

# IA pour le génie logiciel- HAI916I - M2 GL

Responsable : Marianne Huchard

Comprendre comment l'Intelligence Artificielle et le Génie Logiciel s'enrichissent mutuellement

- IA pour le GL
- GL pour l'IA

## En pratique

- Connaître les principes qui sont ou seront prochainement à l'œuvre dans les outils de suivi de projets et de développement, par exemple sous forme d'assistants
- Entrevoir les problématiques des workflows d'analyse de données en termes de génie logiciel (conception, variabilité, maintenance, tests, vérification)

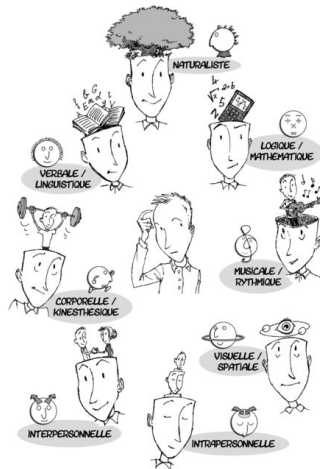
## Modalités de contrôle des connaissances

- Session 1 : 4 projets (25 %) - Ces projets seront sélectionnés parmi les TP réalisés lors des différents cours. Précisions à venir.
- Session 2 : 1 projet (100%) - Précisions à venir.

# Programme prévisionnel

- S37 - Marianne Huchard. (1) Introduction IA et GL : qu'est-ce que l'IA, qu'est-ce que le GL, grandes directions en IA symbolique et sous-symbolique, Exemples de situations en génie logiciel. (2) Rappels sur l'Analyse formelle de concepts.
- S38 - Marianne Huchard. Analyse formelle de concepts et analyse de la variabilité. Construction de Feature Models à partir de structures conceptuelles
- S39 - Nadjib Lazaar. Programmation par contraintes pour la génération de tests (1ere partie)
- S40 - Chouki Tibermacine, Apprentissage automatique par analyse du code
- S41 - Nadjib Lazaar. Programmation par contraintes pour la génération de tests (2e partie)
- S42 - Nadjib Lazaar. Fouille de données pour la localisation de fautes (1ere partie)
- S43 - Nadjib Lazaar. Fouille de données pour la localisation de fautes (2e partie)
- S44 – *Vacances de la faculté des sciences*
- S45 - Quentin Perez, Christelle Urtado, Sylvain Vauttier. Analyse de la contribution architecturale dans les projets OS. Données GitHub, calcul de métrique, analyse statistique des résultats, interprétation (choix des visualisations adaptées)
- S46 - Quentin Perez, Christelle Urtado, Sylvain Vauttier. Classification de tickets dans les Issue Tracking Systems : distinction bug / non bug. Données issues d'IST (Jira), techniques de NLP (TF-IDF), classification supervisée, réseaux de neurones, évaluation (signification des écarts statistiques avec l'état de l'art)
- S47 - Abdelhak-Djamel Seriai. Clustering, q-learning pour la restructuration et la modernisation
- S48 - Marianne Huchard. Analyse relationnelle de concepts et analyse de la variabilité logicielle par séparation des préoccupations
- S49 - Elyes Cherfa. Utilisation des métaheuristiques en GL

# Intelligence multiples, Howard Gardner, 1983, 1996



Source : <https://www.mieux-apprendre.com/lapproche-du-mieux-apprendre/les-outils/les-intelligences-multiples/>

# Intelligence Artificielle : une délicate définition

## Événement fondateur de la discipline

Séminaire d'août 1956, Dartmouth College, John McCarthy, Marvin Minsky, Herbert A. Simon, Allen Newell, Claude Shannon, etc.

## Définitions

- Alan Turing, 1950 (Computing Machinery and Intelligence) : Can machines think ? Turing test (Un humain cherche à distinguer, dans des réponses à des questions qu'il pose, si ces réponses sont données par une machine ou un être humain)
- John McCarthy, 1956 : "It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable".
- Marvin Minsky, 1968 : "the science of making machines do things that would require intelligence if done by men."

## Tâches visées

- Raisonnement, planification, apprentissage, traitement du langage naturel, perception, analyse de sentiments, d'émotions, manipuler des objets.

