A Theory of Developmental Architecture

Juyang Weng
Department of Computer Science and Engineering
Michigan State Univerfsity
East Lansing, MI 48824 USA
weng@cse.msu.edu

Abstract

This paper presents a theory of developmental mental architecture primarily for robots. Six types of architecture are presented, starting with the observation-driven Markov decision process as Type-1. From Type-1 to Type-6, the architecture progressively becomes more complete toward the necessary functions of the autonomously developing agent. Properties of each architecture type are presented.

1. Introduction

A computational system can be specified at one of the *four levels of detail*: (1) constraint, (2) architecture, (3) algorithm, and (4) program, with increasing order of detail from one to the next. Studies in psychology often address issues at the constraint level while many engineering papers discuss systems at the algorithmic level. This paper deals with the *architecture* level. Mental architecture is a very challenging and important subject, but there have been relatively few systematic (agent-wise) investigations.

Supervised and reinforcement learning, based on the Markov Decision Process (MDP) architecture (single- or multi-level), enables a robot to learn autonomously while the environment (including humans) provides labels [11] or rewards [7, 13]. However, the MDP architecture, as explained in the following sections, has fundamental limitations that prevent them to be effective for the developmental robots described in [18].

Several alternative general-purpose architectures have been proposed. Major remarkable ones include Soar proposed by Laird, Newell & Rosenbloom [9], ACT-R by Anderson [2], and the architecture by Albus [1]. Soar and ACT-R incorporated many useful concepts that are necessary for human intelligence. Albus' architecture outline is motivated by neural architecture. The subsumption architecture proposed by Brooks [3] is a biologically motivated architecture component well suited for what is now known as the behavior-based approach.

The architecture models discussed above do not directly address perception, such as vision and audition. Neisser [10] pointed out that any model of vision that is based on spatial computational parallelism alone is doomed to failure. He proposed a two-stage visual process which consists of a pre-attentive phase followed by an attentive phase. Feldman & Ballard [5] proposed a "100-step rule:" A biologically plausible algorithm for immediate vision (one that does not involve slower deliberate thinking) can require no more than 100 steps. John Tsotsos' study [14] on the complexity of *immediate vision* proposed a coarse architecture for a biologically motivated general purpose vision system (for immediate vision). All these architectures are nondevelopmental in the sense that the information processor is not generated through real-time interactions with the environment.

Recently, there has been an onset of efforts on computational studies of autonomous mental development (e.g., the workshop report in [19]). There is a need of studies on the developmental mental architecture. This paper deals with this architecture issue. It does not describe algorithm details, but it provides citations to publications of experimental systems with algorithm detail and support the architecture described here. As this is a theoretical study, not all the capabilities of an architecture have been fully implemented. On the other hand, studies of actual experimental systems cannot replace studies on architecture since the former does not provide the properties of alternative architectures.

The history of studies on mental architecture has shown that this subject is hard to study and challenging to understand. This paper does not mean to solve all the problems and answer all the questions about this subject. This work is just a theoretical step toward the goal. In the following sections, I introduce a series of architectures, from simple to complex, along with the associated properties.

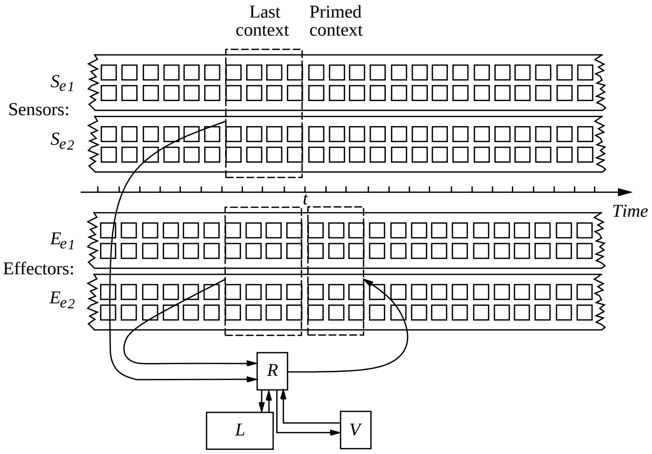

Figure 1. The Type-1 architecture of a multi-sensor multi-effector agent: Observation-driven Markov decision process. Each square in the temporal streams denotes a smallest admissible mask. The Type-1 architecture takes the entire image frame without applying any mask. The block marked with L is a set of context states (prototypes), which are clusters of all observed context vectors $l(t)$.

2. Type-1: Observation-driven MDP

We first need some definitions.

