AGATHE : une architecture générique à base d'agents et d'ontologies pour la collecte d'information sur domaines restreints du Web

B. Espinasse*, S. Fournier* et F. Freitas**

** LSIS UMR CNRS 6168, Universités d'Aix-Marseille, Domaine Universitaire de St Jerôme, F-13997, Marseille Cedex 20, France.*

*** Universidade Federal de Pernambuco, CIn-UFPE, Centro de Informática, Cx Postal 7851 50372-970 Recife, PE, Brasil*

RÉSUMÉ. *La collecte pertinente d'information sur le Web est une tâche très complexe et les moteurs de recherche actuels, reposant sur des méthodes d'indexation et de recherches basées sur des mots-clés, ont de très faibles taux de précision. Les recherches qu'ils réalisent sont essentiellement lexicales statistiques et ne prennent pas en compte leurs contextes sous-jacents. En se limitant à des domaines restreints, la prise en compte de ces contextes est possible et doit conduire à des collectes plus pertinentes. Dans ce papier, est proposée une collecte coopérative d'information à base d'agents logiciels et d'ontologies. Ensuite, une architecture logicielle générique, AGATHE, mettant en œuvre ce type de collecte et permettant le développement de systèmes de collecte relatif à un ou plusieurs domaines, est présentée en détail.*

ABSTRACT. *Relevant information gathering in the Web is a very complex task. The main problem with most information retrieval approaches is neglecting the context of the pages, mainly because search engines are based on keyword-based indexing. Considering restrained domain, it is possible to take into account of this contextwhat should lead to more relevant information gathering. In this paper, a specific cooperative information gathering approach based on the use of software agents and ontologies is proposed. To operationalise this approach, a generic software architecture, named AGATHE, permitting the development of specific restricted-domain information gathering systems is presented in detail.*

MOTS-CLÉS : *système multi-agents, agent logiciel, génie logiciel, collecte coopérative d'information, extraction information, classification, ontologies.*

KEYWORDS: *multi-agents system, software agent, cooperative information gathering, information extraction, classification, ontologies.*

CORIA 2007, pp. 367–383
Saint-Étienne, 28–30 mars 2007

1. Introduction

Du fait de la taille du Web et de la diversité de l'information qui y est accessible, la collecte pertinente d'information sur le Web est une tâche très complexe. Les moteurs de recherche actuels comme Google reposent sur des méthodes d'indexation et de recherche basées sur des mots-clés. Cette approche, bien que robuste, est fondamentalement imprécise, et fournit en général beaucoup de documents non pertinents conduisant à de faibles taux de précision. Cette situation provient principalement du fait que les utilisateurs utilisant ces moteurs ne peuvent seulement faire que des recherches lexicales statistiques, et ne peuvent notamment pas fournir le contexte sous-jacent de leurs besoins en information.

Ainsi la principale faiblesse de la plupart des approches de recherche d'information (RI) sur le Web est qu'elles négligent le contexte des pages. Il est assez illusoire de considérer qu'un seul système peut répondre à l'utilisateur avec un traitement de l'information sur n'importe quel sujet présent sur le Web. Utiliser seulement des mots, des termes et leurs fréquences respectives n'est pas suffisant, pour réaliser une tâche de collecte qu'aucun être humain n'est capable de faire avec des documents, sur n'importe quel sujet, et de plus dans n'importe quelle langue, avec notamment le problème de la polysémie.

Sans la prise en compte explicite du contexte dans lequel est faite la recherche, la plupart des approches de RI actuelles laissent s'échapper toute forme organisée du Web, par exemple des regroupements ou « clusters » spécifiques d'information. Considérer de tels clusters, regroupant des classes de pages avec leur information (par exemple l'ensemble de pages affichant les données d'une bourse, l'ensemble de pages sur des chercheurs, ...), permettraient la reconnaissance et la classification de pages dispersées sur le Web, relatives à un domaine restreint, et par là même faciliter la prise en compte de contexte de recherche.

Avec la prise en compte du contexte, un meilleur traitement de l'information est alors possible. C'est le cas de l'extraction de la plupart des informations des pages d'une même classe (par exemple valeur du dollar, sujets d'intérêt d'un chercheur, ...). Un autre avantage est de permettre aux utilisateurs d'effectuer des interrogations combinant notamment des critères de recherche relatifs à différentes classes de pages, permettant des requêtes complexes (la recherche des papiers publiés dans un certain ensemble de conférences par exemple), et ainsi la réalisation d'applications sophistiquées de collecte d'information sur le Web pour des domaines spécifiques. Avec un cluster « tourisme », par exemple, il serait possible de réaliser un système qui prenne en compte les hôtels, les billets de voyage, et les manifestations culturelles liés à un événement scientifique.

Cette recherche s'intéresse à une approche originale de la collecte d'information sur domaine restreint à partir du Web, avec une prise en compte du contexte de la recherche. Cette approche est originale dans la mesure où elle est basée sur l'usage

d'ontologies permettant de définir le domaine restreint et le contexte, et où elle est coopérative en s'appuyant sur les agents logiciels, pour réaliser notamment l'extraction d'information grâce à ces ontologies, ceci de façon distribuée et coopérative.

