State-of-the-art Schema Mechanisms for Developmental Artificial Intelligence

Olivier L. Georgeon[1][0000−0003−4883−8702],
Filipo S. Perotto[2][0000−0003−2283−4703], Kristinn R. Thórisson[3],
Arash Sheikhlar[3], and Paul Robertson[4][0000−0002−4477−0379]

[1] UR CONFLUENCE: Sciences et Humanites (EA 1598), UCLy, France
`ogeorgeon@univ-catholyon.fr`
[2] ONERA – The French Aerospace Lab – DTIS, Toulouse, France
`filipo.perotto@onera.fr`
[3] Center for Analysis and Design of Intelligent Agents, Reykjavik University, Iceland
`thorisson@ru.is, arash@iiim.is`
[4] DOLL Labs, Lexington, MA, USA
`paulr@dollabs.com`

Abstract. Schema mechanisms are software frameworks designed to reflect learning theories like genetic epistemology, constructivist psychology, and developmental learning. These theories find roots in the work of Jakob von Uexküll on animal behavior, Ernst von Glasersfeld on epistemology, and, more centrally, Jean Piaget ranging from child psychology to theory of knowledge. Gary Drescher coined the generic term schema mechanism and pioneered their implementation in 1991. His work laid the ground to identifying the core challenges: open-ended learning, intrinsic motivation, incremental hierarchical abstraction, sensorimotor grounding, reflexivity, individuation. This paper reviews a selection of the latest schema mechanisms and highlights their contributions to these key challenges, with the goal of unifying their different contributions toward a comprehensive theory of developmental artificial intelligence.

Keywords: schema mechanism · genetic epistemology · constructivist learning · intrinsic motivation · cognitive architectures.

1 Introduction

Throughout his life, Jean Piaget developed and popularized the theory of *genetic epistemology* [27] to account for the genesis of intelligence and knowledge. In parallel, he pioneered *developmental psychology* by establishing the foundational methods for studying mental development in children. Toward the end of his life, he connected genetic epistemology with constructivist epistemology and the work of Ernst von Glasersfeld [11]. Genetic epistemology rests on the key concept of "scheme", closely related to the concept of "functional circles" proposed by Jakob von Uexküll. Piaget defines a scheme as follows:

> "A scheme is a structure or organization of actions as they transfer or generalize in similar or analogous circumstances." ([25], p. 23).

In this paper, we translate the French term *scheme* with the English term *schema* and we use the plural *schemas*. A schema is the basic unit of knowledge that encapsulates the action and its circumstances, that is, a *pattern of interaction*. Genetic epistemology insists on the primacy of interaction as a condition for the emergence of perception and knowledge:

> "Knowledge does not originally arise either from a subject conscious of itself or from objects already constituted (from the subject's point of view) that would impose themselves on the subject. Knowledge results from interactions occurring halfway between the subject and the objects, and thus involving both, but due to a complete un-differentiation and not from exchanges between distinct forms.
>
> If, at the beginning, there is neither a subject, in the epistemic sense of the term, nor objects, conceived as such, nor, above all, invariant instruments of exchange, then the initial problem of knowledge will be to construct such mediators. Starting from the contact zone between one's own body and the objects, these mediators will progressively engage more deeply in both complementary directions toward the exterior and the interior. It is from this dual progressive construction that the joint elaboration of both the subject and the objects depends.
>
> The initial instrument of exchange is not perception, as rationalists too easily conceded to empiricism, but rather action itself, with its much greater plasticity. Certainly, perceptions play an essential role, but they partly depend on action as a whole, and some perceptual mechanisms that one might have thought to be innate or very primitive only emerge at a certain level of object construction." (translated from [26], p. 14-15)

To this day, Piaget's ideas remain central to modern theories of knowledge as well as developmental psychology. Indeed, genetic epistemology seems to offer a materialistic explanation of how human psychology develops from birth. The question whether this explanation is too reductionist remains however open. To our knowledge, Piaget never claimed that his theories could be implemented as a mechanistic process in a computer. The first attempt only began eleven years after his death.

2   Schema mechanisms

Implementing genetic epistemology as a mechanistic process involves reducing Piaget's psychological concept of schema into data structures handled by a com-