# Intelligence Artificielle : Quelques grands courants

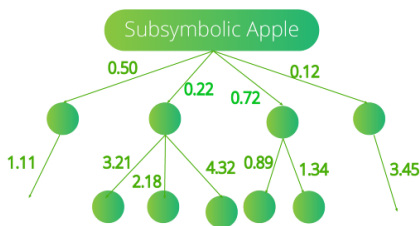
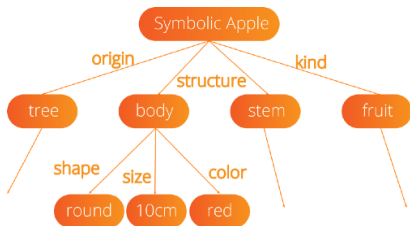
## ● Approches Symboliques

- Système symbolique : symboles (formes physiques), expressions, processus pour les manipuler (A. Newell & P. Simon)
  - Basées principalement sur la logique et l'explicitation des connaissances
  - GOFAI ("Good Old-Fashioned Artificial Intelligence"), John Haugeland, 1985
  - Systèmes à base de connaissances, représentation des connaissances, ontologies, langages du web sémantique, ressources lexicales, dictionnaires
  - Systèmes de raisonnement à base de règles, systèmes experts
  - Systèmes multi-agents basés sur la logique
  - Planification automatique, programmation par contraintes
  - Classification et apprentissage symboliques, règles d'association, analyse formelle de concepts
- + Traitement de données complexes, applicable sur des données de taille modeste, approches supervisées ou non supervisées, résultats explicables, contrôlables, peuvent être débogués
- Explicitation des connaissances qui peut être coûteuse, complexité pratique de certains algorithmes, peu robuste au bruit

# Intelligence Artificielle : Quelques grands courants

- Approches Sub-symboliques (statistiques, connexionnistes)
  - Arbres de décision, random forests
  - Apprentissage Bayésien
  - Deep learning
  - Connectionisme, Réseaux de neurones
  - Latent Dirichlet Allocation, Support Vector Machines
- + Plus robuste au bruit, meilleures performances, peuvent traiter des données massives, plus adapté pour les problèmes de perception et la connexion avec les neurosciences
- Nécessitent des données en quantité importante (approches supervisées), simplification des données nécessaire, résultats peu explicables, résultats que l'on peut biaiser par des données orientées et truquées ou des paramétrages particuliers
- $\Rightarrow$  En développement actuellement : approches Hybrides !

# La synthèse de la pomme (Auteur : Orhan G. Yalcin)



<https://towardsdatascience.com/symbolic-vs-subsymbolic-ai-paradigms-for-ai-explainability-6e3982c6948a>

# Intelligence Artificielle : Aperçu de la classification ACM

- ACM Computing Classification System
  - Machine learning
    - Learning paradigms
    - Learning settings
    - Machine learning approaches
    - Machine learning algorithms
    - Cross-validation
  - Artificial intelligence
    - Natural language processing
    - Knowledge representation and reasoning
    - Planning and scheduling
    - Search methodologies
    - Control methods
    - Philosophical/theoretical foundations of artificial intelligence
    - Distributed artificial intelligence
    - Computer vision

<https://dl.acm.org/ccs>



# Génie Logiciel

*Le génie logiciel ("software engineering" en anglais) repose sur l'application systématique des connaissances, méthodes, compétences scientifiques et technologiques nécessaires à la modélisation, l'implémentation, les tests, la maintenance et la documentation de ces logiciels qui structurent de plus en plus notre société.*

<https://gl.frama.io/manifeste/>

# Génie logiciel : le SWEBOK (IEEE Curriculum for SE)

SWEBOK : The Guide to the Software Engineering Body of Knowledge describes generally accepted knowledge about software engineering

- Software requirements
- Software configuration management
- Software design
- Software engineering management
- Software construction
- Software engineering process
- Software testing
- Software engineering tools and methods
- Software maintenance
- Software quality

<https://www.computer.org/education/bodies-of-knowledge/software-engineering>

# Génie logiciel : Aperçu de la classification ACM

- ACM Computing Classification System
  - Software and its engineering
    - Software organization and properties  
Contextual software domains Software system structures Software functional properties  
Extra-functional properties
    - Software notations and tools  
General programming languages Formal language definitions Compilers Context specific  
languages System description languages Development frameworks and environments Software  
configuration management and version control systems Software libraries and repositories  
Software maintenance tools
    - Software creation and management  
Designing software Software development process management Software development  
techniques Software verification and validation Software post-development issues  
Collaboration in software development Search-based software engineering