Definition 1 *The* internal environment *of an agent is the brain (or "the central nervous system") of the agent. The* external environment *consists of all the remaining parts of the world, including the agent's own body (excluding the brain).*

Definition 2 *An external sensor S_e and an internal sensor S_i are sensors that sense the external and internal environments, respectively. An external effector E_e and an internal effector E_i are effectors that act on the external and internal environments, respectively.*

Fig. 1 illustrates a multi-sensor multi-effector model of agent. The agent $A(t)$ operates at equally spaced discrete time instances $t = 0, 1, ...$ We assume that an image is produced at each time instance by the sensor, independent of the sensing modality, visual, auditory, touch, etc. Without loss of generality, we assume that the agent has two external sensors and two external effectors. Each external sensor S_{ei}, $i = 1, 2$, senses a random multi-dimensional sensory frame $x_e(t) = (x_{e1}(t), x_{e2}(t))$ at each time instance t and the sensed signal is fed into the agent. Each external effector E_{ei}, $i = 1, 2$, receives from the agent an effector frame $a_e(t) = (a_{e1}(t), a_{e2}(t))$ at each time instance t. Note that we change a variable of a vector to its subscript (e.g., change $x(t)$ to x_t) when it is convenient.

Definition 3 (Markov Decision Process) *The Markov decision process (MDP) is as follows. Suppose $S = \{1, 2, ..., n\}$ is a set of n predefined symbolic states that is used to model a part of the world. The state s_t at time t is a random variable taking one of the values in S. Its prior probability distribution is $P(s_0)$. The action a_t is the action of the agent at time t. Let H_t be the random history from time $t = 0$ up to time $t - 1$:*

$$H_t = \{s_{t-1}, s_{t-2}, ..., s_0, a_{t-1}, a_{t-2}, ..., a_0\}.$$

If its conditional state transitional probability $P(s_t \mid H_t)$ satisfies

$$P(s_t \mid H_t) = P(s_t \mid l_t)$$

where l_t is the short last k frames of the history

$$l_t = \{s_{t-1}, s_{t-2}, ..., s_{t-k}, a_{t-1}, a_{t-2}, ..., a_{t-k}\},$$

we call the process as the k-th order Markov decision process (MDP) [7, 13].

In many applications, the state of the world is not directly observable by the agent, or observable but with noise.

Definition 4 (Patially Observable MDP) *If the state s_t of the world is not totally observable to the agent. Instead, there is an observation x_t at time t that depends on the state s_t by an observation probability $P(x_t \mid s_t)$, the process is called partially observable MDP or POMDP [7, 13].*

In contrast, consider the following observation driven Markov Decision Process.

Definition 5 (Type-1) *Let $x_t \in \mathcal{X}$ and $p_t \in \mathcal{P}$ be the observations and outcome covariates (i.e., random vectors) at time t, respectively. Let H_t be the random vector of the entire history:*

$$H_t = \{x_t, x_{t-1}, ..., x_0, p_{t-1}, ..., p_0\}.$$

If its conditional state transitional probability $P(s_t \mid H_t)$ satisfies

$$P(p_t \mid H_t) = P(p_t \mid l_t)$$

where l_k is the last k observations:

$$l_t = \{x_t, x_{t-1}, ..., x_{t-k}, p_{t-1}, ..., p_{t-k}\}$$

as shown in Fig. 1, we call the process as the k-th order Observation-driven Markov Decision Process (MDP)[4]. The Type-1 mental architecture is a k-th order Observation-driven MDP (ODMDP)).

In the developmental ODMDP, the random observations in l_t across time $t = 0, 1, ..., t$ are the source from which the agent automatically generates states in the form of clusters

$l \in \mathcal{L}$, where $\mathcal{L}$ consists of all possible observations of the last contexts $\mathcal{L} = \{l_t \mid 0 \leq t\}$. The predicted consequence p_t consists of predicted action a_t and the predicted value v_t, $p_t = (a_t, v_t)$.

The following are the major differences between a POMDP[7, 13] (or HMM[11]) and an ODMDP:

1. The POMDP models a part of the world using hand-designed states. A state corresponds to an object or event of the modeled part of the world (e.g., a corner). The ODMDP models observed sensory space. Each state corresponds to an observation of the environment (e.g., a view of the corner with other background objects).
2. The states s_t of POMDP are hand-designed but the states of ODMDP can be automatically generated (developed). With the POMDP, the programmer must provide a simulation environment in which the meaning of each state must be hand-designed (for estimating three probability distributions). In contrast, ODMDP does not need to require prior probability and all the probability distribution $P(p_t \mid l_t = l)$ can be estimated incrementally on-the-fly.
3. In the POMDP, there are two layers of probability: the state transition probability $P(s_t \mid x_t, s_{t-1})$ and the state observation probability $P(x_t \mid s_t)$, while the observation-driven MDP has only one layer of probability: $P(p_t \mid l_t)$, enabling a more efficient learning algorithm.