puter. This computer controls an artifact (robot or virtual agent) that inter-
acts with its environment. The computer must process these data structures
through some mechanism that will support the generation of increasingly in-
telligent behavior demonstrating the internal development of the robot. Guerin
and McKenzie [12] proposed the graphical representation of this developmental
process shown in 1.
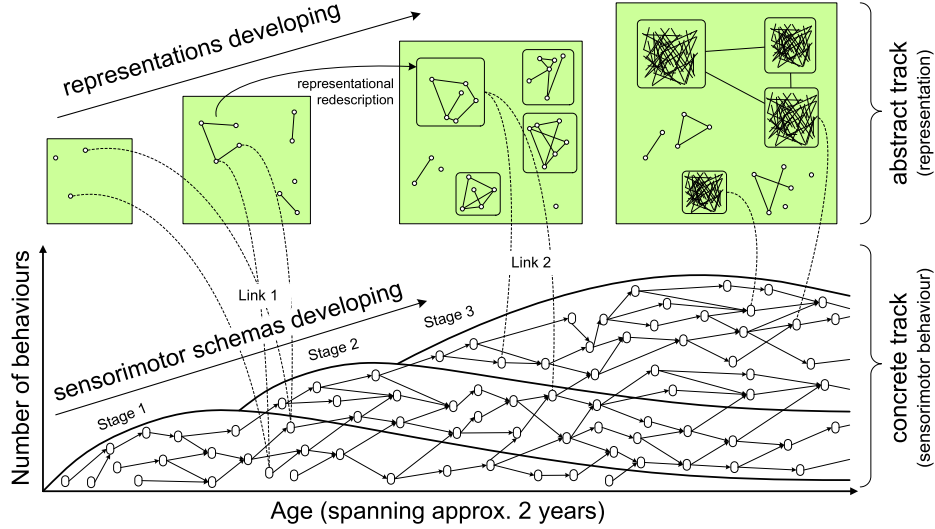
Fig. 1. Conceptual diagram of infant development from [12] Fig. 1. The lower (concrete)
track shows a directed acyclic graph of sensorimotor schemas. A node represents a
newly created schema. An edge has the meaning "is a necessary precursor". Stage 1:
behaviors without objects. Stage 2: behaviors with single objects. Stage 3: object-
object behaviors. The schemas now involve relationship among objects, and locations
and transforms within space. The higher (abstract) track represents representations of
objects by schemas and physical properties influencing their interactions.

Ziemke [40] examined how these views apply to robotics. Oudeyer [23] stressed
the importance of intrinsic motivation.

2.1   Drescher's

Drescher notes that "a constructivist account of the development of intelligence
holds that the difference between the mind of an adult, and that of an infant,
lies in mental structures built by the individual" [6, p. 41]. He pioneered the
first schema mechanism in 1991 by designing these mental structures as schemas
formalized as the tuple (context, action, result) depicted in Fig. 2. The *context*
and the results are sets of binary *items* that can be *positively* or *negatively*
included. A schema's context is satisfied when all the positively included items

are *On* and all the negatively included items are *Off*. A schema's result asserts that, if the action is taken when the context is satisfied, then the positively included items of the results will be turned *On* and its negatively included items will be turned *Off*.
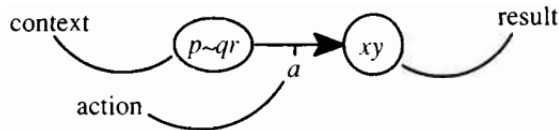
Fig. 2. Schema from [6] Fig. 3.2. The schema noted $p{\sim}qr/a/xy$ asserts that if action a is taken in the context where item p is On, q is Off, and r is On then the items x and y will be turned On.

The agent is initialized with a set of actions, *primitive items*, and *primitive schemas* that encode the agent's initial "innate" behaviors. The activation of a schema consists of initiating its action. An activated schema is said to *succeed* if its predicted results are all in fact obtained, and to *fail* otherwise. The ratio of success and failure is called *reliability* and is memorized for each schema. As the agent interacts with its environment, it records new schemas that associate the observed context together with the action taken and the observed result. Schemas can be chained to achieve goals.

Drescher implemented Piaget's idea of "construction of reality" through a mechanisms that seeks underlying invariant that make the enaction of some schemas reliable. The mechanism identifies *relevant results* made of items that are slightly more frequent than average, and then seeks conditions under which the relevant results follow reliably. Such conditions are found through new schemas that allow recovering the context leading to the relevant result, and then performing the action again to asses the reliability of these new schemas. These new schemas act as "probes" that may evoke the manifestation of a "thing". They are only reliable if the "thing" is present. When such a schema is found, it is considered a "host schema" for a newly spawn "synthetic item" that represents this thing. Drescher notes that "synthetic item thus works backward from a thing's manifestation to define the very thing manifested" [6, p. 83]. This addresses the fundamental problem of *concept invention* in that "a synthetic item is a new element of the system's ontology—an element fundamentally different from the prior contents of the system's conceptual vocabulary [6, p. 81].

The designer defines the agent's goals by assigning *primitive values* to items. The mechanism can identify subgoals as specific sets of items to be activated or deactivated; and it constructs *composite actions* as sequences of schemas used to reach a subgoal. Items also have *instrumental values* and *delegated values* that enter into play in the selection of schemas and composite actions. The agent randomly alternates between explicit goal-pursuit and exploration, with probability defined by the designer.

[4] [13] [37] [18]

2.2   Perotto's

Similar to Drescher's, Perotto's schemas [24] are tuples that associate a context, an action, and an expected result. Perotto, however, introduces a new distinction between two kinds of context variables: hidden variables and observable variables as shown in Fig. 3.
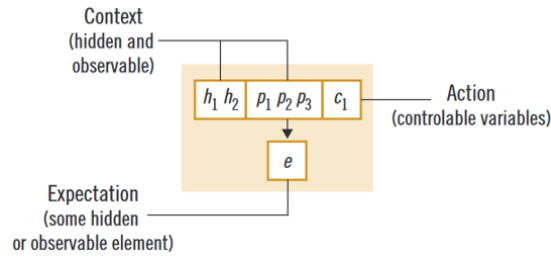
Fig. 3. Perotto's schema from [24] Fig. 2. A schema is composed of three vectors: context containing hidden variables and observable properties $(h_1, h_2, p_1, p_2, p_3)$, action (c_1), and expected result (e).

