



SOAR Architecture **Project**

une architecture cognitive générale



Presented by
ATRAOUI YASMINE
BOUHOUC ROUA
RAVELOSON MAHERINIRINA

SOAR: modéliser l'intelligence humaine et artificielle.

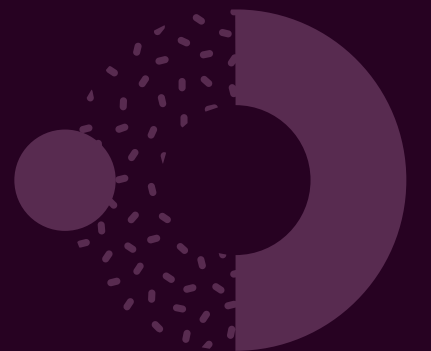


Capacités cognitives fondamentales: perception, mécanismes d'attention, sélection d'actions, mémoire, apprentissage, raisonnement, s'adapter , développer des idées nouvelles

Achitectures cognitives => créer des programmes pour avoir les capacités cognitives ; modéliser l'esprit humain

Système Intelligent: Structure technique fixe .

Architecture Cognitive: inspiré du fonctionnement de l'esprit humain





SOAR

1. Principes fondamentaux



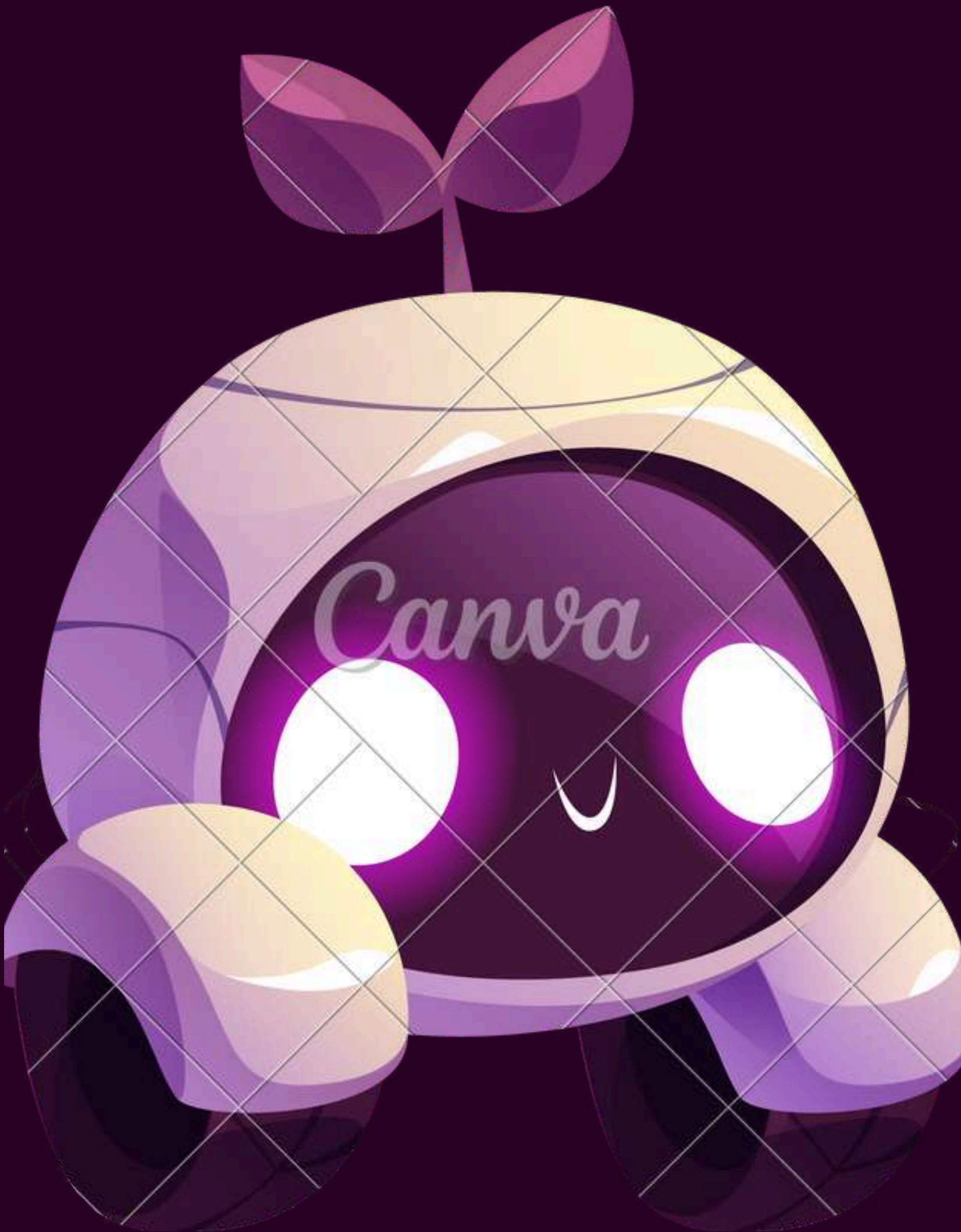
Hypothèse de l'Espace de Problème

Tout comportement orienté vers un but = recherche d'états et d'opérateurs

Un problème = états initiaux + états buts + contraintes de chemin

Un état = ensemble de caractéristiques + valeurs (interne ou externe)

La résolution = une recherche progressive (état par état)



Rationalité limitée

Inspirée de la rationalité humaine : décisions prises avec connaissances et ressources limitées

Soar ne cherche pas la solution optimale, mais une solution efficace

Le système gère les situations d'ignorance ou de conflit de connaissance



Impasses et sous-buts

- Quand Soar ne peut pas décider → impasse
- Il crée alors automatiquement un sous-but pour résoudre cette impasse
- Ce mécanisme évite la recherche combinatoire coûteuse
- Alternance :
 - Traitement rapide (règles de production)
 - Traitement lent (recherche délibérée)





Systeme symbolique

La cognition =
manipulation de
symboles

Les symboles
représentent
concepts, objets,
actions, etc.

