# Guide pratique sur l'intelligence artificielle et la vie artificielle sous GNU/linux

## *Version française du* GNU/Linux AI and Alife HOWTO

## John Eikenberry

<jae@zhar.net>

Pierre-Yves Aimon

Laurent Fallet

Laurent Morretton

Yvon Benoist

Version : 2.3.fr.1.0

| Historique des versions | | |
|---|---|---|
| Version 2.3.fr.1.0 | 2006-10-01 | LF |
| Mise à jour de la traduction française | | |
| Version 1.9.fr.1.0 | 2004-01-15 | LM, LF, PYM |
| Première traduction française | | |
| Version originale | 2006-08-21 | JE |
| Version 2.3. Consulter la section "Quoi de neuf" pour les révisions antérieures. | | |

**Résumé**

Ce guide contient principalement des informations et des liens relatifs à diverses applications, bibliothèques, etc. sur l'IA qui fonctionnent sur la plate-forme GNU/Linux. Tout ceci est libre dans le cadre d'un usage personnel (ou davantage). La page la plus à jour de ce document se trouve à l'adresse http://zhar.net/gnu-linux/howto/

---

**Table des matières**

# Introduction

## But

Le système d’exploitation GNU/Linux a évolué depuis ses origines de système pirate jusqu’à devenir un système UNIX complet, capable de rivaliser avec n’importe quel UNIX commercial. Il est maintenant une base peu onéreuse pour la construction de bonnes stations de travail. Il s’est libéré de ses problèmes de compatibilité matérielle en ayant été porté sur des stations DEC Alpha, Sparcs, Power PC et bien d’autres. Son potentiel de croissance rapide ainsi que son support réseau le rend idéal pour le partage du temps de travail entre plusieurs ordinateurs. En tant que station de travail, il permet toutes sortes de recherches et de développements, y compris les travaux sur l’intelligence artificielle et la vie artificielle.

Le but de ce HOW-TO est de fournir une base permettant de trouver différents logiciels, bibliothèques de code, et tout ce qui peut aider à démarrer (et trouver les ressources) dans le monde de l’intelligence artificielle, de la vie artificielle, etc. Tout cela dans l’optique GNU/Linux.

## Quoi de neuf ?

- v2.4 (en développement) - Nouvelles entrées : Eprover, Player et Robodeb. J’ai divisé la section Agents en deux sous-sections, à savoir Agents Logiciels et Simulateurs et Robotique. Je crains d’être allé un peu vite quand à la répartition, mais c’est une amélioration. MIA trouvé! Cellular, le système d’automate programmable cellulaire.

- v2.3 - Nouvelles entrées : Yamp, pygene, Push, ANNEvolve, dgpf, Golly, IBAL, 3APL, OSCAR, and RobocodeNG. Mise à jour des entrées Yale, Joone, Drone, Biome, ECLiPSe, Xtoys, GECO, Creatures Docking Station et d’autres. J’ai également renommé la section MIA en Manquants et Inactifs qui regroupe maintenant les liens morts ou erronés pour lesquels je ne trouve de remplacement et les projets qui ne sont plus actifs.

- v2.2 - Quelques liens cassés ont été réparés et quelques nouvelles entrées ont du être déplacées dans la section MIA (manquante). J’ai également supprimé une entrée dupliquée. Nouvelles entrées : MASON, spyse, AntWars, OpenSteer, Pyro, Robocode, Trend and Open BEAGLE.

- v2.1 - Nouvelles entrées : NLTK, NEURObjects, KANREN, Neural Networks at your Fingertips, SimWorld, SimAgent, Fuzzy sets for Ada, maxent, Evo, breve and AJA

- v2.0 - Un vérificateur de liens m'a permis de supprimer les liens erronés ou de les remplacer. Voir la liste des entrées supprimées dans la section MIA (me contacter si vous connaissez le nouveau lien). Nouvelles entrées : Yale, DIET Agents, JASA, Jason, Noble Ape, Maude, ECLiPSe, lush, and pygp
- v1.9 - Ajout d'une nouvelle entrée (Bond) et réparation du lien donné plus bas, qui pointe vers la liste dynamique.
- v1.8 - Nettoyage de liens erronés, recherche de nouveaux liens lorsque cela était possible et suppression de ceux qui semblent avoir disparus. Quelques nouvelles entrées ajoutées également : Torch, Aleph, AI Kernel, OpenCyc, HTK, FFLL, JCK, Joone, scnANNlib, GAUL, Cougaar, and RoboTournament
- v1.7 - 9 autres nouvelles entrées ajoutées, une série de liens réparés, quelques éléments enlevés qui ont disparus du net. Nouvelles entrées : SPASS, CNNs, JCASim, Genetic, CAGE, AgentFarms, MATREM, OAA, et UTCS Neural Nets Research Group Software
- v1.6 - Ajout de 9 nouvelles entrées, deux liens réparés et un élément redondant enlevé.
- v1.5 - 26 nouvelles entrées, ainsi que deux ou trois réparations de liens.
- v1.4 - 10 mises à jour effectuées et réparation de liens en rapport avec LISP.
- v1.3 - Pour rattraper le retard, j'ai ajouté plus de 30 nouvelles entrées aujourd'hui et je l'ai soumis au LDP.
- Pas d'enregistrements sur les versions précédentes :(

## Où trouver les logiciels ?

Tous les programmes devraient être disponibles sur Internet (ftp ou http). Les liens vers ces ressources sont fournis dans la description de chaque logiciel. Il y a également une pléthore de logiciels non traités dans ces pages (qui sont généralement indépendants de la plate-forme), situés sur une des ressources listées dans la section lien du site principal (donné plus haut).

## Mises à jour et commentaires

Si vous trouvez des erreurs, que vous êtes au courant des mises à jour de l'un des outils décrit plus loin, ou que vous rencontrez des problèmes pour compiler l'une des applications, écrivez-moi à jae@zhar.net et je verrai ce que je peux faire.

Si vous connaissez des applications sur l'IA/VA, des bibliothèques de classe, etc, envoyez-moi un email au sujet de celles-ci. Mentionnez votre nom, l'adresse ftp ou http où l'on peut trouver ces applications, ainsi qu'un bref résumé/commentaire sur le logiciel (ces informations devraient me rendre les choses beaucoup plus faciles... mais ne vous sentez pas obligés de le faire ;) ).

Je sais que garder cette liste à jour et l'augmenter demande pas mal de travail. Alors s'il vous plait, soyez patient (j'ai aussi d'autres projets en cours). J'espère que vous trouverez ce document utile.

## Copyright/Licence



# Intelligence Artificielle traditionnelle

L'IA traditionnelle se fonde sur tout ce qui tourne autour de la logique, des systèmes de règles, de la linguistique, et du concept de rationalité. Ses racines sont les langages de programmation comme Lisp et Prolog. Les systèmes experts sont les plus grands exemples de réussite de ce paradigme. Un système expert consiste en une base de connaissances détaillée et un système complexe de règles pour l'utiliser.

De tels systèmes ont été utilisés comme support de diagnostiques médicaux et systèmes de vérification de solvabilité.

## Bibliothèques de classes/code d'IA

Ce sont des bibliothèques de classes ou code à utiliser lorsque vous programmez dans le domaine de l'intelligence artificielle. Elles ne sont pas destinées à devenir des applications à part entière, mais plutôt comme des outils afin de créer vos propres applications.

ACL2

- Site web : www.cliki.net/ACL2

ACL2 (A Computational Logic for Applicative Common Lisp) permet de démonter les théorèmes pour des applications industrielles. C'est à la fois de la logique mathématique et un système d'outils pour construire des preuves dans la logique. ACL2 marche avec GCL (GNU Common Lisp).

AI Kernel

- Site Web : aikernel.sourceforge.net
- Site Web Sourceforge : sourceforge.net/projects/aikernel

AI Kernel (noyau d'IA) est un moteur d'intelligence artificielle réutilisable qui utilise le traitement du langage naturel et un modèle Activateur/Contexte afin de permettre le multi-tâche entre les cellules installées.

AI Search II

- Site Web : www.bell-labs.com/topic/books/ooai-book/

En gros, cette bibliothèque offre au programmeur une série d'algorithmes de recherche qui peuvent être utilisés pour résoudre toutes sortes de problèmes différents. L'idée est que lorsqu'il développe un logiciel de résolution de problème, le programmeur devrait pouvoir se concentrer sur la représentation du problème à résoudre, sans avoir à se soucier de l'implémentation du moteur de recherche qui serait précisément utilisé pour mener la recherche. Cette idée a été appliquée grâce à l'implémentation d'une série de classes de recherche qui peuvent être incorporées dans d'autres logiciels à travers des spécifications de dérivation ou d'héritage du C++. Les algorithmes de recherche suivants ont été mis en ½uvre :

- Recherche en profondeur (deep-first) dans un arbre ou un graphe.
- Recherche horizontale (breadth-first) dans un arbre ou un graphe.
- Recherche par coûts uniformes dans un arbre ou un graphe.
- Recherche du meilleur en priorité (best-first).
- Recherche bidirectionnelle en profondeur (bidirectional depth-first) dans un arbre ou un graphe.

- Recherche bidirectionnelle horizontale (bidirectional breadth-first) dans un arbre ou un graphe.
- Recherche ET/OU en profondeur dans un arbre.
- Recherche ET/OU horizontale dans un arbre.

Il existe un livre qui correspond à cette bibliothèque, "Object-Oriented Artificial Instelligence, Using C++".

Aleph

- Site Web : web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/

Ce document fournit des informations de référence sur "A Learning Engine for Proposing Hypotheses" (Aleph : un moteur d'apprentissage pour proposer des hypothèses). Aleph est système de programmation par logique inductive (Inductive Logic Programming, ILP). La fonction d'Aleph est celle de prototype pour explorer des idées. Aleph est un algorithme ILP mis en ½uvre en Prolog par le Dr Ashwin Srinivasav au laboratoire informatique de l'université d'Oxford, et a été écrit spécifiquement pour la compilation avec le compilateur Prolog YAP.

Chess In Lisp (CIL : les échecs en Lisp)

- Site Web : *fait partie de l'archive CLOCC clocc.sourceforge.net

La fondation CIL (les échecs en Lisp) est une implémentation "Common Lisp" de toutes les fonctions principales nécessaires au développement d'applications sur les jeux d'échecs. Le but principal du projet CIL est d'amener les chercheurs en IA intéressés par l'emploi de Lisp à travailler sur le domaine des échecs.

DAI

- Site Web : starship.python.net/crew/gandalf/DNET/AI/

C'est une bibliothèque pour le langage de programmation Python qui fournit une interface orientée objet à l'outil système expert CLIPS. Il inclut une interface pour COOL (CLIPS Object Oriented Langage : Langage orienté objet CLIPS) qui permet de  :

- Examiner, étudier les classes COOL
- Créer et manipuler des instances COOL
- Manipuler les gestionnaires de messages COOL
- Manipuler les modules

FFLL

- Site Web : ffll.sourceforge.net

La bibliothèque gratuite de logique floue (FFLL : Free Fuzzy Logic Library) est une bibliothèque de classe et une API libres de logique floue qui est optimisée pour des applications dont la rapidité est importante, comme les jeux vidéos. FFLL est capable de charger des fichiers conformes au standard IEC 61131-7.

Fuzzy sets for Ada

- Site Web : http://www.dmitry-kazakov.de/ada/fuzzy.htm

  Freshmeat : http://freshmeat.net/projects/fuzzy/

Les ensembles flous pour Ada (Fuzzy sets for Ada) est une librairie fournissant des implémentations de facteurs de confiance pour les opérateurs NOT, AND, OR, XOR, + et *, les ensembles flous classiques avec les opérations sur la théorie des ensembles, les opérations sur la théorie des possibles, les opérations sur les ensembles intuitifs, la logique floue basée sur la théorie des possibles et les ensembles flous intuitifs ; les nombres flous, à la fois entier et à virgule flottante avec leurs opérations arithmétiques classiques, les variables linguistiques et les ensembles de variables linguistiques avec les opérations les concernant. Les entrées/sorties à base de chaînes de caractères sont supportées.

HTK

- Site Web : htk.eng.cam.ac.uk

La boîte à outils du modèle de Markov masqué (HTK : Hidden Markov Model ToolKit) est une boîte à outils portable pour créer et manipuler des modèles de Markov masqué. HTK consiste en une série de modules de bibliothèques et d'outils disponibles sous la forme de sources en C. Ces outils founissent des solutions sophistiquées pour l'analyse vocale, la formation HMM, les tests et l'analyse de résultats. L'application supporte les HMM en utilisant à la fois les fonctions gaussiennes sur les mélanges à densité constante et les distributions discrètes et peut être utilisée pour créer des systèmes HMM complexes. Le logiciel HTK contient une grande quantité d'informations et d'exemples.