Perotto also introduced a hierarchy of schemas called the *anticipatory tree* represented in Fig. 4. Schema are organized in the *anticipatory tree* whose top-level schema describes the agent's highest-level goal and leaf schemas are deciders of actions in the environment.
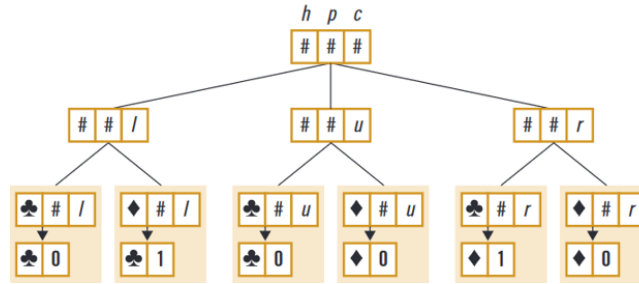
Fig. 4. Example of Perotto's anticipatory tree from [24] Fig. 10: "Final schematic tree for solving the flip problem.".

2.3   AERA

AERA[5] is a unified system that can learn about causal relations cumulatively, from experience [34, 35, 19]. The system's knowledge representation uniquely combines ampliative reasoning, autonomous resource management, explanation-driven compositional knowledge representation and reflection in a synergetic architecture that can grow its knowledge from a small "bootstrapping seed" [32]. Its compositional knowledge contains numerous peewee-size "micro-models" that can be combined in explainable ways to model experience [33].

The methodology behind AERA rests on three fundamental principles [38, 31, 20]: 1. During learning, a learner autonomously creates new knowledge through a variety of methods including hypothesis generation and testing; 2. autonomous learning must create models of causal relations through situated goal-driven reasoning, and 3. self-programming is based on transparent operational semantics based on peewee-size uniform knowledge models. Those familiar with Piaget's [2] fundamental insight that human learning requires creation of new information will recognize the first principle. For this principle, Piaget proposed the idea of schemas – information structures that mediate between perception and action, capturing and controlling experience. An important difference between AERA's knowledge representation and other proposals based on this idea (cf. [1, 6]) is its emphasis on non-axiomatic worlds and the resulting strong requirement for transversal defeasibility of acquired knowledge (cf. [28]) – in other words, that since no certainty about anything learned can be guaranteed, the methods for knowledge creation cannot be based on the assumption of knowledge of a "ground truth". The methodology behind AERA unifies this, and the above listed assumptions, in a coherent theoretical approach, with deep implications for knowledge creation in light of novelty: AERA starts out with only a tiny amount of "bootstrap" (seed) knowledge – the remainder of its knowledge, which soon vastly outsizes the seed[6], is then generated autonomously by the system itself, based on hypotheses that are inspired and verified through experience, using ampliative reasoning mechanisms.

- The knowledge representation in AERA is peewee (relatively small) and compositional; AERA can inspect, deconstruct, and reconstruct its models, as well as create hierarchies of models of its experience.
- The models in AERA are falsifiable, have degrees of confidence, and can be related to specific groups with certain attributes specifying their context.
- The choice of models and reasoning methods at any point in time is dynamic, based among other things on the extent of available resources (models and time).

---

[5] Autocatalytic Endogenous Reflective Architecture; see www.openaera.org – accessed Dec. 6th, 2024.

[6] In our implementation of S1 – an AERA agent that learns to do TV-style interviews, the ratio of the learned knowledge over the seed knowledge, after 20 hours of learning, was 53 [35].

AERA's peewee-size models, similar to e.g. Drescher's [6] schemas and the neuro-symbolic approach presented in [14], represent knowledge by contextualized programs that capture antecedental and successional states of important relationships (e.g. the relationship between a hand and an object that one wants to pick up). Unlike these, AERA places explicit representation of cause-effect relations at its core [36]. During its continuous learning, driven by a self-maintained dynamic goal hierarchy, the most useful (effective and efficient) models of cause-effect relations encountered in the environment are retained, while others are discarded. Based on predictions, informed experimentation, and strategic induction, the system autonomously creates new knowledge cumulatively, effortlessly unifying new information with old, even in light of contradictions. Further separating AERA from other approaches are its atomic elements for knowledge creation, whose operational semantics are transparent to the system's own reasoning, meaning that they can be inspected and learned by the same explicit, defeasible unified reasoning mechanisms as used for learning about the world.

The approach uniquely unifies cognition and meta-cognition in a single, coherent architecture, enabling efficient and effective self-reflection that lays a foundation for a capacity for cognitive growth. The result is a system that can dynamically employ ampliative reasoning, at any point in time, to predict both the immediate and far future, including its own cognitive resource use ("think time").

Key representational components of this approach include [29, 30]:

- **Facts** are statements that represent a cognitive event, e.g. operations happening in AERA itself; facts are the only entity in AERA that have complete certainty (corresponding to Descartes' axiom "I think, therefore I am").
- A **composite State model (CST)** captures a set of simultaneous (timeless) Facts that the AERA system considers to be (potentially) useful and/or necessary for properly predicting an event's consequences; used to e.g. represent a variety of subsets in the world such as spatial relations of objects, the values of a perceived object's properties, such as its position and color, etc.
- A **cognitive event model (CEM)** captures a (hypothesized) causal influence of a Fact on another Fact; used to represent the effect that an action or event in particular circumstances, e.g. a command initiated by AERA at some timestep, has on the state of the world at a later timestep. When a CEM fires it produces a prediction (while the production of a prediction is represented as a Fact, the confidence in the prediction itself is defeasible and is in part based on the confidence of the CEM).
- A **requirement model** (M_{req}) captures the (predicted) requirements/conditions that must be met for a CEM to be relevant in a particular context at a particular time.

