

Introduction à la Recherche d'Information à base d'Agents

Bernard ESPINASSE

Université Paul Cézanne - LSIS UMR CNRS 6168

2008

- **1. Introduction aux agents logiciels**
- **2. Une typologie des agents logiciels (Nwana)**
- **3. Problématique de la RI sur le Web**
- **4. Pourquoi les agents pour la RI**
- **5. Les grandes approches de la RI à base d'agents**

Références bibliographiques (1)

Cours :

- **Drogoul A.** Cours "Intelligence collective, systèmes multi-agents et IAD", Université Paris 6 - LIP6.
- **Gleize M.P.**, Cours "Intelligence collective", Université de Toulouse, IRIT.
- **Nwama H.**, "Software Agents", Knowledge Engineering Review, vol. 11, N°3, pp. 1-40, sept 96, Cambridge University Press.
- **Esfandiari B.**, "Software Agents" Course, University of Carleton, Canada.
- **Chaïb-Draa B.**, Cours "Systèmes multi-agents", Université Laval, Québec, Canada.
- **Finin T. & Labrou Y.**, Tutorial "Agents Communication Languages", ASA/MA.
- **Finin T. & Nicholas C.**, Tutorial "Software Agents for Information Retrieval", CIKM2000.

Articles :

- ...

Références bibliographiques (2)

Livres :

- **Weiss G. - editor (00)**, **Multiagent Systems**, MIT Press.
- **Ferber J. (95)**, **Les systèmes multi-agents**, InterEditions.
- **Singh M. (94)**, **Multiagent Systems**, Springer Verlag.
- **Conte R., Castelfranchi C. (1995)**, **Cognitive and Social Action**, UCL Press.
- **Haddadi A. (95)**, **Communication and Coopération in Agent Systems**, Springer Verlag.
- **O'Hare G.M.P. & Jennings N.R. - editors (96)**, **Foundations of Distributed Artificial Intelligence**, Wiley-Interscience.
- **Bradsham M. - editor (97)**, **Software Agents**, AAAI Press - The MIT Press.
- **Huhns M.N. & Singh M.P. - editors (97)**, **Readings in Agents**, Morgan-Kaufmann.
- ...

Plan

- **1. Introduction aux agents logiciels**
- **2. Une typologie des agents logiciels (Nwana)**
- **3. Problématique de la RI sur le Web**
- **4. Pourquoi les agents pour la RI**
- **5. Les grandes approches de la RI à base d'agents**

1 - Introduction

- De l'IA à l'IAD et aux agents logiciels
- Motivation de l'IAD
- Qu'est-ce qu'un agent logiciel

De l'IA à l'IAD et aux agents logiciels

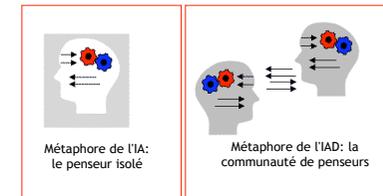
- IAD s'intéresse aux domaines pour lesquels sont inappropriés :

- un seul résolveur de problème
- une seule machine
- ou un seul lieu de traitement

IA: modéliser le savoir des agents (compétence)

+ Distribution : modéliser leurs interactions (organisation sociale)

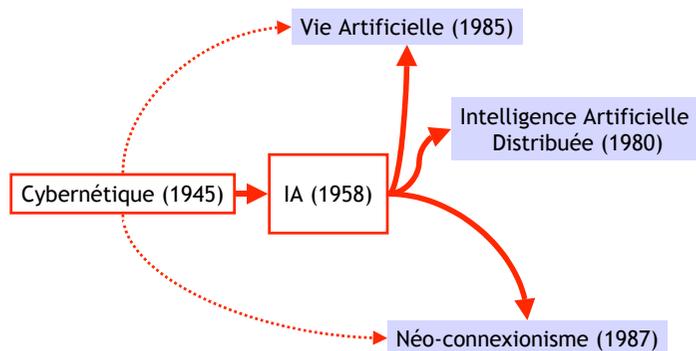
- la connaissance est à la fois **cognitive** et **sociale**



IAD s'intéresse à des systèmes dans lesquels des agents artificiels opèrent collectivement et de façon décentralisée pour accomplir une tâche

Emergence de l'IAD

dans les années 80, émergence de 3 domaines majeurs :



Motivations de l'IAD

Domaines à connaissances multiples et limites des systèmes experts (1960):

- construction de programme informatique capable d'exécuter des tâches complexes
- concentration de l'intelligence dans un système unique
- base de connaissances + moteur d'inférences

=> difficultés :

- expertise, compétences d'individus différents qui, dans la réalité
- communiquent et collaborent à la réalisation d'un but commun

Evolution des technologies :

- multiprocesseurs
- machines parallèles
- machines distribuées et réseaux de machines
- techniques et langages concurrents

Motivations de l'IAD

Problèmes réels complexes et ouverts :

- géographiquement ou logiquement distribués
- asynchrones
- changements continus
- autonomie locale
- interdépendance
- inconsistance

Modélisation de systèmes sociaux ou naturels

Compréhension des interactions entre les humains, simulations comportementales

=> distribuer l'intelligence" dans différentes entités
=> système multi-expert
=> IAD, SMA

L'IAD : un domaine pluridisciplinaire

Domaines de l'informatique concernés :

- intelligence artificielle distribuée (IAD) et systèmes multi-agents (SMA)
- vie artificielle (métaphore biologique)
- réseau de neurones

Autres disciplines concernées :

- psychologie sociale
- sociologie
- dynamique des groupes
- systémique
- biologie
- éthologie
- ergonomie
- ...

Problèmes et concepts majeurs de l'IAD

Les problèmes majeurs de l'IAD :

- décomposition et allocation de tâches
- interaction et communication
- modélisation des autres
- coordination / cohérence
- reconnaître et réduire les différences
- implémentation et évaluation

Les concepts majeurs de l'IAD :

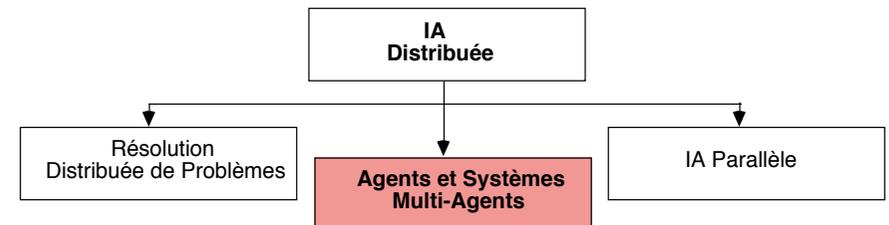
- l'interaction
- l'organisation
- la coopération
- la communication
- la représentation des autres et de soi

...De l'IAD aux Agents Logiciels

A l'origine des agents : les acteurs

Carl Hewitt's actor model (1977)

"a self-contained, interactive and concurrently-executing object, with some encapsulated internal state and which could respond to messages from other similar objects"



Qu'est ce qu'un agent logiciel (software agent)?