In practice, we implemented the regressor R using the Incremental Hierarchical Discriminant Regression (IHDR) [17, 16]. Given any observed (last) context $l(t)$, the regressor R produces multiple consequences (primed contexts) $p_1(t), ..., p_k(t)$ having a high probability:

$$\{p_1(t), ..., p_k(t)\} = R(l(t)). \tag{1}$$

Thus, the regressor R is a mapping from the space of the last context $\mathcal{L}$ to the power set of $\mathcal{P}$:

$$R : \mathcal{L} \mapsto 2^{\mathcal{P}}. \tag{2}$$

R is developed incrementally through the real-time experience.

Therefore, we need a value system $V(t)$ that selects a desirable context from multiple primed ones:

$$V(R(l(t))) = V(\{p_1(t), p_2(t), ..., p_k(t)\}) = p_i(t) \tag{3}$$

where $1 \leq i \leq k$ and k varies according to experience. The value function selects the best consequence $p_i(t)$ that has the best value $v_i(t)$ in $p_i(t) = (a_i(t), v_i(t))$. For example, $V(\{p_1(t), p_2(t), ..., p_k(t)\}) = p_i(t)$ if $i = \arg\max\{v_1(t), v_2(t), ..., v_k(t)\}$.

The real-time Q-learning algorithm[15] can be used to estimate the value of each consequence $p_i(t)$, $i = 1, 2, ..., k$, and the agent selects the one (action) with the highest value. Therefore, the value system V is a mapping from the power set of $\mathcal{P}$ to the space of $\mathcal{P}$:

$$V : 2^{\mathcal{P}} \mapsto \mathcal{P}. \tag{4}$$

3. Type-2: Observation-driven Selective MDP

The Type-1 mental architecture is sensory nonselective in the sense that it is not able to actively select a subpart of relevant information from the sensory frame (intra-modal attention) or to attend a particular modality but not the other (inter-modal attention).

Given a d-dimensional input vector x, the attention can be modeled by an attention mask m, where m is a d-dimensional vector whose elements are either 0 or 1. Suppose that the input vector is $x = (x_1, x_2)$ and the mask is $m = (m_1, m_2)$. Then the corresponding attended input vector is $x' = x \otimes m = (x_1 m_1, x_2 m_2)$, where $\otimes$ denotes vector pointwise product. Not all the masks are admissible. For example, the set of admissible masks consists of circles with different radiuses ρ at different center positions (r_0, c_0) of the image frame. Then, the attention selection effector has three degrees of freedom: (r_0, c_0, ρ).

Without loss of generality in our theoretical discussion, we may assume, as shown in Fig. 1, that there are only four addmissible masks for each image frame at time t, denoted by the upper attention square, the lower attention square, and two trivial masks: no square is selected and both squares are selected, respectively.

Definition 6 (Type-2) *The* Type-2 mental architecture *is a Type-1 architecture, with the addition of an attention selector*

$$T : \mathcal{Y} \times \mathcal{A}_i \mapsto \mathcal{L},$$

as shown in Fig. 2, where $\mathcal{Y}$ is the space of all possible pre-attention contexts $\mathcal{Y} = \{l(t) \mid 0 \leq t\}$, $\mathcal{A}_i$ is the space of all possible attention selections for T, and $\mathcal{L}$ is the space of attention-masked last contexts.

In order to investigate the properties of different architectures, we define a concept called *higher*.

Definition 7 (Higher) *Given a set D of tasks, we say that a developmental architecture A_2 is* higher *than another developmental architecture A_1, if given the same teaching environment E, the architecture A_2 requires statistically fewer teaching examples than A_1, expected over the environment E and over the tasks in D.*

As a convention, we regard environment as part of the specification of a task. For example, a task is more challenging if the environment of the task execution is uncontrolled.