AERA's programming paradigm allows the system itself to inspect these types of information structures, use them "for parts" to create variations, as well as create new ones from scratch. AERA's programming language, Replicode, allows it to autonomously use these building blocks for self-programming, in

the form of coherent plans, predictions, and goals that it can pursue [22, 21, 31]. These mechanisms have been demonstrated for control of virtual robots, language learning and use, and multimodal interaction [35, 29]).

Both CEMs and M_{req}s have a left-hand side (LHS) and a right-hand side (RHS), composed of sets of Facts and constraints (e.g., math equations). The LHS Facts describe (assumed) conditions under which the RHS Facts would be produced. Figure 5 shows how the above components lead to making a prediction. When some sensory Facts match everything in a CEM's LHS, a relevant M_{req} allows the CTS's instantiation, which means the conditions for making a prediction via a CEM are met. A successful instantiation of a CEM means that everything in its RHS will happen (probably – because all knowledge in AERA is defeasible). As the CEM predicts a future state after an event's occurrence, this is verified and marked as a "successful prediction" by AERA only if the prediction matches the state that is observed.
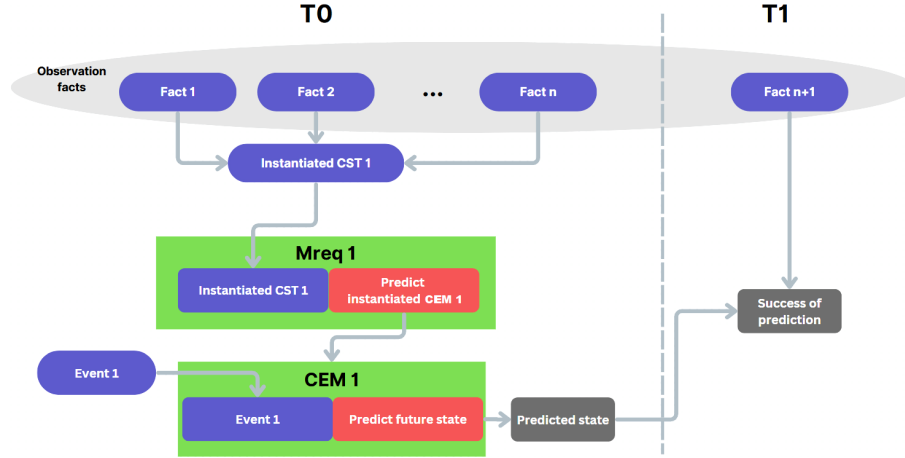
Fig. 5. The connection between CST, M_{req}, and CEM. The observation facts matching $CST1$ allow M_{req} to trigger a prediction by instantiating $CEM1$. AERA checks whether the subsequent observations match the predicted future state.

As an example, the triad of $CST1$, $M_{req}1$, and $CEM1$ can represent the physical movement behavior of an entity when it collides with another entity (e.g., a hand). $CEM1$ represents the dynamics of the movement, connecting the collision event (left-hand side of $CEM1$) to the moved entity's new position (right-hand side of $CEM1$). $CST1$ represents the preconditions of the behavior, such as the entities' position, weight, etc., and the fact that the point of collision and the moved entities' position must initially be the same. $M_{req}1$ predicts that if a striking entity hits another entity with the conditions described by $CST1$, the stricken entity's position will change a specific amount in the next time step.

The interactions of the operational principles of AERA are quite intricate in their entirety; important aspects that are covered elsewhere include induction – the creation of new knowledge – unified abduction-deduction – the creation of plans and explanations – and runtime relevance computations [22]. The unified integration of these operational principles in a single coherent architecture is what gives AERA its power, allowing its seed – initial human-provided knowledge – to be orders of magnitude smaller than what AERA can create on its own through learning from experience.

2.4   The LIDA cognitive architecture

The LIDA cognitive architecture [17] implements a schema mechanisms on top of a perceptual memory module called the Perceptual Associative Memory (PAM). Percepts are not directly received from the environment but constructed in the PAM through an active sensorimotor process. Other memory modules are implemented including episodic memory, declarative memory, and spatial memory. Knowledge is represented as a directed graph called the *node structure*.

LIDA's schemas work as templates with unbound variables in their context, action, and result. Actions may have unbound variables that qualify and modulate how an action is executed. Schemas are instantiated as *behaviors*.