Définitions :

- un système informatique, **situé** dans un environnement, et qui **agit** d'une façon **autonome** et **flexible** pour atteindre des **objectifs** pour lequel il a été conçu [Sycara & Wooldridge]
- une entité computationnelle **active**, avec une **identité persistente**, qui peut **percevoir**, **raisonner** et **agir** dans son **environnement** et qui peut **communiquer** notamment avec d'autres agents [Huhns]
- un composant **logiciel** ou **matériel** qui est capable **d'agir** afin d'accomplir des **tâches** au nom de son **utilisateur** [Nwana]

Diversité des agents logiciels

Plusieurs appellations liées à une **apparence** ou à un **rôle** spécifique :

- **knowbots** (i.e. knowledge-based robots) : ont une connaissance experte dans un domaine particulier (personal assistants, ...)
- **softbots** (software robot)
- **taskbots** (task-based robots) : réalisent des tâches spécifiques
- **userbots**
- **robots** : habitent le monde physique
- **personal agents, personal assistants** : agents assistant un utilisateur spécifique, ...

Mais aussi :

search agents, report agents, presentation agents, navigation agents, role-playing agents, management agents, search and retrieval agents, domain-specific agents, development agents, analysis and design agents, testing agents, packaging agents and help agents

Caractéristiques générales des agents

situé : capable d'agir sur son **environnement** à partir d'**entrées sensorielles** qu'il en reçoit

autonome : capable **d'agir** sans l'intervention d'un tiers (humain ou agent) et de **contrôler** ses propres **actions** ainsi que son **état interne** (autonomie limitée à son comportement dans une société d'agents)

flexibilité :

- **réactif** : capable de percevoir son environnement et de réagir à temps
- **proactif** : capable de prendre l'initiative et être opportuniste au bon moment
- **social** : capable d'interagir avec les autres agents quand la situation l'exige (pour réaliser ses tâches ou coopérer avec eux)

Domaines d'application des SMA

- **Systèmes de production** : ordonnancement d'ateliers, conduite de processus industriels, systèmes multi-capteurs, ...
- **Diagnostic** : diagnostic à multiples niveaux
- **Taches de contrôle** : contrôle du trafic routier, trafic aérien, distribution d'énergies, ...
- **Taches d'interprétation** : interprétation de signaux, reconnaissance de la parole, cristallographie, reconnaissance et compréhension des formes, ...
- **Télécommunications, systèmes de transports, réseaux** : routage, équilibrage de charges, recouvrement d'erreurs, management et surveillance de réseaux, ...
- **Travail collaboratif assisté par ordinateur** : agents assistants, agents médiateurs, workflows, gestion des rendez-vous, personal digital assistants (PDA), ...
- **Robotique distribuée** : planification multi-robot, robots autonomes mobiles, ...
- **Télématique (Internet)** : agents "intelligents", agents d'interface, agents mobiles, ...
- **Simulation de systèmes complexes** : simulation individu-centrée, ...
- **Commerce électronique**
- **Data Mining, ...**

"in 10 years time most new IT development will be affected, and many consumer products will contain embedded agent-based systems" [Guilfoyle 95]

2 – Typologie d'Agents

[Nwana 99]

- Critères de classification
- Attributs majeurs de agents
- Types d'agents
- Attributs secondaires d'un agent

Critères de classification

Agents mobiles / statiques :

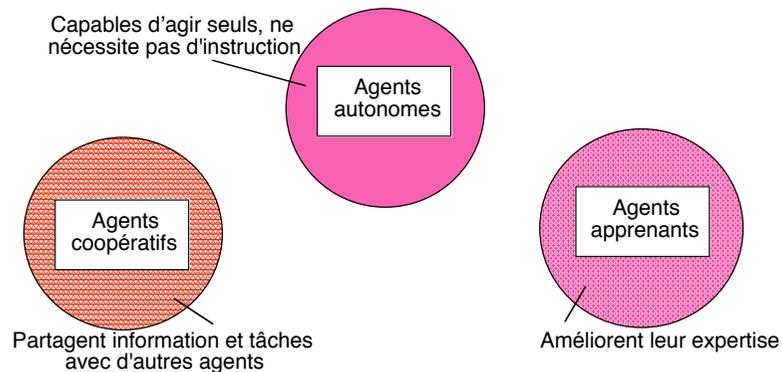
- agent **mobiles** : agents capables de se **déplacer sur un réseau**
- agents **statiques** : agents **pas capables de se déplacer sur un réseau**

Agents collaboratifs / réactifs :

- agents **collaboratifs** :
 - possédant un **modèle de raisonnement symbolique interne**,
 - se **coordonnent** entre eux par **planification** ou **négociation**
- agents **réactifs** :
 - ne possèdent **pas de modèle de raisonnement symbolique interne**,
 - agissent par **stimulis - réponse**
 - ont un **comportement répondant à l'état courant de l'environnement** dans lequel ils sont immergés

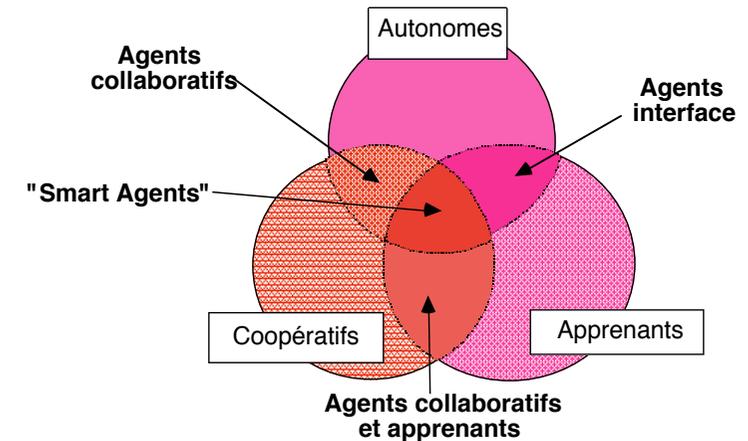
Attributs majeurs des agents [Nwana 99]

- **autonomie** : agents **autonomes**
- **coopération** : agents **coopératifs**
- **apprentissage** : agents **apprenants**



Types d'agents [Nwana 99]

Combinaison des 3 attributs majeurs : 6 types d'agents



Typologie d'agents [Nwana 99]

Par combinaison critères x attributs majeurs, on peut avoir :

- agents **collaboratifs délibératifs statiques**
- agents **collaboratifs réactifs et mobiles**
- agents **interface délibératifs statiques**
- agents **interface réactifs mobiles, ...**

Classification selon le **rôle** de l'agent :

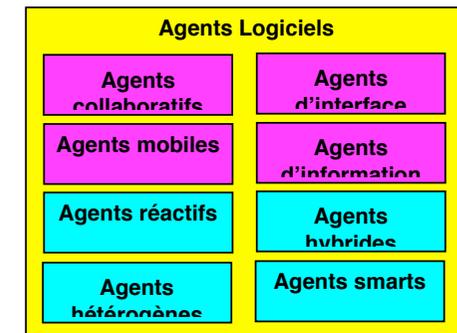
- **agents information ou internet** : en utilisant des moteurs de recherche sur internet comme WebCrawlers, Lycos and Spiders, ils aident à gérer une somme importante d'information
- **agents présentation**
- **agents d'analyse et de conception**
- **agents de test**
- **agents d'emballage**
- **agents d'aide, ...**

Classe des Agents hybrides : combinent plusieurs classes.

