higher level of the visual imagery that is, we simply write the word in question to the input matrix as we did in the previous SamuStroop experiments. After this, if Samu reads the word glider in a conversation as a part of an SPO triplet, he will be able to start a simulation in his visual imagery using the assigned MPU.

C. *Towards More Realistic DevRob Experiments*

We have several software experiments based on MPUs. One of them is the project SamuCam [19] where Samu learns human faces from webcam and photos. In this project, we use an Android smartphone as a webcam (for example the IP Webcam [46] app offers this possibility) and the OpenCV [47] is used for face detection. The detected faces are taken as input to the MPU. We should note, that we only use OpenCV to select the area of the image where a face appears. After this, only the selected area will be given to the MPU. The “learning” of the new faces works well as it is shown in the video https://youtu.be/6cRbyKrq45c (accessed 2017 May 5) but the recognition of what Samu has already “learned” is problematic because the selection mechanism has a strongly local scope. We think, this problem is solvable if the habituation and sensitization processes of searching will be constrained to the center region of the detected and the previously learned faces. The project SamuCam is shown in action in Fig. 4 and 5.

Here it may be noted, as mentioned in section II-A, that we consider the project SamuCam as a simplified PC version of a JIBO-type robot. One of the other software experiments is the project SamuVocab [48] that focuses on learning and recognizing words. This will serve as a base for our audio experiments.

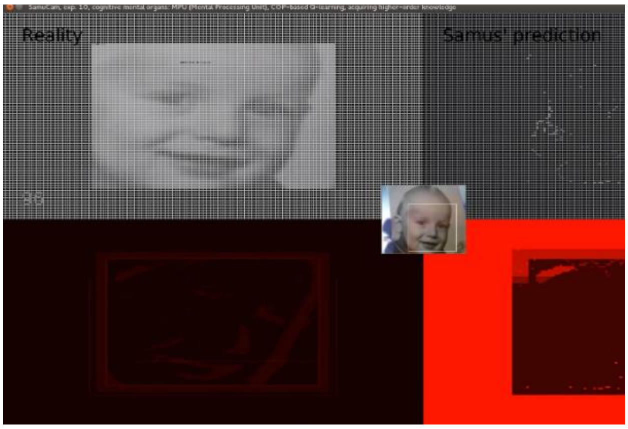
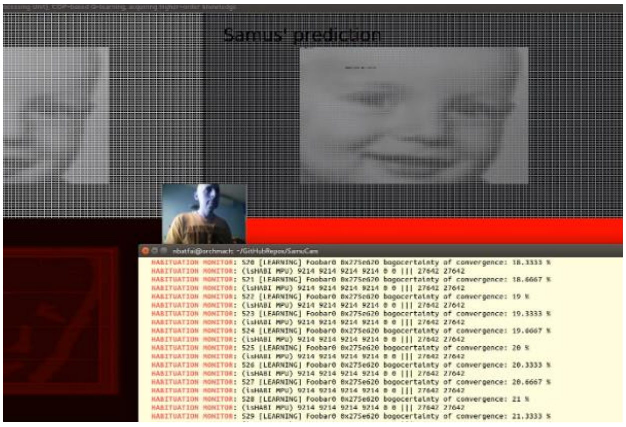