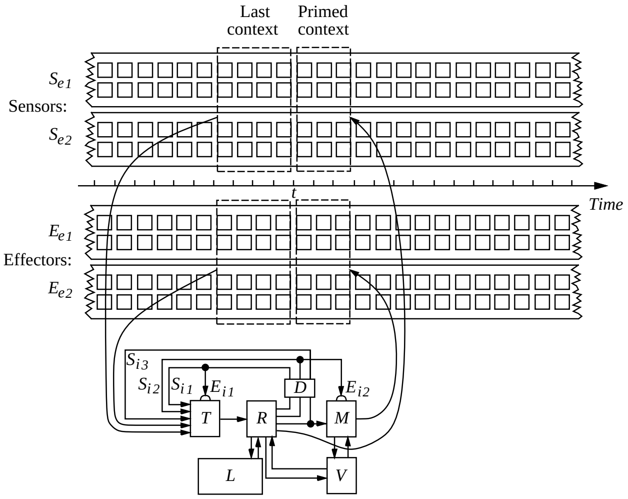

Figure 2. Progressive additions of architecture components from Type-2 to Type-5. Type-2: adding T and E_{i1}. Type-3: Adding M and E_{i2}. Type-4: Adding S_i and primed sensation. The block marked with D is a delay module, which introduces a unit-time delay. Type-5: Developmental T, R, M and V.

Theorem 1 (Existence of higher architecture) *There is at least one class D of tasks and the associated teaching environment E in which the Type-2 architecture is higher than the Type-1 architecture.*

Due to the limit of space, here only a sketch of proof is provided. Construct a set D of tasks whose goal is to classify sensory information in set C (e.g., human body). Without loss of generality, assume that at any time, only one of the attention squares of sensor S_{e1} contains an element (e.g., human body) in C and the other window does not (e.g., natural background that is free of human bodies). The Type-2 architecture enables the teacher to teach the agent to pay attention to S_{e1}, but the Type-1 architecture cannot.

It is not true that a higher architecture can learn faster than a lower architecture in any setting for any tasks. If the environment is such that the entire input vector $x(t)$ contains only elements in C and nothing of the natural background (which is very rare in reality), then the attention selection mechanism enabled by the Type-2 architecture does not help to reduce the number of training examples.

4. Type-3: Observation-driven Selective Rehearsed MDP

The Type-2 architecture does not have a motor mapping M. Therefore, it cannot autonomously rehearse an action sequence to evaluate its consequences without actually carrying out the action sequence. The rehearse is autonomous in that there is no pre-defined program segments that specify when and how to rehearse.

Definition 8 (Type-3) *The* Type-3 mental architecture *is a Type-2 mental architecture, with the addition of an action releaser M:*

$$M : \mathcal{P} \times \mathcal{A}_i \mapsto \mathcal{P},$$

as shown in Fig. 2, where $\mathcal{P}$ is the space of all possible predicted consequences, $\mathcal{A}_i$ is the space of all possible attention selections for M.

The action releaser M is a special case of the more general motor mapping (corresponding to the motor cortex) which also generates representation for frequently practiced action sequences (e.g., using the principal component analysis PCA or independent component analysis ICA) so that smooth action sequences can be generated.

With a traditional MDP with hand-designed states, it is possible to compute all the possible next states and perform planning. The Q-learning method uses the estimated action value $Q(s, a)$ of action a at state s to select the best action $a^* = \max_{a \in A} Q(s, a)$, from the set A of all the possible actions. This best next action a^* maximizes the expected rewards in the future. This kind of approach has two fundamental problems. First, the value system is rigid. No matter what value model is used (finite horizon, time discount model, etc.), the agent cannot autonomously change the way the value is determined[7, 13]. For example, if the time discount model is used, the agent is short-sighted. It prefers small rewards in the near further to faraway but important reward. Second, the agent is not able to learn to predict events (not just value) using the learned experience. For example, fed well and sleep well can be a reasonable goal for a human infant, but a human adult has a more sophisticated value system.

The Type-3 architecture does not suffer from these limitations. However, as long as the predicted action (e.g., drop a cup) is released, the effect that it causes to the external world will result (the broken cup). Can we design an architecture that enables the robot to "consider" and "plan" a significant amount of time ahead before it releases the action?

5. Type-4: Observation-driven SASE MDP

Type-4 architecture is Self-Aware and Self-Effecting (SASE). The term "self" here means the brain, instead of the body of the agent.

Definition 9 (Awareness) *The awareness of a task b in an (internal and external) environment E by an agent A is the capability of the agent to (1) sense various context states s of task b from E and (2) recall the predicted multiple contexts (primed contexts) $p = R(s)$ using the regressor R.*

By definition, the agent must use its sensors, the entry point of its sensory architecture (the input of T), to sense the contexts. In the above definition for awareness, we consider a particular b and an environment E. This is because any awareness has a scope. A person who is aware of the boiling temperature of water in a domain (e.g., in a normal environment) may not necessarily be aware of the boiling temperature of water in another domain (e.g., lower in a low pressure environment). With the above definition, we are ready to address the issue of self-awareness and self-effecting.