<https://dl.acm.org/ccs>

# Intelligence artificielle et génie logiciel

## Une vieille histoire, quelques exemples

- Assistant intelligent d'un programmeur pour écrire, déboguer, corriger, refactorer du code
- Systèmes multi-agents et systèmes à composants : questions identiques sur l'organisation en petites entités autonomes et communicantes, résolution distribuée de problèmes, l'auto-adaptation ou encore la fiabilité et la tolérance aux fautes
- Définition de modèles et d'architectures de logiciels fondés sur des techniques de représentation des connaissances, telles que les graphes de connaissances ou les ontologies
- Interopérabilité entre services Web ou l'automatisation des transformations de modèles ou de méta-modèles appuyées sur des techniques d'alignement de schémas ou d'ontologies
- Génération automatique de compositions de services par des techniques de planification
- Traitement de la langue naturelle appliquée à l'ingénierie des exigences, à l'analyse des identificateurs, à l'analyse ou à la génération de documentations

# Intelligence Artificielle pour le génie logiciel

## Maintenance et évolution

- Comment exploiter les grandes collections de données à disposition pour déduire/prédire des migrations, corrections, etc. ?
- Comment nettoyer les données ?
- Comment faire intervenir le choix du développeur dans une approche initialement pensée comme étant complètement automatique ?
- Quels algorithmes d'IA pour quelles classes de problèmes de maintenance logicielle ?

## Sûreté de fonctionnement

- Comment les techniques d'IA et en particulier celles autour de la prédiction peuvent-elles contribuer à la sûreté de fonctionnement ?
- Comment exploiter les bibliothèques de preuves formelles afin de construire de nouvelles preuves de programmes ?

Crédits à David Delahaye

# Génie logiciel pour l'Intelligence Artificielle

## Développement de systèmes intégrant de l'IA

- Comment prendre en compte dans le développement logiciel, l'utilisation de composants appris (par opposition à des composants spécifiés) ?
- Comment prendre en compte l'utilisateur dans la boucle ?
- Comment gérer l'explication des algorithmes d'IA quand elle existe ?
- Comment s'assurer de la pertinence des entrées du composant ?
- Comment gérer un modèle qui apprend à la volée ?

## Validation et vérification de ces systèmes

- Comment vérifier de tels systèmes ? Quelles approches (certifiées/certifiantes) ? Quels certificats ?
- Comment valider un système dont l'une des caractéristiques est d'évoluer naturellement ?
- Comment garantir, au cours de sa vie, qu'il rende le service attendu par l'utilisateur, avec une qualité au moins identique ?

Crédits à David Delahaye

## Outils ou ressources d'IA

Cette liste n'est pas exhaustive, elle sera complétée au fur et à mesure des cours. Elle vous permet de commencer à explorer ce domaine. L'objectif n'est pas que vous utilisiez tous ces outils mais que vous connaissiez des ressources pour vos besoins présents ou futurs.

### Plate-formes d'IA : apprentissage ou analyse de données

- Weka <https://www.cs.waikato.ac.nz/ml/weka/>
- Scikit-learn <https://scikit-learn.org/stable/>
- Orange data mining <https://orangedatamining.com/>
- PyTorch <https://pytorch.org/>

# Outils ou ressources d'IA (suite)

## Quelques outils d'IA spécialisés dans les domaines de ce module

- Ontologies
  - Protégé <https://protege.stanford.edu/>
  - CoGUI <http://www.lirmm.fr/cogui/>
- Analyse formelle de concepts
  - Page générale de ressources <https://upriss.github.io/fca/fca.html>
  - Utilisés dans ce cours <https://www.lirmm.fr/fca4j/>;  
<http://dataqual.engees.unistra.fr/logiciels/rcaExplore>;  
<http://www.lirmm.fr/cogui/>
- Programmation par contraintes
  - Choco-solver <https://choco-solver.org/>
  - IBEX <http://www.ibex-lib.org/>
- Méta-heuristiques
  - jMetal 5 <https://jmetal.github.io/jMetal/>; <https://github.com/jMetal>



## Travail à réaliser par groupe de 1 à 2 étudiants

- Vous recherchez un article récent dans l'une des grandes conférences du domaine : ESEC/FSE, ASE, ICSE, MODELS portant sur l'application de l'IA au GL. Cet article doit mentionner : un outil d'IA connu ou réalisé par les auteurs et disponible en ligne ; un dataset disponible en ligne. Des articles vous sont donnés sur Moodle, issus de la conférence ESEC/FSE mais vous pouvez en chercher d'autres. Tous les pdfs ne sont pas forcément accessibles en ligne.
- Vous écrivez un ou deux paragraphes décrivant cette approche, et mettez les liens vers les ressources logicielles et de données identifiées.
- Dépôt sous forme d'un unique fichier pdf sur Moodle avant le 19 septembre au soir.
- Moodle : <https://moodle.umontpellier.fr/course/view.php?id=22617>