JCK

- Site Web : www.pms.informatik.uni-muenchen.de/software/jack/

JCK est une nouvelle bibliothèque pour la programmation et la recherche par contraintes pour Java.

- JCK est constitué de trois composants :
- \- JCHR : Java Constraint Handling Rules (règles de manipulation de contraintes Java). Langage de haut niveau pour écrire des résolveurs de contraintes.
- \- JASE : Java Abstract Search Engine (moteur de recherche abstrait Java). Moteur de recherche générique pour JCHR afin de résoudre les problèmes de contraintes.
- \- VisualCHR : outil interactif pour visualiser les calculs JCHR.

Les sources et la documentation sont disponibles à partir du lien ci-dessus.

KANREN

- Site Web : http://kanren.sourceforge.net

KANREN est un système de programmation par logique déclarative avec des relations de premier ordre, inclus dans un sous-ensemble fonctionnel pur de Scheme. Le système est constitué de sémantiques de la théorie des ensembles, de relations vraies, de relations de premier ordre, de variables logiques à portée lexicale, de stratégies en profondeur (deep-first) et itératif en profondeur (iterative deepening). Le système atteint de hautes expressivités et performances sans élimination.

LK

- Site Web : www.cs.utoronto.ca/~neto/research/lk/

LK est une implémentation de l'heuristique de Lin-Kernighan pour le problème du commercial démarcheur et pour le problème de la correspondance parfaite de pondération minimum. Il est spécialement adapté aux figures géométriques en 2D, et a été appliqué dans certains exemples jusqu'à un million de villes. Il comprend aussi des générateurs d'exemples et des scripts Perl pour manipuler les exemples TSPLIB.

Cette implémentation introduit la "compensation groupée efficace", une technique algorithmique expérimentale dont le but est de rendre l'heuristique de Lin-Kernighan plus robuste dans le cas de données groupées.

maxent

- Version Python / C++ : homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html
- Version Java : maxent.sourceforge.net

La boîte à outils entropie maximum fournit un ensemble d'outils et une librairie pour construire des modèles d'entropie maximum (maxent) en Python ou C++. Il contient des modèles d'entropie maximum conditionnels, une estimation des paramètres GIS et L-BFGS, un lissage Gaussien, une API C++, un module d'extension Python, un utilitaire en ligne de commande et une bonne documentation. Une version Java est également disponible.

Nyquist

- Site Web : www-2.cs.cmu.edu/~music/nyquist/

Le projet de musique informatisée (Computer Music Project) au CMU développe la technologie de la musique informatisée, et de la performance interactive pour améliorer l'expérience et la créativité musicale de l'homme. Cet effort interdisciplinaire puise dans la théorie de la musique, dans les sciences cognitives, l'intelligence artificielle, l'apprentissage des machines, l'interaction homme-machine, les systèmes temps réelles, les graphismes et animations informatiques, le multimédia, les langages de programmation, le traitement du signal. Un exemple paradigmatique de ces efforts interdisciplinaires est la création de la performance interactive qui lie l'improvisation musicale de l'homme aux agents informatiques intelligents en temps réel.

OpenCyc

- Site Web : www.opencyc.org
- Site Web alternatif : sourceforge.net/projects/opencyc/

OpenCyc est la version libre de Cyc, la plus grande et la plus complète des bases de connaissance générale ainsi que le plus grand moteur de raisonnement de sens communs. Une ontologie basée sur 6000 concepts en relation avec 60000 affirmations.

Module de logique floue Python

- Site FTP : ftp://ftp.csh.rit.edu/pub/members/retrev/

C'est un simple module python pour la logique floue. Le fichier est 'fuz.tar.gz' dans ce répertoire. L'auteur ambitionne également d'écrire un algorithme de génétique simple de même qu'une bibliothèque sur les réseaux neuronaux. Regarder le fichier '00_index' dans ce répertoire pour obtenir des informations sur la version.

Screamer

- Site Web : www.cis.upenn.edu/~screamer-tools/home.html
- La version la plus récente fait partie de CLOCC : clocc.sourceforge.net

Screamer est une extension de Common Lisp qui ajoute un support pour la programmation non déterministe. Screamer comprend deux niveaux. Le niveau non déterministe basique ajoute un support pour la journalisation et les effets de coté irréalisables. En plus de cet aspect non deterministe, Screamer fournit un langage complet de programmation sous contraintes dans lequel on peut formuler et résoudre un mélange de systèmes sous contraintes numériques ou symboliques. Ensembles, ces deux niveaux étendent Common Lisp avec pratiquement toutes les fonctionnalités de Prolog et des languages de programmation logiques sous contraintes comme CHiP ou CLP(R). En outre, Screamer est complètement compatible avec Common Lisp. Les programmes de Screamer peuvent coexister et interopérer avec d'autres extensions de Common Lisp comme CLOS, CLIM et Iterate.

SPASS

- Site Web : spass.mpi-sb.mpg.de

SPASS : Démontreur de théorèmes automatisé pour des logiques de premier ordre avec égalité.

Si vous êtes intéressés par les démonstrations de théorèmes de logique du premier ordre, l'analyse formelle de programmes, systèmes, protocoles, les approches formelles de projets sur l'IA, les procédures de décisions, la démonstration de théorèmes à logique modale, SPASS peut vous offrir les bonnes fonctionnalités.

ThoughtTreasure

- Site Web : www.signiform.com/tt/htm/tt.htm

ThoughtTreasure est un projet pour créer une base de données de règles de sens communs à utiliser dans n'importe quelle application. Il consiste en une base de donnnées d'un peu plus de 100 000 règles et d'une API C pour l'intégrer dans vos applications. Les emballeurs Python, Perl, Java et TCL sont déjà disponibles.

Torch

- Site Web : www.torch.ch

Torch est une bibliothèque d'apprentissage écrit en C++. Son but est de fournir l'état de l'art des meilleurs algorithmes. Il est, et sera toujours en développement.

- Plusieurs méthodes basées sur les gradients, y compris des perceptions de niveau multiple, fonctions sur des bases radiales et un mélange de compétences. Plusieurs petits modules (modules linéaires, modules Tanh, modules SoftMax, ...) peuvent être reliés ensembles.
- Machine de vecteur de support, pour la classification et la regression.
- Distributions logicielles, comprenant Kmeans, des modèles de mélanges Gaussiens, des modèles de Markov Masqués, des classificateurs de Baye, et des classes pour la reconnaissance vocale avec entrainement incorporé.
- Modèles ensemblistes tels que Bagging et Adaboost.
- Modèles non paramétriques comme les k plus proches voisins, la regression de Parzen et l'estimateur de densité de Parzen.

Torch est une bibliothèque libre dont les auteurs encouragent tout le monde à développer de nouveaux logiciels qui seront inclus dans les versions futures sur le site officiel.

## Kits logiciels d'IA, applications, etc.

Voici plusieurs applications, kits logiciels, etc. ayant pour but la recherche dans le domaine de l'intelligence artificielle. Leur facilité d'utilisation peut varier, dans la mesure où ils ont été conçus pour satisfaire des intérêts de recherche particuliers, plus que comme un logiciel commercial sorti de l'étagère.

ASA - Adaptive Simulated Annealing

- Site web : www.ingber.com/#ASA-CODE
- Site FTP : ftp.ingber.com

ASA (Adaptive Simulated Annealing : Recuit simulé adaptatif) est un puissant algorithme en code C d'optimisation globale particulièrement pratique pour les systèmes non linéaires et/ou stochastiques.

ASA est développé pour trouver statistiquement le meilleur ajustement global d'une fonction coût non linéaire et non convexe sur un espace de dimension D. Cet algorithme permet un schéma de refroidissement du recuit pour une température T exponentiellement décroissante dans le temps recuit k, T=T_0 exp(-c k^1/D). L'introduction du renouvellement de recuit permet également une adaptation aux changements de sensibilités dans un espace paramétrique multi-dimensionnel. Cet schéma de refroidissement de recuit est plus rapide que le recuit de Cauchy rapide, où T = T_0/k, et encore plus rapide que le recuit de Boltzmann, où T = T_0/ln k.

Babylon

- Site FTP : ftp.gmd.de/gmd/ai-research/Software/Babylon/

BABYLON est un environnement modulaire, configurable et hybride pour le développement de systèmes experts. Ses caractéristiques comprennent les objets, les règles avec chaînage avant et arrière, la logique (Prolog) et les contraintes. BABYLON est codé et incorporé en Common Lisp.

cfengine

- Site web : www.iu.hio.no/cfengine/

Cfengine, ou "moteur de configuration" est un langage de très haut niveau pour la construction de systèmes experts qui administrent et configurent de grands réseaux informatiques. Cfengine utilise l'idée de classes et une forme primitive d'intelligence pour définir et automatiser la configuration de grands systèmes de façon la plus économique possible. Cfengine est conçu pour être une part des systèmes immunisés informatiques.

CLEARS

- Site web : ????? (Quelqu'un sait-il encore où trouver le site ?)

Le système CLEARS est un environnement graphique interactif pour les sémantiques de calcul. L'outil permet l'exploration et la comparaison de différents formalismes sémantiques ainsi que de leur interaction avec la syntaxe. Cela permet à l'utilisateur d'obtenir une idée de l'étendue des possibilités en matière de construction sémantique, ainsi qu'où se trouve la convergence réelle entre les théories.

CLIPS

- Site web : www.ghg.net/clips/CLIPS.html

CLIPS est un outil productif de système expert de développement et de livraison qui fournit un environnement complet pour la construction de systèmes experts basés sur des règles ou des objets.

CLIPS fournit un outil cohésif pour la gestion d'une grande variété de connaissances compatible avec trois différents paradigmes de programmation : basé règles, orienté objet et procédural. La programmation basée sur les règles permet de représenter les connaissances comme des heuristiques, ou règles empiriques, qui spécifient une série d'actions à entreprendre pour une situation donnée. La programmation orientée objet permet à des sytèmes complexes d'être modelés comme composants modulaires (qui peuvent être facilement réutilisés pour modeler d'autres systèmes ou pour créer de nouveaux composants). Les capacités de programmation procédurale fournies par CLIPS sont similaires à celles trouvées dans les langages comme le C, le Pascal, l'Ada et le Lisp.

Eprover

- Site web : http://www.eprover.org/
- Site web : http://www4.informatik.tu-muenchen.de/~schulz/WORK/eprover.html

:TODO:

FOOL et FOX

- Site web : rhaug.de/fool/
- Site FTP : ftp.informatik.uni-oldenburg.de/pub/fool

FOOL signifie organisateur flou Oldenburg [Fuzzy Organizer OLdenburg]. C'est le résultat d'un projet de l'Université d'Oldenburg. FOOL est une interface utilisateur graphique pour développer les règles de base floues. FOOL va vous aider à inventer et maintenir une base de données qui spécifie le comportement d'un contrôleur flou ou quelque chose de similaire.

FOX est un petit mais puissant moteur flou qui lit la base de données, lit des valeurs d'entrée et calcule la nouvelle valeur de contrôle.

FUF et SURGE

- Site web : www.cs.bgu.ac.il/research/projects/surge/index.htm
- Site FTP : ftp.cs.bgu.ac.il/pub/fuf

FUF est une implémentation étendue du formalisme des grammaires d'unification fonctionnelles (FUGs : functional unification grammars) introduit par Martin Kay, spécialisé dans la tâche de génération de langage naturel. Il ajoute les fonctionnalités suivantes au formalisme de base :

- Types et héritage.
- Installation de contrôle étendu (maintient de l'objectif, recherche inverse intelligente).
- Syntaxe modulaire.

Ces extensions permettent le développement de grandes grammaires qui peuvent être traitées efficacement et peuvent être maintenues à jour et comprises plus facilement. SURGE est une grande grammaire syntaxique de réalisations de l'anglais, écrite en FUF. SURGE est développée pour servir de composant de génération syntaxique de boîte noire dans un système de génération plus grand qui encapsule une connaissance riche de la syntaxe anglaise. SURGE peut aussi être utilisé comme plate-forme pour l'exploration de l'écriture de la grammaire dans une perspective de génération.

The Grammar Workbench

- Site web : ??? www.cs.kun.nl/agfl

Semble être obsolète ??? N'est plus sur le site, bien que le projet parent se poursuive toujours.

The Grammar Workbench (L'atelier de grammaire), ou GWB en abrégé, est un environnement pour le développement aisé de grammaires Affix dans le formalisme AGFL. Ses buts sont :

- de permettre à l'utilisateur d'introduire, inspecter et modifier une grammaire ;
- d'effectuer des vérifications de cohérence sur la grammaire ;

- de calculer des propriétés de la grammaire ;
- de générer des phrases d'exemple ;
- d'aider à effectuer des transformations sur la grammaire.