Ce papier introduit tout d'abord l'intérêt de restreindre la collecte d'information sur le Web à des domaines restreints, l'intérêt d'utiliser des agents logiciels et des ontologies pour réaliser des systèmes de collectes intelligents. Ensuite est présentée une architecture logicielle générique, le système AGATHE (Agents information GATHEring). Ce système doit permettre le développement de systèmes de collecte intelligents sur le Web sur un ou plusieurs domaines. L'architecture AGATHE est tout d'abord présentée de façon générale puis chacun de ses sous-systèmes constitutifs, sous-système de recherche, d'extraction et d'utilisation, est décrit plus en détail. Enfin, une conclusion fait un bilan de cette recherche et présente les perspectives associées.

2. Collecte d'information sur domaine restreint à base d'agents et d'ontologies

Dans cette section, sont développés tout d'abord l'intérêt de la restriction de domaine en recherche et extraction d'information, ensuite l'intérêt de l'usage d'agents logiciels, puis des ontologies en collecte d'information sur le Web.

2.1. Recherche et extraction d'information sur domaine restreint

La principale leçon apprise en intelligence artificielle (IA) dans les années 70 a été que l'opérationnalisation de la connaissance est valable seulement sur des domaines restreints, ce qui a conduit au succès relatif des systèmes experts. Cette constatation est aussi valable pour la recherche d'information (RI).

En effet, l'évaluation des systèmes de RI est principalement effectuée sur des corpus homogènes dont les textes portent sur un seul sujet et viennent souvent de la même source, et non pas d'ensembles de textes dont le contenu et les styles sont variables comme c'est le cas de ceux disponibles sur le Web. Ce fait est d'ailleurs à l'origine du développement en RI de moteurs de recherche spécialisés (Steele, 2001). Ils agissent sur des domaines suffisamment spécifiques du Web, tels que celui des nouvelles quotidiennes (Newstracker - www.newstracker.com), ou des pages personnelles (HPSearch - http://hpsearch.uni-trier.de/).

Un autre argument plaidant pour une restriction de domaine en RI concerne l'extraction d'information (EI). D'une façon générale, l'EI concerne l'extraction d'information à partir d'une collection de documents textuels (Muslea, 1999) (Embley, 1998). L'EI est un nouveau champ de recherche de la RI dont l'objectif consiste à extraire des données appropriées à partir de classes spécifiques de pages issues du Web (Gaizauskas et Robertson, 1997). L'EI est la tâche d'identification de fragments spécifiques d'un document constituant le noyau de son contenu sémantique

(Kushmerick, 1999a). Le but principal de l'EI est de construire des bases de données (relatives à des domaines restreints) rassemblant de l'information provenant de nombreuses pages Web issues de sites géographiquement distribués, bases de données qui seront ensuite interrogées par des utilisateurs.

2.2. Agents logiciels et collecte coopérative d'information

Les agents logiciels ont évolués à partir des systèmes multi-agents (SMA), qui constituent une des trois principales branches de l'intelligence artificielle distribuée (IAD), les deux autres étant respectivement la résolution de problèmes distribués et l'intelligence artificielle parallèle (Nwana, 96). Les agents logiciels héritent par conséquent des propriétés et potentialités associées habituellement au domaine de l'IAD. Par exemple, ils héritent de la modularité, de la vitesse d'exécution (due au parallélisme) et de la fiabilité (grâce à la redondance) des systèmes de ce domaine. D'autres propriétés venant de l'IA classique viennent aussi s'ajouter permettant notamment une manipulation de la connaissance, une maintenance aisée, une réutilisation et une indépendance aux plates-formes (Huhns et Singh, 1994).

Les SMA et les agents logiciels ont tout d'abord été utilisés afin de résoudre des problèmes intrinsèquement distribués et de simuler des phénomènes complexes. Ils sont dorénavant aussi utilisés afin de développer des systèmes de recherche et de classification de l'information. De tels systèmes donne la possibilité de concevoir des outils robustes et adaptatifs. La FIPA (Foundation For Physical Agents) propose de nombreux standards et recommandations facilitant l'utilisation du paradigme agent dans des applications réelles. Plusieurs méthodologies ont aussi été proposées permettant d'accompagner les phases de conception et d'implémentation contribuant ainsi à la constitution d'un génie logiciel orienté agents.