Theorem 2 (Necessary conditions of self-awareness) *Suppose an agent is aware of its mental activities (sensations and actions) about a task b in an environment E. Then the following must be true: (1) It senses such mental activities using its sensors. (2) It feeds the sensed signal into its perceptual entry point just like that for external sensors.*

Proof: Point (1) is true because, according to the definition for the awareness of an object, the agent must sense the object using its sensors. Point (2) is true because the status of the object must be sensed and fed into the entry point for sensors for proper perception and recall of the primed contexts. □

Based on the last two theorems, let us examine the issue of self-awareness more closely. If an agent runs a learning algorithm (e.g., the Q-learning algorithm to be explained later) but it does not sense the voluntary decision process using its sensors which are linked to its entry point for sensors, the agent is not aware of its own algorithm. For the same reason, humans do not sense the way their primary cortex works and, therefore, normally they are not aware of their own earlier visual processing. However, the voluntary part of the mental decision process does require a conscious, willful decision.

The traditional model of agent has a fundamental flaw. The model is for an agent that perceives only the external environment and acts on the external environment. It does not sense its internal "brain's" activities. In other words, its internal decision process is neither a target of its own cognition nor a subject for the agent to explain when the agent is sufficiently mature.

The human brain allows the thinker to sense what he is thinking about without performing an overt action. For example, visual attention is a self-aware and self-effecting internal action (see, e.g., Kandel, et al.[8], pp. 396 - 403). Motivated by neuroscience, the mathematical model of the *self-aware and self-effecting* (SASE) agent is defined as follows.

Definition 10 *A self-aware and self-effecting (SASE) agent has internal sensors S_i and internal effectors E_i for this internal (brain) environment, in addition to its external sensors S_e and external effectors E_e for its external environment (outside brain). The regressor R takes signals from S_i and S_e and generates internal and external actions for E_i and E_e, respectively.*

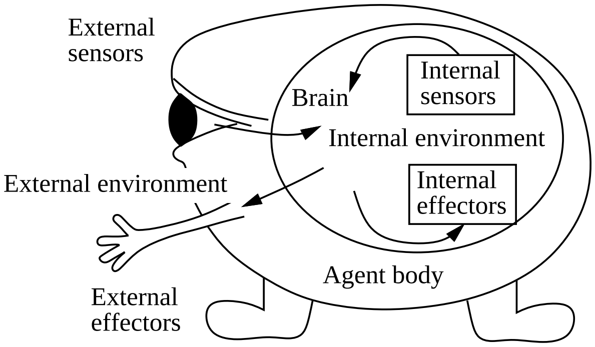

Figure 3. A self-aware self-effecting (SASE) agent. It interacts with not only the external environment but also its own internal (brain) environment: the representation of the brain itself.

Fig. 3 shows the illustration of a SASE agent.

The major design principles for a SASE agent include:

1. A SASE agent must have a sensor for each of its voluntary external effectors, so that it can sense what each effector is doing. For example, the muscle spindles sense the tension of human muscles to tell the position of the arms.

2. A SASE agent must have internal effectors and internal sensors for its voluntary internal effectors. For example, it needs a set of internal attention effectors to select the most relevant part of sensory information for later processing, which eventually leads to voluntary actions for attention selection.

3. A SASE agent needs pre-motor areas, where it stores information about the control of the effectors, but the signal in the pre-motor area is not sent to the effectors unless an action release signal is issued. The effector signals in the pre-motor areas are also sensed by internal sensors so that the robot can "talk to itself" internally.

It is important to note that not all the internal brain representations are sensed by the brain itself. Early processing actions are typically not sensed.

Definition 11 (Type-4) *The* Type-4 mental architecture *is a Type-3 mental architecture, but additionally, the internal voluntary decision is sensed by the internal sensors S_i and* *the sensed signals are fed into the entry point of sensors, i.e., the entry point of the attention selector T. In order to recall the effects of the voluntary actions, not only the expected reward value is estimated by the value system, but also the primed context which includes not only the primed action, but also the primed sensation.*

The architecture illustrated in Fig. 2 is a Type-4 architecture. Two voluntary internal actions are modeled by E_{i1} for attention selection, and by E_{i2} for action release. Both internal actions are sensed by the internal (virtual) sensors S_{i1} and S_{i2}, respectively. The rehearsed external action (not released) is sensed by the virtual internal sensor S_{i3}.

