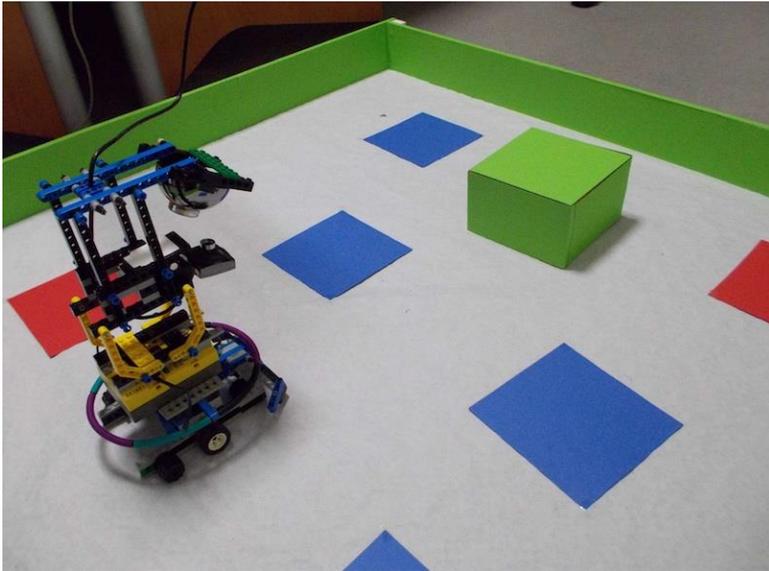


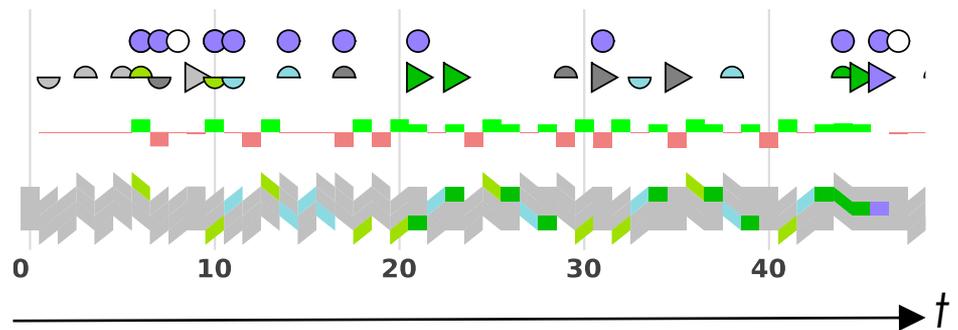
Intelligence Artificielle Développementale



27 Septembre 2023

Olivier.georgeon@gmail.com

<http://www.oliviergeorgeon.com>



Déroulement du cours

UE « IA et Cognition » (30h): Marie Lefevre

**Module « IA Développementale » (12h)
Olivier Georgeon**

- Lundi 18 septembre: 3h
- Lundi 25 septembre: 3h
- **Mercredi 27 septembre: 3h**
- Lundi 2 octobre: 3h

- Contrôle des connaissances:
 - TD par groupe de 2: 40% de la note de contrôle continu
 - Examen final: 7 points sur 20

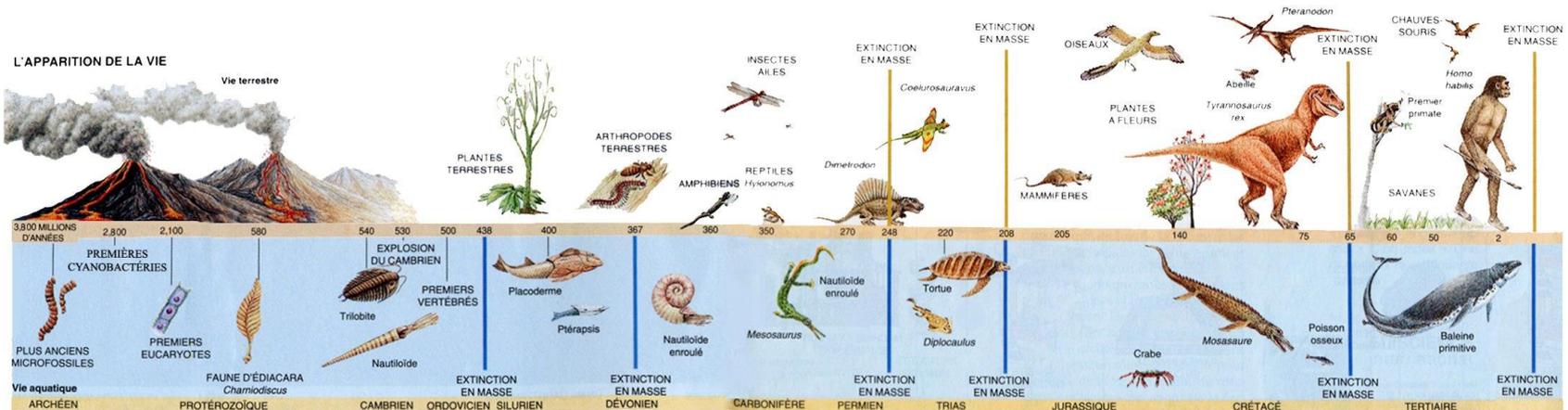
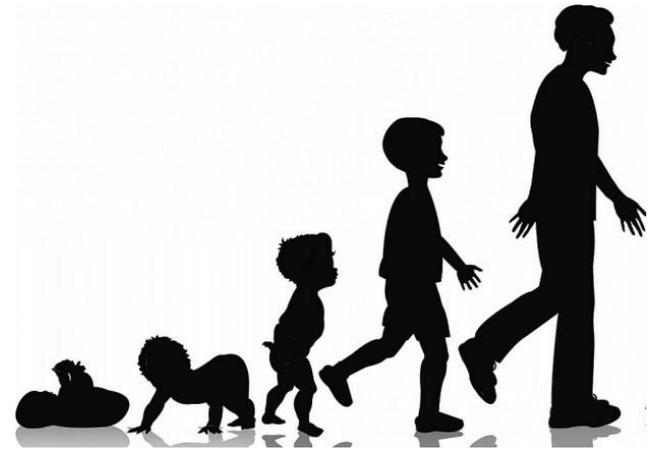
Théorie du développement