Le concept de « collecte coopérative d'information » sur le Web, proposé par Oates (Oates *et al*., 1994), est associé à une manipulation de l'information disponible sur le Web basée sur de la connaissance. Oates définit la collecte de l'information (Information Gathering) comme la conjonction de l'acquisition et de la recherche de l'information, et il propose d'utiliser des systèmes multi-agents coopératifs permettant d'intégrer et d'élaborer des « clusters » d'information de qualité. La résolution distribuée de problème est alors un moyen pour les agents de découvrir des clusters pertinents d'information. D'autres travaux de recherches (Ambite et Knoblock, 1997) préconisent l'usage d'agents pour la collecte d'information : les bases de données d'une grande bibliothèque numérique ont été groupées en classes hiérarchiques, chaque classe possédant son propre agent avec la connaissance explicite à son sujet. Ces agents construisent les plans de recherche qui améliorent l'efficacité dans le processus de recherche. L'utilisation d'un tel outil sur le Web nécessite un appariement correct des pages récupérés sur le Web à ces classes et d'en extraire l'information pour alimenter ces bases de données.

2.3. Collecte coopérative d'information et ontologies

En collecte d'information sur le Web, la combinaison de l'approche multi-agents avec de la connaissance déclarative, conduisant à l'usage d'ontologies, s'avère pertinente pour le développement de systèmes intelligents de collecte.

Cette combinaison se justifie tout d'abord en raison des avantages classiques des solutions déclaratives sur celles qui sont procédurales. Les solutions déclaratives fournissent une plus grande intégration de l'approche ontologique avec une traduction plus directe de la connaissance de domaine. Les tâches d'extraction et de classification sur le Web, qui impliquent des données semi structurées ou non structurées, exigent des changements fréquents de la solution. Avec cette déclarativité de la connaissance, de tels changements peuvent être facilement pris en compte, sans recompilation de code ou d'arrêt d'exécution. Ce qui constitue un avantage notable d'extensibilité.

La grande expressivité du mode déclaratif de représentation de la connaissance est aussi un avantage majeur en collecte d'information sur le Web. En plus des possibilités d'inférences, quand les concepts impliqués dans ces tâches (par exemple les entités du cluster, les groupes fonctionnels, les représentations de page Web, etc.) sont définis de façon déclarative, ces concepts peuvent être organisés en structures connues sous le nom d'ontologie.

L'utilisation des ontologies apporte beaucoup d'avantages (Gruber, 1995). Tout d'abord en permettant généralement l'héritage multiple, elles procurent un avantage d'expressivité sur des implémentations orientées objets. Elles permettent aussi la création de modèles de communication de haut niveau, connu sous le nom de « pair-à-pair », dans lesquels les concepts définis, comme une connaissance de domaine, sont communs aux agents communicants, jouant le rôle du vocabulaire partagé pour la communication entre agents.

L'usage d'ontologies dans le développement de système de collecte sur le Web augmente aussi leur flexibilité. Les entités d'un cluster (la connaissance de domaine) peuvent être définies avec la granularité appropriée représentant les différences subtiles de hiérarchie entre les entités.

Enfin, représentée de façon déclarative, la connaissance sur les pages Web, et les conditions sous lesquelles elles sont considérées pour représenter un exemple d'une entité, n'est pas limitée aux termes, aux mots-clés et aux statistiques. Elle concerne n'importe quel fait pouvant distinguer une classe de pages à d'autres classes, par exemple des faits impliquant la structure de page, de régions probables où trouver l'information appropriée à extraire, les concepts contenus et la signification de la phrase, ceci par l'utilisation de technique de traitement du langage naturel.

2.4. Conclusion

Dans le développement de systèmes de collecte d'information de domaine restreint sur le Web, une approche combinant l'usage d'agents logiciels et d'ontologies apparaît pertinente. C'est cette approche que nous avons retenu pour concevoir et réaliser le système AGATHE, une architecture logicielle générique permettant le développement de systèmes de collecte intelligents sur le Web relatif à un ou plusieurs domaines. Les sections suivantes seront consacrées à la présentation de cette architecture générique.

3. Présentation générale du système AGATHE

Le système AGATHE est une architecture logicielle générique permettant le développement de systèmes de collecte d'information sur le Web sur un ou plusieurs domaines restreints. AGATHE met en œuvre une collecte coopérative d'information à base d'agents et d'ontologies. Ce système prend en compte des contextes de recherche en considérant des regroupements de pages Web relatifs à des domaines spécifiques (par exemple la recherche académique, le tourisme, …).

Dans cette section, sont tout d'abord présentés les objectifs du système AGATHE, son architecture générale mettant en évidence trois principaux sous systèmes, son fonctionnement général, et enfin quelques détails d'implémentation sont fournis.

3.1. Objectifs du système AGATHE

La structure du Web et son contenu évoluent continuellement. Des informations apparaissent et disparaissent et de nouveaux concepts et outils sont créés et changent très rapidement. Les systèmes de collecte d'information doivent pouvoir suivre cette évolution. Ces systèmes doivent être extensibles, adaptatifs et flexibles pour exploiter l'information issue du Web.