The regressor R maps each attended context $l \in \mathcal{L}$ to a set of multiple primed contexts, from which the value system selects a single primed context $p \in \mathcal{P}$. In other words, the composite function of R followed by V gives a mapping: $V \circ R : \mathcal{L} \mapsto \mathcal{P}$. With a SASE agent, both external context (sensed by S_e) and internal context (sensed by S_i) are available in l.

Through a consecutive time series $t = 1, 2, ..., k$, the composite function $V \circ R$ performs a series of reasoning, represented by the the regression sequence:

$$s = ((l_1, p_1), (l_2, p_2), ..., (l_k, p_k)) \tag{5}$$

where each regression pair (l_i, p_i) is an input-output pair of the composite function $V \circ R$, $p_i = V \circ R(l_i)$, $i = 1, 2, ..., k$. The link between two consecutive regression pairs can be realized by two paths, the external path and the internal path, denoted by e and i respectively. Symbolically, the reasoning process can be represented by the following composite reasoning sequence:

$$s = \left((l_1, p_1), \left[\begin{array}{c} e_1 \\ i_1 \end{array}\right], (l_2, p_2), \left[\begin{array}{c} e_2 \\ i_2 \end{array}\right], ..., (l_k, p_k), \left[\begin{array}{c} e_k \\ i_k \end{array}\right]\right) \tag{6}$$

where

$$\left[\begin{array}{c} e_i \\ i_i \end{array}\right], i = 1, 2, ..., k$$

represents the parallel external and internal paths. Whether the result of external and internal paths are taken into account at any time t by the regressor depends on the attention selection in T.

Definition 12 (External and external reasoning process) *There are three types of reasoning processes, external, internal, and mixed, corresponding to the attention in which the attention module T attends to external, internal or both, respectively.*

From the above discussion, we have the following summary:

- Type-1 through Type-3 architectures allow the agent to perform external reasoning processes, but not internal reasoning as defined above.

- A Type-4 architecture is able to execute external, internal, and mixed reasoning processes.

Theorem 3 *The Type-4 architecture allows internal reasoning to realize the following kinds of learning (1) nonassociative learning, (2) classical conditioning, and (3) instrumental conditioning.*

Proof: First we prove the nonassociative learning[12]. The nonassociative learning occurs when the agent is exposed to stimulus because of the history of similar or dissimilar stimuli. Sensitization and habituation are two well known examples of nonassociative. In Eq. (6), the nonassociative learning can be accomplished by the link (l_i, p_i) realized by the composition of regression R and the value system: $p_i = V \circ R(l_i)$. The value system plays a central role. For example, the action (e.g., looking at another direction) that is predicted to generate more novel stimuli then alternative action (e.g., continue looking after repeated exposure to the similar stimuli), the former action is selected by the value system V from the alternative actions predicted by R.

Next, we prove the case for classical conditioning. In classical conditioning, a conditional stimulus CS (e.g., tone) is repeatedly paired with unconditional stimulus US (e.g., food) that elicits unconditional response UR (e.g., salivation). In this case, l_i = CS, l_{i+1} = US, and p_{i+1} = salivation, for all the time instances i where the event occurs. The Q-learning used by the value system V backpropagates repeatedly the primed action p_{i+1} through time i, so that l_i primes p_i = salivation even in the absence of l_{i+1}.

Finally, we establish the case for instrumental conditioning. When l_i stimulus is present, two actions a_1 and a_2 are predicted, $(a_1, a_2) = R(l_i)$. According to past experience, a_1 has a low value and a_2 has a higher value, using, e.g., Q-learning by the value system V. Thus, a_2 is selected by V. □.

For a realization of the nonassociative learning, the classical conditioning, and the instrumental conditioning, using the Type-4 architecture, see Huang & Weng [6], Zhang & Weng [20], and Huang & Weng [6], respectively. The instrumental conditioning has been known as reinforcement learning in the machine learning community and has been very widely studied using the traditional MDP architecture [7] [13]. A major novelty here is that a single architecture realizes all three types of learning.

There are many more complex internal mental activities. We addresses a typical complex activity known as autonomous planning. Planning has been conducted extensively using the traditional MDP architecture, based on the

Q-learning mechanism (i.e., time discounted value propagation) [15]. However, Q-learning based planning has a major drawback: It prefers immediate small rewards to future large rewards. One can program the planner in such a way so that only the final goal produces a reward and intermediate goals do not. However, such a task-specific setting is too inflexible for the general setting of mental development, where various rewards are generated from the real world at different stages and it is impossible for the programmer to write a different program for a different planning task (due to the task non-specific nature of autonomous mental development Weng et al. [18].)