Types d'agents [Nwana 99]

Principaux types d'agents :

- **Collaborative agents**
- **Interface agents**
- **Mobile agents**
- **Information/Internet agents**
- **Reactive agents**
- **Hybrid agents**
- **Smart Agents**



Attributs secondaires d'un agent

- la **véracité** : il ne communique pas de mauvaises informations sans le savoir
- le **bénévolat** : il n'a pas de buts incompatibles et essaiera de faire ce qu'on attend de lui
- la **rationalité** : il agira de sorte à atteindre ses objectifs, au moins dans la limite de ses convictions
- la **versatilité** : il poursuit plusieurs buts et il s'engage dans de plusieurs tâches
- la **persistance** : il est temporellement continu ou il se dégrade au fil du temps
- les **attitudes émotionnelles** : en a-t-il assez d'être sollicité ? quel rôle a l'émotion dans l'élaboration d'agents (Bates, 1994)?
- les **attitudes mentales** : a-t-il des attitudes mentales comme des croyances, des convictions, des désirs et des intentions (BDI agents - Rao & Georgeff 95)

Agents collaboratifs

- **autonomes** et **coopèrent** entre eux
- **négocient** avec les autres agents en vue d'atteindre des ententes lors de la résolution distribuée de problèmes
- possèdent des **capacités d'apprentissage limitées**

Finalités :

- résoudre des **problèmes** :
 - "**naturellement**" **distribués** (réseaux de capteurs distribués (DVMT - Durfee et al., 1987), control aérien, ...)
 - "**trop grands**" ou "**trop complexes**" pour un seul agent
 - associés à des **sources d'information** ou **d'expertise distribuées**
- permettre **interconnectivité** et **interopérabilité** de **systèmes existants** (legacy systems)
- accroître :
 - la **modularité** (et ainsi réduire la complexité),
 - la **vitesse** (par parallélisme),
 - la **fiabilité** (par redondance),
 - la **flexibilité** (à partir d'une organisation plus modulaire) et
 - la **réutilisabilité** au niveau de la connaissance (donc partageabilité des ressources)
- **autres finalités** : la **compréhension d'interactions** dans des sociétés humaines

Agents collaboratifs : systèmes, théories et modèles

Quelques systèmes connus :

- **PLEIADE** : [Sycara et al. 1995]
- **DVMT** : [Durfee et al. 1987]
- **MACE** : [Gasser et al. 1987]
- **ADEPT** : [O'Brien et al. 1996]
- **GRATE/GRATE*** : [Jennings 1993]
- **ARCHON** project [Wittig, 1992; Jennings *et. al.*, 1993], ...

Théories et modèles :

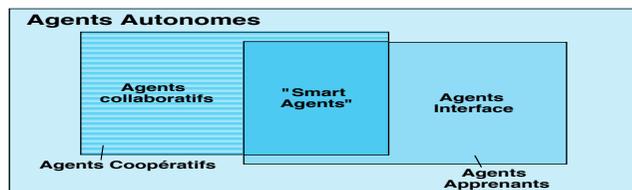
- **Théorie raisonnable de l'intention de Cohen & Levesque (90)**
- **Architecture BDI** (Rao & Georgeff 92) : caractérisation d'un agent rationnel en terme d'attitudes mentales de Croyances, Désirs et Intentions (Beliefs, Desires and Intentions) - usage de **logiques épistémiques**
- **AOP (Agent Oriented programming - Shoham 93)** : un état mental d'un agent est décrit par ses croyances, décisions, capacités et obligations - langage basé sur des opérateurs épistémiques et modaux.
- **Autres contributions** sur les attitudes mentales : IRMA [Bratman et al. 88], GRATE/GRATE* [Jennings 93], ...

Agents collaboratifs : principaux défis [Nwana]

- **disposer d'une véritable ingénierie** : nécessité de méthodes et d'outils de conception et d'implémentation de systèmes à base d'agents collaboratifs [Smith 96b]
- **importance de la coordination entre agents** : nécessité de disposer de véritables théorie de la coordination évitant anarchie et blocage (travaux formels et expérimentaux encore indispensables)
- **importance de la stabilité**, la "**scalability**" et de la **performance** : demande encore beaucoup de travaux de recherche
- **systèmes existants** (legacy systems): besoin de définir des techniques et méthodologies pour intégrer agents et systèmes existants
- **apprentissage** : comment ces systèmes apprennent ? l'apprentissage ne conduit-il pas à l'instabilité ? quelles sont les architectures appropriées pour les différents types de problèmes? est-on sûr que les agents ne passent pas plus de temps à apprendre qu'à réaliser ce pour quoi ils sont fait?
- **évaluation des systèmes d'agents collaboratifs** : pas encore abordé, comment vérifier et valider leurs spécifications fonctionnelles ?

Agents interface

- accent sur l'**autonomie**, l'**apprentissage** dans l'exécution de tâches pour des utilisateurs
- fournissent une assistance **proactive** à l'utilisateur pour une application précise
- leurs interactions avec les autres agents sont en général limitées



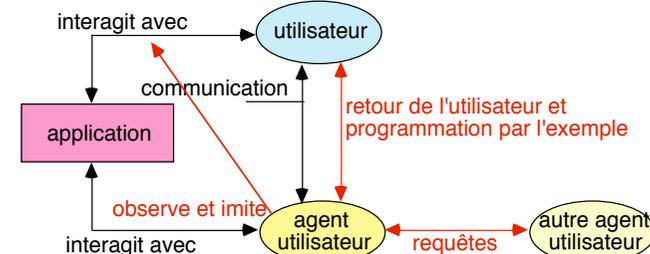
Agents interfaces = assistant personnel collaborant avec l'utilisateur

Finalités :