GSM Suite

- Site web : www.ibiblio.org/pub/Linux/apps/graphics/draw/

The GSM Suite (la suite GSM) est une série de programmes pour l'utilisation des machines à état fini en mode graphique. Cette suite consiste en des programmes qui éditent, compilent, et impriment des machines d'état. La suite comprend un programme d'édition, gsmedit, un compilateur, gsm2cc, qui produit une implémentation C++ d'une machine d'état, un générateur PostScript, gsm2ps, et deux autres programmes mineurs. GSM est sous licence publique GNU et est donc libre d'utilisation selon les termes de cette licence.

Isabelle

- Site web : isabelle.in.tum.de

Isabelle est un démontreur de théorème générique populaire développé à l'Université de Cambridge et au TU Munich. Les logiques existantes comme Isabelle/HOL fournissent un environnement de démontreur de théorème prêt à être utilisé pour d'assez grandes applications. Isabelle peut également servir comme châssis pour un prototypage rapide de systèmes déductifs. Il est présenté avec une grande bibliothèque comprenant Isabelle/HOL (logique classique d'ordre supérieur), Isabelle/HOLCF (Logique de Scott pour des fonctions calculatoire avec HOL), Isabelle/FOL (logique du premier ordre classique et intuitive), et Isabelle/ZF (ensemble de théories de Zermelo-Fraenkel au dessus de FOL).

Jess, the Java Expert System Shell

- Site web : herzberg.ca.sandia.gov/jess/

Jess (le shell système expert Java) est un clone du populaire shell de système expert CLIPS, entièrement écrit en Java. Avec Jess, vous pouvez à votre convenance donner à vos applets l'habilité à raisonner. Jess est compatible avec toutes les versions de Java, à partir de la version 1.0.2. Jess implémente les constructeurs suivants de CLIPS : defrules, deffunctions, defglobals, deffacts, et deftemplates.

learn

- Site web : www.ibiblio.org/pub/Linux/apps/cai/

Learn (apprendre) est programme d'apprentissage vocal avec un modèle de mémoire.

LISA

- Site web : lisa.sourceforge.net

LISA (Lisp-based Intelligent Software Agents : agents logiciels intelligents basés sur Lisp) est un système de règles de production lourdement influencé par JESS (Java Expert System Shell). Il a en son squelette un moteur de raisonnement basé sur l'algorithme de filtrage de Rete. LISA fournit également l'habilité à raisonner sur d'ordinaires objets CLOS.

NICOLE

- Site web : nicole.sourceforge.net

NICOLE (Nearly Intelligent Computer Operated Language Examiner) est une théorie ou une expérimentation qui, si on donne à un ordinateur suffisamment de combinaisons sur la manière dont les mots, groupes de mots ou phrases sont liés les uns aux autres, peut vous répondre. C'est un essai pour simuler une conversation en apprenant comment les mots se rapportent à d'autres mots. Un humain communique avec NICOLE via le clavier et NICOLE répond avec ses propres phrases qui sont automatiquement générées, basées sur ce que NICOLE a stocké dans sa base de données. Chaque nouvelle phrase qui a été écrite et que NICOLE ne connaît pas est ajoutée à la base de donnée de NICOLE, ce qui étend la base de connaissances de NICOLE.

NLTK

- Site web : nltk.sourceforge.net

NLTK, la boîte à outils de langage naturel (the Natural Language Toolkit), est une suite de librairies et de programmes en Python pour le traitement statistique et symbolique du langage naturel. NLTK contient des démonstrations graphiques et des échantillons de données. Il est accompagné d'une documentation fournie, y compris des tutoriels explicant les concepts sous-jacents des tâches de traitement du langage réalisées par l'outil.

NLTK convient parfaitement aux étudiants qui apprennent le traitement du langage naturel, ou qui réalisent une recherche dans le TLN ou un domaine approchant, tels que la linguistique, les sciences cognitives, l'intelligence artificielle, l'extraction de données, et l'apprentissage par machines. NLTK a été utilisé avec succès comme outil d'apprentissage, outil d'étude individuel, et comme une plateforme de prototypage et de construction de systèmes de recherche.

Otter : Système de déduction automatisé (An Automated Deduction System)

- Site web : www-unix.mcs.anl.gov/AR/otter/

Notre système de déduction automatisé actuel Otter est conçu pour démontrer les théorèmes de logique du premier ordre avec une égalité. Les règles d'inférence d'Otter sont basées sur la résolution et la paramodulation, et il comprend des procédures pour la réécriture de termes, l'ordonnancement de termes, la complétion Knut-Bendix, le pesage, et les stratégies pour diriger et restreindre les recherches de preuves. Otter peut aussi être utilisé comme un calculateur symbolique et a un système de programmation équationnelle incorporé.

PVS

- Site web :pvs.csl.sri.com

PVS est un système de vérification : c'est à dire un langage de spécification intégré à des outils de support et à un démontreur de théorème. Il est prévu pour capturer la plus sophistiquée des méthodes formelles mécanisées et pour être suffisamment robuste pour pouvoir être utilisé pour des applications significatives. PVS est un prototype de recherche : il évolue et s'améliore au fur et à mesure que nous développons ou appliquons de nouvelles capacités, et lorsque la charge en utilisation réelle expose de nouveaux besoins.

SNePS

- Site web : www.cse.buffalo.edu/sneps
- Site FTP : ftp.cse.buffalo.edu/pub/sneps

Le but à long terme du groupe de recherche SNePS est la conception et la construction d'un langage naturel utilisant un agent cognitif informatisé, et d'effectuer la recherche dans l'intelligence artificielle, les linguistiques calculatoires, et la science cognitive nécessaire pour cette entreprise. Les trois parties sur lesquelles le groupe met l'accent sont la représentation du savoir, le raisonnement, et la compréhension et la génération du langage naturel. Le groupe est largement connu pour son développement du système de représentation/raisonnement du savoir SNePS, et pour Cassie, son agent cognitif informatisé.

Soar

- Site web : sitemaker.umich.edu/soar

Soar a été développé pour être une architecture cognitive générale. Nous avons l'ultime intention de permettre à l'architecture Soar de :

- travailler sur l'ensemble des taches attendues d'un agent intelligent, des routines jusqu'aux problèmes extrêmement difficiles avec réponse ouverte ;
- représenter et utiliser les formes appropriées du savoir, comme les formes procédurales, déclaratives, épisodiques et possiblement iconiques ;
- se servir de tout l'éventail des méthodes de résolution de problèmes ;
- interagir avec le monde extérieur et
- apprendre tous les aspects des tâches et leurs performances sur celles-ci.

En d'autres termes, notre intention est de donner à Soar toutes les compétences requises d'un agent globalement intelligent.

TCM

- Site web : wwwhome.cs.utwente.nl/~tcm/
- Site FTP : ftp.cs.utwente.nl/pub/tcm/

TCM (Toolkit for Conceptual Modeling : boîte à outil pour la modélisation conceptuelle) est notre suite d'éditeurs graphiques. TCM contient des éditeurs graphiques pour des diagrammes entité-relation, des diagrammes classe-relation, des diagrammes de flux de données et d'évènements, des diagrammes de transition d'état, des diagrammes de structure de procédé de Jackson et des diagrammes de réseau systèmes, d'arbres de décomposition de fonction et d'éditeurs de tables variés, tels qu'un éditeur de table fonction-entité ou un éditeur de table de décomposition de fonction. TCM est facile d'accès et procède à de nombreuses vérifications de consistance, dont certaines sont effectuées immédiatement, d'autre pendant la requête.

Yale

- Site web : yale.cs.uni-dortmund.de/index.html

Yale (Yet Another Learning Environment : Encore un autre environnement d'apprentissage) est un environnement pour les expériences d'apprentissage des machines. Les expériences peuvent être faites d'un grand nombre d'opérateurs emboîtés les uns dans les autres de manière arbitraire et leur installation est décrite par des fichiers XML qui peuvent être aisément créés à l'aide de l'interface graphique. Les applications de YALE couvrent à la fois les tâches d'apprentissage de recherche et du monde réel.

WEKA

- Site web : lucy.cs.waikato.ac.nz/~ml

WEKA (Waikato Environnement for Knowledge Analysis : Environnement Waikato pour l'analyse de connaissance) est la pointe de la technologie de l'application des techniques d'apprentissage de machines à des problèmes pratiques. C'est un banc d'essai logiciel complet qui permet d'analyser les données du monde réel. Il intègre différents outils d'apprentissage machine avec un squelette identique et une interface utilisateur uniforme. Il a été conçu selon la méthode de "plus c'est simple, mieux c'est" qui permet à l'utilisateur d'expérimenter de façon interactive avec un outil d'apprentissage machine simple avant de se tourner vers des solutions plus complexes.

# Connexionnisme

Le connexionnisme est un terme technique pour un groupe de techniques liées. Ces techniques comprennent des notions comme les Réseaux de Neurones Artificiels, des Réseaux Sémantiques et quelques autres idées similaires. Dans cette section je me concentre essentiellement sur les réseaux de neurones (bien que je cherche aussi des ressources sur les autres techniques). Les réseaux neuronaux sont des programmes conçus pour simuler le fonctionnement du cerveau. Ils consistent en un réseau de petits n½uds basés sur les mathématiques, qui fonctionnent ensemble afin de former des schémas d'information. Ils ont un potentiel énorme et semblent actuellement rencontrer un grand succès dans le traitement de l'image et le contrôle de robots.

## Bibliothèques de classe/code sur le connexionnisme

Ce sont des bibliothèques de code ou de classes à utiliser dans la programmation dans le domaine du connexionnisme. Elles ne sont pas destinés à être des applications autonomes, mais plus des applications pour construire vos propres applications.

Logiciel de modélisation Bayésienne flexible

- Site web : www.cs.utoronto.ca/~radford/fbm.software.html

Ce logiciel implémente les modèles Bayésiens flexibles pour des applications de régression et de classification qui sont basés sur les réseaux neuronaux à perception multi-couche ou sur les transformations Gaussiennes. L'implémentation utilise les méthodes de Monte Carlo de chaîne de Markov. Les modules logiciels qui sont compatibles avec l'échantillonnage des chaînes de Markov sont incluses dans la distribution, et peuvent être utiles à d'autres applications.

BELIEF

- Site web : www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/reasonng/probabl/belief/

BELIEF (croyance) est une implémentation Common Lisp de la fusion et de l'algorithme de propagation de Dempster et Kong pour des Modèles de Fonction de Croyances Graphique et l'algorithme de Lauritzen et Spiegelhalter pour des Modèles Probabilistes Graphiques. Il comprend du code pour manipuler les modèles de Croyance graphiques comme les réseaux Bayésiens et les diagrammes de Relevance (un sous-ensemble des diagrammes d'Influence) en utilisant à la fois les fonctions de croyance et les probabilités comme des représentations basiques de l'incertitude. Il utilise la version de Shenoy et de Shafer de cet algorithme, donc l'une de ses caractéristiques uniques est le fait qu'il supporte à la fois les distributions de probabilité et les fonctions de croyance. Il possède aussi un support limité des modèles de second ordre (distributions de probabilité sur les paramètres).

bpnn.py

- Site web : arctrix.com/nas/python/bpnn.py

Un simple ANN à propagation arrière en Python.

CNNs

- Site web : www.isi.ee.ethz.ch/~haenggi/CNNsim.html
- Un site plus récent : www.isi.ee.ethz.ch/~haenggi/CNNsim_adv_manual.html
- Version Java : www.ce.unipr.it/research/pardis/CNN/cnn.html

Réseaux neuronaux cellulaires (CNN : Cellular Neural Networks) est un paradigme d'informatique parallèle massive défini dans des espaces discrets de dimension N.

CONICAL

- Site web : strout.net/conical/

CONICAL est une bibliothèque de class en C++ pour construire des simulations communes à la neuroscience informatique. Actuellement, elles se concentrent sur la modélisation compartimentale, avec des caractéristiques proches de GENESIS et NEURON. Un modèle de NEURON est fait de compartiments, souvent avec une forme cylindrique. Quand ils sont assez petits, ces cylindres à possibilité d'extension peuvent approximer quasiment toutes les géométries. Les classes futures pourront accepter les cinématiques de réaction-diffusion et bien plus. Une caractéristique clef de CONICAL est sa compatibilité à travers les plates-formes, il a été entièrement codéveloppé et testé sous Unix, DOS et Mac OS.

L'architecture neuronale de Jet

- Site web : www.voltar-confed.org/jneural/

L'architecture neuronale de Jet est un châssis C++ pour faire des projets de réseau neuronaux. Le but de ce projet était de faire une architecture neuronale rapide et flexible qui n'est pas limité à une sorte de réseau, et de s'assurer que l'utilisateur final pourra facilement écrire des applications utiles. De même, toute la documentation est facile d'accès.