Ontogénèse vs. phylogénèse

LA développementale:
Ontogénèse.

Ontogénèse

Phylogénèse

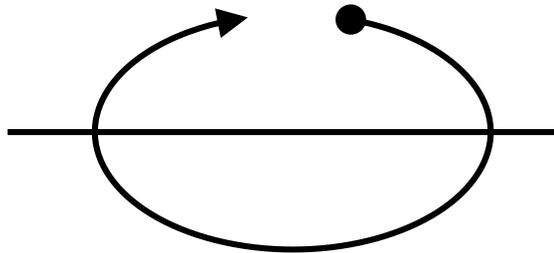


Couplage structurel

On parle de **couplage structurel** chaque fois que survient une histoire d'interactions récurrentes responsables d'une congruence structurelle entre deux systèmes ou plus.

(Maturana & Varela, 1989, p. 78)

Domaine interactionnel: Spécification du couplage

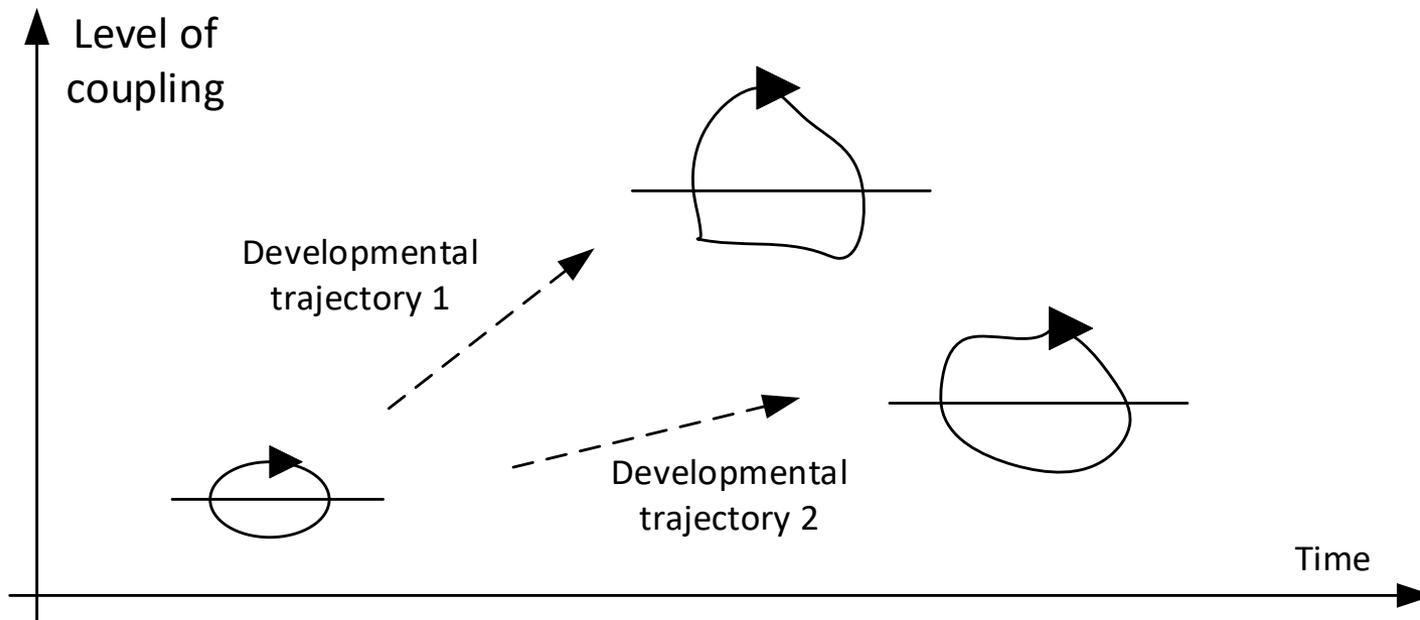


Couplage

Sujet
/
Objet

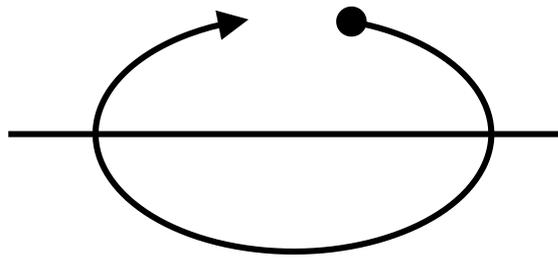
Apprentissage développemental

Capacité d'un sujet a développer individuellement sa structure interne et son couplage avec son environnement. (Théories de Piaget).