LIDA implements *affective valence*. Affective valence can bias behavior selection towards desirable outcome. Liking and disliking are distinguished from wanting and dreading with the argument that they relate to distinct neural pathways in the brain [15]. Wanting and dreading are qualified by *incentive salience* of behaviors.
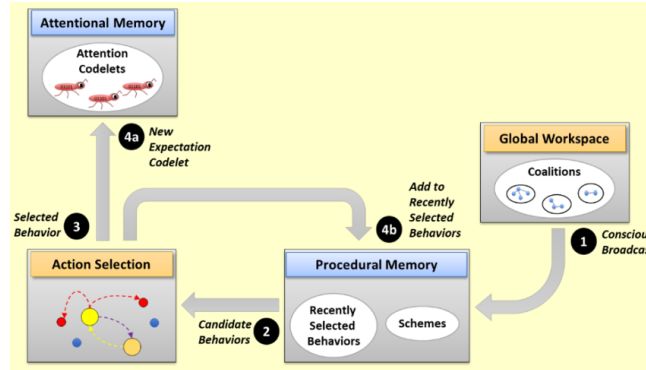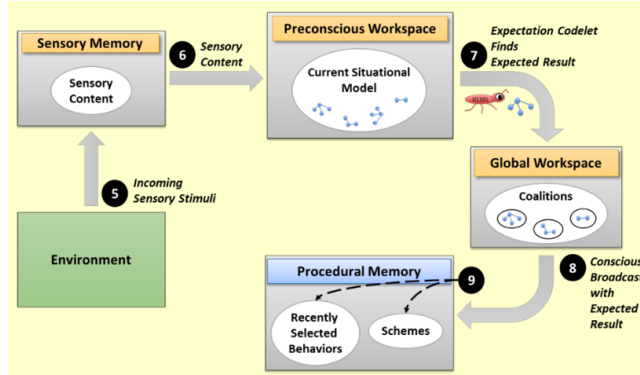

Fig. 6. Initiation of LIDA's procedural learning process ([17] Fig. 10)

Fig. 7. Continuing process of selectionist and instructionalist procedural learning that was initiated by Action Selection ([17] Fig. 11)
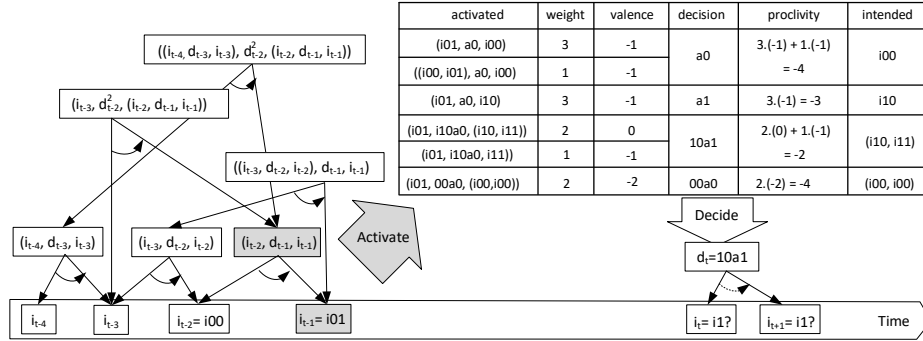
2.5   Georgeon's

Similar to Perotto, Georgeon modeled schemas as tuples (pre-condition, decision, post-condition), and organized them hierarchically.

In contrast with Drescher's and Perotto's mechanisms, however, pre-conditions and post-conditions are not properties of the world (hidden or observed) but are other schemas learned previously. The mechanism learns new schemas from the bottom up, with higher-level schemas made of a sequence of two previously-learned lower-level schemas, as illustrated in Fig. 8. The mechanism is not initialized with a top-level goal. Instead, it is initialized with a predefined set of low-level primitive schemas that define the agent's basic possibilities of interaction. In a robot, primitive schemas are hard-coded control loops involving actuators and sensory feedback.

The schema learning mechanism and selection works as follows. At the end of time step t, the agent records or reinforces the schemas:

- $(i_{t-2}, d_{t-1}, i_{t-1})$
- $((i_{t-3}, d_{t-2}, i_{t-2}), d_{t-1}, i_{t-1})$
- $(i_{t-3}, d^2, (i_{t-2}, d_{t-1}, i_{t-1}))$
- $((i_{t-4}, d_{t-3}, i_{t-3}), d^2, (i_{t-2}, d_{t-1}, i_{t-1}))$

If it does not yet exist, the new decision d^2 is constructed different from the decision d_{t-2} that was actually made at time $t-2$. For example, if the agent made decision $d_{t-2} = a0$ and enacted interaction $i_{t-2} = i00$, and then made decision $d_{t-1} = a0$, and enacted interaction $i_{t-1} = i01$, the agent learns the new decision $d^2 = i00a0$ consisting of trying to enact the interaction $i_t = i00$ and then do action $a_{t+1} = a0$. When decision d^2 has been selected and successfully enacted, the mechanism learns higher-level schemas on top of it. The rate of schema construction being constant, the number of schemas grows linearly with time. Older and unused schemas can be forgotten.