- ils apportent une **assistance à un utilisateur** : par ex. dans l'apprentissage de l'usage d'une application particulière (un tableur, un OS (Unix), ...)
- ils **déchargent l'utilisateur** et le développeur de tâches laborieuses
- ils **peuvent s'adapter** aux habitudes de l'utilisateur et à ses préférences
- ils peuvent aider plusieurs **utilisateurs à se partager leur savoir-faire**

Fonctionnement des agents interfaces: [Maes 94]

- il **observe** et **surveille** les actions de l'utilisateur dans l'interface (regarde par dessus son épaule), apprend de nouveaux raccourcis et suggère de meilleure façon de réaliser une tâche
- il agit comme un assistant personnel autonome qui **coopère** avec l'utilisateur en accomplissant certaines tâches dans l'application



- il **apprend** à mieux assister un utilisateur de différentes façons :
 - en **observant** et en **imitant** l'utilisateur
 - en recevant un **retour** positif ou négatif de l'utilisateur
 - en recevant des **instructions explicites** de l'utilisateur
 - en **demandant des conseils** à d'autres agents

Applications des agents interfaces (1) [Maes 94]

- **assistants** [Kozierok & Maes 93] :
 - l'agent **Calendar** assiste son utilisateur dans la planification de réunions impliquant acceptations, rejets, ordonnancement, négociation et réordonnement des horaires
 - il apprend ses préférences et ses engagements (elle n'aime pas les réunions le mercredi, il les préfère le matin, ...)
- **guides** [Liebermann 95] :
 - l'agent **Letizia** apporte une assistance dans la recherche sur le Web en conduisant, en coopération avec l'utilisateur, une recherche en largeur d'abord concurrente pour des sites pouvant intéresser celui-ci
 - il découvre ses intentions à partir de son comportement
- **aide-mémoire** [Rhodes & Starner 96] :
 - l'agent **Remembrance**, associé à l'éditeur Emacs,
 - assiste un utilisateur dans la rédaction de documents (e-mail, articles, rapports, ...) en pouvant lui ramener sur un mot clé donné, les documents les plus pertinents que l'utilisateur a déjà dans la mémoire de son ordinateur

Applications des agents interfaces (2) [Maes 94]

- **filtrage** [Sheth & Maes 93]:
 - l'agent de filtrage de nouvelles **NewT** aide l'utilisateur à filtrer et sélectionner des nouvelles dans un flux continu de nouvelles
 - il s'agit de créer des agents spécialisés (sports, finance, ...) et de les entraîner par renforcement en exploitant le contenu, les mots clés, l'auteur et la source (algorithme génétiques)
- **assistant d'achat/vente** [Chavez & Maes 96] :
 - **Kasbah** représente une place de marché (un site web) où des agents agissent pour leur propriétaires,
 - peuvent filtrer la annonces et retenir celles qui intéressent leur propriétaires et négocier, acheter et vendre des articles
- ...

Agents interface : défis majeurs [Nwana]

- **efficacité** :

démontrer que la connaissance apprise avec des agents interface peut vraiment réduire la charge de travail d'utilisateur
- **techniques d'apprentissage** :

déterminer **quelles** sont les techniques d'apprentissage qui sont préférables, pour **quels** domaines, et **pourquoi**
- **garantir la «vie privée»** des utilisateurs
- **capacité des agents** :

étendre les agents interface afin qu'ils soient capables de **négocier** avec d'autres agents
- **domaines d'application** :

étendre si possible le champ d'application à d'autres domaines innovants comme les **loisirs et la maison**

Agents Mobiles et pour Internet

l'agent autonome possédant une **identité** capable de **voyager sur un réseau**

agent mobile = unité logicielle capable de :

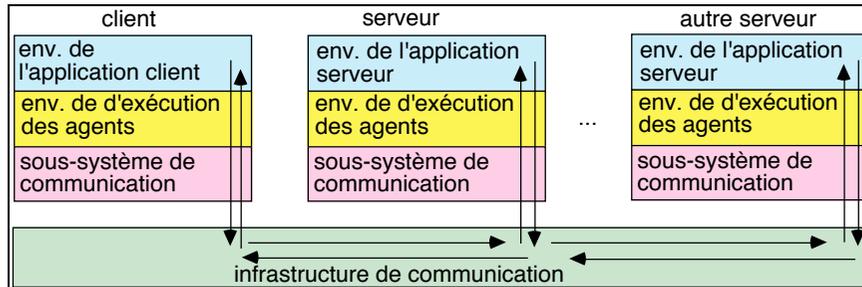
- **errer dans des réseaux** (locaux ou globaux comme le WWW)
- **migrer** d'une machine à une autre
- **interagir** avec différentes machines
- **collecter de l'information** pour son propriétaire
- **revenir "à la maison"** en ayant fait le travail fixé par son propriétaire

Finalités :

- **réduction de la charge du réseau** :
 - ils consomment moins de ressources du réseau en déplaçant le calcul vers les données au lieu de l'inverse
 - ils ne nécessitent pas une connexion continue entre machines
- interaction asynchrone : ils peuvent remplacer des communications (ex : documents mobiles)
- paradigme de programmation intéressant : ils cachent les canaux de communication mais pas la localisation de l'exécution et facilitent ainsi le déploiement d'application réparties

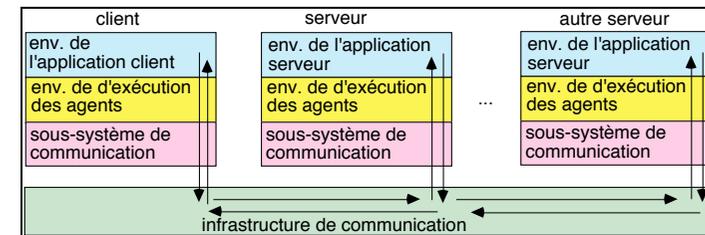
Fonctionnement des agents mobiles

revient à une **migration de process** déplaçant l'agent d'un client qui émet une requête (pour information, transaction ou mail) vers un serveur capable de satisfaire la requête :



- 1- **l'application** passe des infos à l'environnement d'exécution des **agents** via l'**API**
- 2 - **initialisation d'une instance d'agent** (process, thread)
- 3 - l'agent **exécute une instruction créant un nouveau process fils** s'exécutant dans un message exprimé sous une forme indépendante de la machine

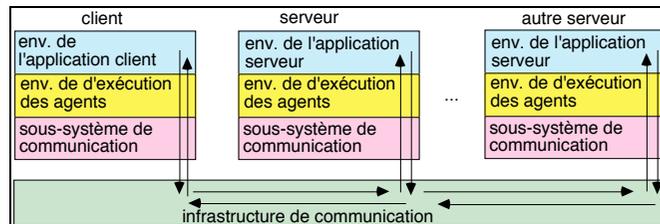
Agents mobiles : fonctionnement (2)