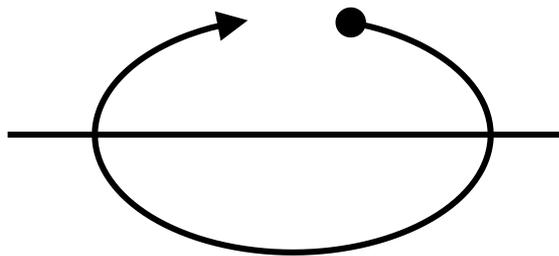
Constitutive autonomy

Dans un système artificiel: dissocier
« couplage cognitif » vs « couplage physique »



*Decisional mechanism
/
Software Interface*

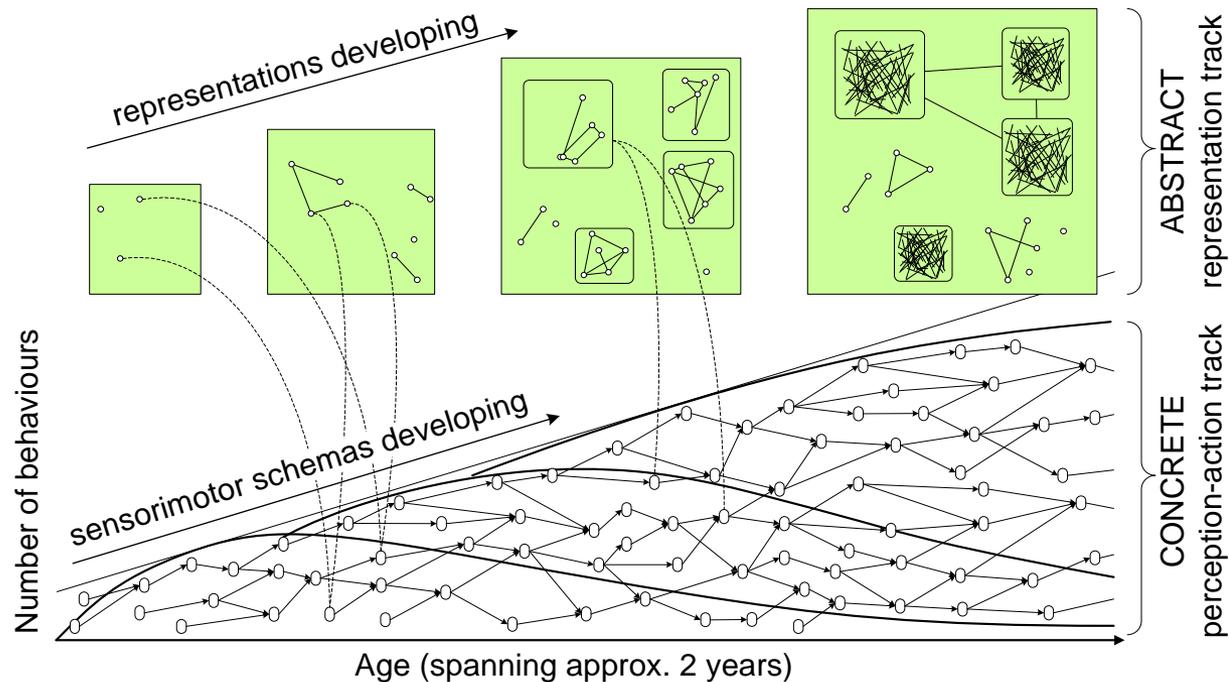
↑
Evolve



*Software
/
World*

Fixe

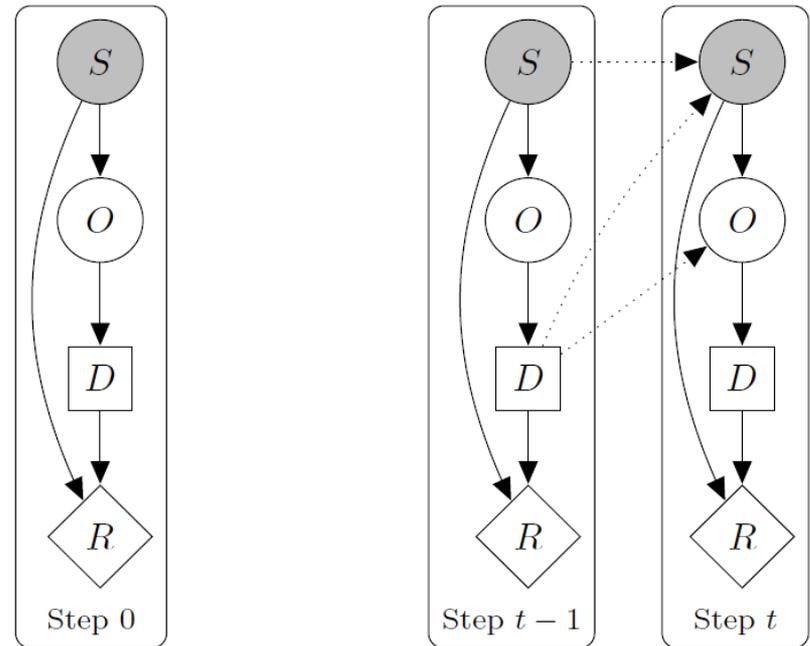
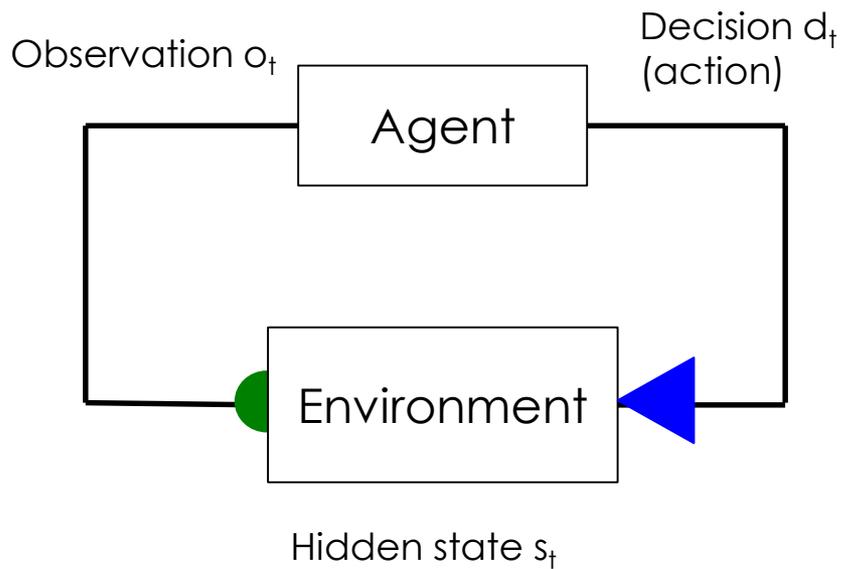
