

L'architecture SOAR - Résumé

- Créée en 1981 par John Laird, Paul Rosenbloom (Développeurs), et Allen Newell (Prof. Référent)
- Principes fondamentaux :
 - Système à base de connaissances : l'agent va stocker les connaissances apprises sous forme d'état cognitif qui vont être mémorisés, voire réutilisés dans le futur.
 - Repose sur des règles : permet à l'agent d'agir en conséquence suivant les conditions présentes dans l'état cognitif de l'agent. Pour les tâches les plus complexes, cette application de règles peut se faire en parallèle afin de faciliter l'exécution.
 - Modulaire : les différents modules sont séparés en fonction de leur rôle à jouer au sein de l'agent. Facilite le développement de nouvelles fonctionnalités.
 - Opérateurs : l'agent utilise des opérateurs sur les données intérieures aux états, pour avoir des résultats et parvenir à formuler des conclusions, ou bien des actions.
 - Apprentissage en continu : les expériences vécues et les résultats obtenus permettent à l'agent de continuellement développer sa base de connaissance comme un humain.
 - Flexibilité : il vient s'adapter en fonction de ses connaissances afin de réaliser au mieux la tâche qui lui est confiée. Il peut très bien se baser sur les anciennes tâches réalisées pour réussir une tâche plus complexe.
- Composantes de l'architecture :
 - Capteurs divers
 - Mémoires à long terme, mises à jour par le système :
 - Mémoire Sémantique : Contient les données sous forme de faits, organisées selon un réseau sémantique. Elle fournit au système des éléments clés pour la prise de décision « Quelle est la situation ? »
 - Mémoire procédurale : Stocke les règles et permet de guider le comportement du système en fournissant les règles nécessaires. « Comment effectuer une action ? »
 - Mémoire épisodique : Enregistre l'état de la mémoire régulièrement et permet de revenir en arrière. « Quelles actions-ai-je déjà effectué dans cette situation ? »
 - Mémoire de travail symbolique :
 - Contient : l'objectif visé, les données reçues des mémoires à long terme et des capteurs, les règles actives et les actions en cours.
 - Point central de la prise de décision
 - Interagit avec l'ensemble des modules de l'architecture
- Cycle de traitement :
 1. Sélection d'un objectif
 2. Sélection des règles
 3. Application des règles
 4. Mise à jour de la mémoire de travail
 5. Evaluation des résultats
 6. Apprentissage et adaptation
- Différences avec d'autres architectures d'ia :
 - Architecture Soar issue d'un « patron » d'architectures : ACT-R
 - Soar -> nombre de buffers & taille de chunks illimités / ACT-R : nombre de buffers fixe
 - Soar -> mémoire : un élément = plusieurs règles / ACT-R : un élément = une règle

- Soar -> possibilité de retour en cas d'échec / ACT-R : pas de retour (mémoire)
- Soar -> plus de modularité & flexibilité pour apprentissage et adaptation / ACT-R : axé apprentissage et mémorisation à long terme
- Plusieurs applications de Soar dans la société d'aujourd'hui :
 - Militaire : RGATOR (véhicule de repérage pour les missions militaires « sans pilote »)
 - Robotique diverse : Robo-Soar & Splinter
 - Superordinateur : Watson (IBM) -> capable de répondre à des questions et de comprendre les situations de manière rapide
 - NL-Soar (Natural language)
 - Jeux-vidéos : utilisés pour les PNJ & tout autre objet devant imiter le comportement humain
- Avantages :
 - Bons résultats quant à l'étude du raisonnement humain
 - Versatile : modulaire, flexible et conçu pour s'adapter à tout type de contexte
 - Prise en charge de la perception
 - Apprentissage par instruction
- Inconvénients :
 - Densité et complexité du système
 - Apprentissage limité
 - Besoin de matériel pour la perception
 - Modèle « lourd » (complet) et donc long à développer
 - Problèmes métaphysiques quant à la comparaison à l'humain
- Débat :
 - Manque de puissance ou mauvais modèle ?
 - Comment peut-on savoir si c'est un problème de puissance ? Il faut match la puissance pour savoir si on utilise un bon ou un mauvais modèle.
 - Si il y a un manque de puissance c'est parce qu'il n'y a pas d'agent qui est mis dans le développement de ce genre d'architecture parce que les architectures existantes "suffisent".
 - Il y a forcément un manque de puissance avec la quantité de données à calculer en temps réel. En terme de consommation, un humain c'est h24 une quantité énorme de ressources.
 - On cherche des résultats et les architectures qu'on utilise actuellement en donnent de bons donc on ne cherche pas à changer de modèle.
 - Un cerveau humain arrive à faire tous ces calculs alors on devrait théoriquement pouvoir le faire, c'est possiblement un problème d'architecture.
 - Common Model of Cognition, un pas de plus vers l'IA forte ?
 - Une IA cognitive est plus crédible que les autres pour être considérée comme une IA forte mais le simple fait d'avoir une architecture cognitive ne suffit pas à avoir une IA forte. C'est un pas vers l'IA forte mais un petit (à supposé que l'IA forte soit faisable).
 - L'architecture cognitive ne reproduit pas toute la façade émotionnelle de l'humain.
 - L'architecture cognitive, une étape nécessaire ou qui se contourne ?
 - La cognition est nécessaire à la conscience donc elle est nécessaire pour avoir une IA forte, mais jamais on y arrivera directement sans contourner le problème.

- Il faudrait comprendre comment fonctionne la conscience et l'esprit humain pour pouvoir la reproduire, on ne doit donc pas la reproduire mais la simuler / contourner le problème.