Reliés entre eux →
structures complexes
(réseaux sémantiques,
ontologies...)

En mémoire de travail
:

Chaque objet =
identifiant + triplets
attribut-valeur

Ex : Café → {état:
moulu, quantité: 2
cuillères}



Règles de production



01

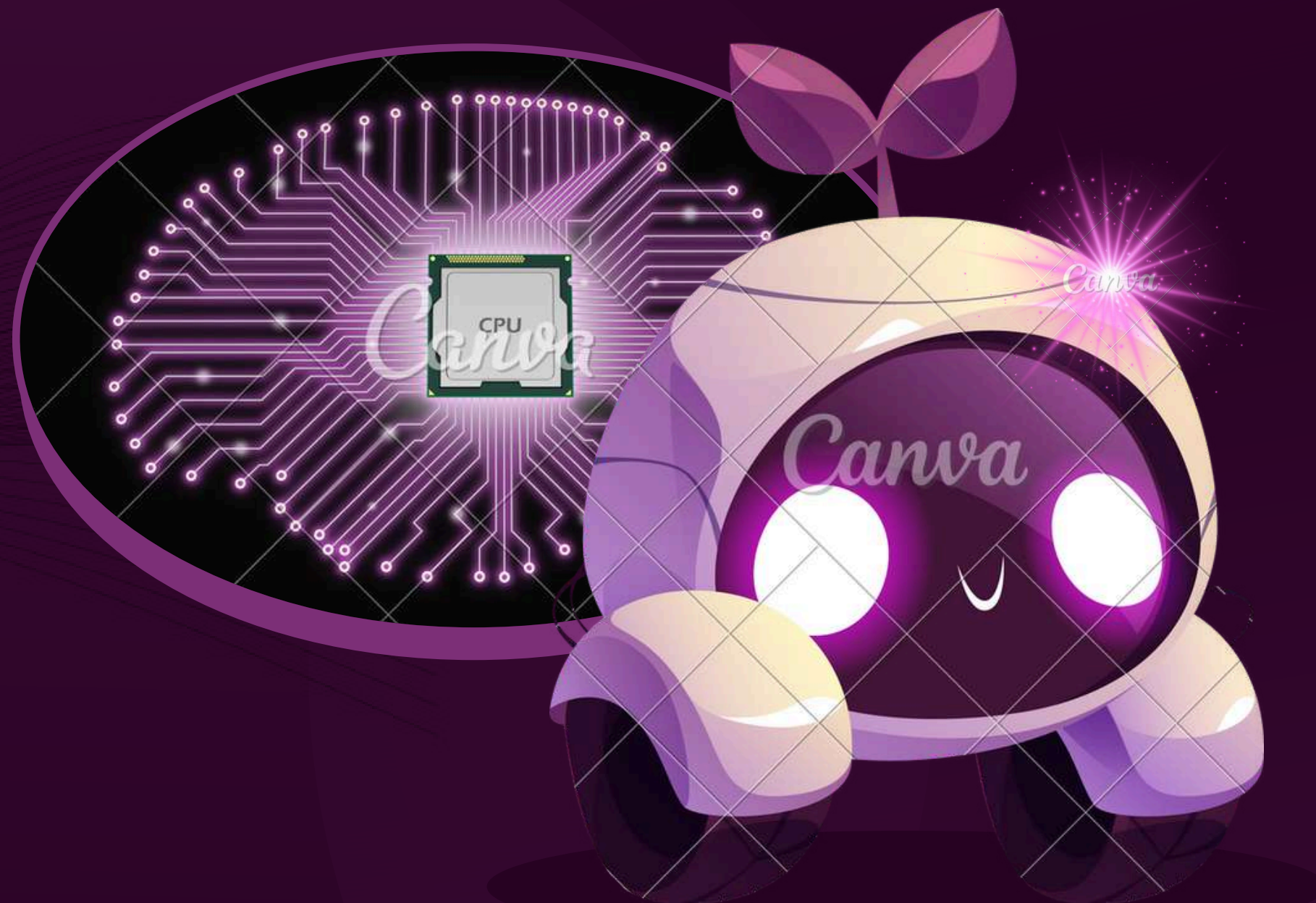
Règles = mécanisme
de décision de Soar

02

Forme : IF (c) →
THEN (a)