Construire des connaissances à partir de régularités d'interactions (schèmes sensorimoteurs)



Conceptual diagram of developmental learning (Guerin, Krüger, and Kraft, 2013)

Formalisation

POMDP

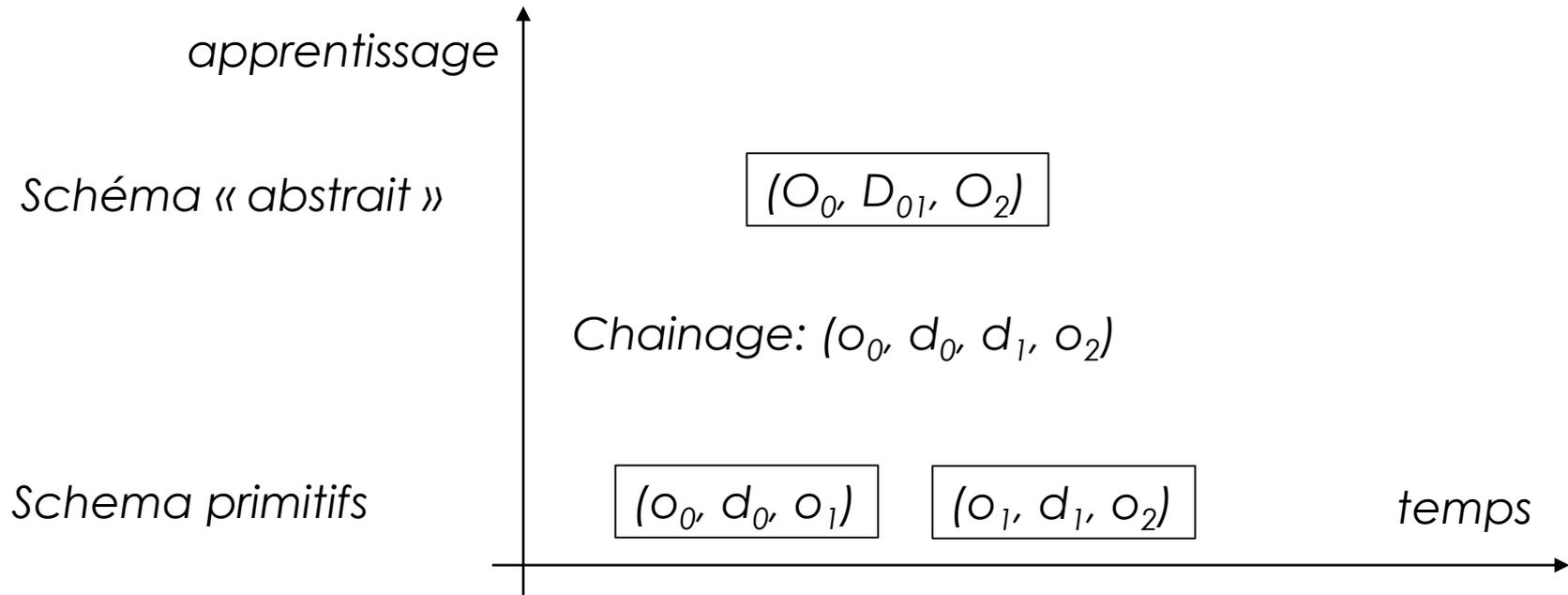


History at time t : $h_t = \{o_0, d_0, o_1, d_1, \dots, o_{t-1}, d_{t-1}, o_t\} \in H_t$