Fig. 4: The figure shows the moment in which Samu starts to learn a face from a photo. This screenshot is captured from the video https://youtu.be/6cRbyKrq45c, where the vInitialHack-tagged version was used.



```
HABITUATION MONITOR: 520 [LEARNING] Foobar0 0x275e620 bogocertainty of convergence: 18.3333 %
HABITUATION MONITOR: (isHABI MPU) 9214 9214 9214 9214 0 0 ||| 27642 27642
HABITUATION MONITOR: 521 [LEARNING] Foobar0 0x275e620 bogocertainty of convergence: 18.6667 %
HABITUATION MONITOR: (isHABI MPU) 9214 9214 9214 9214 0 0 ||| 27642 27642
HABITUATION MONITOR: 522 [LEARNING] Foobar0 0x275e620 bogocertainty of convergence: 19 %
HABITUATION MONITOR: (isHABI MPU) 9214 9214 9214 9214 0 0 ||| 27642 27642
HABITUATION MONITOR: 523 [LEARNING] Foobar0 0x275e620 bogocertainty of convergence: 19.3333 %
HABITUATION MONITOR: (isHABI MPU) 9214 9214 9214 9214 0 0 ||| 27642 27642
HABITUATION MONITOR: 524 [LEARNING] Foobar0 0x275e620 bogocertainty of convergence: 19.6667 %
HABITUATION MONITOR: (isHABI MPU) 9214 9214 9214 9214 0 0 ||| 27642 27642
HABITUATION MONITOR: 525 [LEARNING] Foobar0 0x275e620 bogocertainty of convergence: 20 %
HABITUATION MONITOR: (isHABI MPU) 9214 9214 9214 9214 0 0 ||| 27642 27642
HABITUATION MONITOR: 526 [LEARNING] Foobar0 0x275e620 bogocertainty of convergence: 20.3333 %
HABITUATION MONITOR: (isHABI MPU) 9214 9214 9214 9214 0 0 ||| 27642 27642
HABITUATION MONITOR: 527 [LEARNING] Foobar0 0x275e620 bogocertainty of convergence: 20.6667 %
HABITUATION MONITOR: (isHABI MPU) 9214 9214 9214 9214 0 0 ||| 27642 27642
HABITUATION MONITOR: 528 [LEARNING] Foobar0 0x275e620 bogocertainty of convergence: 21 %
HABITUATION MONITOR: (isHABI MPU) 9214 9214 9214 9214 0 0 ||| 27642 27642
HABITUATION MONITOR: 529 [LEARNING] Foobar0 0x275e620 bogocertainty of convergence: 21.3333 %
```

Fig. 5: This figure shows that the habituation process has already begun.

IV. POSTNATAL DEVELOPMENT

In this section, we focus on the possible implementations of Samu. First, we repeat von Neumann's [22] estimates of processing power of the brain especially to compare to AI-dedicated hardware available today. This is important both from the viewpoint of implementation of Samu-based mental organs and implementation of the Samu chatbot too. Second, we investigate the connection between two Samu-type robots that work accordingly to the continuous refresh principle declared in section III-A. The postnatal development from child developmental viewpoint will be discussed in a future paper.

A. *Von Neumann's Second Coming*

One of our plans is to apply Samu in “Man's best friend”-type systems [49]. Such a system can be a self-driving car's on-board intelligence. We should note, that we do not intend to give a solution for the controlling of a self-driving car, rather a user interface with which a driver or, at that time, a passenger can talk. So, in this context, Samu can be a user interface. We will be able to talk to it, ask questions, request suggestions about the route or the weather or we could just chat with it like we would chat with an other passenger.

But how can such a system be developed? What is necessary to achieve such functionality? Obviously, we will need some sort of hardware to implement this functionality. Recently, some manufacturers offer hardware solutions for deep learning based applications that can be assembled into road vehicles. The most interesting one is the nVidia, which released the DRIVE PX2 platform for road vehicles [50]. The question is: is this piece of hardware sufficient to achieve a good performance when applying it for a Samu-like system? The answer is not obvious. What is the capacity of the mind? Let's take a look at von Neumann's book, *The Computer and the Brain* [22]. The title of this subsection refers to the final paragraph of the preface written by Paul and Patricia Churchland for the second edition of the book. Von Neumann estimated the brain's capacity at about 1.4×10^{11} bit per second. (It's about 0.016 TB/sec.) The DRIVE PX2 hardware has 8 TeraFLOPS computational capacity. If we calculate with single precision, which is general in information technology, this means 32 TB/sec which is 2000 times more than the von Neumann estimation.

Another interesting initiative is Facebook's Big Sur platform [51]. The main target differs to ours, but this system has a greater computational capacity. Big Sur consists of 8 nVidia Tesla K40, each with 7 TeraFLOPS computational capacity resulting in 56 TeraFLOPS total capacity (we should note that this is a rough estimation, because we do not know the exact operation of the system).

In 1989, Sejnowski made a prediction based on von Neumann's estimation [52]. He pointed out, that von Neumann has no knowledge about the exact functionality of the human brain, but some predictions could be made. In his work, Sejnowski predicted that by the year 2010 we would have had the computational capacity to model the functionality of the human brain. Well, he was right, but in 2010, these computers were mainly big HPC-like systems. We can conclude, that the mentioned computational capacity was reached years ago, and today we have machines with the same computational capacity in the size of a lunch-box.