Joone

- Site web : joone.sourceforge.net

Joone est un châssis de réseau neuronal pour créer, entraîner et tester des réseaux neuronaux. Le but est de créer un environnement distribué pour des utilisateurs enthousiastes ou professionnels, basé à la fois sur JavaSpaces et sur les dernières technologies Java. Joone est composé d'un moteur central qui est le fulscrum de toute application qui existe déjà ou qui va être développé. Le moteur neuronal est modulaire, de taille réglable, multi-tâches et robuste. Tout le monde peut écrire de nouveaux modules pour implémenter de nouveaux algorithmes ou de nouvelles architectures à partir des simples composants distribués avec le moteur central. L'idée principale est de créer les bases pour promouvoir un million d'applications sur l'IA qui résolveront les problèmes autour du c½ur de la structure.

Classe Matrice

- Site FTP : ftp.cs.ucla.edu/pub/

Une classe Matrice C++ simple, rapide et efficace conçue pour les scientifiques et les ingénieurs. La classe Matrice est tout à fait adaptée à des applications avec des algorithmes mathématiques complexes. A titre d'exemple de l'utilité la classe Matrice, elle a été utilisé pour implémenter l'algorithme de rétropropagation de l'erreur pour un réseau neuronal artificiel à réaction positive multi-couche.

Neural Networks at your Fingertips (Réseaux Neuronaux sur le bout des doigts)

- Site web : http://www.neural-networks-at-your-fingertips.com/

:TODO:

NEURObjects

- Site web : http://www.disi.unige.it/person/ValentiniG/NEURObjects/

:TODO:

Pulcinella

- Site web : iridia.ulb.ac.be/pulcinella/Welcome.html

Pulcinella est écrit en Common Lisp, et semble être une bibliothèque de fonctions Lisp pour la créer, modifier et évaluer des systèmes valués. Alternativement, l'utilisateur peut choisir d'interagir avec Pulcinella via une interface graphique (disponible uniquement dans le CL Allegro). Pulcinella fournit les primitives pour construire et évaluer des modèles incertains en accord avec plusieurs calculs incertains, comprenant la théorie de la probabilité, la théorie de la possibilité, et la théorie des fonctions de croyance de Dempster-Shafer, et la théorie de la possibilité par Zadeh, Dubois et Prade. Un manuel est disponible sur demande.

ScnANNlib

- Site web : www.sentinelchicken.org/projects/scnANNlib/

SCN Artificial Neural Network Library (Bibliothèque de réseau neuronal artificiel SCN) fournit un programmateur avec une API simple orientée objet pour construire des ANNs. Actuellement, la bibliothèque est compatible avec les réseaux non récursifs avec un nombre de couches arbitraires, chacune avec un nombre de n½uds arbitraires. Des installations existent pour s'entraîner avidement, et il existe des plans pour étendre gracieusement les fonctionnalités de la bibliothèque dans ses versions futures.

Applications du UTCS Neural Nets Research Group

- Site web : www.cs.utexas.edu/users/nn/pages/software/software.html

Cette entrée, un peu différemment des autres, est une référence à une collection de logiciels plutôt qu'à une seule application. Elle a été développée par le UTCS Neural Net Research Group (Groupe de recherche sur les réseaux neuronaux UTCS). Voici un résumé des logiciels disponibles  :

- Traitement du langage naturel
  - MIR - Prototypage rapide basé sur TCL/TK pour le traitement des phrases
  - SPEC - Analyse grammaticale des phrases complexes
  - DISCERN - Traitement d'histoires basées sur les scripts, comprenant :
    - PROC - Analyse grammaticale, génération, réponse à des questions
    - HFM - Organisation de la mémoire épisodique
    - DISLEX - Traitement lexical
    - DISCERN - Le modèle intégré complet
  - FGREPNET - Apprentissage des représentations distribuées
- Auto-organisation
  - LISSOM - Carte auto-organisante avec des connections latérales.
  - FM - Cartes auto-organisantes génériques
- Neuroévolution
  - Sous-populations disciplinées (ESP : Enforced Sub-Populations) pour les taches de décision séquentielle.
    - Equilibrage de pôle double non markoviens.
  - Neuroévolution adaptive et symbiotique (SANE : Symbiotic, Adaptive NeuroEvolution, prédécesseur d'ESP)
    - JavaSANE - Application Java pour appliquer SANE à de nouvelles tâches.

        - SANE-C - Version C, prédécesseur de JavaSANE.
        - Equilibrage de pôles - SANE de niveau neuronal sur la tâche d'équilibrage de pôle.
    - Programme de neuroévolution d'augmentation de topologies (NEAT : NeuroEvolution of Augmenting Topologies) pour l'évolution de réseau neuronaux utilisant une structure.

Divers réseaux neuronaux (C++)

- Site web : www.dontveter.com/nnsoft/nnsoft.html

Exemples de codes de réseaux neuronaux tirés du livre The Pattern Recognition Basics of AI. Ce sont des exemples simples de codes de ces divers réseaux neuronaux. Ils fonctionnent correctement comme un bon point de départ pour des expérimentations simples et pour apprendre à quoi ressemble le code derrière les simulateurs. Les types de réseaux disponibles sur le site (codés en C++) sont :

- le paquetage Backprop
- Les algorithmes du plus proche voisin
- L'algorithme d'activation interactive
- Les algorithmes des machines de Hopfield et de Boltzman
- Le classificateur de schéma linéaire
- ART I
- La mémoire associative bi-directionnelle
- Le réseau à rétropropagation et à réaction positive.

## A venir

:TODO:

# Informatique évolutive

L'informatique évolutive est en réalité un terme général pour un grand nombre de techniques de programmation, y compris des algorithmes génétiques, systèmes adaptatifs complexes, programmation évolutive, etc. L'initiative principale de toutes ces techniques est l'idée d'évolution. L'idée qu'un programme pouvant évoluer vers un certain objectif puisse être écrit. Cet objectif peut être n'importe où entre la résolution d'un problème scientifique et la victoire dans un jeu.

:TODO:

# A venir

:TODO:

# Les agents

Aussi connus sous le nom d'agents intelligents ou simplement agents, ce domaine de recherche en IA traite de petites applications destinées à aider l'utilisateur(trice) dans son travail. Ils peuvent être mobiles (capables de stopper leur exécution sur une machine et de la relancer sur une autre) ou statiques (en demeurant dans une seule machine). Ils sont le plus souvent spécifiques à une tâche (donc relativement simples) et conçus pour aider l'utilisateur à la manière d'un assistant. Jusqu'à présent, l'utilisation la plus courante (c'est à dire la plus largement répandue) de ce type d'application concerne les robots d'indexation, que beaucoup de moteurs de recherche utilisent (ex.: webcrawler).

Agent

- Site FTP : www.cpan.org/modules/by-category/23_Miscellaneous_Modules/Agent/

Agent est le prototype d'un système "Information Agent". Il est à la fois multiplateforme et multilangue, car il stocke les informations récoltées dans de simples chaînes de caractères condensées. Il peut être condensé et transporté à travers le réseau dans n'importe quel format, car l'agent peut s'autobloquer dans son état initial.

agentTool

- Site Web : en.afit.af.mil/ai/agentool.htm
- Site de téléchargement : en.afit.af.mil/ai/registration.htm

Autre cadre de développement d'agent conçu en Java. Il est assez unique dans le fait qu'il s'appuie sur l'utilisation d'une interface graphique pour concevoir le système qui va "synthétiser semi-automatiquement des systèmes multiagents pour répondre à ces exigences". Il vous faut un navigateur compatible java pour pouvoir télécharger le logiciel. :P

Aglets Workbench

- Site Web : www.trl.ibm.com/aglets/index_e.htm

Un aglet est un objet Java qui peut se déplacer d'un hôte à un autre sur Internet. Cela veut dire qu'un aglet s'exécutant sur une machine peut brusquement arrêter son exécution, s'expédier sur une machine distante, et reprendre son déroulement sur son nouvel hôte. Quand l'aglet se déplace, il emporte aussi bien son code source que son état (ses données). Un mécanisme de sécurité intégré sécurise l'accueil d'aglets d'origine incertaine pour l'hôte. L'API d'Aglet Java (J-AAPI) est un standard proposé au public pour interfacer les aglets et leur environnement. J-AAPI contient des fonctions pour initialiser les aglets, gérer les messages et expédier, retirer, activer ou désactiver, cloner, ou se débarrasser de l'aglet. J-AAPI est simple, souple, et stable. Les développeurs d'applications peuvent coder des aglets multiplateformes en considérant qu'ils peuvent fonctionner sur n'importe quelle machine supportant J-AAPI.

A.L.I.C.E.

- Site Web : www.alicebot.org

Le logiciel ALICE implémente l'AIML (Artificial Intelligence Markup Language), un langage à balise non standard en évolution, permettant de créer des robots de "chat". La caractéristique de base de sa conception est minimaliste. Comparé à d'autres langages de robot de "chat", l'AIML est peut-être le plus simple. Le langage de filtrage est très simple, ne permettant par exemple qu'un seul caractère joker ('*') de comparaison par filtre. L'AIML est un langage XML, ce qui implique qu'il obéit à certaines méta-règles grammaticales. Le choix d'une syntaxe XML rend possible l'intégration avec d'autres outils tels que des éditeurs XML. Une autre motivation du choix du XML est son apparence familière, particulièrement pour les personnes expérimentée en HTML.

Ara

- Site Web : wwwagss.informatik.uni-kl.de/Projekte/Ara/index_e.html

Ara est une plateforme d'exécution sûre et portable d'agents mobiles dans les réseaux hétérogènes. Dans ce contexte, les agents mobiles sont des programmes ayant la capacité de changer de machine hôte pendant leur exécution tout en préservant leur état interne. Cela leur permet de gérer des opérations localement, plutôt que de devoir les effectuer à distance. Par rapport aux plateformes du même type, le but précis d'Ara est de fournir des fonctionnalités complètes d'agent mobile, tout en maintenant autant que possible des modèles et langages de programmation établis.

BattleBots

- Site Web : www.bluefire.nu/droidbattles/

Jeu de programmation d'IA où l'on conçoit un robot en choisissant du matériel et en programmant son unité centrale, puis en faisant des compétitions avec d'autres robots. Les compétitions peuvent engager des équipes et des règles spéciales pour un jeu.

Le matériel disponible pour concevoir votre robot comprend des armes, un moteur, des scanners, un microprocesseur, etc. Le langage de programmation dépend du type de votre CPU et est similaire à un langage d'assemblage.

Bee-gent

- Site Web : www2.toshiba.co.jp/beegent/index.htm

Bee-gent est un nouveau type de cadre de développement car c'est un système d'agent à 100%. Par opposition à d'autres systèmes qui ne font qu'une utilisation partielle des agents, Bee-gent "agentifie" complètement la communication qui a lieu entre les applications logicielles. Les applications deviennent des agents, et tous les messages sont transportés par les agents. Ainsi, Bee-gent permet aux développeurs de construire des systèmes ouverts, distribués et souples, qui permettent une utilisation optimale des applications existantes.

Bond

- Site Web : bond.cs.ucf.edu/

Encore un autre système d'agent en Java...

Bond est un système d'objets distribués conçu en Java et un cadre d'agent. Il met en application un logiciel médian basé sur des messages et des services associés tels qu'un annuaire, la persistance, la surveillance et la sécurité. Bond permet de construire facilement des applications réparties multiagents. Une autre application future de Bond est le Laboratoire Virtuel supportant l'annotation d'informations et l'informatique répartie à grande échelle.

Cadaver

- Site Web : www.erikyyy.de/cadaver/

Cadaver simule en temps réel des cyborgs et la nature. Le champ de bataille est constitué de forêts, de blé, d'eau, d'herbe, de carcasses (bien sûr) et de plein d'autres choses. Le serveur de jeu contrôle le jeu et les règles. Vous démarrez le serveur et connectez quelques clients. Les clients communiquent avec le serveur en utilisant un protocole très primitif. Ils peuvent donner l'ordre aux cyborgs de moissonner le blé, d'attaquer des ennemis ou de couper du bois. Le jeu n'est pas destiné à être joué par des humains! Il y a beaucoup trop de choses à contrôler. Réservé aux experts : vous pouvez uniquement vous connecter au serveur par telnet et entrer les commandes manuellement. Mais l'idée est plutôt que vous écriviez des clients dotés d'une intelligence artificielle pour battre les autres intelligences artificielles. Vous pouvez sélectionner un langage (et un système d'exploitation) de votre choix pour effectuer cette tâche. Il n'y a qu'à écrire un programme qui communique sur les canaux d'entrée et de sortie standard. Puis vous pouvez employer des programmes tels que "socket" pour connecter vos clients au serveur. Il n'est PAS nécessaire d'écrire du code TCP/IP, bien que je l'ai fait :) La bataille ne sera pas ennuyeuse, donc vous disposez du client appelé "spyboss" qui affiche graphiquement l'action à l'écran.