Decision at time t : $d_t := \sigma(h_t)$, $\sigma \in \Sigma$

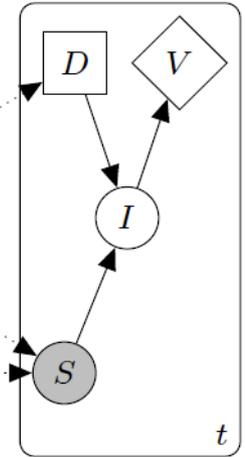
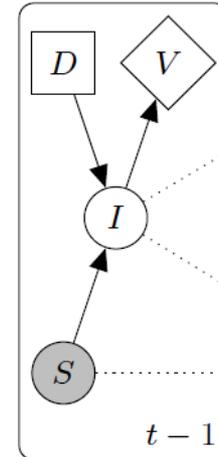
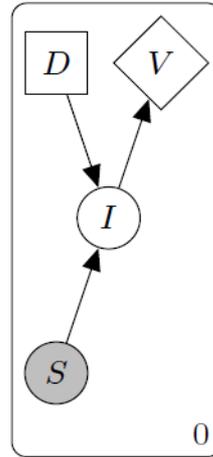
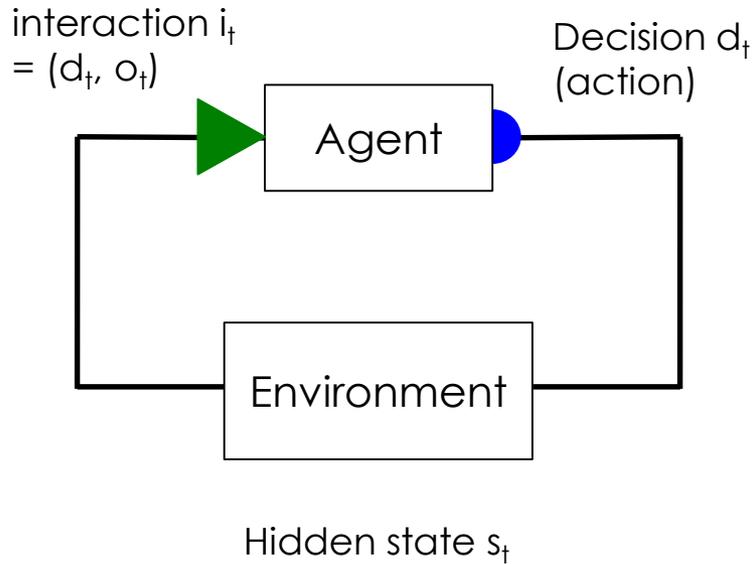
Schema learning

Apprentissage bottom-up par chainage et abstraction



Drescher (1991 « The made-up mind... »), Thórisson (2020 « seed programmed learning... »), Perrotto (2013, « computational constructivist model... »)