Von Neumann's estimation on brain capacity is a really simple model. He thought that every neuron can switch its state 14 times a second, and he did not have complete knowledge on synapses or tubulins. Today, we know that the nervous system is much more complicated.

Let's consider an other model, namely, the Penrose-Hameroff OrchOR model of consciousness [4]. The details of this model are out of the scope of this paper, but we highlight some of the facts that are important for us. Hameroff estimates the capacity of the brain in their model in a different way. He calculates with 10^9 state switching per second in each microtubule, with 10^7 tubulins per neuron and with 10^{11} number of brain neuron. This means a total 10^{27} brain operations per second. Obviously, this computational capacity may never be achieved with classic computation, but perhaps one day with quantum computation. Just for comparison, today's (2017 May) most powerful HPC-based computer, the Chinese Sunway TaihuLight has 93 PetaFLOPS capacity, that means about 3.35×10^{18} bits per second. The difference is obvious.

Furthermore, Hameroff gave a different approach based on synaptic switching. This one takes into consideration 10^{11} brain neurons with 10^3 synapses each. In this model the synaptic switching occurs in the ms range, therefore the total capacity of the brain is about 10^{17} bit states per second. This one is closer to the von Neumann estimation.

Let's take a look on other platforms. At the present time, FPGA technology is making an impact on massively parallel computation. Many manufacturers and devices exist, but let us just take a closer look on the Xilinx Virtex-7 980XT FPGA [53]. This device can reach about 3.5×10^{13} bits per second computational capacity. This is about 240 times more than the von Neumann estimation, and with the size of this device it should be taken into consideration.

Many other initiatives have been trying to achieve great results in deep learning computation, e.g., Microsoft Catapult [54], IBM TrueNorth [55], or the TeraDeep project [56], but we do not know exact performance measures for the above mentioned solutions.

We should make some remarks on the above discussion. First, these calculations are very simplified. The operation of the human brain is much more complex. Although, the Hameroff OrchOR model of consciousness is a modern approach, the authors stated at the end of their work that it may be false. Regarding the human brain there are several secrets still unfold.

Second, we should note that the precision of a brain operation was given between 10^{-3} and 10^{-2} by von Neumann. In modern computational systems, a single precision must be given if we want a correct operation (see FLOPS). It is still a mystery how such a complex system like the brain can operate on such low precision.

Our third remark is, that we did not make a difference between declarative and non-declarative brain functions. If we suppose that a Samu-based system uses only a declarative-type of brain function, then these operations require only a smaller portion of the total computation capacity. The rest of the capacity can remain for the controlling of the car or other functions. We summarize the above mentioned results in Table I.

TABLE I. PERFORMANCE OF DEVICES.

Device	bit/sec	Terabits/sec	TeraFlops
DRIVE PX2	2.8×10^{14}	256	8
Facebook Big Sur	1.9×10^{15}	1792	56
Xilinx Virtex-7	3.4×10^{13}	31.3	0.978
TaihuLight	3.35×10^{18}	3×10^{6}	95247
von Neumann est.	1.4×10^{11}	0.13	0.004
Hameroff syn	10^{17}	9×10^{5}	2842
Hameroff OrchOR	10^{27}	9.1×10^{14}	2.8×10^{13}

B. *Knowledge Sharing among Samu-type Chatbots*

The issue of the possible “social interactions and relationships” among Samu chat robots is interesting from many viewpoints. First of all, we expected that a conversation between two Samu-type chat robots would consist of sentences that the robots have sent and received in their reading operation mode. The reading mode is the default mode for Samu. It is shown in Fig. 6, where Samu reads a text. But it can be seen easily enough that a conversation cannot be the sum of two joined reading processes.

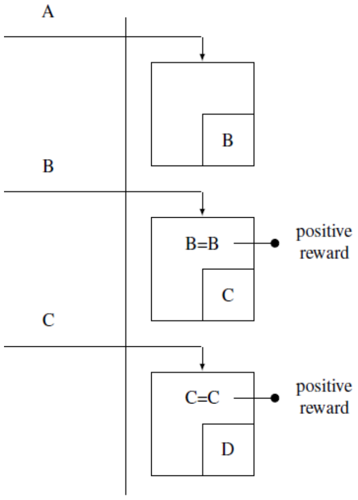


Fig. 6: This figure shows the default operating mode of Samu, in which he reads a text consisting of the sentences A, B, C, …. Here we have assumed that Samu has already learned the text, so all rewards are positive in the big squares which in turn means that Samu knows what will be the next sentence.