Cougaar

- Site Web : www.cougaar.org/

Cougaar est une architecture Java pour la construction d'applications à base d'agents distribués à grande échelle. C'est le produit de plusieurs années de recherche dans le cadre du projet DARPA sur les systèmes d'agents à grande échelle, et il inclut non seulement l'architecture du noyau mais aussi un ensemble de composants de démonstration, de visualisation et d'administration pour simplifier le développement d'applications complexes et distribuées. [Encore un autre système d'agents conçu en Java - ed.]

D'Agent (anciennement AGENT TCL)

- Site Web : agent.cs.dartmouth.edu/software/agent2.0/
- Site FTP : agent.cs.dartmouth.edu/pub/agents/

Un agent transportable est un programme qui peut migrer d'une machine à une autre dans un réseau hétérogène. Le programme choisit où et quand il va migrer. Il peut suspendre son exécution à un certain point, se déplacer sur une autre machine et reprendre l'exécution sur la nouvelle machine. Par exemple, un agent transportant un courrier électronique migre d'abord vers le routeur et ensuite vers la boîte mail du destinataire. L'agent peut effectuer arbitrairement un traitement complexe sur chaque machine afin de s'assurer que le message atteint le bon destinataire.

Dunce

- Site Web : www.boswa.com/misc/dunce/

Dunce est un simple robot de "chat" (IA conversationnelle) et un langage pour programmer ce type de robots. Il emploie un filtrage d'expressions rationnelles de base et un mécanisme semi-neuronal de lancement de règles et de réponses (avec des cycles accélération / ralentissement d'activité).

FIPA-OS

- Site Web : fipa-os.sourceforge.net/
- Site Web secondaire : www.nortelnetworks.com/products/announcements/fipa/

FIPA-OS est l'implémentation "open source" des éléments obligatoires contenus dans la spécification FIPA nécessaires à l'interopérabilité des agents. En plus d'être compatible avec le concept d'interopérabilité de FIPA, FIPA-OS fournit également une architecture basée sur des composants permettant le développement d'agents spécifiques à un domaine, pouvant utiliser les services des agents de la plateforme FIPA. Il est implémenté en Java.

FishMarket

- Site Web : www.iiia.csic.es/Projects/fishmarket/newindex.html

FM - Le projet FishMarket mené par l' "Artificial Intelligence Research Institute" (IIIA-CSIC) tente d'apporter sa contribution dans cette direction en développant FM, une maison de ventes aux enchères électronique gérée par des agents, qui a été transformée en banc d'essai pour les marchés d'enchères électroniques. Le cadre, conçu et mis en application comme extension de FM96.5 (une version Java de la maison de ventes de Fishmarket), permet de définir des scénarii de transactions basés sur les enchères à la criée (enchères hollandaises). FM fournit un cadre où les concepteurs d'agent peuvent réaliser des expériences contrôlées de telle manière qu'une multitude de scénarii de marchés expérimentaux - que nous considérons comme scénarii de compétition du fait de la nature concurrentielle du domaine - d'un réalisme et d'une complexité plus ou moins forts peuvent être établis, activés, et enregistrés ; et de telle manière que les agents hétérogènes (humains et logiciels) d'échange (acheteur et vendeur) peuvent être comparés, accordés et évalués.

GNU Robots

- Site Web : www.gnu.org/software/robots/robots.html

GNU Robots est un jeu / divertissement où il vous faut construire un programme pour un petit robot, puis le regarder explorer un monde. Le monde en question est rempli de méchants qui peuvent vous blesser, d'objets dans lesquels vous pouvez frapper, et de nourriture que vous pouvez manger. Le but de ce jeu est de collecter autant de prix que possible avant d'être tué par un ennemi ou d'être à court d'énergie. Les robots peuvent être écrits en script "Guile" ou en utilisant une interface graphique.

Grasshopper

- Site Web : www.grasshopper.de/

Autre système d'agent en Java. Possède de nombreuses fonctionnalités et est activement développé. Commercial mais libre. Orienté à l'origine vers les systèmes embarqués.

Hive

- Site Web : hive.sourceforge.net/

Hive est une plateforme logicielle en Java pour créer des applications réparties. En utilisant Hive, les programmeurs peuvent facilement créer des systèmes qui se connectent et utilisent des données provenant d'internet. Au fond, Hive est un environnement d'évolution pour les agents distribués, qui communiquent et se déplacent pour accomplir des tâches. Nous essayons de rendre Internet vivant.

ICM

- Site Web : www.nar.fujitsulabs.com/
- Site sur SourceForge : sourceforge.net/projects/networkagent/

Inter-Agent Communication Model (ICM, Modèle de Communication Inter-agents) est un mécanisme de communication pouvant être utilisé pour envoyer des messages entre agents en mode asynchrone. Son domaine d'application prévu est le mécanisme de transport pour les langages de communication d'agent (ACLs), comme KQML et ACL de FIPA.

Jacomma

- Site Web : jacomma.sourceforge.net/
- Site sur SourceForge : sourceforge.net/projects/jacomma/

Jacomma est une plateforme / un cadre de développement pour développer des agents d'information distribués, mobiles et réactifs, avec des capacités de communication hétérogène, en Java et Jpython.

Jacomma fournit un cadre de développement et un environnement d'exécution, qui repose sur l'infrastructure "Inter-Agent Communication Model". ICM définit un protocole de communication, une architecture de stockage et d'expédition de messages, et une infrastructure de communication bas niveau pour l'échange de messages. La communication est réellement asynchrone, basée sur des sockets TCP.

Il y a une rubrique sur ICM dans ce guide, vous pouvez également le trouver via un lien à partir de ce site.

Jade

- Site Web : sharon.cselt.it/projects/jade/

JADE (Java Agent DEvelopment Framework) est un cadre de développement logiciel complètement implémenté en langage Java. Cela simplifie l'implémentation de systèmes multiagents à travers un logiciel médian dont on est assuré qu'il est conforme aux spécifications de la FIPA, et à travers un ensemble d'outils permettant le débuggage et la phase de déploiement.

La plateforme d'agents peut être distribuée entre diverses machines (qui n'ont même pas besoin de partager le même système d'exploitation) et sa configuration peut être réalisée au moyen d'une interface graphique distante. La configuration peut même être changée au cours de l'exécution, en déplaçant les agents d'une machine à une autre, au fur et à mesure que cela est nécessaire.

JAM Agent

- Site Web : www.marcush.net/IRS/irs_downloads.html

JAM accepte aussi bien les modes de raisonnement descendant (à base de buts à atteindre) que les modes de raisonnement ascendant (à partir des données). JAM choisit ses objectifs et plans selon la priorité maximale si le raisonnement métaniveau n'est pas utilisé, ou sur les plans de raisonnement métaniveau développés par l'utilisateur si ceux-ci existent. Pour JAM, la conceptualisation des objectifs et l'accomplissement des objectifs est définie plus classiquement (UMPRS est plus basé sur les performances comportementales que véritablement sur les objectifs) et fait la distinction entre les plans pour atteindre un objectif et les plans qui encodent simplement les comportements. Les objectifs-type implémentés incluent la réalisation (atteindre un état donné du système), la maintenance (atteindre de nouveau cet état donné du système), et la performance. L'exécution simultanée d'objectifs multiples est possible, avec une capacité de suspension et de reprise pour chaque processus d'objectif (c-à-d d'intention). Les plans de JAM ont des propriétés explicites de condition à priori et de fonctionnement qui restreignent leur applicabilité, une propriété de condition à posteriori et une section des propriétés du plan sont utilisées pour définir des caractéristiques spécifiques à un plan ou un domaine. Les constructions de plan disponibles incluent : l'ordonnancement, l'itération, les objectifs partiels, les segments de plan atomiques (c-à-d non-interruptible), l'exécution conditionnelle non déterministe et déterministe à n-branche, l'exécution parallèle de segments de plan multiples, la synchronisation basée sur l'état donné du système ou les objectifs, une section de manipulation des défaillances explicite, et la définition de primitives de fonction Java au moyen de sa construction dans JAM aussi bien que l'invocation de membres de classe prédéfinies (c-à-d propriétaire) via les capacités de réflexion de Java, sans avoir à le construire dans JAM.

JATLite

- Site Web : java.stanford.edu/

JATLite fournit un ensemble de paquetages Java qui facilite la conception de systèmes multiagents en Java. JATLite fournit seulement un ensemble petit et léger de paquetages, de sorte que les développeurs puissent se débrouiller avec tous les paquetages avec un minimum d'efforts. Pour plus de souplesse, JATLite fournit quatre couches différentes, d'une implémentation abstraite à une implémentation "Router". Un utilisateur peut accéder à n'importe quelle couche fournie. Chaque couche est basée sur un ensemble de postulats différents. L'utilisateur peut choisir la couche appropriée selon les postulats de la couche et de l'application de l'utilisateur. La page d'introduction présente les fonctionnalités de JATLite et l'ensemble des postulats de chaque couche.

JATLiteBeans

- Site Web : waitaki.otago.ac.nz/JATLiteBean/
- Interface de JATLite améliorée et maniabilité accrue, comprenant l'envoi, la réception et l'analyse lexicale des messages KQML

- Architecture extensible pour la gestion des messages et l'administration "thread of control" (NdT : unité d'exécution) des agents
- Fonctions utiles pour l'analyse lexicale du contenu de messages KQML simples
- JATLiteBean permet la transmission automatique aux agents animateurs de l'information sur les capacités des agents
- Gestion automatique, facultative de la directive "forward"
- Analyseur lexical des fichiers génériques de configuration
- Vérificateur syntaxique de KQML

Java(tm) Agent Template

- Site Web : www-cdr.stanford.edu/ABE/JavaAgent.html

Le JAT fournit un modèle pleinement fonctionnel, écrit entièrement en langage Java, qui permet de construire des agents logiciels qui communiquent en poste-à-poste ("peer-to-peer") avec une communauté d'autres agents répartis sur Internet. Bien que des portions du code utilisé dans chaque agent soient portables, les agents JAT ne sont pas mobiles mais restent statiques sur une seule machine. Ce comportement est en opposition à beaucoup d'autres technologies "agent". (Cependant, en utilisant le RMI Java, les agents JAT pourraient se déplacer dynamiquement sur une machine extérieure par l'intermédiaire d'un agent résidant sur cette seconde machine). Actuellement, tous les messages d'agent emploient KQML comme protocole de haut niveau ou comme emballeur de message. Le JAT possède une fonctionnalité pour échanger dynamiquement les "Ressources", qui peuvent inclure des classes Java (ex.: nouveaux langages ou interpréteurs, services distants, etc.), des fichiers de données et des informations intégrés dans les messages KQML.

Khepera Simulator

- Site Web : diwww.epfl.ch/lami/team/michel/khep-sim/index.html

Khepera Simulator est un paquetage logiciel du domaine public écrit par Olivier MICHEL pendant la préparation de son doctorat, au Laboratoire I3S, URA 1376 du CNRS et de l'Université de Nice-Sophia Antipolis, en France. Il permet d'écrire son propre contrôleur en langage C ou C++ pour le robot mobile Khepera, de le tester dans un environnement simulé, et comporte une belle interface graphique X11 colorée. De plus, si vous possédez un robot Khepera, le simulateur peut commander le vrai robot en utilisant le même algorithme de contrôle. Il est principalement destiné aux chercheurs étudiant les agents autonomes.

Lyntin

- Site Web : lyntin.sourceforge.net/

Lyntin est un client "Mud" (Multi-User Dialog, dialogue multi-utilisateur) extensible et un cadre de développement d'agents autonomes, ou de robots, aussi bien que tout ce qui approche le "mudding" en général. Lyntin est basé sur du Python, langage de programmation amusant, dynamique et orienté objet, basé sur TinTin++, un beau client "mud".

Mole

- Site Web : mole.informatik.uni-stuttgart.de/

Mole est un système d'agent compatible avec les agents mobiles programmés en Java. Les agents de Mole sont un regroupement d'objets, qui n'ont pas de référence vers l'extérieur, et qui dans l'ensemble travaillent sur des tâches données par l'utilisateur ou par d'autres agents. Ils ont la capacité d'évoluer sur un réseau d'"emplacements" de façon autonome. Ces "emplacements" sont une abstraction de n½uds réels, qui existent dans le réseau sous-jacent. Ils peuvent accéder à des ressources de nature locale, en communiquant avec les agents dédiés représentant ces services. Les agents sont capables d'utiliser les services fournis par d'autres agents et de fournir également des services.