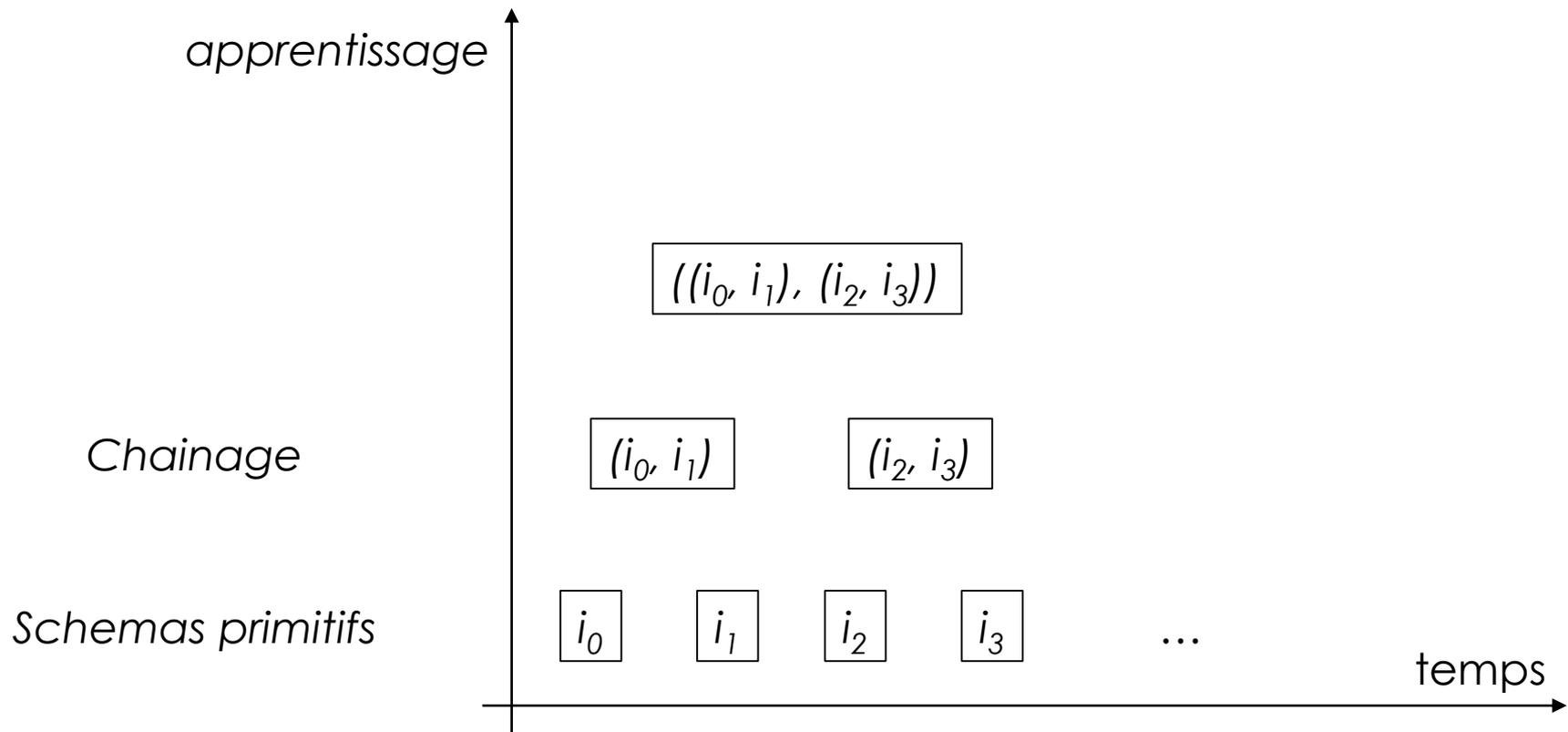
Enactive MDP



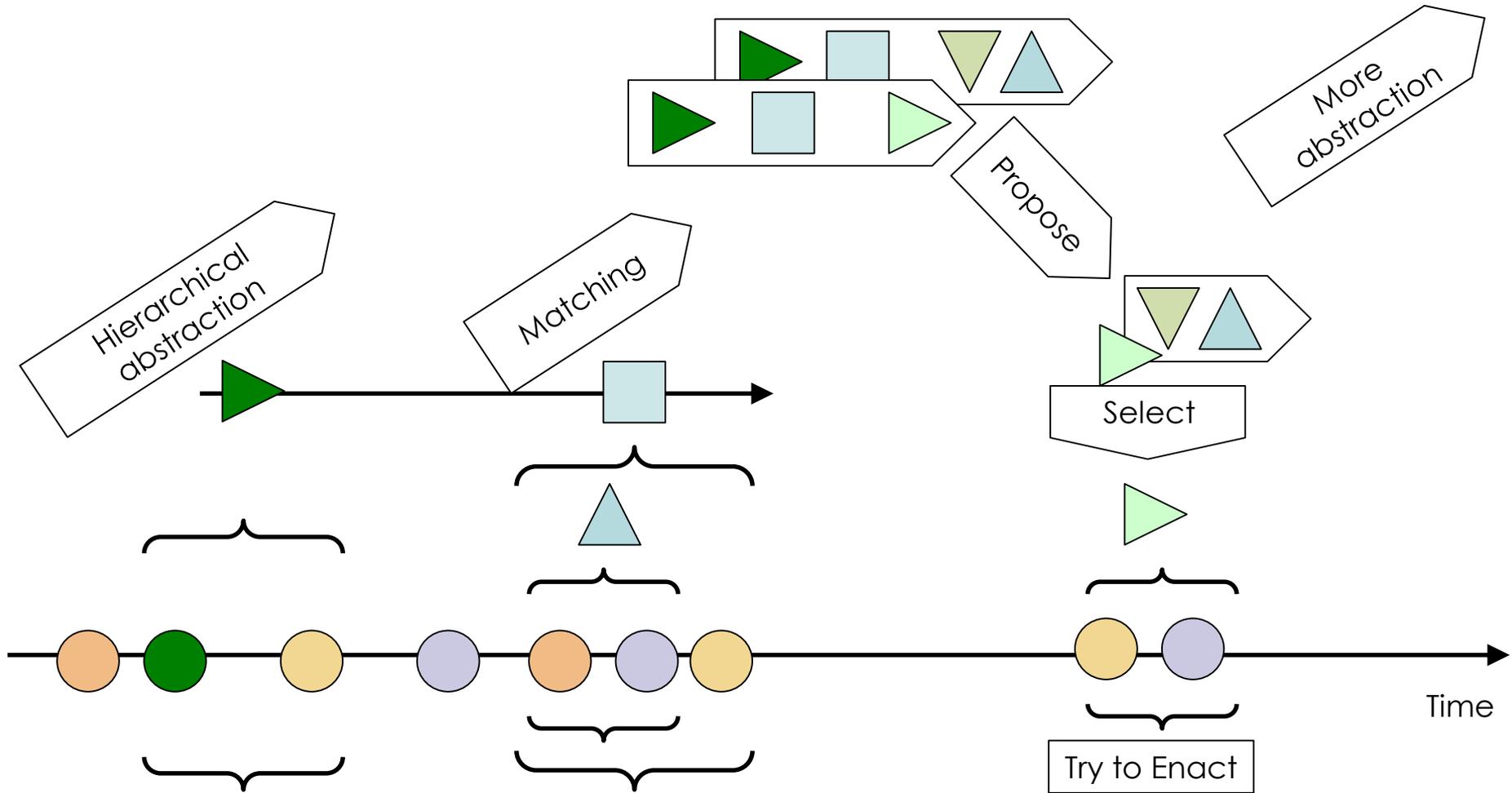
History at time t : $h_t = \{i_0, \dots, i_{t-1}\}$