((i_{t-4}, d_{t-3}, i_{t-3}), d²_{t-2}, (i_{t-2}, d_{t-1}, i_{t-1}))

(i_{t-3}, d²_{t-2}, (i_{t-2}, d_{t-1}, i_{t-1}))

((i_{t-3}, d_{t-2}, i_{t-2}), d_{t-1}, i_{t-1})

(i_{t-4}, d_{t-3}, i_{t-3})   (i_{t-3}, d_{t-2}, i_{t-2})   (i_{t-2}, d_{t-1}, i_{t-1})

Activate

i_{t-4}   i_{t-3}   i_{t-2}= i00   i_{t-1}= i01

activated	weight	valence	decision	proclivity	intended
(i01, a0, i00)	3	-1	a0	3.(-1) + 1.(-1) = -4	i00
((i00, i01), a0, i00)	1	-1			
(i01, a0, i10)	3	-1	a1	3.(-1) = -3	i10
(i01, i10a0, (i10, i11))	2	0	10a1	2.(0) + 1.(-1) = -2	(i10, i11)
(i01, i10a0, i11))	1	-1			
(i01, 00a0, (i00,i00))	2	-2	00a0	2.(-2) = -4	(i00, i00)

Decide

d_t=10a1
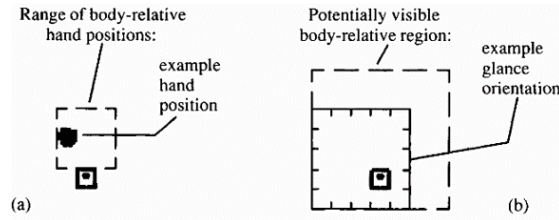
i_t= i1?   i_{t+1}= i1?   Time

Fig. 8. Schema learning and selection. Schemas are nested tuples: (pre-schema, decision, post-schema). Over time, new decisions and new schemas are learned from the bottom up. Recently enacted schemas (in gray) activate the previously-learned higher-level schema whose pre-schema they match. Activated schemas propose their post-schemas with a proclivity value calculated from the activation weight and the expected valence. The schema with the highest proclivity is selected to try to enact.

In essence, the agent learns habits of interaction that are organized hierarchically, with shorter sequential habits learned first, and longer sequences of habits made of sequences of shorter habits later learned on top of them. Such habit learning relates to Woolford's *sensorimotor sequence reiterator* [39].
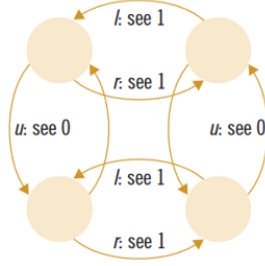
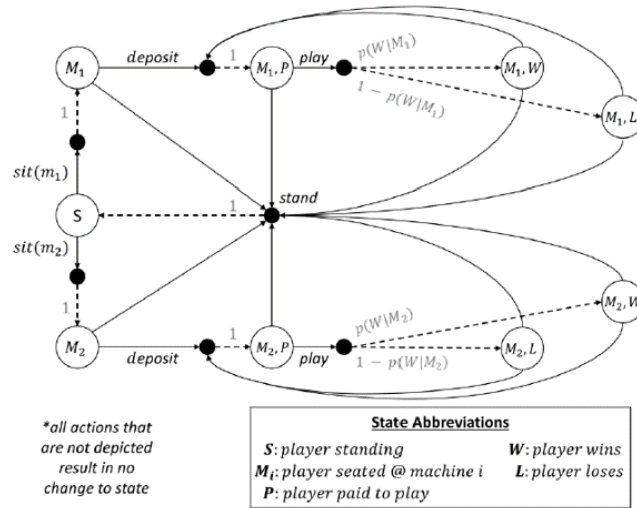## 3   Benchmarks

Fig. 9 shows Drescher's experimental settings.

Range of body-relative hand positions: example hand position

Potentially visible body-relative region: example glance orientation

(a)   (b)

Fig. 9. Drescher's benchmark from [6] Fig.6.1 "Hand and glance ranges".
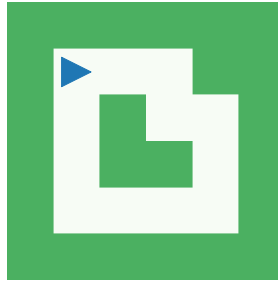
Fig. 10 shows Perotto's experimental settings [24].

Fig. 12 shows Georgeon' experimental settings [10]. This experimentation demonstrates a self-programming effect [9]. Video Demo [7].

Fig. 10. Perotto's benchmark from [24] Fig.9 "The hyper-flip problem".

Fig. 11. The multi-armed bandit environment used to assess LIDA's schema Mechanisms ([16] Fig. 5)

Fig. 12. Georgeon's benchmark (Adapted from [10]). The agent can move forward to an empty cell, bump into a wall (green cell), turn to the left or to right by 90°, or "feel" the cell in front, to the left, or to the right. Sensory signal is a single-bit feedback from the action.

4 Schema mechanisms and theory of knowledge

Table 1 shows a comparison of the schema mechanisms presented in this paper.

Table 1. Comparison of schema mechanisms

	Drescher's	Perotto's	AERA	LIDA	Georgeon's
Chaining to goal	✓	✓	✓	✓	
Marginal attribution	✓	✓	✓	✓	
Hierarchical goals		✓	✓		
Non-representational datum				✓	✓
Intrinsic motivation				✓	✓
Sequential schemas				✓	✓
Active perception					✓
3D spatial schemas				✓	✓
Reflexivity			✓	✓	
Defeasibility of acquired knowledge			✓		
Self-programming			✓		✓

Bettoni has criticized Drescher's schema mechanism in stating that "Drescher's Constructivism is not Piaget's Constructivism, mainly because of its tacit acceptance of *cognitive dogmatism*" ([3], p. 6). Bettoni describes cognitive dogmatism as taking for granted that "patterns and structures of objects, attributes, relations, etc. [...] be as much as possible true copies of 'original' objects, attributes, relations etc. in the world" ([3], p. 1). Indeed, theories of enaction as well as of radical constructivism have insisted that we should not take the sensory signals as representational items of an alleged reality.