Narval

- Site Web : www.logilab.fr/narval.html et www.logilab.org/projects/narval

Narval signifie Assistant Réseau pour la Visualisation de Ressources et Langage d'Agent ("Network Assistant for Ressources Visualization and Agent Language"). C'est un assistant personnel en réseau basé sur l'intelligence artificielle et les technologies d'agent. Il exécute des recettes (séquences d'actions) pour effectuer ses tâches. Il est très facile de définir une nouvelle action grâce à du XML et de l'implémenter en utilisant Python. Les recettes peuvent être codées et déboguées au moyen d'une interface graphique.

NeL

- Site Web : www.nevrax.org/

NeL est en fait une bibliothèque de développement de jeux (pour les jeux multijoueurs massifs), mais je le signale ici car il devrait inclure une bibliothèque d'IA assez considérable. Voici une notice tirée du livre blanc :

Le but de cette bibliothèque d'IA est de fournir une approche pragmatique pour créer une plateforme d'agents distribués. Elle est centrée sur les agents ; des entités individuelles communiquant indépendamment de l'endroit, en utilisant un modèle action-réaction.

OAA

- Site Web : www.ai.sri.com/~oaa/

"Open Agent Architecture" (Architecture Ouverte d'Agents) est un cadre dans lequel une communauté d'agents logiciels s'exécutant sur des machines distribuées peuvent travailler ensemble sur des tâches assignées par des participants de la communauté, humains ou non. La coopération distribuée et la communication haut-niveau sont deux idées centrales à l'origine d'OAA.

Il définit un langage de communication inter-agent et est compatible avec de multiples plateformes et langages de programmation.

PAI

- Site Web : utenti.quipo.it/claudioscordino/pai.html

PAI ("Programmable Artificial Intelligence", Intelligence Artificielle Programmable) est un programme capable de soutenir une conversation dans sa langue maternelle, l'anglais. Ecrit en C++.

Penguin!

- Site FTP : www.cpan.org/modules/by-category/23_Miscellaneous_Modules/Penguin/FSG/

Penguin est un module Perl version 5. Il vous fournira un ensemble de fonctions qui vous permettront de :

- envoyer du code Perl crypté, avec signature numérique vers une machine distante où il sera exécuté.
- recevoir du code et, selon le signataire du message, l'exécuter dans un compartiment arbitrairement sécurisé et limité.

La combinaison de ces fonctions permet le codage direct en Perl d'algorithmes qui permettent un commerce électronique sûr, des agents mobiles de récupération d'information, des applications d'aide à la navigation en contenu temps réel sur Internet ("live content"), le calcul distribué à répartition de charge, la mise à jour de logiciel à distance, l'administration de machines distantes, la propagation d'information basée sur le contenu, les applications de données partagées à travers Internet, les générateurs d'applications réseau, et ainsi de suite.

Ps-i

- Site Web : ps-i.sourceforge.net/

Ps-i est un environnement permettant d'effectuer des simulations basées sur des agents. Il est multiplateforme, et un exécutable est disponible pour Win32. Ses fonctionnalités sont :

- langage déclaratif pour la spécification des modèles
- langage de script Tcl/Tk conforme aux normes industrielles avec optimisation intégrée des sous-programmes, évaluation spéculative, de plus les utilisateurs du compilateur JIT xf86 pourront créer des modèles complexes sans dégradation de performance
- interface conviviale
- sauvegarde et restauration du déroulement du programme
- changement des paramètres du modèle à la volée
- visualisation des données : affichage d'agent par champs avec de multiples formes et couleurs, fenêtre de statistiques, visionneur d'agents, explorateur de sous-programmes et outil de mise en surbrillance d'agents

RealTimeBattle

- Site Web : www.lysator.liu.se/realtimebattle/

RealTimeBattle (Bataille Temps Réel) est un jeu de programmation dans lequel les robots, contrôlés par des programmes, se battent les uns contre les autres. L'objectif est de détruire ses ennemis, à l'aide du radar pour observer les alentours et du canon pour tirer.

- Le jeu se déroule en temps réel, le programme du robot s'exécutant en tant que processus fils de RealTimeBattle.
- Les robots communiquent avec le programme principal via l'entrée et la sortie standard.
- Les robots peuvent être construits dans presque n'importe quel langage de programmation.
- Autant de robots que voulu peuvent concourir simultanément si le système d'exploitation le permet.
- Un langage de transmission de messages simple est utilisé pour communiquer, ce qui facilite le début de la construction de robots.
- Les robots se comportent comme des objets physiques réels.
- Vous pouvez créer vos propres arènes.
- Hautement configurable.

Remembrance Agents

- Site Web : www.remem.org/

Remembrance Agents (les Agents du Souvenir) sont un ensemble d'applications qui veillent sur l'utilisateur et suggèrent des informations concernant la situation actuelle. Tandis que les moyens mnémoniques basés sur des requêtes aident à la remémoration directe, les agents de souvenir sont une mémoire associative étendue. Par exemple, la version du traitement de texte de RA met continuellement à jour une liste de documents en rapport avec ce qui est saisi ou lu dans un tampon d'emacs. Ces documents suggérés peuvent être n'importe quel fichier texte qui pourrait être en rapport avec ce que vous êtres en train d'écrire ou de lire. Ils pourraient être de vieux courriers électroniques liés au courrier que vous êtes en train de lire, ou des résumés de la recherche et des articles de journaux qui traitent du même sujet que ce que vous écrivez.

RoboTournament

- Site Web : robotournament.sourceforge.net/

RoboTournament est un jeu inspiré de RoboRally où les joueurs programment leurs robots pour vaincre leurs adversaires. Les fonctionnalités de RoboTournament sont : types de jeu multiples : match à mort, rallye et capture de drapeau. Multijoueurs via TCP/IP, six armes y compris le BFG, un éditeur de cartes, et une grande variété d'éléments de tableau de bord.

SimRobot

- Site Web : www.informatik.uni-bremen.de/simrobot/index_e.htm

- Site FTP : ftp.uni-bremen.de/pub/ZKW/INFORM/simrobot/

SimRobot est un programme de simulation de robots avec capteurs dans un environnement 3D. Ecrit en C++, il fonctionne sous Unix et X11 et requiert la boîte à outils graphique Xview.

- Simulation de la cinématique du robot
- Définition de scène construite hiérarchiquement par l'intermédiaire d'une langue de définition simple
- Divers capteurs intégrés : appareil photo, ½il à facettes, mesure de distances, capteur de luminosité, etc.
- Les objets sont définis comme des polyèdres
- Emetteur défini abstraitement : peut être interprété par exemple comme lumière ou bruit
- Images d'appareil-photo calculées selon les algorithmes de raytracing ou de Z-buffer que l'on connaît par l'infographie
- Interface logicielle de capteur/moteur spécifique pour communiquer avec la simulation
- Mappage de textures sur les surfaces de l'objet : bitmaps dans divers formats
- Visualisation complète de la scène : rendu en fil de fer sans lignes cachées, ni valeurs de capteur ou d'acteur
- Le contrôle des agents et l'opération dans l'environnement sont aussi bien interactifs que traités par lots
- Détection des collisions
- Capacité d'extension grâce à des types d'objet définis par l'utilisateur
- Communication par socket possible vers, par exemple, le logiciel de traitement d'image Khoros

Sulawesi

- Site Web : wearables.essex.ac.uk/sulawesi/

Un cadre de développement appelé Sulawesi a été conçu et implémenté pour aborder ce qui est considéré comme des défis importants dans les interfaces utilisateur tout usage [NdT : en anglais le terme wearable est utilisé, pour exprimer la possibilité d'un système que l'on peut emporter sur soi, intégré ou non aux vêtements] : la capacité d'accepter une entrée venant de n'importe quel nombre de modalités, et d'effectuer au besoin une traduction vers n'importe quel nombre de sorties modales. Cela est fait principalement au moyen d'un ensemble d'agents qui agissent par anticipation sur l'entrée.

TclRobots

- Site Web : www.nyx.net/~tpoindex/

TclRobots est un jeu de programmation, semblable à « Core War ». Pour jouer à TclRobots, vous devez écrire un programme en Tcl qui contrôle un robot. La mission du robot est de survivre à une bataille contre d'autres robots. Deux, trois ou quatre robots combattent pendant une bataille, chacun utilisant un programme différent (ou éventuellement le même programme dans différents robots). Chaque robot est équipé avec un radiomètre à balayage, un canon, et un mécanisme de déplacement. Un duel ne s'arrête que lorsqu'il ne reste qu'un seul robot opérationnel. Les robots peuvent combattre individuellement, ou par équipe. Un tournoi peut avoir lieu avec n'importe quel nombre de programmes de robot, chaque robot jouant contre tous les autres dans une poule, en face à face individuel. Un simulateur de bataille est disponible pour aider à la mise au point des programmes de robot.

Le programme TclRobots fournit un environnement physique, imposant certains paramètres de jeu auxquels tous les robots doivent adhérer. TclRobots fournit également une vue de la bataille, et une interface de contrôle pour l'utilisateur. Conditions pour utiliser TclRobots : un interpréteur "wish" conçu pour Tcl 7.4 et Tk 4.0.

TKQML

- Site Web : www.csee.umbc.edu/tkqml/

TKQML est une application / un ajout KQML à Tcl/Tk, ce qui permet aux systèmes basés sur Tcl de communiquer facilement avec un puissant langage de communication d'agent.

The Tocoma Project

- Site Web : www.tacoma.cs.uit.no/

Un agent est un processus qui peut migrer à travers un réseau informatique dans le but de satisfaire des requêtes émises par des clients. Les agents sont une manière attrayante de décrire les calculs distribués sur tout le réseau.

Le projet TACOMA se concentre sur le support des systèmes d'exploitation pour les agents et sur la manière dont les agents peuvent être utilisés pour résoudre des problèmes traditionnellement destinés aux systèmes d'exploitation. Nous avons implémenté une série de systèmes prototypes pour accueillir des agents.

TACOMA version 1.2 est basé sur UNIX et TCP. Le système est compatible avec des agents écrits en C, Tcl/Tk, Perl, Python, et Scheme (Elk). Il est implémenté en C. Cette version de TACOMA a été rendue publique en Avril 1996.

Nous nous concentrons actuellement sur les problèmes d'hétérogénéité, de tolérance aux anomalies, de sécurité et de gestion. En outre, plusieurs applications de TACOMA sont en construction. Nous avons implémenté StormCast 4.0, un vaste système en réseau de surveillance météorologique sur Internet, en utilisant TACOMA et Java. Nous sommes actuellement en cours d'évaluation de l'application, et nous prévoyons de construire une nouvelle version de StormCast qui devrait être achevée en Juin 1997.

Ummon

- Site Web : www.advogato.org/proj/Ummon/

Ummon est un robot conversationnel avancé et "open source". Le principe général du robot est qu'il ne possède aucune connaissance initiale en vocabulaire ou grammaire ; il apprend tout à la volée. De nombreuses techniques d'IA seront explorées au cours du développement d'Ummon pour arriver à une communication "humaine" réaliste avec le support de différentes personnalités paramétrables.

UMPRS Agent

- Site Web : www.marcush.net/IRS/

UMPRS supporte le mode de raisonnement descendant, avec objectifs à atteindre et il choisit ses objectifs et plan selon la priorité maximale. L'exécution de plusieurs buts simultanément est possible, ainsi que la suspension et reprise pour chaque tâche de but (c-à-d d'intention). Les plans d'UMPRS ont un attribut intégré de précondition/moteur d'exécution qui restreint leur applicabilité. Les plans de construction disponibles comprennent : l'ordonnancement, l'itération, la définition de sous-objectifs, les blocs atomiques (c-à-d non-interruptible), l'exécution conditionnelle déterministe à n-branches, une section de gestion de pannes explicite, et la définition de fonction primitive en C++.

Virtual Secretary Project (ViSe)

(Tcl/Tk)

- Site Web : www.vise.cs.uit.no/vise/

Ce qui motive le projet Secrétaire Virtuelle est la construction d'agents logiciels intelligents basés sur des modèles utilisateurs, qui pourraient dans la plupart des cas remplacer les hommes pour des tâches de secrétariat ; ces agents seraient basés sur l'informatique mobile moderne et les réseaux informatiques. Le projet comporte 2 différentes phases : la première phase (ViSe1) se concentre sur le filtrage des informations et la migration de processus, son but étant de créer un environnement sécurisé pour les agents logiciels en utilisant le concept des modèles utilisateur ; la seconde phase (ViSe2) est axée sur la coopération efficace et intelligente des agents dans un environnement distribué, son but étant de construire des agents coopératifs pour obtenir un niveau d'intelligence élevé. (codé en Tcl/TclX/Tix/Tk)

VWORLD

- Site Web : zhar.net/projects/vworld/

Vworld est un environnement simulé, écrit en prolog, pour la recherche avec des agents autonomes. Il est actuellement plus ou moins en version bêta. Il fonctionne bien avec SWI-prolog, et devrait fonctionner avec Quitnus-prolog, avec seulement quelques modifications. Il est destiné à être utilisé comme outil pédagogique pour des projets d'élèves traitant de prolog et des agents autonomes. Il est livré avec trois mondes ou environnements de démonstration, ainsi qu'avec des exemples d'agents qui leurs sont destinés. Il y a deux version maintenant. L'une d'elles est écrite pour SWI-prolog et l'autre pour LPA-prolog. Globalement, la documentation est faite (dans l'optique d'un cadre étudiant/professeur), et une interface graphique est prévue.