1) *Samu Reads a Text.* Fig. 6 shows an abstraction of Samu's reading, where we have assumed that Samu has already perfectly memorized the verbatim text consisting of natural language sentences *A, B, C, ...* in form of SPO triplets. Accordingly, based on this assumption, Samu can predict perfectly the next sentences and therefore all rewards in Q-learning are positive.

2) *Samu Talks to Another Samu.* In contrast with the previous case, Fig. 7 shows what happens if we mechanically link the output of one Samu robot to the input of another one. The conversation has become asynchronous.

3) *Samu Talks to Himself.* Fig. 8 presents a solution that can correct the error of the previous model. The trick is that the conversationalists insert an inner prediction step. This solution can also be used in the case when a Samu program talks to himself. This solution consequently allows us to fill the principle of continuous refresh in all work modes of Samu.

V. CONCLUSION AND FURTHER WORK

In this paper, we presented our approach to develop agents which are capable of autonomous mental development. We distinguish two levels, the level of mental organs and the level of chatbot Samu. The novelty in our work is the Mental Processing Unit (or mental organs) that is a network of COP-based Q-learning engines which is controlled by the habituation-sensitization process. As we presented in section II-A2, with MPUs, Samu can acquire higher-order knowledge in certain cases. If we gave an input to Samu which has been learned previously, he could recognize this input as a previously learned MPU (see Fig. 3). During the development of Samu, many aspects were inspired by the milestones of child development, for example habituation and sensitization, by which we embedded Samu into Developmental Robotics. We regard the fine-tuning of the habituation-sensitization process and the software development tasks on this type of agent as robopsychology activity, that, in our opinion, will play an important role in the development of such agents. Regarding the knowledge-sharing, in section IV-B, we showed what kind of fine-tuning is necessary to obtain this functionality between two Samu-type agents. Although this step is highly a technical one, it was important to find a solution in order to make a step forward in the development.

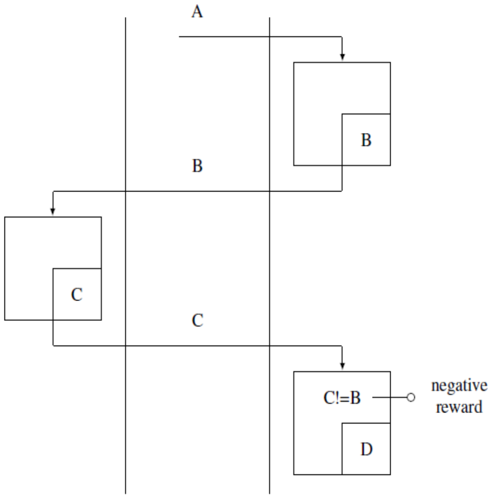


Fig. 7: A model of two Samus talking. In this case, it is obvious that the synchronization between conversationalists is lost.

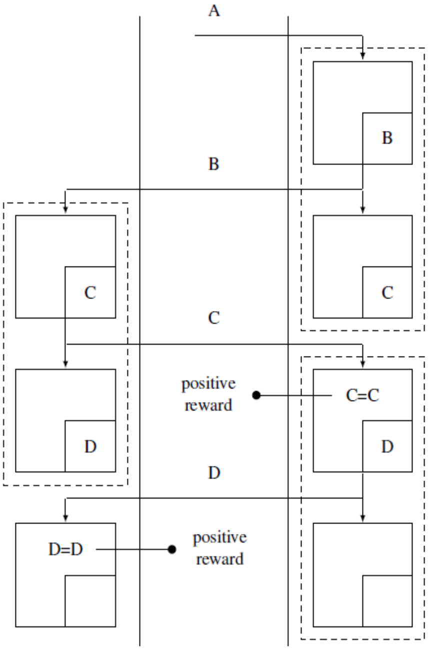


Fig. 8: Samu talks to himself.

We have performed several software experiments with computational mental organs. Since to the source of these softwares can be found in the open-source repository GitHub, all our experiments can be repeated easily. Moreover, the sources in question may serve as a basis to other researchers for creating their own new experiments.