03

Cycle continu :
reconnaissance + action

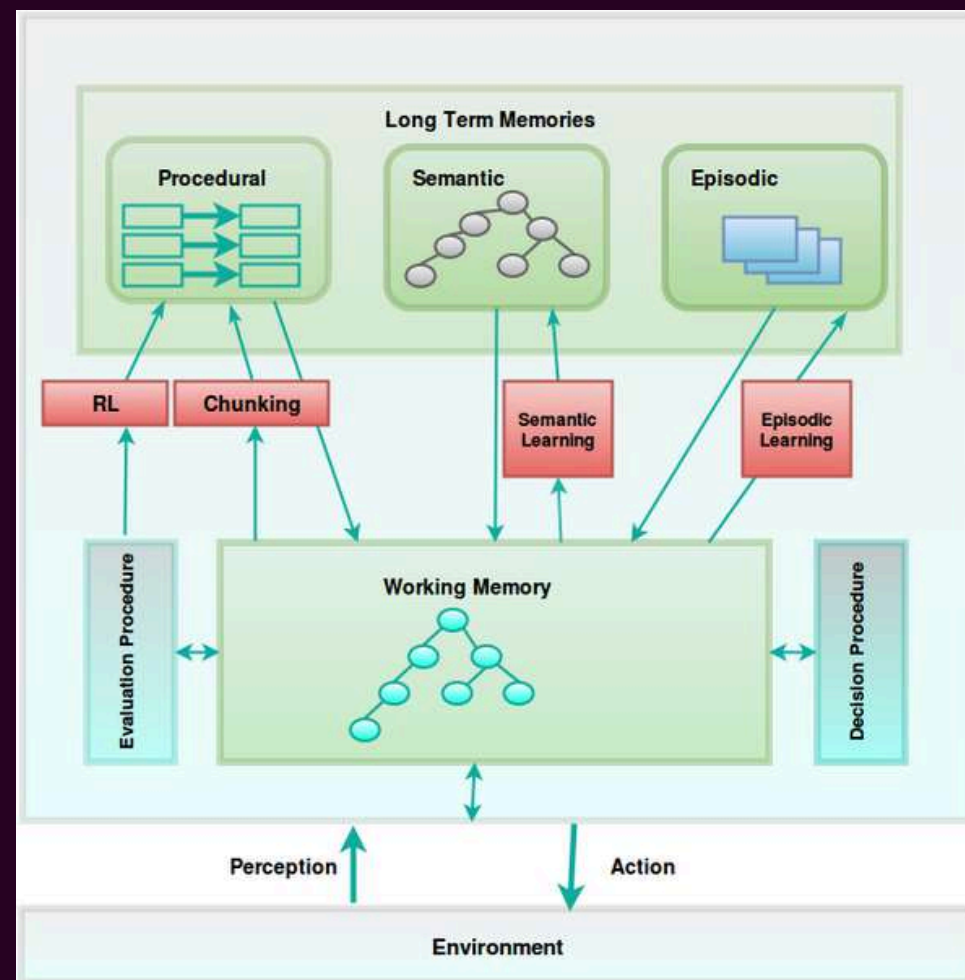




SOAR

1. Principes fondamentaux
2. Structure et fonctionnement de l'architecture

3 composants principales



1 Mémoire de travail

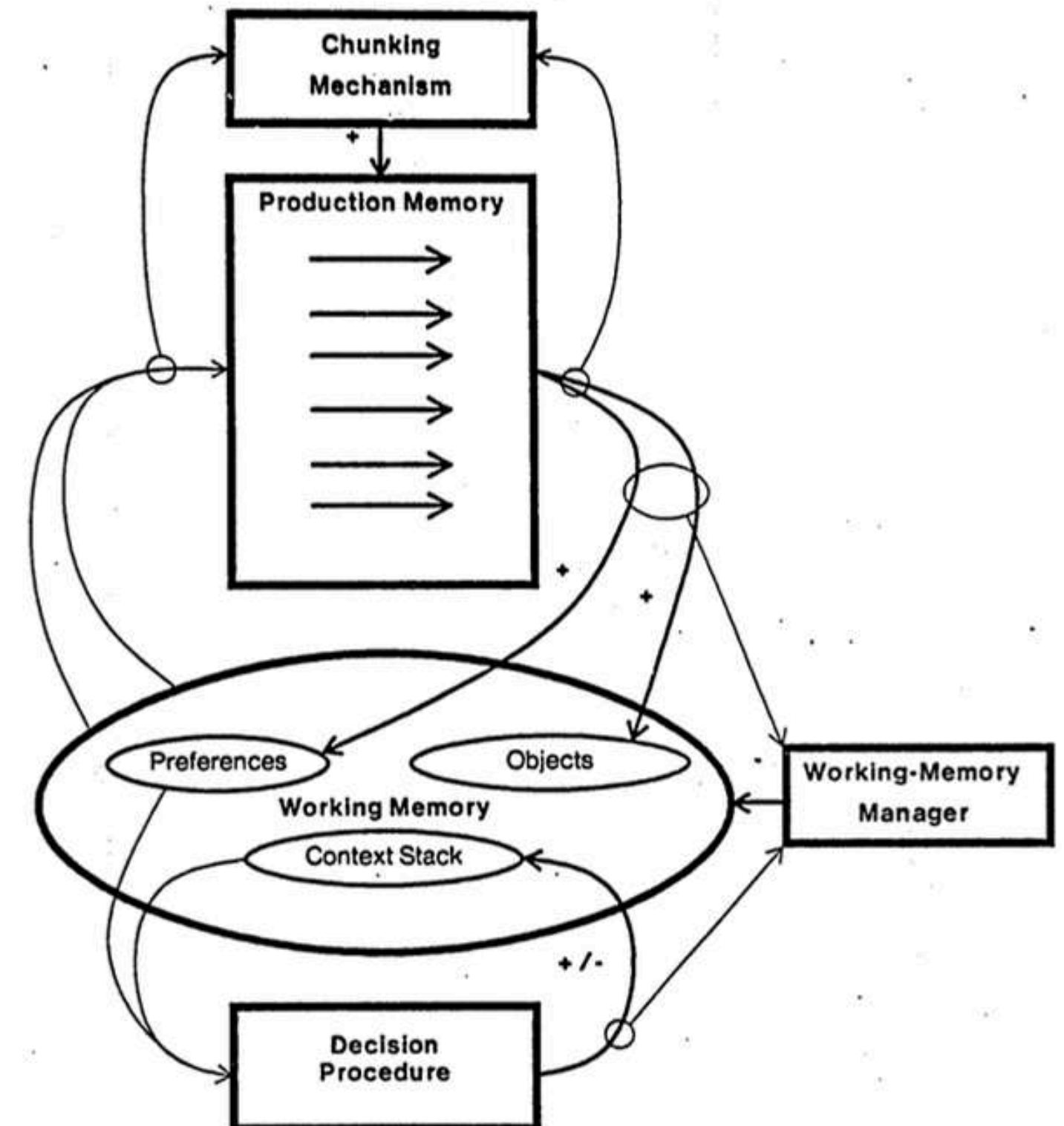
2 Mémoire à long terme
Mémoire procédurale
Mémoire déclarative ou sémantique
Mémoire épisodique