WebMate

- Site Web : www-2.cs.cmu.edu/~softagents/webmate/

WebMate est un agent personnel pour la recherche et la navigation sur le World-Wide Web. Il vous accompagne quand vous surfez sur Internet et vous fournit ce que vous désirez. Ses fonctionnalités comprennent :

- Amélioration des recherches, dont la recherche parallèle, affinage des mots-clés de la recherche en utilisant notre technologie d'extraction de mots-clés appropriés, retour d'information pertinente, etc.
- Assistant à la navigation, comprenant l'apprentissage de votre centre d'intérêt actuel, avec recommandation de nouveaux URLs selon votre profil et les ressources sélectionnées, gestion de vos signets de Netscape ou IE, envoie de la page actuellement affichée à vos amis, etc.
- Navigation hors ligne, y compris le téléchargement des pages suivant la page affichée pour une navigation hors ligne.
- Filtrage des en-têtes HTTP, comprenant l'enregistrement des en-têtes HTTP et de toutes les transactions effectuées entre le navigateur et les serveurs WWW, etc.
- Vérification de la page HTML et détection des erreurs et liens morts, etc.
- Programmation en Java, indépendant du système d'exploitation, fonctionnant en multiprocessus

Zeus

- Site Web : more.btexact.com/projects/agents/zeus/

La construction de systèmes multiagents implique une longue période de développement et exige des solutions à quelques difficultés techniques considérables. C'est ce qui a motivé le développement de la boîte à outils ZEUS, qui fournit une bibliothèque de composants logiciels et des outils qui facilitent la conception, le développement et le déploiement rapide de systèmes d'agent.

# Les langages de programmation

Bien que n'importe quel langage de programmation puisse être utilisé pour la recherche sur la vie ou l'intelligence artificielle, les langages de programmation décrits ci-dessous sont, si ils n'ont pas été spécifiquement créés pour cela, utilisés intensivement pour la programmation d'intelligence artificielle.

Allegro CL

- Site Web : www.franz.com/

Version linux gratuite de l'environnement de développement Lisp de Franz Inc. Vous pouvez la télécharger ou ils vous expédieront un CD gratuitement (vous n'avez même pas à payer de frais d'expédition). Elle est généralement considérée comme l'une des meilleures plateformes Lisp.

APRIL

- Site Web : sourceforge.net/projects/networkagent/ et www.nar.fujitsulabs.com/april/index.html

APRIL est un langage de programmation symbolique conçu pour le codage de systèmes basés sur des agents, mobiles et distribués dans l'environnement Internet. Il a des fonctionnalités avancées telles qu'un sous-langage de macro, l'envoi et la réception asynchrone de messages, la mobilité du code, le filtrage, des fonctions d'ordre supérieur et la déclaration typée de variables. Ce langage est compilé en pseudo-code binaire qui est ensuite interprété par le moteur d'exécution APRIL. APRIL a maintenant besoin qu'InterAgent Communications Model (ICM) soit installé avant de pouvoir être installé à son tour. [Ed. ICM est disponible sur le même site web]

Ciao Prolog

- Site Web : www.clip.dia.fi.upm.es/Software/Ciao/

Ciao est un système complet Prolog englobant ISO-Prolog avec une nouvelle conception modulaire qui permet à la fois la restriction et l'extension du langage. Parmi les extensions de Ciao, on trouve actuellement les arguments nommés (enregistrements), les fonctions d'ordre supérieur, les contraintes, les objets, les prédicats persistants, une bonne base pour l'exécution distribuée (d'agents) et la concurrence. Les bibliothèques supportent aussi la programmation WWW, les sockets, et les interfaces externes (C, Java, Tcl/Tk, les bases de données relationnelles, etc.). Un environnement pour Emacs, un compilateur autonome, et un interpréteur de commandes haut niveau sont également fournis.

DHARMI

- Site Web : megazone.bigpanda.com/~wolf/DHARMI/

DHARMI est un langage spatial de haut niveau facile à utiliser dont les composants sont administrés de manière transparente par un processus en tâche de fond appelé "the Habitat". Comme son nom l'indique, ce langage a été conçu pour faire des prototypes de modélisation et la gestion des données actives. Les programmes peuvent être modifiés pendant leur exécution. Cela est rendu possible en estompant la distinction entre le code source, le programme et les données.

ECLiPSe

- Site Web : eclipse.crosscoreop.com/eclipse/

:TODO:

ECoLisp

- Site Web : www.di.unipi.it/~attardi/software.html

ECoLisp ("Embeddable Common Lisp", Lisp Commun Embarquable) est une implémentation de "Common Lisp" conçue pour être insérable dans des applications écrites en C. ECL utilise des conventions d'appel C standard pour les fonctions Lisp compilées, ce qui permet aux programmes C d'appeler facilement des fonctions Lisp et vice versa. Aucune interface de fonctions externes n'est requise : les données peuvent être échangées entre le C et le Lisp sans besoin de conversion. ECL est basé sur un "Common Runtime Support (CRS)" (Appui d'Exécution Commun) qui fournit des fonctionnalités de base pour la gestion de la mémoire, le chargement et le vidage dynamique des images binaires, et le soutien de processus multiple d'exécution. Le CRS est intégré à une bibliothèque qui peut être liée au code de l'application. L'ECL est modulaire : les modules principaux sont les outils de développement du programme (niveau supérieur, débogueur, trace d'une exécution, exécution pas à pas), le compilateur, et CLOS. Une version native de CLOS est disponible dans ECL : on peut de configurer ECL avec ou sans CLOS. Une version d'exécution d'ECL peut être construite avec les modules requis par l'application seulement. Le compilateur ECL compile du Lisp vers le C, puis fait appel au compilateur GCC pour créer les binaires.

ESTEREL

- Site Web : www-sop.inria.fr/esterel.org/

Esterel est à la fois un langage de programmation, dédié à la programmation de systèmes réactifs, et un compilateur qui traduit les programmes Esterel en machine à états finis. Il est particulièrement bien adapté à la programmation de systèmes réactifs, y compris les systèmes temps-réel et les automates de commande.

Seul l'exécutable est disponible pour le compilateur de langage. :P

Gödel

- Site Web : www.cs.bris.ac.uk/~bowers/goedel.html

Gödel est un langage de programmation déclaratif à usage global, de la famille des langages de programmation de logique. C'est un langage fortement typé, le système de typage étant basé sur de la logique multitri avec du polymorphisme paramétrique. Il a un système de module. Gödel supporte les entiers à précision infinie, les nombres rationnels à précision infinie, et les réels à virgule flottante également. Il peut résoudre les contraintes sur des domaines finis d'entiers et également les contraintes rationnelles linéaires. Il supporte le traitement des ensembles finis. Il a également une règle de calcul flexible et un opérateur d'élagage qui généralise la validation des langages de programmation de logique concourante. Un accent considérable est mis sur l'infrastructure méta-logique de Gödel qui fournit un support significatif pour les méta-programmes qui font de l'analyse, de la transformation, de la compilation, de la vérification, du débogage, et ainsi de suite.

CLisp (Lisp)

- Site Web : clisp.sourceforge.net/
- Site Web officiel : clisp.cons.org/

CLisp est une implémentation de "Common Lisp" réalisée par Bruno Haible et Michael Stoll. Il supporte principalement le Lisp décrit par Common LISP: The Language (2nd edition) et le standard Common Lisp ANSI. CLisp comprend un interpréteur, un byte-compilateur, un grand sous ensemble de CLOS ("Object-Oriented Lisp", Lisp Orienté Objet), une interface de langage externe et, pour certaines machines, un éditeur d'écran.

On choisit la langue d'interface utilisateur (anglais, allemand, français) au moment de l'exécution. Les principaux paquetages qui fonctionnent dans CLisp incluent CLX et Garnet. CLisp ne requiert que 2 Mo de mémoire seulement.

CMU Common Lisp

- Site Web : www.cons.org/cmucl/
- Site Web : ww.telent.net/lisp/howto.html

CMU Common Lisp est un environnement de programmation Common Lisp de "puissance industrielle" de domaine public. Une grande partie des changements de X3j13 ont été incorporés dans CMU CL. Partout où cela était possible, ceci a été fait afin de permettre de manière transparente l'utilisation soit du CLtL1, soit du CL ANSI proposé. Les nouvelles fonctionnalités les plus intéressantes pour les utilisateurs sont probablement les fonctions SETF, LOOP et la macro WITH-COMPILATION-UNIT.

GCL (Lisp)

- Site FTP : ftp.ma.utexas.edu/pub/gcl/

GNU Common Lisp (GCL) possède un compilateur et un interpréteur pour Common Lisp. Il était connu auparavant sous le nom de Kyoto Common Lisp. Il est très portable et extrêmement efficace sur une large gamme d'applications. Il soutient la comparaison au niveau performance avec les Lisps commerciaux sur un grand nombre de démontreurs de théorèmes et de systèmes d'algèbre symbolique. Il supporte la spécification CLtL1 mais évolue vers la définition proposée de norme ANSI. GCL compile vers du C et utilise ensuite les compilateurs C d'optimisation de code natif (ex.: GCC). Une fonction comportant un nombre fixe d'arguments et une valeur se transforme en une fonction C du même nombre d'arguments, renvoyant une valeur, GCL a donc une efficacité maximale sur de tels appels. Il a un garbage collector [NdT : ramasse-miettes] classique qui offre une grande liberté au compilateur C pour mettre les valeurs Lisp dans des registres arbitraires.

Il possède un débogueur Lisp niveau source pour le code interprété, avec affichage du code source dans une fenêtre Emacs. Son outil de profilage (basé sur les outils de profilage C) compte les appels de fonction et le temps passé dans chaque fonction.

GNU Prolog

- Site Web : pauillac.inria.fr/~diaz/gnu-prolog/
- Site Web : www.gnu.org/software/prolog/prolog.html

GNU Prolog est un compilateur Prolog gratuit, et résolveur de contraintes sur les domaines finis, développé par Daniel Diaz. GNU Prolog accepte des programmes en Prolog+contraintes et produit des exécutables natifs (comme gcc le fait à partir de code source C). L'exécutable obtenu est alors autonome. La taille de cet exécutable peut être relativement petite puisque GNU Prolog peut éviter de lier le code de la plupart des prédicats intégrés inutilisés. Les performances de GNU Prolog sont très encourageantes (comparables à celles des systèmes commerciaux).

En plus de la compilation en code exécutable natif, GNU Prolog offre un interpréteur interactif classique (haut niveau) avec un débogueur.

La partie Prolog se conforme aux standards ISO pour Prolog avec beaucoup d'extensions très utiles à l'usage (variables globales, interface système d'exploitation, sockets, ...).

GNU Prolog comprend également un efficace résolveur de contraintes sur Domaines Finis (FD), ce qui ouvre la programmation logique par contraintes à l'utilisateur, en combinant la puissance de la programmation par contraintes à la déclarativité de la programmation logique.

IBAL

- Site Web : www.eecs.harvard.edu/~avi/IBAL/

:TODO:

lush

- Site Web : lush.sourceforge.net/

:TODO:

Maude

- Site Web : maude.cs.uiuc.edu/

:TODO:

Mercury

- Site Web : www.cs.mu.oz.au/research/mercury/

Mercury est un nouveau langage de programmation logique, purement déclaratif. Tout comme Prolog et d'autres langages de programmation logique existants, c'est un langage très haut niveau qui permet aux programmeurs de se concentrer sur le problème plutôt que sur les détails bas niveau tels que la gestion de la mémoire. Contrairement à Prolog, qui est orienté vers la programmation exploratoire, Mercury est conçu pour la construction de grands systèmes logiciels fiables et efficaces par des équipes de programmeurs. Par conséquent, programmer en Mercury apporte une touche différente par rapport à la programmation en Prolog.

Mozart

- Site Web : www.mozart-oz.org/

Le système Mozart fournit un support de pointe dans 2 domaines  : le calcul distribué ouvert et l'inférence basée sur les contraintes. Mozart implémente Oz, un langage concourant orienté objet avec synchronisation des flux de données. Oz combine la programmation distribuée et concourante avec l'inférence logique basée sur les contraintes, ce qui en fait un choix excellent pour le développement des systèmes multiagents. Mozart est une plateforme idéale aussi bien pour les applications distribuées polyvalentes que pour les problèmes difficiles nécessitant une optimisation sophistiquée et des capacités d'inférence. Nous avons développé des applications de planification et de gestion du temps, de placement et de configuration, de représentation du langage naturel et des connaissances, de systèmes multiagent et d'outils de collaboration sophistiqués.