Decision at time t : $d_t := \sigma(h_t)$

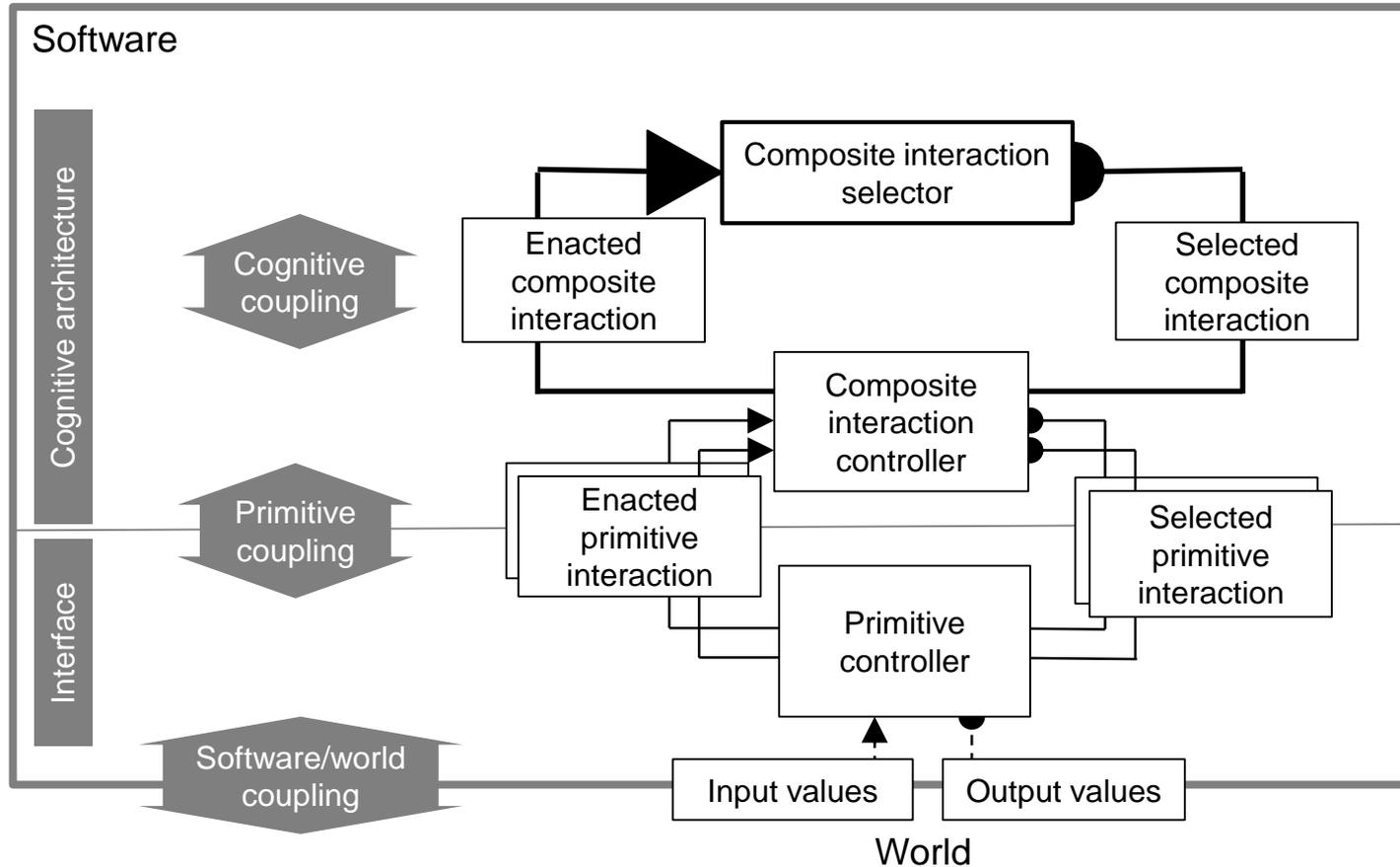
Bottom up interaction chaining



Apprentissage de régularités hiérarchique



Evolution du couplage cognitif



Georgeon O., Riegler A. (2019). Constitutive Autonomy through Motorsensory Self-Programming. Cognitive Systems Research, 58: 366-374.

Et avec des transformers ?