The game of Mastermind provides an emblematic example in which the observation is not representational. Player 2 attempts to infer a hidden combination of colored pegs ("hidden state") by proposing guesses ("actions"), which Player 1 responds to with feedback pegs ("observation"). Black pegs indicate a correct color in the correct position, while white pegs indicate a correct color in the wrong position. Since the observation depends on the action, there exist no function or stochastic distribution that map the state to the observation. The absence of such function or distribution is expressed in cognitive terms by that the observation is not "representational" of the state.

Software to play Mastermind have been proposed using diverse techniques such as entropy measure and evolutionary algorithms [5]. These techniques, however, require that the semantics of the feedback is known beforehand. We propose the *constructivist Mastermind analogy* that likens a general learner to someone playing a giant game of mastermind where they start with no knowledge of the hidden combination and even the semantics of actions and feedback. The player may never find the hidden combination or goal but may survive and strive for some time in a satisfactory *knowledge niche*.

As reviewed in Section 3, most schema mechanisms have been tested in settings in which the sensory signal is representational. Nonetheless, the fact that

they have not been tested with non-representational sensory signal does not mean that they would not work or could not be adapted to such settings. Georgeon's mechanism may constitute an illustration of that. In Georgeon's schemas, the sensory data is feedback from action rather than beeing representational. The pre-condition and post-condition of schemas are are just other schemas all the way down to non-representational primitive schemas. In essence, the agent knows its current context by possibilities of interaction rather than by representational data.

Georgeon's schema mechanism is not targeted at reaching a predefined goal. Since sensory data does not represent the environment's state, the agent cannot have a goal represented as an environment state. Instead, the agent's behavior is driven by the expected valence of each decision. The calculation of expected valence may incorporate predefined preferences for some primitive schemas or different intrinsic motivation principles such as an estimation of information gained.

5   Conclusion

Beyond the first stage called by Piaget the *sensorimotor level*, comes a second stage he calls the *first level of pre-operative thought* ([26], p. 30). It is at this second stage that the subject begins to differentiate itself from the objects. The subject becomes able to manipulate the schemas by thought. Piaget formulates this process as follows:

> "On top of simple actions that ensure direct interdependence be-
> tween the subject and objects, in certain cases, a new type of action
> is superimposed, one that is internalized and more precisely con-
> ceptualized: for example, in addition to being able to move from
> A to B, the subject acquires the ability to mentally represent this
> movement from A to B and to evoke, through thought, other move-
> ments" (translated from [26], p. 30).

Future studies on schema mechanisms will have to address this second stage to tackle fundamental questions of knowledge abstraction. We expect this will involve a cognitive architecture that has the capacity to represent spatial proper-ties of schemas, and to simulate the enaction of schema in different spatial frames of reference. LIDA and ECA [8] constitute initial attempts in this direction .

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Arbib, M.A.: The handbook of brain theory and neural networks. MIT press (2003)
2. Berlyne, D., Piaget, J., Piercy, M.: The Psychology of intelligence. Routledge (2003)
3. Bettoni, M.C.: Made-up minds: A constructivist approach to artificial intelligence - a book review **6**(3), 234–240. https://doi.org/10.3233/AIC-1993-63-413
4. Chaput, H.: The constructivist learning architecture: A model of cognitive development for robust autonomous robots
5. Cotta, C., Merelo Guervós, J.J., Mora García, A.M., Runarsson, T.P.: Entropy-driven evolutionary approaches to the mastermind problem. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) Parallel Problem Solving from Nature, PPSN XI. pp. 421–431. Springer. https://doi.org/10.1007/978-3-642-15871-1_43
6. Drescher, G.L.: Made-up minds: a constructivist approach to artificial intelligence. Artificial intelligence, MIT Press
7. Georgeon, O.L.: The small loop challenge (2012), https://youtu.be/Mstl8BRCbSI
8. Georgeon, O.L., Lurie, D., Robertson, P.: Artificial enactive inference in three-dimensional world. Cognitive Systems Research **86**, 101234 (2024). https://doi.org/10.1016/j.cogsys.2024.101234
9. Georgeon, O.L., Riegler, A.: CASH only: Constitutive autonomy through motorsensory self-programming **58**, 366–374 (2019). https://doi.org/10.1016/j.cogsys.2019.08.006
10. Georgeon, O.L., Ritter, F.E.: An intrinsically-motivated schema mechanism to model and simulate emergent cognition **15-16**, 73–92. https://doi.org/10.1016/j.cogsys.2011.07.003, https://linkinghub.elsevier.com/retrieve/pii/S1389041711000398
11. Glasersfeld, E.v.: Radical constructivism: a way of knowing and learning. No. 6 in Studies in mathematics education series, Falmer Press, reprinted edn.
12. Guerin, F., Kruger, N., Kraft, D.: A survey of the ontogeny of tool use: From sensorimotor experience to planning **5**(1), 18–45. https://doi.org/10.1109/TAMD.2012.2209879, http://ieeexplore.ieee.org/document/6248675/
13. Guerin, F., McKenzie, D.: A piagetian model of early sensorimotor development. In: Proceedings of the Eighth International Conference on Epigenetic Robotics. pp. 29–36. Lund University Cognitive Studies, Kognitionsforskning, Lunds universitet, issue: -
14. Komrusch, S., Minsky, H.: Symbolic guidance for constructivist learning by neural model. In: Proceedings of the Third International Workshop on Self-Supervised Learning. pp. 63–76. PMLR, https://proceedings.mlr.press/v192/komrusch22a.html, ISSN: 2640-3498
15. Kringelbach, M., Berridge, K.: The functional neuroanatomy of pleasure and happiness **9**, 579–87 (2010)
16. Kugele, S.: Constructivist procedural learning for grounded cognitive agents (in press)
17. Kugele, S., Franklin, S.: Learning in LIDA **66**, 176–200 (2021). https://doi.org/10.1016/j.cogsys.2020.11.001
18. Miller, M.S.P.: Building Minds with Patterns. Independently published
19. Nivel, E., Þórisson, K.R., Thórisson, K.R., Dindo, H., Pezzulo, G., Rodriguez, M., Corbato, C., Steunebrink, B., Ognibene, D., Chella, A., et al.: Autocatalytic endogenous reflective architecture (2013)