SWI Prolog

- Site Web : www.swi-prolog.org/

SWI est une version gratuite de Prolog dans la famille Prolog Edinburgh (la rendant de ce fait très similaire à Quintus et à beaucoup d'autres versions), qui contient : une grande bibliothèque de prédicats intégrés, un système de modules, un garbage collector, une interface bidirectionnelle avec le langage C, et beaucoup d'autres dispositifs encore. Il est censé être un langage éducatif, donc son code compilé n'est pas le plus rapide. Bien que sa similarité avec Quintus permette un portage facile.

XPCE est librement et gratuitement disponible en exécutable pour la version Linux de SWI-Prolog. XPCE est un paquetage/environnement de développement d'interface graphique (GUI) X-window orienté objet.

Push

- Site Web : hampshire.edu/lspector/push.html

:TODO:

Kali Scheme

- Site Web : www.neci.nj.nec.com/PLS/Kali.html

Kali Scheme est une implémentation distribuée de Scheme qui permet une transmission efficace d'objets d'ordre supérieur tels que les fermetures et continuations. L'intégration d'équipements de communication distribuée dans un langage de programmation évolué engendre un certain nombre d'abstractions et de paradigmes nouveaux pour l'informatique distribuée. Parmi ces derniers nous avons les politiques de migration des processus et l'équilibrage de charge, personnalisés par l'utilisateur, les calculs distribués liés incrémentalement, les agents, et les applications client-serveur paramétrées. Kali Scheme supporte la simultanéité et la communication, grâce aux procédures et continuations de première classe. Il intègre les procédures et continuations dans un cadre distribué basé sur des messages qui permet à n'importe quel objet Scheme (y compris les vecteurs de code) d'être envoyé et reçu dans un message.

RScheme

- Site Web : www.rscheme.org/
- Site FTP : ftp.rscheme.org/pub/rscheme/

RScheme est une version orientée objet étendue du dialecte Scheme de Lisp. RScheme est librement redistribuable, et il offre des performances raisonnables en dépit de son extraordinaire portabilité. RScheme peut être compilé en C, et le C peut ensuite être compilé avec un compilateur C habituel pour générer du code machine. Cependant, par défaut, RScheme compile en code objet qui est interprété par une machine virtuelle (un moteur d'exécution). Cela assure une compilation rapide et permet de conserver un code de taille réduite. En général, nous recommandons l'utilisation du système de génération de code objet (activé par défaut), et de ne compiler en code machine que vos programmes dont le temps d'exécution est critique. Cela permet un bon compromis taille mémoire/temps d'exécution. (voir le site web pour plus de détails)

Scheme 48

- Site Web : s48.org

Scheme 48 est une implémentation de Scheme basée sur une architecture de machine virtuelle. Scheme 48 est conçu pour être simple, souple, fiable, et rapide. Il devrait être facilement portable vers des machines 32 bits qui ont POSIX et le support du C ANSI. En plus de l'environnement de développement et des procédures usuelles intégrées de Scheme, les logiciels de bibliothèques incluent le support des macros hygiéniques (comme cela est décrit dans le rapport Scheme Revised^4), du multitâche, des enregistrements, de la gestion des exceptions, des tables de hachage, des tableaux, des pointeurs nuls, et du FORMAT. Scheme 48 implémente et exploite un système expérimental de modules vaguement dérivé de Standard ML et de Scheme Xerox. L'environnement de développement supporte les changements interactifs de modules et d'interfaces.

SCM (Scheme)

- Site Web : swissnet.ai.mit.edu/~jaffer/SCM.html

SCM est conforme à la révision 5 du rapport sur l'Algorithmic Language Scheme (le langage algorithmique Scheme) et à la spécification IEEE P1178. SCM est écrit en C. Il utilise les utilitaires suivants (tous disponibles sur le site ftp) :

- SLIB (Standard Scheme Library) est une bibliothèque Scheme portable qui est prévue pour fournir des fonctions de compatibilité et des utilitaires pour toutes les implémentations Scheme standard, y compris SCM, Chez, Elk, Gambit, MacScheme, MITScheme, scheme->C, Scheme 48, T3.1, et VSCM, et est disponible sous le nom de fichier slib2c0.tar.gz. Ecrit par Aubrey Jaffer ;
- JACAL est un système symbolique de mathématiques écrit en Scheme, et est disponible sous le nom de fichier jacal1a7.tar.gz ;
- Des interfaces vers les bibliothèques standards comprenant un filtrage d'expressions régulières de chaînes de caractères REGEX et le paquetage de gestion d'écran CURSES ;
- Sont disponibles pour être ajoutés des paquetages parmi lesquels un débogueur interactif, une base de données, des composants X-window, des traitements graphiques BGI, Motif, et des paquetages Open-Windows ;
- Un compilateur (HOBBIT, disponible séparément) et la liaison dynamique des modules compilés.

Shift

- Site Web : www.path.berkeley.edu/shift/

Shift est un langage de programmation permettant de décrire les réseaux dynamiques d'automates hybrides. De tels systèmes sont constitués de composants qui peuvent être créés, interconnectés et détruits au fur et à mesure que le système évolue. Les composants montrent un comportement hybride, constitué de phases continues séparées par des transitions à événements discrets. Les composants peuvent évoluer indépendamment, ou ils peuvent interagir par leurs entrées, leurs sorties et les événements exportés. Le réseau d'interaction lui-même peut évoluer.

YAP Prolog

- Site Web : www.ncc.up.pt/~vsc/Yap/

YAP est un compilateur Prolog à haute performance développé par LIACC/Universidade do Porto. Son moteur Prolog est basé sur WAM (Warren Abstract Machine, Machine Abstraite Warren), avec plusieurs optimisations pour de meilleures performances. YAP suit la tradition d'Edinburgh, et est en grande partie compatible avec Prolog DEC-10, Quintus Prolog, et particulièrement avec le C-Prolog. Le travail sur la version la plus récente de YAP tends vers plusieurs objectifs :

- la portabilité : le système tout entier est maintenant écrit en C. YAP compile sur les machines 32 bits les plus courantes, tels que les Sun et les PC sous Linux, et sur les machines 64 bits, les Alphas sur lesquelles tournent OSF Unix et Linux.

- la performance : nous avons optimisé l'émulateur pour obtenir des performances comparables voire meilleures que celles des systèmes Prolog les plus connus. En réalité, la version actuelle de YAP est plus puissante que l'originale, écrite en assembleur.

- la robustesse : nous avons testé le système avec un grand échantillonnage d'applications Prolog.

- l'extensibilité : YAP a dès l'origine été conçu structurellement pour encapsuler la manipulation de termes. Ces principes étaient utilisés, par exemple, pour implémenter une interface C simple et puissante. La nouvelle version de YAP étend ces principes pour adapter les extensions à l'algorithme d'unification, qui, selon nous, sera utile pour l'implémentation d'extensions telles que la programmation de contraintes.

- la complétude : YAP fournit depuis longtemps la plupart des installations intégrées que l'on attend d'une implémentation Prolog Edinburgh. Celles-ci comprennent les fonctionnalités d'entrées/sorties, les opérations de bases de données, et les modules. Le travail sur YAP a maintenant pour objectif d'être compatible avec le standard Prolog.

- l'ouverture : nous voudrions faire en sorte que les nouveaux développements de YAP soient ouverts à la communauté des utilisateurs.

- la recherche : YAP est un projet moteur pour la recherche dans et en dehors de notre groupe. Actuellement la recherche continue sur le parallélisme et la tabulation, et nous avons commencé à travailler sur la prise en compte de la manipulation des contraintes.

# Manquant et Inactif

Ceci est l'espace pour les entrées anciennes ou erronées. Le chapitre MIA est destinée aux entrées pour lesquelles je n'ai plus de page d'accueil valide. Si vous avez la moindre information sur l'emplacement actuel de ces logiciels, merci de me le faire savoir. La section Inactifs est destinée aux projets qui ne semblent plus maintenus. Les déplacer ici me permet de conserver les chapitres principaux propres ; les parties intéressées peuvent malgré tout me corriger, auquel cas je les déplacerai à nouveau.

## MIA - Projets ne disposant pas de liens.

:TODO:

IDEAL

- Site web : yoda.cis.temple.edu:8080/ideal/

IDEAL est un banc d'essai pour le travail sur les diagrammes d'infuence et les réseaux Bayésiens. Il comprend des algorithmes d'inférence variés pour les réseaux de croyance et des algorithmes d'évaluation pour les diagrammes d'inluence. Il contient des installations pour créer et éditer les diagrammes d'influence et les réseaux de croyance.

IDEAL est écrit purement en Common Lisp et ainsi il tournera en Common Lisp sur n'importe quelle plate-forme. L'accent dans l'écriture d'IDEAL a été mis sur la clarté du code et sur la fourniture d'abstractions de programmation de haut niveau. Il est ainsi tout à fait adapté aux implémentations expérimentales qui nécessitent ou étendent la technologie de réseau de croyance.

Au plus haut niveau, IDEAL peut être utilisé comme une bibliothèque de sous-routine qui fournit l'évaluation du diagramme d'inférence et d'influence du réseau de croyance dans un paquetage logiciel. Le code est documenté dans un manuel détaillé et ainsi il est aussi possible de travailler à un niveau inférieur sur les extensions des méthodes du réseau de croyance.

IDEAL est fourni avec une interface graphique optionnelle écrite en CLIM. Si votre Common Lisp possède aussi CLIM, vous pouvez lancer l'interface graphique.

Symbolic Probabilistic Inference (SPI)

- Site FTP : ftp.engr.orst.edu/pub/dambrosi/spi/
- Paper (ijar-94.ps) : ftp.engr.orst.edu/pub/dambrosi/

Contient des bibliothèques de fonction Common Lisp pour implémenter des réseaux bayésiens de type SPI. La documentation est très limitée. Caractéristiques :

- Probabilités, Utilitaires du langage d'expression locale, explications, modèles dynamiques, et une GUI basée sur TCL/TK.

## Projets inactifs.

EMA-XPS - A Hybrid Graphic Expert System Shell

- Site web : www.iai.uni-wuppertal.de/EMA-XPS

EMA-XPS est un interpréteur de commande [shell] de système expert graphique hybride basé sur le shell orienté ASCII Babylon 2.3 du Centre National Allemand de Recherche pour les Sciences Informatiques (GMD). En plus de la puissance d'IA de Babylon (représentation de données orientées objet, règles de chaînage avant et arrière - regroupables en séries, en clauses de Horn, et en réseaux de contraintes), une interface graphique basée sur le système X11 Window System et la OSF/Motif Widget Library ont été fournies.

PDKB

- Site Web : lynx.eaze.net/~pdkb/web/
- Site SourceForge : sourceforge.net/projects/pdkb/

La banque de connaissances sur le domaine publique (PDKB : Public Domain Knowledge Bank) est une banque de connaissances sur l'intelligence artificielle des lois et règles de sens commun. Elle est basée sur l'ontologie de niveau supérieur de Cyc (CUO : Cyc Upper Ontology) et le langage MELD.

QUANT1

- Site Web : linux.irk.ru/projects/QUANT/

Quant/1 signifie QUANTifieur de type 1. Son but est d'être une alternative pour les systèmes semblables à Prolog (ou résolutionnaux). Les principales caractéristiques comprennent la non-nécéssité d'éliminer les quantifieurs, la scolémisation, la facilité de compréhension, les opérations sur des formules de grande échelle, l'acceptation de fomules 'nonHorn', l'approfondissement iterratif. La bibliothèque réelle implémentée dans ce projet est appelée ATPPCF (Automatic Theorem Prover in calculus of Positively Constructed Formulae : preuve automatique du théoreme dans le calcul de formules construites positivement)

ATPPCF deviendra une bibliothèque (moteur d'inférence) et une extension du langage de calculs de prédiction comme un nouveau langage logique. Cette bibliothèque sera incorporable dans d'autres programmes, comme TCL, Python, Perl. La méthode d'inférance primaire du moteur sera la "recherche d'inférances dans le langage des formules construites positivement (PCF : Positively Constructed Formulas)" (comme un sous-ensemble de calculs de prédicats bien traduits dans les deux sens). Le langage sera utilisé comme langage de script pour le moteur. Mais il y aura la possibilité de le remplacer par des extension de langage du programme principal.

TresBel

- Site FTP : iridia.ulb.ac.be/pub/hongxu/software/

Bibliothèques comprenant du code Common Lisp (Allegro) pour les fonctions de croyance (c-à-d le raisonnement probant de Dempster-Shafer) comme la représentation de l'incertain. Documentation pauvre. Possède une GUI limitée.