

# Synthèse SOAR

Alexandre Faure, Arthur Desbiaux, Emilien Komlenovic, Valentin Cuzin-Rambaud

October 2024

## 1 Introduction

C'est en 1950 que commencent les débuts de la recherche sur la cognition, mais la date la plus importante, c'est bien les années 1980. Cette année marque le début des recherches sur la cognition unifiée (implémenter un comportement humain dans un robot, si on y arrive, alors on aura compris le cerveau humain), SOAR souhaite participer à cette recherche avec son architecture hybride (symbolique et émergente). On remarquera que durant les 40 dernières années[2], plus de 300 architectures cognitives ont été développées, pour seulement moins de 1/3 encore actives.

## 2 Problématique

Comment représenter une architecture cognitive ? [5]

Simuler un processus cognitif humain, c'est vouloir représenter le processus de réflexion d'un humain. SOAR (*State, Operator And Result*) a pour ambition de répondre à cela. C'est une architecture orientée but avec 3 types de mémoire (sémantique, épisodique, procédurale) et plusieurs processus de prises de décisions dont le chunking.

## 3 L'architecture

À partir de perceptions réelles, on peut soit agir directement via le SVS, notamment amélioré en 2012 par le Robot SOAR [4], soit stocker dans la working memory une représentation symbolique du monde. La prise de décision du Soar se base sur la LTM (Long Term Memory), afin de choisir un opérateur qui réalisera l'action correspondante et actualisera l'état courant dans la working memory vers le prochain état. La LTM est composé à la base de la mémoire procédurale, c'est un ensemble de règles de type if then. Lors d'une impasse durant la prise de décision, par exemple entre deux opérateurs, nous subdivisons le problème en générant un sous-espace avec un état qui représente toutes les informations permettant de résoudre cette impasse, alors nous choisissons un opérateur dans ce sous-espace. On appelle le chunking le mécanisme de création de règle à partir des impasses. Le SOAR à eux plusieurs évolutions que nous expliquerons plus tard.

## 4 Code

En SOAR, les fonctions se comportent de manière conditionnelle, selon un schéma "si/alors". La première partie de la fonction définit les états dans lesquels l'environnement doit se trouver pour pouvoir activer la fonction, tandis que la seconde partie décrit les changements à opérer sur l'environnement. Ensuite, le moteur d'inférence de SOAR se charge de créer toutes les règles possibles jusqu'à atteindre l'état but.

## 5 Résultat

Dans les applications de SOAR on retrouve notamment :

- NL SOAR 1994 [3]: Agent en simulation capable de comprendre du langage naturel. Utilisation à des fins militaires Bon résultat et très réactif.
- NTD SOAR 1994 [1]: Agent testé dans le réel, directeur de test pour le lancement de navette spatiale pour la NASA. Utilisation de NL SOAR combiné avec des modules de perception
- SC-SOAR 2012: Utilisation de SOAR pour jouer à Starcraft. Possibilité de battre un ordinateur en difficile.

## 6 Evolution

L'architecture du SOAR à grandement évoluer, à travers le temps, notamment en ajoutant différent type de mémoire LTM. La mémoire Sémantique est une mémoire contenant une ensemble de fait, elle a l'avantage d'être rapide en accès de lecture et d'écriture. La mémoire Épisodique permet de représenter le souvenir, en retenant des épisodes passés. Un épisode est un ensemble d'états et d'actions qui ont été effectués par le système. L'ajout d'apprentissage par renforcement pour créer des règles de pondération d'opérateur à partir des récompenses des états est un autre moyen de remplir la mémoire procédurale.

### 6.1 ROBOT SOAR 2012

Le ROBOT SOAR[4] est un robot qui se déplace dans une simulation en temps réels. Pour parvenir au temps réel, la prise de décision se sert principalement de la mémoire sémantique et au court-circuitage présent dans le SVS. En effet, en prenant de l'information continue, on n'est pas obligé de faire une représentation symbolique du monde, on peut agir directement sur les moteurs. Pour cela, on utilise un arbre de décision qui se base sur la donnée continue pour choisir quel moteur actionner.