Comme nous l'avons déjà évoqué, cette architecture logicielle tire profit du génie logiciel orienté agents (qui sera par la suite étendu à l'usage de services Web), ceci afin d'assurer flexibilité et réutilisabilité. Le point de départ de cette architecture est un prototype déjà réalisé, le système MASTER-Web (Multi-Agent System for Text Extraction and Retrieval over the Web) (Freitas *et al.*, 2000, 2001, 2003). Ce système adoptait déjà l'approche agent et utilisait aussi des ontologies pour réaliser des tâches de classification et d'extraction d'information sur le Web, sur un seul domaine de recherche.

L'architecture AGATHE reprend les techniques de classification et d'extraction à base d'ontologies de MASTER-Web, en les déployant sur une organisation d'agents logiciels plus efficace, avec des agents plus nombreux et plus spécialisés. De plus AGATHE permet de traiter plusieurs domaines de recherche simultanément, et

dispose de plus de mécanismes de recommandation inter domaines sophistiqués, et une mise en œuvre plus aisée. Enfin, dans son développement logiciel orienté agents, AGATHE respecte toutes les recommandations de la FIPA (Foundation for Intelligent Physical Agents).

L'architecture AGATHE doit permettre de développer des systèmes de collecte d'information sur le Web sur des domaines restreints pouvant être progressivement étendus. Pour le développement d'AGATHE, le domaine restreint de recherche de départ est celui de la recherche académique, plus précisément la tenue d'événements scientifiques (conférences ou workshops internationaux). La collecte sera ensuite élargie à d'autres domaines restreints. Ainsi, la collecte d'information relative à la tenue d'événements scientifiques internationaux, avec les informations pertinentes associées (Appel à contribution (CFP), appel à participation, titre, sponsors, événements conjoints, lieu, thèmes, dates importantes, programme, session, …), sera étendue au domaine du tourisme et/ou du transport, ceci afin de prévoir un déplacement lié à une participation à un événement scientifique particulier (possibilités de voyage, d'hébergement, de visites touristiques , etc. …). A chacun de ces domaines est associée une ontologie spécifique.

3.2. Architecture et fonctionnement général d'AGATHE

L'architecture générale du système AGATHE, illustrée à la figure 1, s'articule autour de trois principaux sous-systèmes en interaction :

– Le *sous-système de Recherche* (SSR) est chargé de l'interrogation des moteurs de recherche externes sur le Web (comme Google) afin d'obtenir des pages Web qui seront traitées par le sous-système d'Extraction.

– Le *sous-système d'Extraction* (SSE) composé de différents « cluster d'extraction » (CE), chacun spécialisé dans le traitement de pages Web sur un domaine spécifique (comme celui de la recherche académique, ou celui du tourisme).

– Le *sous-système d'utilisation* (SSU) assure le stockage des données extraites à partir des pages Web traitées par le sous-système d'Extraction, et fournit une interface d'interrogation pour des utilisateurs, pouvant être des humains ou d'autres agents logiciels.

Ces trois sous-systèmes principaux du système AGATHE sont des systèmes multi-agents composés d'agents logiciels (agents informationnels) plus ou moins intelligents. Certains d'entre eux utilisent des ontologies pour réaliser les tâches pour lesquelles ils sont conçus. Chacun de ces sous-systèmes est étudié plus en détail dans les sections suivantes.

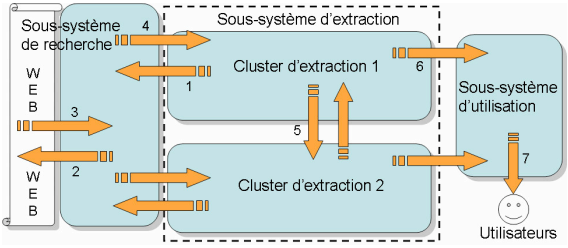

Figure 1 : Architecture générale d'AGATHE.

Le fonctionnement d'AGATHE est illustré sur la figure 1 par les flèches d'interaction numérotées entre ses différents sous-systèmes :

– flèche 1 : Un cluster d'extraction du SSE demande une recherche pour des pages particulières au SSR.

– flèches 2 et 3 : Le SSR travaille comme un méta robot de recherche, recherchant des pages Web, en interrogeant des moteurs de recherche existants comme Google, Altavista ou d'autres.

– flèche 4 : Ces pages sont ensuite transmises au SSE, plus précisément à l'agent du cluster d'extraction qui en avait fait la demande (flèche 1).

– flèche 5 : Si nécessaire, des recommandations sont envoyées par le cluster considéré à d'autres clusters d'extraction, ceci afin de leur proposer des pages qui peuvent potentiellement les intéresser.

– flèche 6 : L'information extraite est transmise au SSU, afin d'être stockée dans une base de données spécifique, et qui sera accessible pour interrogation par les utilisateurs (flèche 7).

3.3. Détails d'implémentation

Le système AGATHE est actuellement en cours d'implémentation. Il est déployé dans l'environnement ECLIPSE en utilisant la plate-forme multi-agents Jade (Jade, 2006). Les agents informationnels sont développés avec le moteur d'inférences Jess (Jess, 2006). Le développent multi-agents respecte les recommandations de la FIPA (FIPA, 2000), notamment le langage de communication retenu est ACL-FIPA. Pour la saisie et la manipulation d'ontologies, l'environnement Protégé (Protégé, 2006) a été retenu et l'exploitation des ontologies par les agents informationnels est faite via le composant JessTab (Eriksson, 2003).