3 Système de production

Fonctionnement

Cycle de traitement cognitif

- 50 ms par cycle
- Grande partie du travail en parallèle
- Goulot d'entrangementement sériel
- Acte délibératif

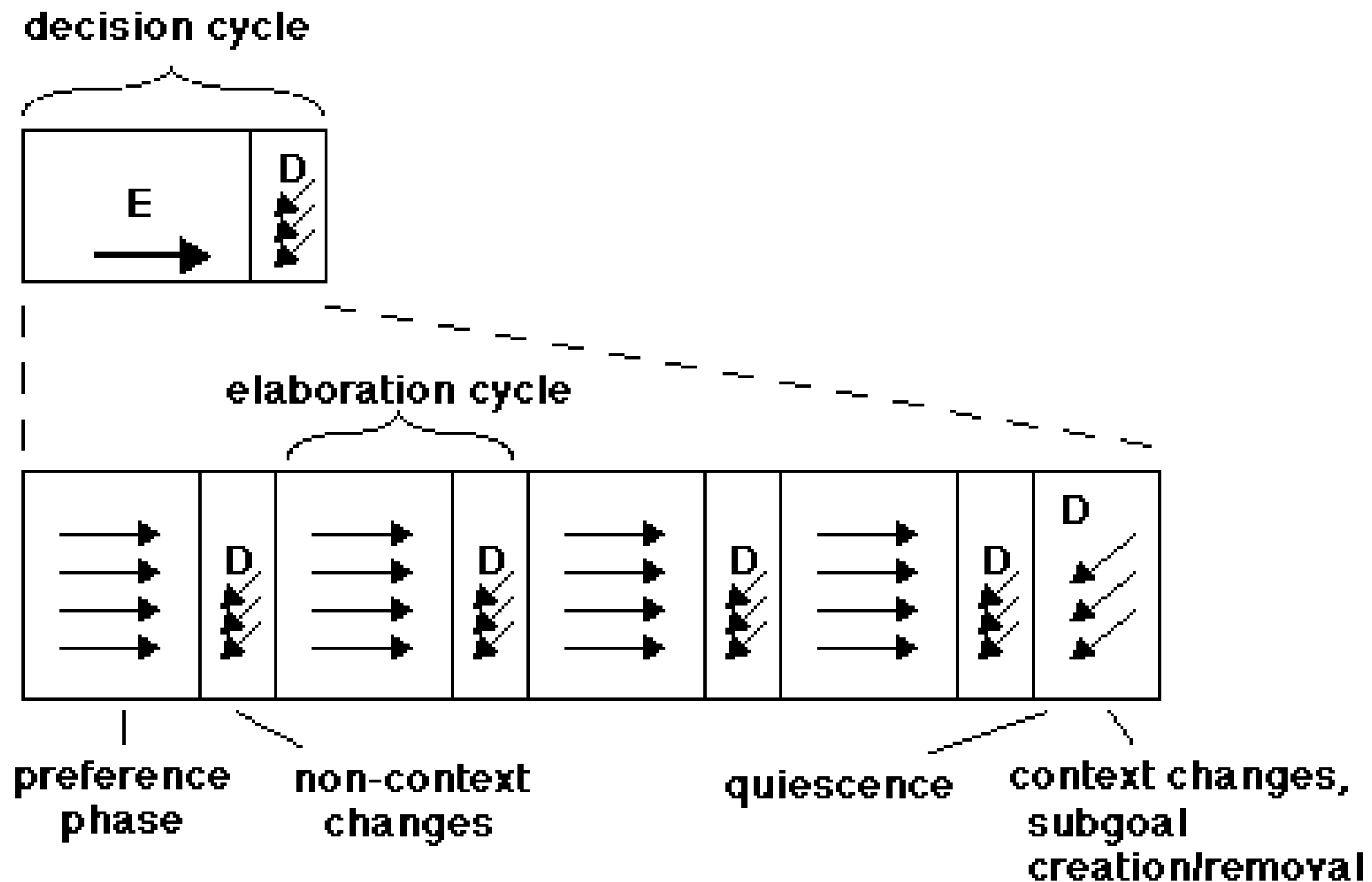