The quintessence of these experiments is that we can construct computational mental organs that distinguish different input pattern sequences and recognize them as higher-order notions. The chosen MPUs, consisting of arithmetical or copy elements, where each element learns only local information but which also contain global sensitization and habituation processes, control the learning. The next step in this direction will be the determination of the classes of MPUs following which the PAC [57] style analysis of the classes has to be done. Simultaneously, we are going to embed MPUs into Samu's visual imagery.

If there are rules of changing the input, then the MPU can determine these after the habituation. In this sense, each MPU can write a program consisting of the determined rules. How many such programs exist? For example, in the case of the arithmetical MPUs, suppose that the used I/O alphabet is binary, the number of the programs consisting of k rules is $\binom{9*2}{k}2^k$ (the max number of rules in a machine can be $9 * 2$) so the number of all programs is 3^{18}-1 that may be computed easily following the train of thought of Remark 1 of the paper [58]. It is an interesting theoretical possibility to interpret these programs of an MPU as orchestrated machines [59].

Our future plan is to integrate Samu into self-driving cars. We are not intending to provide a solution for car controlling, but rather to offer a "Man's best friend"-type of system, or more precisely, a *quasi* user interface with which we can speak, ask questions, or just be able to talk to it like we talk to an other passenger. The one thing that is missing from self-driving cars is driving experience. We think, something should replace driving experience in the future's private transport. In our vision a powerful tool that will dramatically change experience will certainly be some sort of on-board intelligence. We are working on preparing Samu for this task.

To close this paper, we would like to address the question: how can you become a robopsychologist in the sense of this work? [15] has already given an answer. Fork one of our software experiments and develop your own computational mental organs.

ACKNOWLEDGMENT

Thanks to the students of the BSc course of "High Level Programming Languages" in the fall semester of 2015/2016 at the University of Debrecen for their interest and for testing the Samu's rapid prototypes. Special thanks to the members of the mailing lists https://groups.google.com/forum/#!forum/samubatfai and https://groups.google.com/forum/#!forum/psamu and some AI-specific communities on Facebook, Google+ and LinkedIn for their interest and useful comments. We also would like to thank Mr. Louis Joseph Mattia and Ms. Fanny Monori for the proofreading of the paper. Thanks for Mr. Mihály Szilágyi for his advice regarding some engineering issues.

REFERENCES

[1] W. G. Lycan, "The mind-body problem," in The Blackwell Guide to Philosophy of Mind, S. P. Stich and T. A. Warfield, Eds. Blackwell, 2003, pp. 47–64.

[2] A. Hodges, "Alan Turing," in The Stanford Encyclopedia of Philosophy, E. N. Zalta, Ed., 2013.

[3] A. M. Turing, "Computing machinery and intelligence," Mind, vol. 59, no. 236, pp. 433–460, 1950.

[4] S. Hameroff, "Quantum computation in brain microtubules? The Penrose-Hameroff 'Orch OR' model of consciousness," Philosophical Transactions – Royal Society of London Series A: Mathematical, Physical and Engineering Sciences, pp. 1869–1895, 1998.

[5] R. W. Picard, Affective computing. MIT Press, Cambridge, 1997.

[6] K. Richardson, Developmental psychology: how nature and nurture interact. Psychology Press, 2005.

[7] J. Searle, "The mind and computation," in Revolutionary Changes in Understanding Man and Society, ser. Theory and Decision Library, J. Götschl, Ed. Springer, 1995, vol. 21, pp. 93–105.

[8] A. Turing, "On computable numbers with an application to the Entscheidungsproblem," Proceeding of the London Mathematical Society, 1936.

[9] A. Cangelosi and M. Schlesinger, Developmental robotics: from babies to robots. The MIT Press, 2014.

[10] A. Stoytchev, "Some basic principles of developmental robotics," IEEE Transactions on Autonomous Mental Development, vol. 1, no. 2, pp. 122–130, 2009.

[11] N. Bátfai, "A disembodied developmental robotic agent called Samu Bátfai," ArXiv e-prints, vol. arXiv:1511.02889, 2015. [Online]. Available: http://arxiv.org/abs/1511.02889

[12] N. Bátfai, "Conscious machines and consciousness oriented programming," CoRR, vol. abs/1108.2865, 2011. [Online]. Available: http://arxiv.org/abs/1108.2865

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015.