4. Le sous-système de recherche (SSR)

Comme illustré par la figure 2, le sous-système de recherche (SSR) reçoit des requêtes provenant des clusters d'extraction (CE) du sous système d'extraction (SSE). Ces requêtes permettent, via des moteurs de recherche, de récupérer des pages Web en HTML. Par exemple, un agent d'extraction "article" du cluster d'extraction "science", effectue une requête afin d'acquérir des pages contenant les termes tel que "introduction", "related work", "conclusion". Le SSR transfert cette requête aux différents moteurs de recherche et rassemble les pages Web obtenus. Ces dernières sont retournées vers le cluster approprié qui s'occupe ensuite de les transmettre aux agents demandeur.

Trois types d'agent contribuent à la récupération des informations : (i) les *agents de recherche* ciblant le Web dans son ensemble au travers des moteurs de recherche traditionnels (Google, Yahoo, Altavista, …), (ii) les *agents ressources* effectuant des recherches à travers des moteurs de recherche spécifiques comme DBLP et CITESEER, et enfin (iii) *l'agent superviseur* qui supervise les deux premiers types d'agents. Chacun de ces agents est décrit en détail dans les sous-sections suivantes.

4.1. L'agent de recherche

L'agent de recherche transmet aux moteurs de recherche traditionnels les requêtes qui lui proviennent des clusters d'extraction du SSE. Après avoir fusionné les résultats obtenus par les différents moteurs sollicités, cet agent retourne les résultats aux clusters d'extraction concernés du SSE.

4.2. L'agent de ressource

L'agent de ressource est relativement similaire à l'agent de recherche si ce n'est qu'il transmet les requêtes issues du SSE vers des ressources spécialisées du Web. Par exemple, pour des requêtes concernant la recherche académique de telles ressources peuvent être des sites comme CITESEER dans le cas de publications, DBLP dans le cas de recherche à propos d'auteurs où même des services Web spécifiques.

4.3. L'agent superviseur

L'agent superviseur assure la supervision du SSR. Plusieurs rôles lui sont dévolus. Tout d'abord, il est là pour recevoir les requêtes en provenance SSE. En fonction de la charge de chacun des agents de recherche et de ressources, il va gérer l'allocation des requêtes. Ensuite, il est capable en fonction des besoins et de la charge des agents de recherche et de ressource, d'en créer ou d'en supprimer. Enfin, pour des raisons de performance, il est aussi capable de réaliser la migration d'agents vers d'autres unités de calcul moins surchargée. L'agent superviseur gère aussi des inscriptions donnant ainsi la possibilité au SSE de recevoir périodiquement

les résultats d'une requête spécifique sans avoir à en reformuler la demande. Cette méthode est appelée "push".

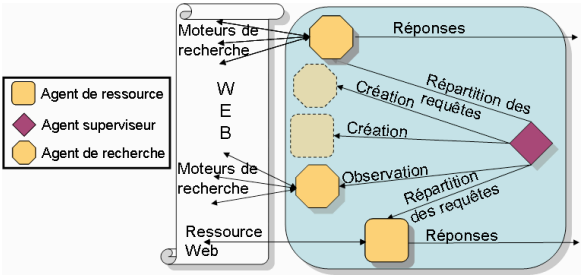

Figure 2 : L'architecture interne du SSR.

5. Le sous-système d'extraction (SSE)

Comme déjà mentionné, AGATHE reprend les techniques de classification et d'extraction à base d'ontologies de MASTER-Web spécialisés (Freitas *et al.*, 2000, 2001, 2003) en les déployant sur une organisation d'agents logiciels plus efficace, avec des agents plus nombreux et plus spécialisés. En effet MASTER-Web est composé d'un seul type d'agent réalisant de nombreuses tâches de classification et d'extraction à l'aide d'une ontologie de domaine.

L'architecture AGATHE permet de traiter plusieurs domaines de recherche simultanément, et dispose de plus de mécanismes de recommandation inter domaines sophistiqués, et une mise en œuvre plus facile. AGATHE peut ainsi intégrer plusieurs clusters d'extraction, et elle est capable de traiter des informations contextualisées possédant des ramifications vers d'autres domaines. Rappelons que chaque cluster d'extraction est rattaché à un unique domaine défini par une ontologie spécifique. Des relations entre domaines peuvent exister, par exemple le concept « d'appel à participation » va contenir des informations spécifiques à un événement scientifique, mais peut aussi contenir des informations touristiques de l'événement (hôtels, événements culturels, moyens de locomotion …). Une page Web contenant ce concept va ainsi pouvoir être traitée par un cluster s'occupant des évènements scientifiques, et aussi par un cluster s'occupant des aspects touristiques.