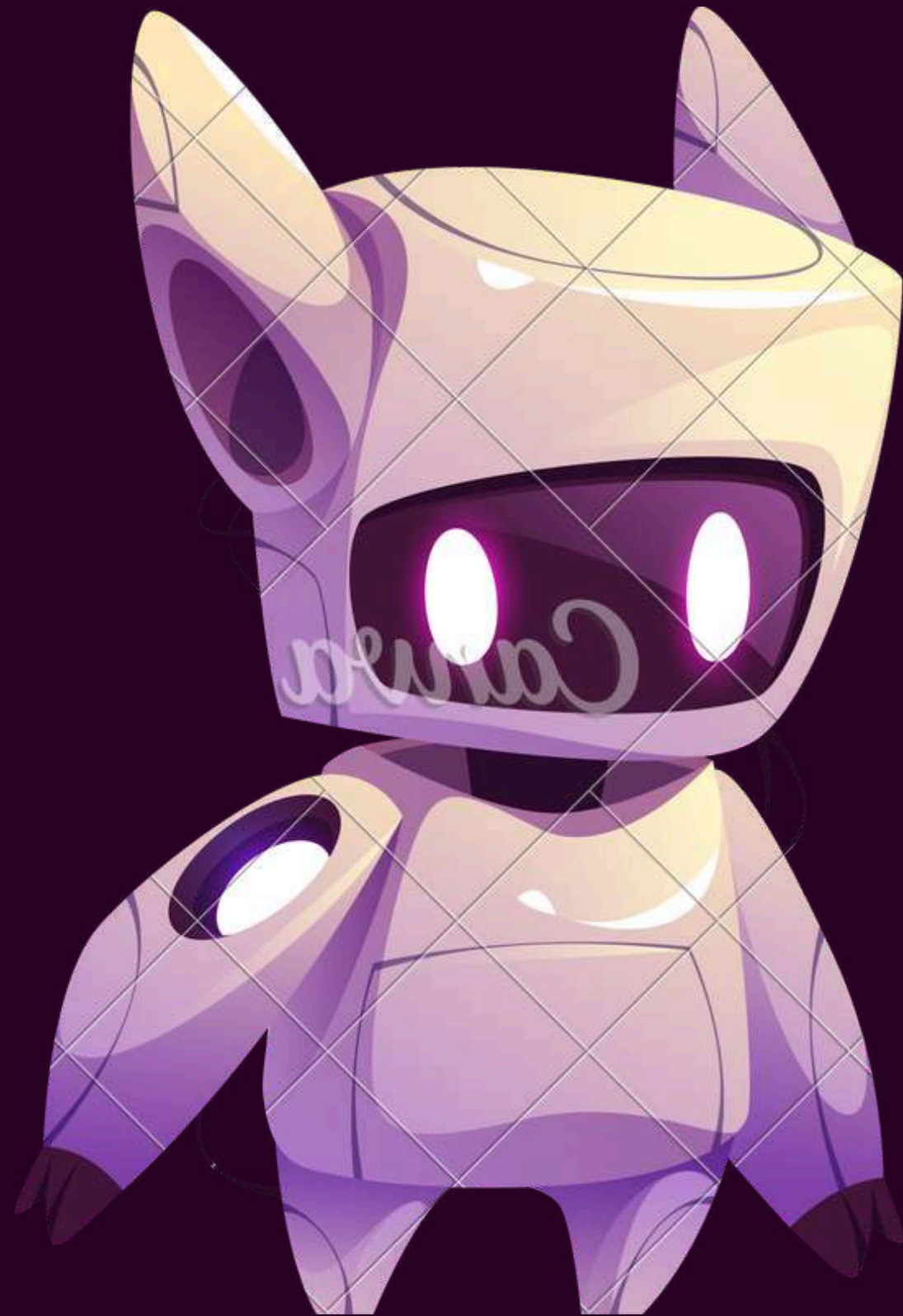
Cycle de traitement cognitif



2 phases:

- Phase d'élaboration
- Phase de décision





Mécanismes de Contrôle et d'Apprentissage

2 mécanisme clés:

- Impasse et Sous-Objectifs
- Apprentissage par Chunking



SOAR

1. Principes fondamentaux
2. Structure de l'architecture
3. Apprentissage par *Chunking*

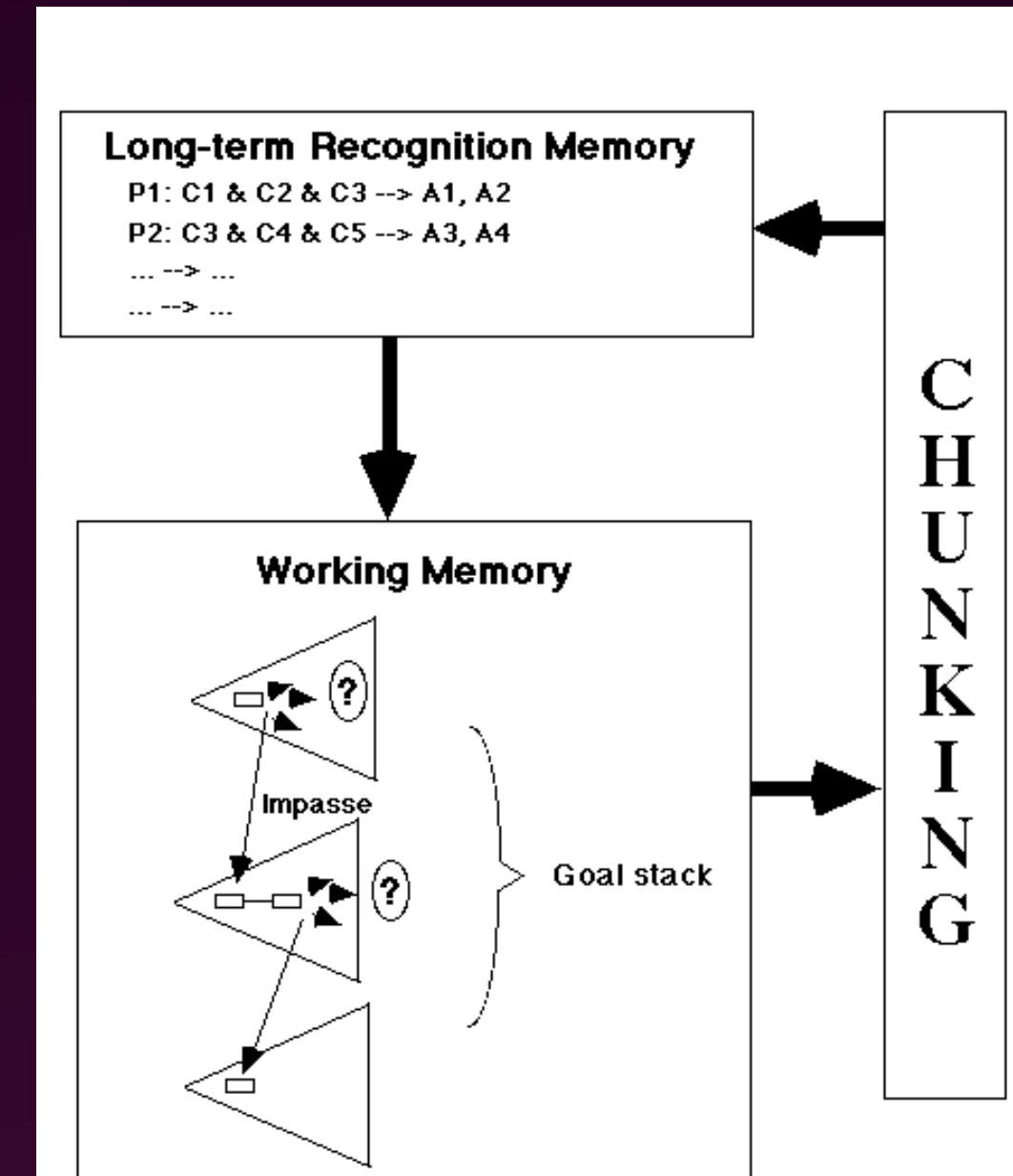
PRINCIPE

CHUNKING



Le chunking = morcellement /
agrégation d'informations

Étroitement lié à la modélisation de
la mémoire humaine



De la mémoire humaine au chunking

- **Introduit par George Miller (1956)**
- **L'humain apprend en formant et stockant des chunks :**
 - → ensembles structurés de connaissances déjà acquises
 - Un chunk = un symbole représentant un motif d'autres symboles
- **Permet d'expliquer :**
 - pourquoi la capacité de la mémoire de travail reste constante
 - comment on regroupe plusieurs éléments en une unité significative
 - donc → moins de charge mentale, plus d'efficacité



La Loi de Puissance de la Pratique

- Plus on répète une tâche → plus on devient rapide et efficace
- Les gains sont importants au début, puis décroissent avec le temps
- T = temps nécessaire
- B = temps initial
- N = nombre d'essais
- α = taux d'apprentissage
- En clair : la pratique améliore la performance, mais avec rendement décroissant

$$T = B \times N^{-\alpha}$$



Le Chunking dans Soar

Proposé par Allen Newell & Paul Rosenbloom (1980s)
Le chunking = mécanisme d'apprentissage
fondamental de Soar

Processus automatique : transforme les expériences en
règles permanentes (productions)

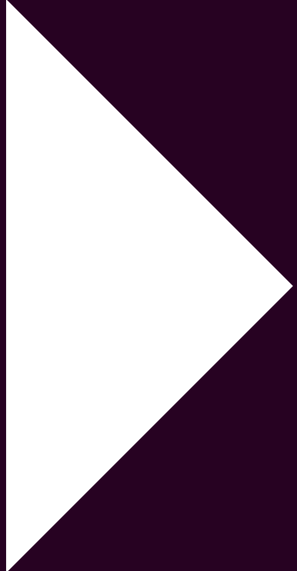
Se déclenche lorsqu'une impasse (manque de
connaissance) survient → création d'un sous-but



Fonctionnement du Chunking

- Résolution de sous-buts → expérience de résolution de problème
- À la fin : création d'un chunk (nouvelle règle) qui :
 - résume le processus complexe du sous-but
 - est ajouté à la mémoire procédurale
 - se réutilise automatiquement à l'avenir
- Résultat : moins de recherche, plus de reconnaissance rapide

Généralisation implicite dans Soar (Chunking)

- 
- Concept clé :
 - Soar crée un chunk (règle) uniquement avec les aspects pertinents de la situation.
 - Tout ce qui n'a pas été utilisé pour atteindre le but est ignoré automatiquement.
 - Permet de réutiliser la règle dans de nouvelles situations similaires.

Exemple (Eight Puzzle) :

Les positions exactes des tuiles ne sont pas retenues si inutiles.

Seules les relations importantes (ex. T1 à gauche de T2) sont conservées.

Le chunk peut donc s'appliquer à d'autres configurations jamais vues.