Theorem 4 *The Type-4 architecture allows internal reasoning to realize autonomous planning.*

Proof. Autonomous planning requires first an accumulation of experiences so that alternative condition-action pairs are learned. Suppose that there are two plans according to the experiences: The execution path of the plan (a) is recalled as:

$$(l_1, p_{a,1}), (l_{a,2}, p_{a,2}), ..., (l_{a,i}, p_{a,i}))$$

and that of the plan (b) is recalled as:

$$(l_1, p_{b,1}), (l_{b,2}, p_{b,2}), ..., (l_{b,j}, p_{b,j})).$$

Both lead to a completion of the task. Both plans are recalled sequentially using only the internal path, i path instead of e in Eq. (6). Finally the value of $p_{a,i}$ is compared with that of $p_{b,j}$. The value system decides which value is better and so chooses the corresponding plan (a) or (b). The association of a to the primed action in $p_{a,1}$ and b with that in $p_{b,1}$ is represented by "talking to itself:" For example, the selected plan in $p_{a,i}$ as part of the last context in l, which primes the first action in $p_{a,1}$. The similar process takes place for plan (b). □.

I expect that early demonstration of autonomous planning is possible in a restricted (simplified) natural setting. Anywhere any-time planning in uncontrolled natural settings is possible after a significant amount of "living experience."

One might think at this point that the internal process looks like "thinking." However, the internal process defined here is not equivalent to autonomous thinking that is fundamental to human intelligence. A necessary piece for thinking is development.

6. Type-5: Developmental observation-driven SASE MDP

Definition 13 (DOSASE MDP) *The developmental observation-drive SASE MDP (DOSASE MDP) has an architecture Type-4 or higher, that satisfies the following requirements:*

1. *During the programming time, the tasks that the agent will learn are unknown to the programmer.*

2. *The agent $A(t)$ starts to run at $t = 0$ under the guidance of its developmental program P_d. After the birth, the brain of the agent is not accessible to humans.*

3. *Human teachers can only affect the agent $A(t)$ as a part of its environment through its sensors and effectors recursively: At any time $t = 0, t = 1, ...$, its observation vector at time t is the last context $l(t)$. The output from $A(t)$ at time t is its selected primed context $p(t) \in \mathcal{P}$. $A(t-1)$ is updated to $A(t)$, including T, R (and L), M, and V.*

In contrast with the traditional MDP, the DOSASE MDP $(A(t), P_d)$ is developmental in the sense that the developing program P_d does not require a given estimate of the *a priori* probability distribution $P(l)$ for all $l \in \mathcal{L}$, nor even a given set of states. Consequently, P_d does not require a given estimate for the state observation probability $P(l_t \mid l_{t-1})$ nor that for the state observation probability $P(x_t \mid l_t)$.

When the number of states is very large, it is practically sufficient to keep only track of the states that have a high probability, instead of estimating probability of all the states, which is too computationally expensive to reach the real-time speed. In HMM, this technique is called beam forming.

7. Type-6: Multi-level DOSASE MDP

The Type-5 architecture has only one sensorimotor level, although each mapping T, R, and M have multiple levels in their own internal structure. We call it a sensorimotor level because the pathway from T through R up to M corresponds to a pathway from sensory input to motor output. The primed context of such a level can be fed into another sensorimotor level for the following reasons:

1. Abstraction. While a low level is tightly linked to fine time steps, the higher levels become more "abstract," in the sense that the higher level clusters of context states are coarser in temporal granularity and grouped more according to actively attended events.

2. Self-generated context: Allow voluntarily generated motor actions to serve as context input to the higher level. Thus the agent is able to "talk to itself" at a more abstract level.

3. Enabling a higher degree of sensory integration. It is not practical to integrate all the receptors in a human body by a single attention selection module T, because otherwise, e.g., the attention is too complex.