D'une façon générale, le SSE génère tout d'abord les requêtes de pages Web envoyées au SSR, puis et c'est sa tâche majeure, traite les résultats de ces requêtes provenant du SSR. Cette tâche est composée de plusieurs sous tâches : validation, classification fonctionnelle et extraction d'information.

Comme le montre la figure 3, chacun des clusters d'extraction est un système multi-agents réalisant la classification des pages Web obtenues et l'extraction d'informations à partir de ces pages. Les clusters d'extraction sont composés de

différents types d'agents : des *agents extracteurs*, des *agents préparateurs*, un *agent superviseur*, un *agent de recommandation* et un *agent de stockage*.

A Certains de ces agents (agent extracteurs, agents préparateurs et agent de recommandation) sont associées des ontologies manipulées à l'aide de Protégé. La figure 4 est issue de cet outil et concerne les concepts liés aux événements scientifiques. La plupart de ces agents sont des agents logiciels « cognitifs » développés avec le moteur d'inférence Jess. Leurs comportements sont pour l'essentiel spécifiés à l'aide de règles de production et ils utilisent des ontologies grâce à JessTab, un composant de Protégé. Ces cinq agents composant le cluster d'extraction et par là même le SSE, sont présentés et détaillés dans les sous-sections qui suivent. La figure 5 illustre les interactions entre ces différents agents.

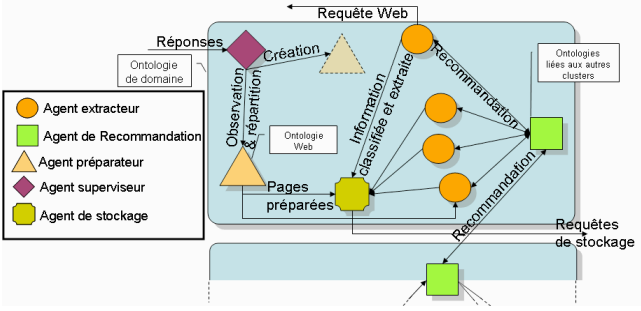

Figure 3 : L'architecture interne du SSE

5.1. Les agents préparateurs

Les agents préparateurs reçoivent les pages Web en provenance SSR. Ils sont créés par l'agent superviseur du même cluster et peuvent être supprimés par lui lorsqu'ils ne sont plus utilisés. Ces agents réalisent un premier traitement des pages Web reçues facilitant par la suite, le traitement d'extraction d'information. Ce traitement préliminaire consiste en la séquence des tâches suivantes :

– Une *validation* consistant à vérifier si les pages Web sont valides. C'est à dire si elles sont au format HTML, elles sont accessibles, et si elles n'ont pas déjà été stockées dans la base de données. Les pages ne vérifiant pas ces conditions ne seront pas traitées par la suite.

– Un *pre-processing* identifiant le contenu, le titre, les liens et les emails disponibles dans la page en utilisant des techniques issues du domaine de l'RI (Recherche d'Informations) et éventuellement des techniques basées sur le langage naturel.

– Une *classification fonctionnelle*. Cette tâche, à base de connaissances, utilise le moteur d'inférence Jess et exploite une ontologie définissant des concepts liés au Web. Une partie de cette ontologie est présentée par la figure 4. En utilisant la base

de connaissances formée de règles de production décrivant l'ontologie associée au Web, l'agent préparateur va classifier les pages Web selon un aspect fonctionnel. Les catégories fonctionnelles ainsi obtenues sont les suivantes : message, liste (de liens, de personnes, d'ouvrages), les pages auxiliaires (pages contenant des informations mais ne représentant pas une entité caractéristique comme par exemple une page concernant les sujets d'intérêts d'une conférence mais séparée de la page principale), les pages choisies pour l'extraction et enfin les pages non valides.

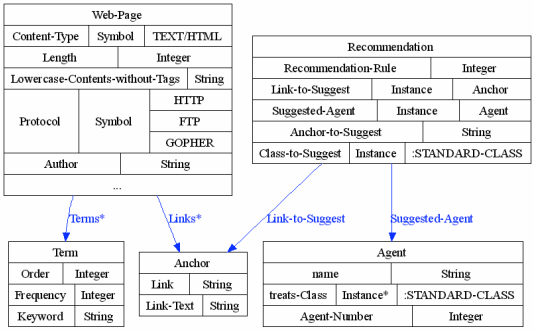

Figure 4 : Principales classes, slots et relations de l'ontologie du Web utilisée.

La figure 5 illustre l'enchaînement des tâches réalisées par l'agent préparateur. L'agent préparateur reçoit (1) de l'agent superviseur les pages HTML brutes (sans traitement préalable). Commence alors la tâche de validation de ces pages (2). Alors soit l'agent commence la tâche de pre-processing (3), soit la page est invalide et est envoyée à l'agent de stockage (4) afin d'éviter de répéter le traitement dans le futur. La tâche de pre-processing étant réalisée, la tâche de classification fonctionnelle est effectuée (5). Cette tâche produit des pages prêtes pour la phase d'extraction et de classification et ces dernières sont envoyées aux agents extracteurs (6). Dans le cas où la page est invalide, celle-ci est envoyée à l'agent de stockage (7).