Effets sur la performance

- Enregistrement de l'expérience → création d'un chunk
- Accélération de la performance → activation immédiate des chunks appris
- Modélisation de la Loi de Puissance :
 - moins d'étapes délibérées
 - passage d'un comportement lent → automatique et fluide



Chunking et EBL (Explanation-Based Learning)

Le chunking = une forme d'apprentissage par généralisation

L'EBL :

reçoit un exemple, un but, et une théorie du domaine
explique pourquoi l'exemple fonctionne
généralise cette explication en une règle réutilisable

Le système n'apprend pas du nouveau contenu, mais
améliore son efficacité

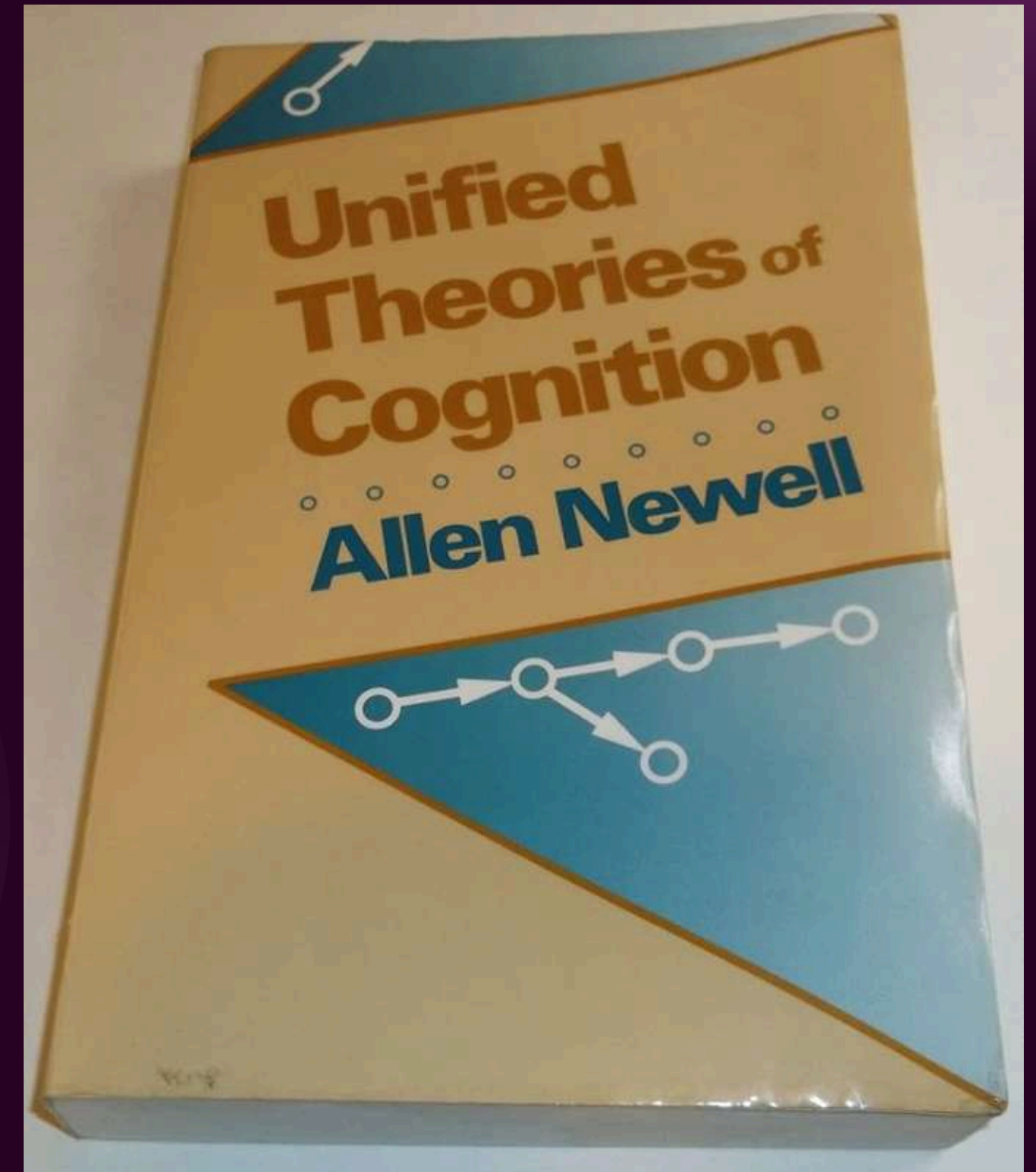


Allen Newell et la Théorie Unifiée de la Cognition

Allen Newell, à l'origine du projet SOAR, a formulé l'idée d'une théorie unifiée de la cognition.

Selon lui, toutes les capacités cognitives humaines dont langage, mémoire, raisonnement, résolution de problèmes qui proviennent d'un même système : l'esprit humain.

Si toutes ces fonctions partagent la même base, leurs mécanismes internes doivent être compatibles et unifiés.



Naissance de SOAR

SOAR est né de cette hypothèse :

trouver un ensemble fixe de mécanismes et de structures fondamentales capables d'expliquer tous les comportements cognitifs, de manière calculable par ordinateur.

Newell résume cette idée par la formule :