3 - l'agent **exécute une instruction créant un nouveau process fils** s'exécutant dans un message exprimé sous une forme indépendante de la machine :

- le **message envoyé** à sa destination finale ou à des destinations intermédiaires prenant en charge le transfert selon son contenu ("Semantic Routing")
- le **message arrive au serveur** destinataire où il est livré à l'environnement d'exécution de l'agent par le sous-système de messagerie du serveur
- dans l'environnement d'exécution de l'agent, le message reçu est **reconstitué en un exécutable** (process ou thread)
- **l'exécution du programme agent client se poursuit** avec l'instruction suivante

Agents mobiles : fonctionnement (3)



- 4 - pendant son exécution sur le serveur, l'agent peut :
 - **terminer** son exécution
 - **passer l'information** qu'il a reçu de l'application client à l'application serveur
 - effectuer une **action de récupération de données** sur le serveur
 - **déterminer** s'il doit **rendre visite à un autre serveur** :
 - si le service nécessaire n'est pas accessible ou mal adapté
 - en fonction les données qu'il a reçues du serveur sur lequel il se trouve
 - **répéter le processus de migration** en lançant un nouveau processus fils pouvant ramener l'agent à son client d'origine ou l'envoyer à un autre serveur client
 - se **suspendre** sur le serveur en attendant qu'un événement approprié ne le réveille : **agent "résident"** sur le serveur (permanent si l'utilisateur souhaite un service répété)

Caractéristiques des agents mobiles

Mobilité :

- **Forte mobilité**: **code** et **données** sont transférés sur le serveur (host) - langages de script en plus qu'un simple transfert d'objet
- **Faible mobilité**: seules les **données** sont transférées Mobility - transfert de code Java sur une machine virtuelle

remarque :

- un **code mobile** (comme les applets) voyage généralement juste d'un point A à un point B
- les **agents mobiles** ont un **itinéraire** et peuvent voyager dans plusieurs sites de façon séquentielle.

Asynchrone :

- un agent envoyé, **pas nécessaire d'attendre les résultats** : programmer l'agent pour qu'il attende aussi longtemps que nécessaire
- **pas nécessaire de rester connecté au réseau** : un agent peut attendre que l'on soit reconnecté au réseau avant de faire son rapport

Sécurité, 3 types de menaces :

- un **agent** peut être **malveillant** sur le host
- le **host** peut être **malveillant** pour l'agent
- l'**agent** peut causer des **dégâts au réseau**
 - ➔ **authentication** (en utilisant le certificat X509) et **encryptage** (avec SSL)

Application des agents mobiles

Caractéristiques générales des applications :

- **équilibre** de charge dynamique
- **déploiement dynamique** de services
- **systèmes connectés** de façon **intermittente**

Applications spécifiques :

• Collecte de données à partir de sources multiples :

ils peuvent collecter des informations réparties sur un grand nombre d'ordinateurs reliés à un réseau :

Ex : archivage réseau : un agent mobile visite le réseau, collecte l'information sur l'état d'archivage de chaque disque et retourner à son point d'origine pour faire un rapport

• Recherche et filtrage :

à la place de l'utilisateur, ils peuvent visiter de nombreux sites du Web, identifier l'information accessible à chaque site, et construire un index de liens vers des éléments d'information qui répondent à certains critères.

Application des agents mobiles (2)

• **Surveillance** : ils peuvent être envoyés pour attendre que certains types d'informations soient disponibles :

Ex : un agent peut aller sur une place boursière, attendre qu'une action particulière atteigne un certain prix puis en acheter à la place de l'utilisateur

• **Distribution ciblée d'informations** : les agents peuvent distribuer des nouvelles interactives ou de la publicité à des destinataires ciblés

• **Négociation** : ils peuvent obtenir des informations en interaction avec d'autres agents.

Ex : en planification de réunion, un agent mobile consulte les agents représentatifs de chacune de ces personnes concernées et négocie une heure de la réunion

• **Commerce électronique** :

- **Parallélisme** : les agents mobiles peuvent se déplacer de noeud en noeud et peuvent se reproduire en sous-agents
- **Loisirs** : les agents représentent les joueurs qui rivalisent les uns avec les autres, à la place des joueurs

• **Data-mining, ...**

Plateformes de développement d'agents mobiles

Langages de développement :

- **Java class libraries** : tire avantage de la machine virtuelle universelle Java (plateformes Aglet, concordia, Voyager, Grasshopper, D'Agents)
- **langages de script** : tire avantage de langages interprétés (plateformes D'Agents, ARA, Tacoma)

Plate-forme Aglet (IBM) pour le développement d'agents mobiles :

- programmé en **Java**, il peut **stopper** son exécution, se **déplacer** dans un réseau et **poursuivre** son exécution sur un autre serveur hôte (ses données et son code sont mobiles)
- **autonomes** : lorsqu'on le lance, il décide seul où il ira et ce qu'il fera, il décide individuellement de répondre à une requête ou non, ...
- à priori **ni intelligent** (agents intelligents) **ni représentatif de l'utilisateur** (agents logiciels interface) mais peut le devenir

Agents mobiles : défis majeurs [Wayner 95 & Nwana]

• **Transport**: comment un agent fait sa valise et se déplace de place en place ?

• **Authentication**: comment être sûr que l'agent est bien celui qu'il dit être ? comment savoir s'il a navigué sur divers réseaux sans être infecté par un virus ?

• **Secret**: comment être sûr que votre agent préserve votre intimité ? que personne ne manipule pour ses intérêts votre propre agent ? comment être sûr que votre agent n'est pas mort et que son contenu 'core-dumped' ?

• **Sécurité**: comment vous protéger contre les virus ? comment éviter qu'un agent extérieur ne vienne chez vous consommer toute votre CPU dans une boucle sans fin ?

• **Paiement**: comment l'agent paiera des services ? comment être sûr qu'il ne sera pas pris d'une folie furieuse qui conduira à une facture outrageuse ?

• **Performance** : quel sera l'effet d'avoir des centaines, milliers et millions d'agents sur un réseau ?

• **Services d'interopérabilité, de communication et de courtier**: comment se fournir les adresses des divers services offerts ? comment exécuter un agent écrit dans un certain langage sur un moteur écrit dans un autre langage ? comment publier ou souscrire à des services donnés ?