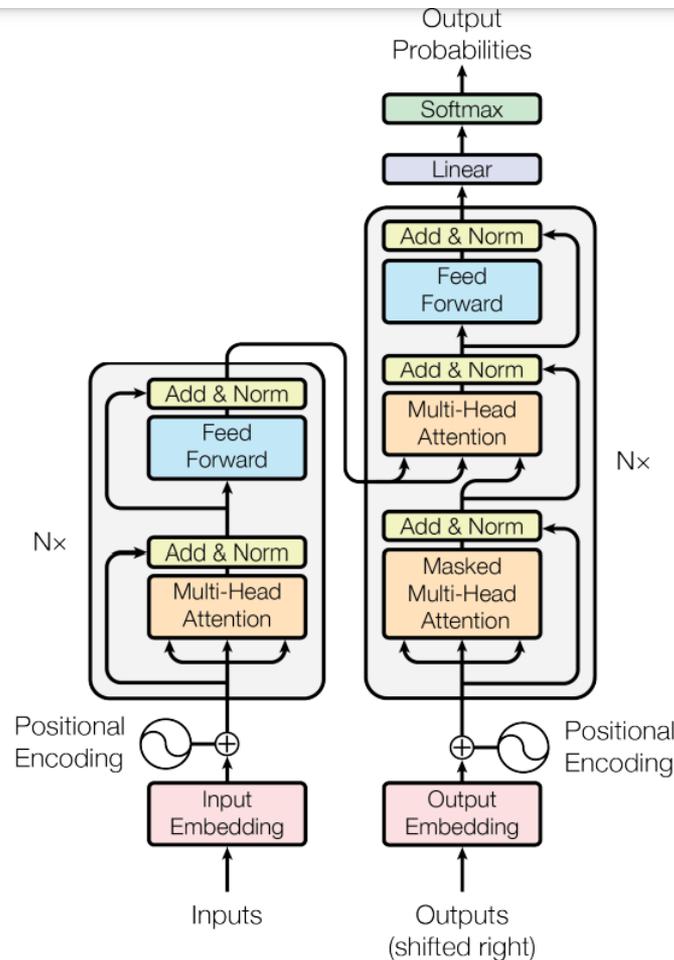


Figure 1: The Transformer - model architecture.

- Pourrait prédire les prochaines séquences d'interaction
- Mais :
 - avec une dépendance temporelle fixée a priori
 - Avec des niveaux hiérarchiques fixés a priori

- <https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/>
- <https://towardsdatascience.com/transformers-explained-visually-part-1-overview-of-functionality-95a6dd460452>
- <https://colab.research.google.com/drive/1SiF0KZJp75rUeetKOWqpsA8clmHP6jMg?usp=sharing>

Auto programmation

Un système auto-programmant est un système capable d'apprendre du code qu'il peut ré-exécuter.

Cela pose de nombreuses questions:

- Quel jeu d'instructions?
- Quel moteur d'exécution?
- Quelle finalité?
 - Pourquoi apprendre un programme plutôt qu'un autre?

Travaux dirigés

Séance 3

Consignes pour les TP

- Par groupe de 2.
- Rendre un seul rapport à la fin .PDF *ou* .IPYNB ?
- Indiquer bien le nom des deux membres du groupe
- Envoyer par mail à olivier.georgeon@gmail.com pour le **vendredi 17 novembre 2023** 23h59
- Pour chaque agent
 - Décrire les principes de l'algorithme que vous avez implémenté
 - Inclure des captures d'écran des traces affichées à la console dans différents environnements
 - Expliquer les comportements obtenus en vous appuyant sur les traces.
- Conclure sur ce que vous retirez de cette expérience et suggestions de comment aller plus loin (Activité 4)

Setup

Suivre la procédure écrite ici <https://github.com/OlivierGeorgeon/TestROS/wiki/Implementer-un-agent-rudimentaire>

Créer un nouveau projet python dans votre environnement de développement Python favori (par exemple Pycharm) contenant le fichier world.py. Vous avez deux méthode possibles :

- Cloner le repository <https://github.com/OlivierGeorgeon/TestROS>
- Créer un nouveau projet et copier le fichiers world.py

Exécuter world.py et vérifiez que vous obtenez la trace d'interaction montrée en Figure 1 sur <https://github.com/OlivierGeorgeon/TestROS/wiki/Implementer-un-agent-rudimentaire>)

Agent 1

Dans le fichier world.py, modifier la class Agent pour créer l'Agent 1 en suivant les instructions :

<https://github.com/OlivierGeorgeon/TestROS/wiki/Agent-1>

Tester votre agent dans Environment1 puis dans Environment2 en commentant et décommentant les lignes appropriés (lignes 70 et 71 dans le fichier world.py initial)

Agent 2

Créer l'Agent2 en suivant les instructions :

<https://github.com/OlivierGeorgeon/TestROS/wiki/Agent-2>

Tester votre agent dans Environment1 puis dans Environment2 en commentant et décommentant les lignes appropriés (lignes 70 et 71 dans le fichier world.py initial)

Modifier la table des valences d'interaction.

Agent 3

Créer l'Agent3 en suivant les instructions :

<https://github.com/OlivierGeorgeon/TestROS/wiki/Agent-3>

Tester votre agent dans l'environnement TurtlePy

Agent 4

Créer l'Agent4 en suivant les instructions :

<https://github.com/OlivierGeorgeon/TestROS/wiki/Agent-4>

Tester votre agent dans les environnements 1, 2 et 3 et montrer qu'il sais s'adapter à chacun de ces trois environnements.