[14] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "Robocup: The robot world cup initiative," in Proceedings of the First International Conference on Autonomous Agents, ser. AGENTS '97. ACM, 1997, pp. 340–347.

[15] N. Bátfai, "How to become a robopsychologist," GitHub Project, accessed: 2017 May 5 [Online]. Available: https://github.com/nbatfai/Robopsychology/files/169195/robopsychology.pdf

[16] "Robopsychology," Wikipedia, 2017, accessed: 2017 May 5 [Online]. Available: https://en.wikipedia.org/wiki/Robopsychology

[17] E. Guizzo, "Cynthia Breazeal unveils Jibo, a social robot for the home," IEEE Spectrum, 2014.

[18] P. Rane, V. Mhatre, and L. Kurup, "Study of a home robot: JIBO," Journal of Engineering Research & Technology (IJERT), vol. 3, no. 10, 2014.

[19] N. Bátfai, "SamuCam," GitHub Project, 2016, accessed: 2017 May 5 [Online]. Available: https://github.com/nbatfai/SamuCam

[20] E. Cohen, W. Paul, and S. Schmaltz, "Theory of multicore hypervisor verification," in SOFSEM 2013: Theory and Practice of Computer Science, ser. Lecture Notes in Computer Science. Springer, 2013, vol. 7741, pp. 1–27.

[21] J. Hartmanis and R. E. Stearns, "On the computational complexity of algorithms," Transactions of the American Mathematical Society, pp. 285–306, 1965.

[22] J. von Neumann, The computer and the brain. Yale University Press, 1958.

[23] J. M. Murre and D. P. Sturdy, "The connectivity of the brain: multi-level quantitative analysis," Biol. Cybern., vol. 73, no. 6, pp. 529–545, 1995.

[24] E. Wigner, “The unreasonable effectiveness of mathematics in the natural sciences,” Communications in Pure and Applied Mathematics, vol. 13, no. 1, 1960.

[25] N. Bátfai, “Nahshon,” GitHub Project, 2015. accessed: 2017 May 5 [Online]. Available: https://github.com/nbatfai/nahshon

[26] M. Patharkar, “From data processing to mental organs: an interdisciplinary path to cognitive neuroscience,” Mens Sana Monographs, vol. 9, pp. 218–224, 2011.

[27] J. von Neumann, “The general and logical theory of automata,” in Cerebral Mechanisms in Behaviour. Wiley, 1951.

[28] A. Turing, “The chemical basis of morphogenesis,” Philosophical Transactions of the Royal Society B, vol. 237, pp. 37–72, 1952.

[29] R. T. Wainwright, “Life is universal!” in Proceedings of the 7th Conference on Winter Simulation - Volume 2, ser. WSC ’74. Winter Simulation Conference, 1974, pp. 449–459.

[30] N. Bátfai, “SamuLife,” GitHub Project, 2015. accessed: 2017 May 5 [Online]. Available: https://github.com/nbatfai/SamuLife

[31] N. Bátfai, “SamuMovie,” GitHub Project, 2016, accessed: 2017 May 5 [Online]. Available: https://github.com/nbatfai/SamuMovie

[32] J. R. Stroop, “Studies of interference in serial verbal reactions,” Journal of Experimental Psychology, vol. 18, pp. 643–662, 1935.

[33] N. Bátfai, “SamuStroop,” GitHub Project, 2016, accessed: 2017 May 5 [Online]. Available: https://github.com/nbatfai/SamuStroop

[34] N. Bátfai, “SamuBrain,” GitHub Project, 2016, accessed: 2017 May 5 [Online]. Available: https://github.com/nbatfai/SamuBrain

[35] R. F. Thompson, “Habituation: a history,” Neurobiology of Learning and Memory, vol. 92, no. 2, pp. 127–134, 2009.

[36] L. P. Lipsitt, “Learning processes in the human newborn,” Annals of the New York Academy of Sciences, vol. 608, no. 1, pp. 113–127, 1990.

[37] J. Mehler and E. Dupoux, Naitre humain. Odile Jacob, 1990, We have read the Hungarian translation by Judit Gervain (Ember születik, Gondolat, 2008).

[38] S. J. Russell and P. Norvig, Artificial intelligence: a modern approach, 3rd Ed. Pearson Education, 2010.