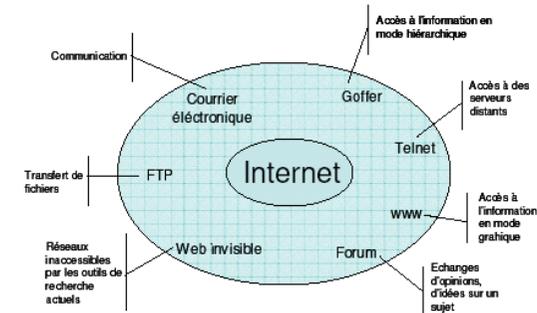
3. Problématique de la RI sur le Web

- Diversité et quantité de documents accessibles sur le Web
- Principaux outils de RI /Web existants
- Faiblesses des outils de RI /Web existants

Problématique de la RI sur le Web

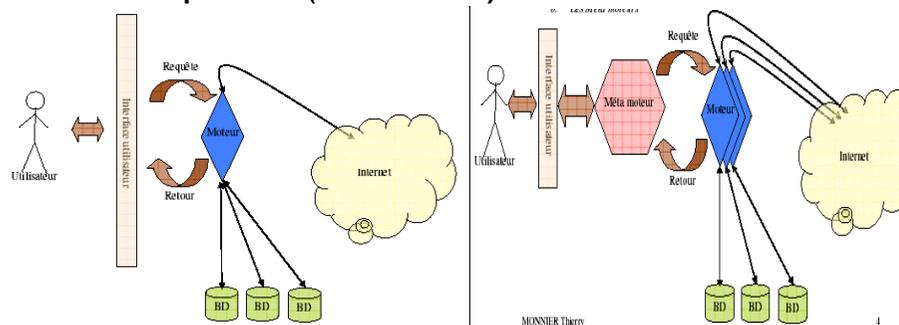
Diversité et quantité de documents accessibles sur le Web

- **Quantité** : Croissance exponentielle du nombre des ressources électroniques mis à la disposition des internautes.
- **Diversité** : multimédias , fichiers sur serveurs FTP, « news letters » issues de « news groups » etc.



Principaux outils de RI /Web existants

- **Moteurs généralistes** (Altavista)
- **Méta-moteurs** :
 - Thématiques hiérarchiques (ou annuaires, type Yahoo)
 - Disciplinaires (biolinks.com)



Ces outils peu satisfaisants pour une grande majorité d'internautes (ils passent plus de 70% de leur temps à cette recherche)

Faiblesses des outils de RI /Web existants (1)

• Atomicité de la page HTML :

Outils actuels : niveau de granularité pour une recherche donnée est en général la **page HTML**.

Or les informations pertinentes pour l'utilisateur se trouvent dans des pages qui contiennent des suppléments d'informations => parcours manuel de la page pour accéder aux informations qui lui sont utiles.

• Indépendance des documents :

Outils actuels : pages HTML indexées indépendamment les unes des autres et perdent ainsi leur contexte (liens vers autres pages)

=> résultat d'une recherche ne peut être qu'un ensemble de pages indépendantes les unes des autres : pas de réponse complète fournie

Or très rare que les informations utiles soient localisées sur une seule page, en général elles sont dispersées sur un ensemble de pages.

En général les outils actuels ne fournissent par de réponses complètes à une requête donnée.

Faiblesses des outils de RI /Web existants (2)

- Différenciation des liens :

Outils actuels : distinguent uniquement les liens internes des liens externes : **ils ne différencient pas les liens selon leur sémantique.**

Or, grandes **différences entre les liens** de publicité, les liens vers du contenu, les liens d'aide à la navigation, ...

- Différenciation des pages :

Outils actuels : indexation des pages HTML, se fait de même façon pour toutes les pages HTML du Web

Or il **existe beaucoup de différences entre ces pages**. Comme il existe des différences entre les paragraphes d'un même livre, il existe des différences entre une page de lien, une page personnelle, ...

Faiblesses des outils de RI /Web existants (3)

- Perte sémantique lors de la construction d'une requête :

Outils actuels : il est **impossible de donner de la sémantique à une requête**

Ex : On désire connaître les livres de science fiction édités en 1999 par Pocket : on devrait entrer les mots clés suivants « *pocket* » « *1999* » « *édition* » « *science fiction* » mais toute la sémantique qui relie les mots est perdue.

Ainsi le moteur ne saura pas que 1999 est la date d'édition des livres, que science fiction est le genre du livre, que pocket est l'éditeur, ...

- Origine des données :

Les outils actuels de recherche sont **incapables d'évaluer le degré de vérité** des données trouvées.

Approche à base d'agent permet de dépasser certaines de ces faiblesses

4. Pourquoi les agents pour la RI

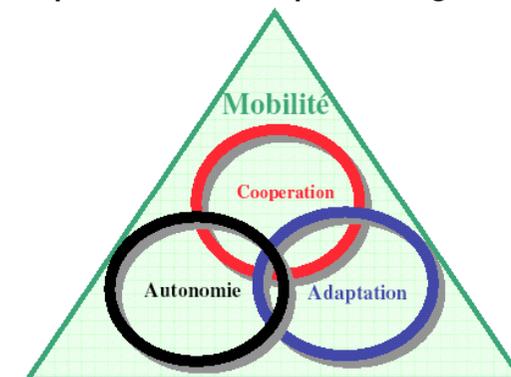
- intérêts de la collaboration, de l'autonomie, de l'adaptation et de mobilité des agents logiciels pour la RI /Web

- Tâches de la RI concernées par les agents

Caractéristiques des agents intelligents

Les agents intelligents sont particulièrement bien adaptés à la RI sur le Web

Rappel des principales caractéristiques des agents



Collaboration et RI

Constituer un système de RI **composé d'agents qui collaborent en échangeant de l'information entre eux** (usage de langages de communication comme FIPA ACL ou KQML)

Doit faciliter :

- la **découverte de nouvelles sources d'informations**,
- le **découpage modulaire du système de RI** sur un ensemble d'agents doit rendre le système :
 - plus **souple** et il est facile de l'étendre (scalable par ajout de nouveaux agents)
 - plus **tolérant** aux erreurs
 - plus **performant** car il est possible d'affecter un agent à un domaine en le dotant d'une parfaite connaissance de ce domaine notamment à travers une ontologie

Autonomie et RI

L'autonomie des agents est **primordiale** dans la RI elle permet de **faciliter le travail de l'utilisateur en agissant à sa place** :

- **Pendant la collecte** :

Lorsque l'agent collecte de l'information il peut selon ses connaissances de l'utilisateur (**profil**) **affiner lui-même la requête** de celui-ci permettant de récupérer de l'information encore plus pertinente

- **Lors du rendu de l'information** :

Le plus gros problème à l'heure actuel est le tri de l'information utile, avant de restituer l'information à l'utilisateur, l'agent doit :

- **réaliser ce tri**, il doit donc de lui-même
- **discriminer les informations pertinentes** de celles qui ne le sont pas.

Adaptation et RI

La capacité d'adaptation des agents leur permet :

- **d'évoluer avec leur environnement et**
- **d'apprendre de leur utilisateur**

permettant ainsi de **cibler de plus en plus précisément les informations utiles**

Les agents deviennent ainsi bien **plus compétitifs que les simples moteurs de recherche** qui eux n'évoluent pas en fonction de l'utilisateur

On peut ainsi introduire les notions de **personnalisation** et de **recommandation** à travers des **profils utilisateurs construits par les agents**.