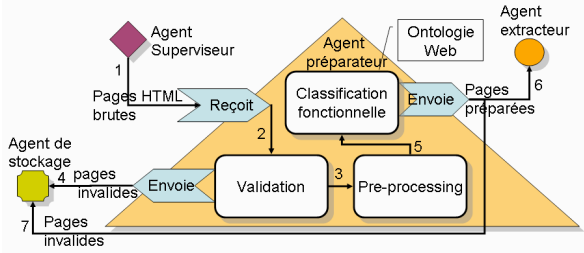

Figure 5 : L'agent préparateur et ses tâches internes.

5.2. Les agents extracteurs

Le cluster d'extraction est aussi composé de plusieurs agents extracteurs. Chacun de ces agents est associé à un élément de l'ontologie de domaine associé au cluster. La figure 6 présente une partie de l'ontologie de domaine relative aux événements scientifiques.

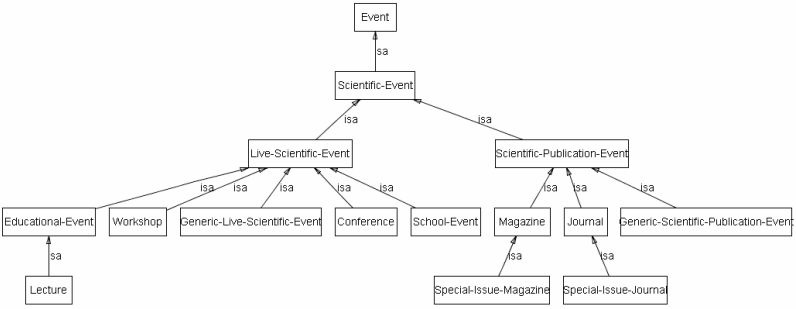

Figure 6 : Une partie de l'ontologie concernant les événements scientifiques.

La tâche spécifique de ces agents est d'effectuer une *classification sémantique* des pages reçues, ainsi qu'une *extraction des informations* qu'elles contiennent. Chacun des agents extracteurs est associé à un concept particulier de l'ontologie de domaine associé au cluster d'extraction considéré. Par exemple, un des agents extracteurs s'occupe du concept « appel à communication ». Tout comme l'agent préparateur, l'agent extracteur utilise aussi pour exploiter ces ontologies les outils Jess et JessTab.

Prenons l'exemple du domaine des événements scientifiques. Un agent extracteur peut ainsi être associé comme nous l'avons vu précédemment au concept « appel à participation » appartenant à l'ontologie décrivant les événements scientifiques. Les pages qu'un tel agent traite proviennent de requêtes utilisant, par exemple, les mots clefs « Call for Papers ». Cet agent va utiliser pour la classification sémantique de ces pages et l'extraction des informations qui y sont continues, les éléments rattachés à ce concept, par exemple les termes : titre, sujet, comité d'organisation …. L'information extraite est ensuite transmise à l'agent de stockage.

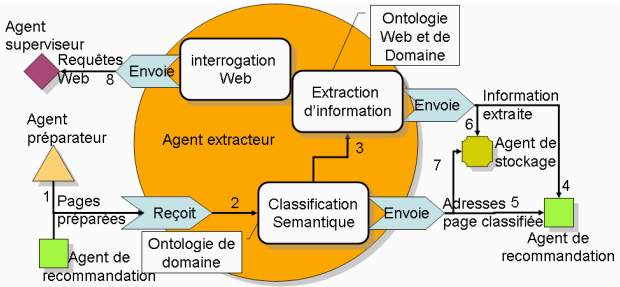

Figure 7 : L'agent extracteur et ses tâches internes.

La figure 7 illustre l'enchaînement de tâches réalisées par l'agent extracteur. L'agent extracteur reçoit les pages préparées de l'agent préparateur (1) et effectue la *classification sémantique* (2). Les adresses des pages classifiées et les classes auxquelles elles appartiennent, sont envoyées à l'agent de stockage (3), et à l'agent de recommandation (4). Ensuite, l'agent extracteur réalise la tâche *d'extraction d'information* des pages (5). Ces informations sont aussi envoyées à l'agent de stockage (6) et à l'agent de recommandation (7). Indépendamment des autres tâches, l'agent extracteur a la possibilité *d'interroger le Web* en envoyant une requête à l'agent superviseur du SSR (8).

5.3. L'agent de recommandation

L'agent de recommandation, illustré par la figure 8, reçoit les pages traitées par des agents extracteurs et recommande certaines de ces pages à d'autres agents extracteurs du même cluster ou à des agents de recommandation d'autres clusters. Pour ce dernier cas, l'agent de recommandation a aussi connaissance d'une partie des ontologies liées aux autres clusters que le sien.