### 6.2 LUCIA 2016

Lucia[6] se base sur la recherche cognitive linguistique. La recherche linguistique est une partie de la recherche qui réalise du formalisme sur la langue. LUCIA utilise des règles ECG (*Embodied Construction Grammar*) et quelques règles pré-codées. À partir de cela LUCIA, génère les nouvelles règles qui lui permettent de comprendre la langue naturelle, même plusieurs langues qui sont syntaxiquement différentes, comme l'anglais et l'espagnol. Exemple de règle ECG : *Transitive Command* qui est un sous-cas de l'impératif, elle est constitué d'un verbe et d'un objet. En 2024 il y a beaucoup de recherche sur les LLMs (*Large language model*) qui se basent sur LUCIA.

## 7 Limitations

Au cours de nos recherches, nous avons constaté plusieurs problèmes limitant le développement de SOAR. D'une part, le faible nombre de contributions sur GitHub et le nombre peu élevé de recherches Google à ce sujet, ainsi que le fait que la documentation pas à jour, constitue de réel limite pour son développement. D'autre part, en ce qui concerne la structure même de SOAR, nous avons observé que sa mémoire se remplit très rapidement au fil du temps, ce qui surcharge le système et affecte son temps de réaction. De plus, environ 50 % des règles créées ne sont finalement pas utilisées[7]. Enfin, il est important de noter que la majorité des projets utilisant SOAR ne sont développés qu'en simulation.

## 8 Conclusion

Durant le débat, nous avons discuté de différents sujets, nous allons faire un petit récapitulatif des points importants. Avant l'apprentissage par renforcement, SOAR utilisait la métaconnaissance (connaissance sur les connaissances) pour choisir des règles. Nous nous sommes posé la question de savoir si SOAR pourrait devenir dans le futur une IA forte. Nous nous sommes demandé quel était le but d'avoir une IA forte. Nous pouvons simplement faire plein de modules qui sont spécialisés dans des tâches précises. SOAR a depuis le début un problème sur lequel les chercheurs travaillent, qui est le problème de remplissage de la mémoire. Nous pouvons faire un lien avec le *Frame Problem*. C'est justement une des raisons pour lesquelles SOAR est principalement utilisé en simulation. SOAR est une architecture "Boite à outils" qui n'est pas encore très utilisée à l'heure actuelle, mais c'était aussi le cas des réseaux de neurones et qui sont maintenant très utilisés, depuis que la puissance de calcul a grandement augmenté. qu'en sera-t-il du SOAR à l'avenir ?

## References

- [1] Jill Fain Lehman Greg Nelson and Bonnie E. John. Integrating cognitive capabilities in a real-timetas. [https://escholarship.org/content/qt8pb598rm/qt8pb598rm\\_n.oSplash\\_c39a6328547e123241fa7b18e67d4205.pdf?t=sh6i7s](https://escholarship.org/content/qt8pb598rm/qt8pb598rm_n.oSplash_c39a6328547e123241fa7b18e67d4205.pdf?t=sh6i7s), 1994.
- [2] John K. Tsotsos Iuliia Kotseruba. A review of 40 years in cognitive architecture research core cognitive abilities and practical applications. <https://arxiv.org/pdf/1610.08602>, 2018.

- [3] Julie Van Dyke Jill Fain Lehman and Nancy Green. Reactive natural language processing: Comprehension and generation in the air combat domain. <https://cdn.aaai.org/Symposia/Fall/1995/FS-95-05/FS95-05-011.pdf>, 1994.
- [4] Shiwali Mohan Joseph Z. Xu John E. Laird, Keegan R. Kinkade. Cognitive robotics using the soar cognitive architecture. <https://cdn.aaai.org/ocs/ws/ws0828/5221-22704-1-PB.pdf>, 2012.
- [5] John E. Laird. Introduction to the soar cognitive architecture1. <https://soar.eecs.umich.edu/home/About/>, 2022.
- [6] John E. Laird Peter Lindes. Toward integrating cognitive linguistics and cognitive language processing. *Conference on Cognitive Modeling*, 2016.
- [7] J. Gregory Trafton William G. Kennedy. Long-term symbolic learning in soar and act-r. <https://apps.dtic.mil/sti/pdfs/ADA478972.pdf>, 2007.