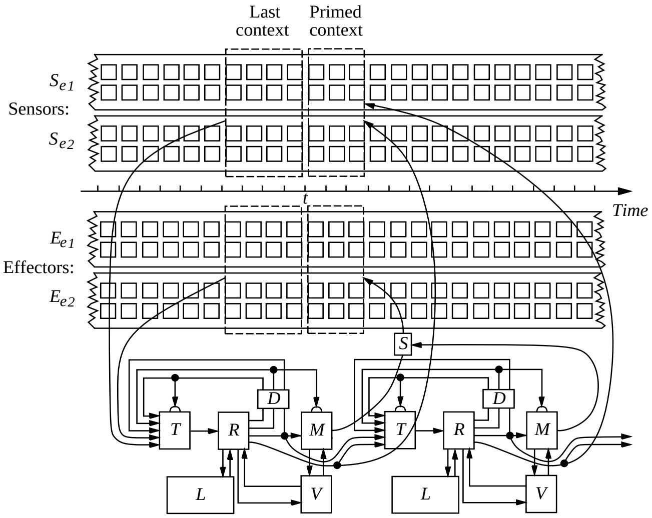

Figure 4. The Type-6 architecture.

Definition 14 (Type-6) *The Type-6 mental architecture is composed of several levels of Type-5 architecture. The primed contexts from a lower-level system is fed into the sensory input of the higher-level system.*

Fig. 4 illustrates the Type-6 architecture. The input to the attention selector $T^{(2)}$ at level 2 includes the primed context $p(t) = (x_p(t), a_p(t))$ from level 1, where $x_p(t)$ and $a_p(t)$ are primed sensation and primed action, respectively. One or multiple levels can feed their primed contexts into the next higher level for sensory integration.

We have systematically introduced six types of architectures. Although the order at which new capabilities are added to the previous type is primarily a design choice, the order used here is motivated by a relatively large payoff in capability with a minimal addition of the architecture complexity.

8. Conclusions

The observation-driven MDP (Type-1), seems more suited for autonomous mental development than the traditional MDP. This paper provides Type-1, through Type-5 (DOSASE MDP), up to Type-6 (multilevel DOSASE MDP). A DOSASE MDP can perform nonassociative learning, classical conditioning, instrumental conditioning and planning. The realization of higher capabilities has yet to be demonstrated.

References

[1] J. S. Albus. Outline for a theory of intelligence. *IEEE Trans. Systems, Man and Cybernetics*, 21(3):473–509, May/June 1991.

[2] J. R. Anderson. *Rules of the Mind*. Lawrence Erlbaum, Mahwah, New Jersey, 1993.

[3] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, March 1986.

[4] D. R. Cox. Statistical analysis of time series: Some recent developments. *Scand. J. Statist.*, 8(2):93–115, 1981.

[5] J. A. Feldman and D. H. Ballard. Connectionist models and their properties. *Cognitive Science*, 6(3):205–254, 1982.

[6] X. Huang and J. Weng. Novelty and reinforcement learning in the value system of developmental robots. In *Proc. Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems (EPIROB'02)*, pages 47–55, Edinburgh, Scotland, August 10 - 11 2002.

[7] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[8] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, editors. *Principles of Neural Science*. McGraw-Hill, New York, 4th edition, 2000.

[9] J. E. Laird, A. Newell, and P. S. Rosenbloom. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33:1–64, 1987.

[10] U. Neisser. *Cognitive Psychology*. Appleton-Century-Crofts, New York, 1967.

[11] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of IEEE*, 77(2):257–286, 1989.

[12] L. R. Squire and E. R. Kandel. *Memory: From Mind to Molecules*. Scientific American Library, New York, 1999.

[13] R. S. Sutton and A. Barto. *Reinforcement Learning*. The MIT Press, Cambridge, Massachusetts, 1998.

[14] J. K. Tsotsos. Analyzing vision at the complexity level. *Behavioral and Brain Sciences*, 13:423–469, 1990.

[15] C. Watkins. Q-learning. *Artificial Intelligence*, 8:55–67, 1992.

[16] J. Weng and W. Hwang. Online image classification using IHDR. *International Journal on Document Analysis and Recognition*, 5(2-3):118–125, 2002.

[17] J. Weng and W. S. Hwang. An incremental learning algorithm with automatically derived discriminating features. In *Proc. Asian Conference on Computer Vision*, pages 426 – 431, Taipei, Taiwan, Jan. 8-9 2000.

[18] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen. Autonomous mental development by robots and animals. *Science*, 291(5504):599–600, 2001.

[19] J. Weng and I. Stockman. Autonomous mental development: Workshop on development and learning. *AI Magazine*, 23(2):95–98, 2002.

[20] Y. Zhang and J. Weng. Action chaining by a developmental robot with a value system. In *Proc. IEEE 2nd International Conference on Development and Learning (ICDL 2002)*, pages 53–60, MIT, Cambridge, Massachusetts, June 12-15 2002.