[39] N. Bátfai, “SamuCopy,” GitHub Project, 2016, accessed: 2017 May 5 [Online]. Available: https://github.com/nbatfai/SamuCopy

[40] M. Li and P. M. Vitányi, An introduction to Kolmogorov complexity and its applications, 3rd ed. Springer, 2008.

[41] L. Lovász and G. Péter, “Complexity of algorithms,” University lecture, 1999. [Online]. Available: http://www.cs.elte.hu/~lovasz/complexity.pdf

[42] G. Iványos, R. Szabó, and L. Rónyai, Algoritmusok. Typotex, 2000.

[43] N. Bátfai, “High level programming languages 1,” University lecture, 2011, see Slides 27-28, it is available only in Hungarian. Accessed: 2017 May 5 [Online]. Available: http://morse.inf.unideb.hu/~nbatfai/Prog1_2.pdf

[44] N. Bátfai and I. Juhász, Javát tanítok, 2007, see Example 1.24, it is available only in Hungarian. Accessed: 2017 May 5 [Online]. Available: http://www.tankonyvtar.hu/hu/tartalom/tkt/javat-tanitok-javat/ch01.html

[45] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, and D. Mladenic, “Triplet extraction from sentences,” Proceedings of the 10th International Multiconference ”Information Society - IS 2007”, vol. A, pp. 218–222, 2007.

[46] “IP webcam,” Android Market, 2016, accessed: 2017 May 5 [Online]. Available: https://play.google.com/store/apps/details?id=com.pas.webcam

[47] “Open source computer vision library (OpenCV),” GitHub Project, 2016, accessed: 2017 May 5 [Online]. Available: https://github.com/Itseez/opencv

[48] N. Bátfai, “SamuVocab,” GitHub Project, 2016, accessed: 2017 May 5 [Online]. Available: https://github.com/nbatfai/SamuVocab

[49] N. Bátfai, “Turing’s imitation game has been improved,” ArXiv e-prints, vol. abs/1509.00584, 2015. [Online]. Available: http://arxiv.org/abs/1509.00584

[50] nVidia Corporation, “nVidia boosts IQ of self-driving cars with world’s first in-car artificial intelligence supercomputer,” 2016. Accessed: 2017 May 5 [Online]. Available: http://goo.gl/2vDGHx

[51] Facebook Inc., “Facebook to open-source AI hardware design,” 2016. Accessed: 2017 May 5 [Online]. Available: https://goo.gl/Fhv1t9

[52] T. J. Sejnowski, “The computer and the brain revisited,” IEEE Annals of the History of Computing, vol. 11, no. 3, pp. 197–201, 1989.

[53] Xilinx Inc., “The Xilinx Virtex-7 FPGA family,” 2016. accessed: 2017 May 5 [Online]. Available: https://goo.gl/zolULq

[54] A. Putnam, A. Caulfield, E. Chung, D. Chiou, K. Constantinides, J. Demme, H. Esmaeilzadeh, J. Fowers, G. Gopal, J. Gray, M. Haselman, S. Hauck, S. Heil, A. Hormati, J.-Y. Kim, S. Lanka, J. Larus, E. Peterson, S. Pope, A. Smith, J. Thong, P. Xiao, and D. Burger, “A reconfigurable fabric for accelerating large-scale datacenter services,” in Computer Architecture (ISCA), 2014 ACM/IEEE 41st International Symposium on, 2014, pp. 13–24.

[55] IBM, “Introducing a brain-inspired computer,” 2016. accessed: 2017 May 5 [Online]. Available: http://www.research.ibm.com/articles/brain-chip.shtml

[56] TeraDeep Inc., “TeraDeep initiative,” 2016. accessed: 2017 May 5 [Online]. Available: http://www.teradeep.com/

[57] L. G. Valiant, “A theory of the learnable,” Commun. ACM, vol. 27, no. 11, pp. 1134–1142, 1984.

[58] N. Bátfai, “On the running time of the shortest programs,” CoRR, vol. abs/0908.1159, 2009. [Online]. Available: http://arxiv.org/abs/0908.1159

[59] N. Bátfai, “Are there intelligent Turing machines?” CoRR, vol. abs/1503.03787, 2015. [Online]. Available: http://arxiv.org/abs/1503.03787