20. Nivel, E., Thórisson, K.R.: Self-programming: Operationalizing autonomy. In: 2nd Conference on Artificial General Intelligence (2009). pp. 212–217. Atlantis Press (2009)
21. Nivel, E., Thórisson, K.R.: Replicode: A constructivist programming paradigm and language. Technical RUTR-SCS13001, Reykjavik University School of Computer Science (2013)
22. Nivel, E., Thórisson, K.R.: Towards a programming paradigm for control systems with high levels of existential autonomy. In: International Conference on Artificial General Intelligence. pp. 78–87. Springer (2013)
23. Oudeyer, P.Y., Kaplan, F., Hafner, V.V.: Intrinsic motivation systems for autonomous mental development **11**(2), 265–286 (2007). https://doi.org/10.1109/TEVC.2006.890271
24. Perotto, F.S.: A computational constructivist model as an anticipatory learning mechanism for coupled agent–environment systems **9**
25. Piaget, J.: La naissance de l'intelligence chez l'enfant. Actualités pédagogiques et psychologiques, Delachaux et Niestlé
26. Piaget, J.: L'épistémologie génétique. Quadrige, PUF
27. Piaget, J.: The Principles of Genetic Epistemology. Psychology Press
28. Pollock, J.L.: Defeasible reasoning. Cognitive science **11**(4), 481–518 (1987)
29. Sheikhlar, A.: Autonomous causal generalization (2024)
30. Sheikhlar, A., Thórisson, K.R.: Causal generalization via goal-driven analogy. In: International Conference on Artificial General Intelligence. pp. 165–175. Springer (2024)
31. Thórisson, K.R.: A new constructivist ai: from manual methods to self-constructive systems. In: Theoretical Foundations of Artificial General Intelligence, pp. 145–171. Springer (2012)
32. Thórisson, K.R.: Seed-programmed autonomous general learning. In: International Workshop on Self-Supervised Learning. pp. 32–61. PMLR (2020)
33. Thórisson, K.R.: The" explanation hypothesis" in general self-supervised learning. In: IWSSL. pp. 5–27 (2021)
34. Thórisson, K.R., Bieger, J., Li, X., Wang, P.: Cumulative learning. In: Artificial General Intelligence: 12th International Conference, AGI 2019, Shenzhen, China, August 6–9, 2019, Proceedings 12. pp. 198–208. Springer (2019)
35. Thórisson, K.R., Nivel, N., Steunebrink, B.R., Helgason, H.P., Pezzulo, G., Sanz Bravo, R., Schmidhuber, J., Dindo, H., Rodríguez Hernández, M., Chella, A., et al.: Autonomous acquisition of natural situated communication. IADIS International Journal on Computer Science And Information Systems **9**(2), 115–131 (2014)
36. Thórisson, K.R., Talbot, A.: Cumulative learning with causal-relational models. In: Artificial General Intelligence: 11th International Conference, AGI 2018, Prague, Czech Republic, August 22-25, 2018, Proceedings 11. pp. 227–237. Springer (2018)
37. Thórisson, K.R.: A new constructivist AI: From manual methods to self-constructive systems. In: Wang, P., Goertzel, B. (eds.) Theoretical Foundations of Artificial General Intelligence, vol. 4, pp. 145–171. Atlantis Press, series Title: Atlantis Thinking Machines
38. Thórisson, K.R.: Seed-programmed autonomous general learning. In: Proceedings of the First International Workshop on Self-Supervised Learnin. Proceedings of Machine Learning Research, vol. 131, pp. 32–61. http://proceedings.mlr.press/v131/
39. Woolford, F.M.G., Egbert, M.D.: A precarious sensorimotor sequence reiterator for modelling enactive habits. In: ALIFE 2020: The 2020 Conference on Artificial Life. pp. 771–779. MIT Press (2020). https://doi.org/10.1162/isal_a_00249

40. Ziemke, T.: The construction of 'reality' in the robot: Constructivist perspectives on situated artificial intelligence and adaptive robotics **6**(1), 163–233. https://doi.org/10.1023/A:1011394317088, https://doi.org/10.1023/A:1011394317088