Mobilité et RI

Les agents de recherche sont souvent des **agents mobiles** permettant de **recueillir plus facilement de l'information** en parcourant à la place de l'utilisateur les pages web :

« **puisque l'information ne vient pas à toi c'est à toi d'aller à l'information** »

Intérêt des agents pour la RI

Usage de capacités d'agents dans différentes tâches de la RI :

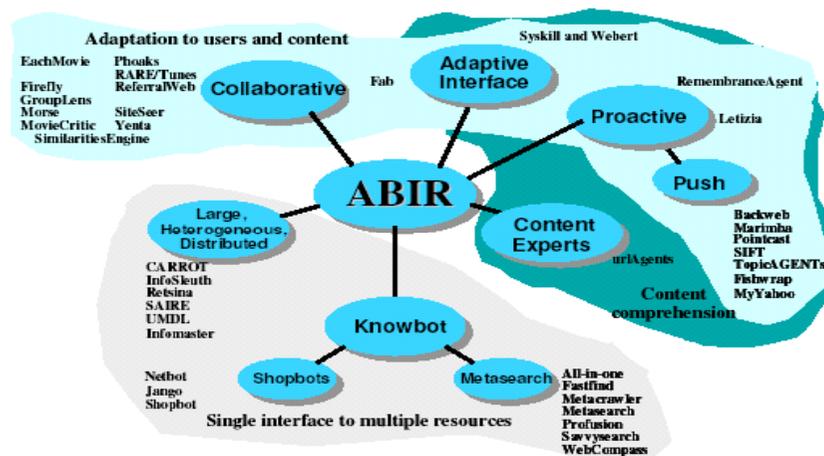
	Adaptation	Cooperation	Autonomy
10. Relevance Feedback	■	■	
9. Information Extraction		■	
8. Multimedia Retrieval		■	
7. Effective Retrieval	■	■	
6. Routing & Filtering	■	■	■
5. Interfaces & Browsing	■	■	
4. Term Expansion			
3. Efficiency & Flexibility	■	■	■
2. Distributed IR		■	■
1. Integrated solutions	■	■	■

5. Les grandes approches de la RI à base d'agents

- Les grandes approches de la RI à base d'agents
- Approche «KnowBot»
- Approche «Adaptative Information Retrieval»
- Approche « Proactive »
- Approche « Collaborative »
- Approche basée sur la négociation
- Restina (Reusable Task Structure-based Intelligent Network Agents)

Différentes approches en RI à base d'agents

Il existe différentes approches dans l' « ABIR » (Agent Based Information Retrieval - Finin & Nicholas 98):



Approche «KnowBot»

Approche permettant à l'aide d'une seule requête, d'accéder à de multiples sources de données

Dans cette approche :

- les agents représentent l'utilisateur devant différents moteurs de recherche sur lesquels ils lancent des requêtes pour récolter de l'information
- forte autonomie des agents : des agents peuvent être capable :
 - d'affiner leurs requêtes en fonction de ce que retourne les moteurs de recherche,
 - de fusionner et filtrer les résultats avant de les rendre à l'utilisateur.

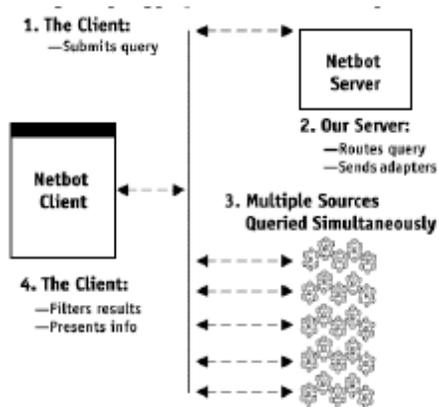
Exemples de KnowBot :

MétaCrawler, SavvySearch, Shopbot, BargainFinder, Netbot, ...

Approche «KnowBot» : exemple de Netbot (1)

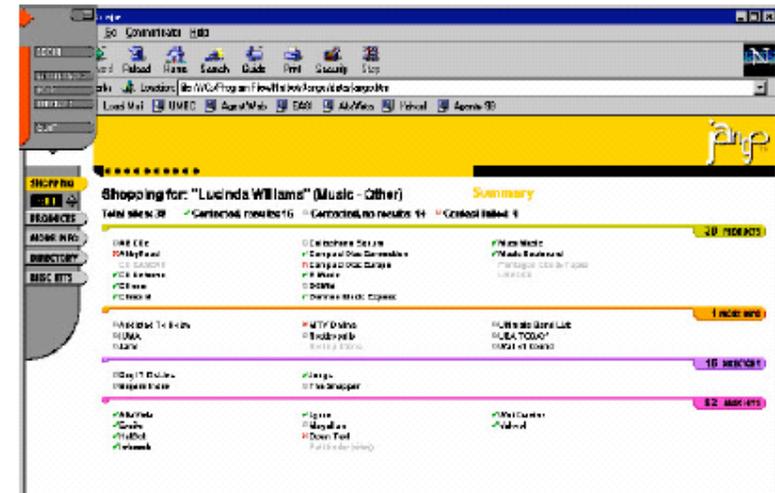
Netbot s'intéresse à la recherche de produits sur Internet

Principe :



Approche «KnowBot» : exemple de Netbot (2)

Interface Web de Netbot :



Approche «Adaptive Information Retrieval»

Approche basée sur le principe de la **recommandation de documents à l'utilisateur**, à partir d'un **profil utilisateur construit par les agents**.

Elle est décomposée en 3 grandes étapes :

- **Collecte** d'informations par le système multi agents,
- **Filtrage et sélection** des informations pertinentes et
- **Rendu** de l'information à l'utilisateur

Notons que l'étape de **requêtage** de l'utilisateur a ici disparu, en effet cette méthode est une méthode « **push** » ou l'agent va de lui-même chercher de l'information et le proposer à l'utilisateur (**recommandation**).