Comportement = Architecture + Contenu

L

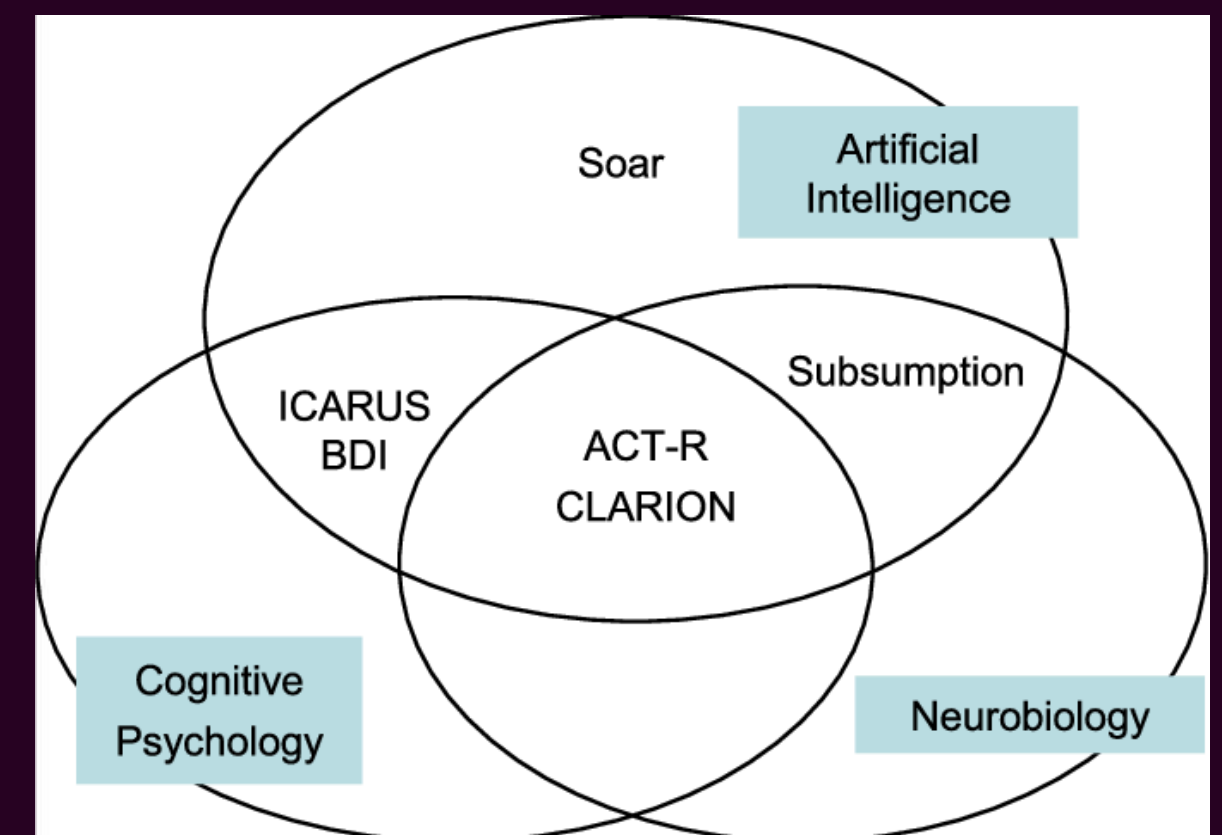


Domaines d'Application

SOAR a été utilisé dans de nombreux domaines :

- Systèmes experts : R1-Soar (configuration d'ordinateurs VAX).
- Médecine : Neomycin-Soar (diagnostic clinique).
- Robotique : MOUTBOT (simulation militaire et contrôle d'agents).
- Langage naturel : NL-Soar (compréhension et génération du langage).
- Psychologie cognitive : simulation du raisonnement humain (syllogismes, séquences).

Soar	Decision cycle selects appropriate actions, bringing system closer to goal
ACT-R	Planing by creating subgoals
ICARUS	Planning through instantiation of skills
BDI	Predefined plans stored in the plan library
Subsumption	Implicit planning by task decomposition
CLARION	Planning by beam search in the Q -value space



Classification des Architectures Cognitives

Les architectures cognitives se distinguent par la nature de leurs représentations :

- Symbolique : raisonnement logique sur des symboles discrets (cognitivism classique).
- Subsymbolique : apprentissage distribué (réseaux neuronaux, poids, activations).
- Hybride : combinaison des deux approches.



Classification des Architectures Cognitives

À l'origine, SOAR était symbolique pur, mais ses versions récentes comme SOAR9 intègrent des mécanismes subsymboliques pour le rendre plus flexible et réaliste.

Architecture	Representation	Working memory	Long term memory
Soar	Symbolic	Contains perceptual input, states, and production rules relevant to current situation	Contains procedural, declarative (semantics), and episodic memory
ACT-R	Symbolic	Contains goal, perception, relevant knowledge, and motor action in the various buffers	Contains declarative knowledge in declarative module and procedural knowledge in production system
ICARUS	Symbolic	Consists of perceptual buffer, belief memory, and short term skill memory	Procedural knowledge stored in long term skill memory, declarative knowledge in long term conceptual memory
BDI	Symbolic	Belief as working memory	Plans as long term memory
Subsumption	No explicit representation	No explicit working memory	Heuristics within each layer
CLARION	Symbolic + subsymbolic	As temporary information storage	Procedural at bottom level Declarative at top level

Conclusion

Est-ce qu'on peut simuler vraiment la cognition humaine avec SOAR?

La rationalité limitée est au cœur de Soar :
Les décisions sont prises avec peu de ressources et des infos incomplètes.
L'IA ne cherche pas la perfection, mais agit de manière raisonnable et efficace

**Les erreurs humaines doivent-elles être reproduites dans une IA pour la rendre plus crédible,
ou au contraire éliminées pour améliorer la prise de décision?**

Soar suppose qu'un seul ensemble de principes (l'Architecture) peut expliquer tous les comportements cognitifs. Cette quête d'une Théorie Unifiée de la Cognition est-elle réalisable, ou impose-t-elle des compromis qui limitent la modélisation précise des phénomènes cognitifs ?

Est-ce que le Chunking suffit pour simuler la cognition humaine?

Le raisonnement séquentiel permet-il de prioriser correctement entre plusieurs objectifs conflictuels, sans créer du biais?

Et l'apprentissage par expérience unique ou chunking permet-il de généraliser des principes moraux à de nouveaux contextes ?