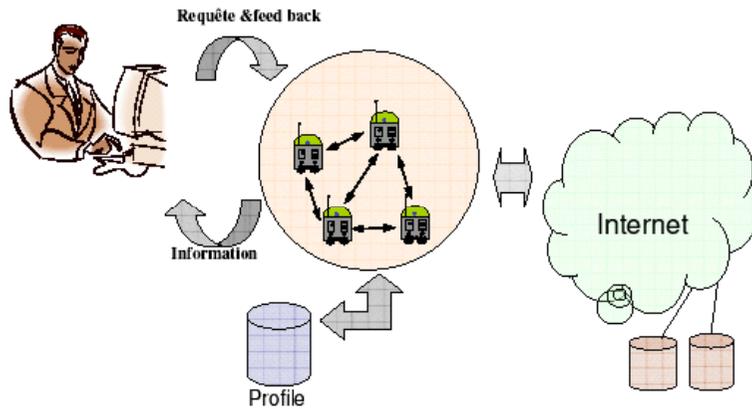
Afin de recommander des informations pertinentes aux utilisateurs un **profil** est construit pour chaque utilisateur :

- sert à **renseigner les agents sur les domaines de recherche** auxquels s'intéresse l'utilisateur,

- est **remis continuellement à jour par les agents** afin d'avoir une description des domaines d'intérêt de l'utilisateur de plus en plus précise.
- les **mise à jour du profil peuvent se faire par retour de l'utilisateur** sur la **qualité** des documents qui lui ont été recommandés.

Approche «Adaptative Information Retrieval»

Le schéma décrivant le **processus de recommandation** se déroule du requêtage de l'utilisateur jusqu'à la mise à jour de son profil :



Approche « Proactive »

Dans cette approche, **les agents parcourent** le Web comme l'utilisateur le fait sauf que le **parcours se fait en largeur** alors que le parcours normal des utilisateurs se fait en profondeur.

- Les agents **récoltent** ainsi de l'information qu'ils **jugent utile** et cela grâce à un ensemble **d'heuristiques**.
- Lorsque des informations utiles sont trouvées, elles sont classées par **ordre de pertinences** et **restituées à l'utilisateur**.

L'un des systèmes les plus célèbres basé sur cette approche est **Letizia**

Approche « Collaborative »

Cette approche est l'une des approches les plus efficaces en RI :

- A **chaque utilisateur** lui est affecté un agent qui gère son **profil**
- Les **agents font des comparaisons entre les profils utilisateurs** afin d'identifier **les utilisateurs qui ont les mêmes centres d'intérêts**
- Lorsque 2 utilisateurs (U1 et U2) ont des profils similaires : **U1 se voit recommander par son agent personnel les documents qu'il n'a pas visité et que U2 a jugé intéressants** (il en sera de même pour U2, il se verra recommander les documents de U1, qu'il n'a pas vu).

Ainsi tous les utilisateurs qui ont un centre d'intérêt commun collaborent au travers de leur agent.

Bien que simple, cette approche assure que les documents recommandés soient pertinents.

les utilisateurs peuvent donner un feedback sur les recommandations de leur agent pour que celui-ci puisse améliorer le profile utilisateur.

Approche basée sur la négociation

Dans cette approche, on identifie 3 types d'agents :

les « user agents » :

- Ils sont du côté de l'utilisateur ils gardent en mémoire ce dont a besoin chaque utilisateur.
- Afin de rechercher les bonnes informations puis lorsque la collecte d'informations est faite c'est eux qui **fusionnerons** les données et les **classerons** afin de les restituer à l'utilisateur

les « broker agents » :

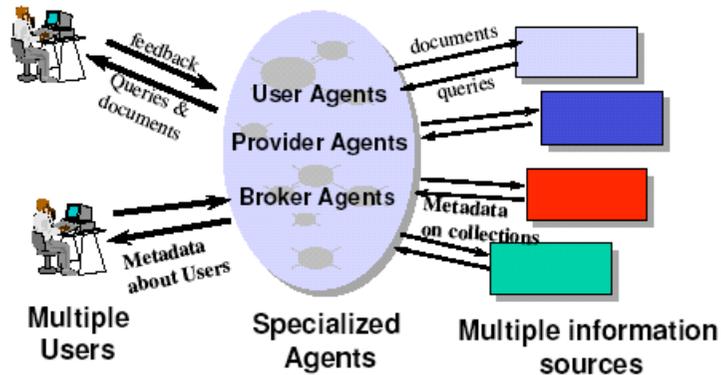
- ils **facilitent la communication entre les différents types d'agents** et permettent de router les requêtes vers les « provider agent » adéquats.

les « provider agents » :

- ils **s'occupent des requêtes vers les sources de données** afin de collecter l'information.

Approche basée sur la négociation

Lorsque les informations sont collectées, elles remontent la chaîne « provider agents », « broker agents », « user agents » pour être restituées à l'utilisateur :

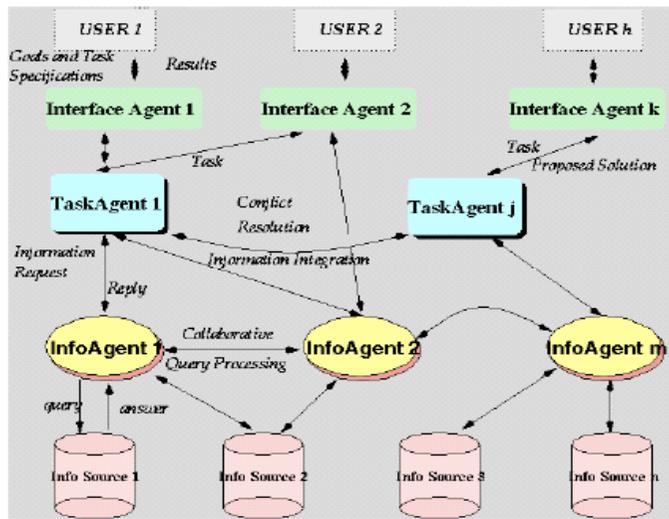


Restina (Reusable Task Structure-based Intelligent Network Agents)

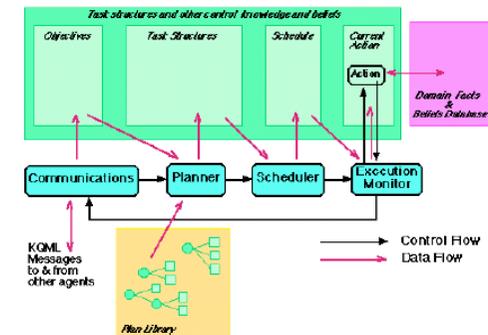
Cette architecture elle aussi différencie plusieurs types d'agents qui travaillent à différents niveaux :

- Les « **interface agents** » :
 - Ils représentent les utilisateurs et peuvent communiquer entre eux
- Les « **task agents** » :
 - ils doivent atteindre un certain nombre de buts et doivent pour cela réaliser un certain nombre de tâches.
 - ils travaillent soit dans des domaines spécifiques soit dans des domaines indépendants
- Les « **information agents** » :
 - Ils collaborent entre eux et exploitent les différentes sources de données

Restina : architecture



Restina : architecture d'un agent



- un module se charge de l'action qui est en cours de réalisation
- un module s'occupe de l'enchaînement des différentes tâches,
- un module possède une description précise des tâches et des objectifs auxquels ces tâches sont associées enfin
- un module s'occupent de recenser l'ensemble des différents objectifs que doit atteindre l'agent.