La figure 8 présente l'enchaînement de tâches effectuées par l'agent de recommandation. Il reçoit l'information extraite et les pages classifiées d'agents extracteur (1), et des pages de recommandation d'agents de recommandation appartenant à d'autres clusters d'extraction (2). L'agent exécute la tâche concernant les recommandations vers les autres clusters d'extraction en utilisant les pages reçues au préalable (3,4). Les pages identifiées comme pouvant faire l'objet de recommandation sont transmises aux agents de recommandation des autres clusters (6) en fonction des concepts identifiés au sein de ces pages. Ensuite, il accomplit la tâche permettant d'identifier (5) les pages recommandées pour les agents extracteurs se trouvant dans le même cluster (7). Les autres pages sont envoyées à l'agent de stockage afin d'éviter tout traitement futur inutile (8).

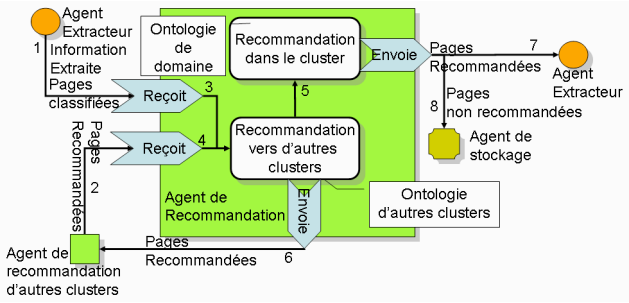

Figure 8 : L'agent de recommandation et ses tâches internes

Pour illustrer l'échange de recommandation, considérons deux clusters d'extraction pour lesquels il existe un lien contextuel. Prenons par exemple un cluster d'extraction responsable des événements scientifiques et un autre responsable du tourisme. Des informations récupérées au sein d'appels à communication peuvent faire référence à des informations concernant les hôtels disponibles et les moyens de transports. Les pages liées à ces informations ne peuvent évidemment pas être traitées par le cluster d'extraction relatif aux événements scientifiques. Le cluster d'extraction en charge du tourisme possède, par contre, lui tous les concepts nécessaires au traitement de telles pages.

5.4. L'agent de stockage

L'agent de stockage est en charge de récolter les informations extraites et classifiées. Il ne réalise pas directement l'action de stocker ces informations dans la base de données ; il prépare le stockage. Il traite l'information qu'il reçoit afin de la conformer au format approprié de stockage selon la structure de la base de données. Ensuite, les données formatées sont envoyées au SSU afin d'être physiquement stockées, et de pouvoir être exploitées par les utilisateurs.

5.5. L'agent superviseur

L'agent superviseur du cluster d'extraction possède des fonctionnalités similaires à celui du sous-système de recherche. Il supervise l'activité des agents préparateurs. Il reçoit les résultats des requêtes envoyées par le sous-système de recherche. En fonction de la charge des agents préparateurs, il peut en créer ou en détruire et répartir les requêtes reçues entre eux.

6. Le sous-système d'utilisation d'AGATHE (SSU)

Le sous-système d'utilisation (SSU), illustré à la figure 9, permet les interactions entre AGATHE et les utilisateurs. Ces derniers exploitent les résultats fournis par

AGATHE par interrogation d'une base de données relationnelle stockant les informations extraites.

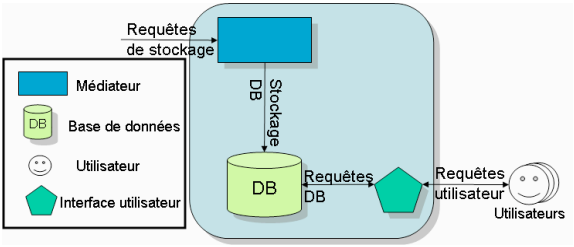

Figure 9 : Architecture interne du SSU

Le SSU est composé de deux principaux composants que sont : le médiateur et l'interface utilisateur. Le premier a la tâche de mettre à jour la base de données en utilisant les informations extraites par le SSE via les agents de stockage. Le second est le support des interactions entre le système AGATHE et les utilisateurs.

7. Conclusion

En se limitant à des domaines restreints, la prise en compte de contextes de recherche d'information est possible et doit conduire à des collectes plus pertinentes. Dans ce papier, a été proposée une collecte coopérative d'information à base d'agents logiciels et d'ontologies (ontologie du Web et ontologies de domaine), et son opérationnalisation au travers d'une architecture logicielle générique, AGATHE, mettant en œuvre ce type de collecte et permettant le développement de systèmes de collecte relatif à un ou plusieurs domaines, est présentée en détail.

AGATHE est composée de trois principaux sous-systèmes en interaction, un sous-système recherche réalisant les requêtes sur le Web au travers de moteurs de recherche, un sous-système d'extraction assurant une classification puis une extraction d'information des pages récupérées du Web et enfin d'un sous-système d'utilisation permettant à des utilisateurs d'exploiter ces informations extraites stockées dans une base de données relationnelle. L'architecture logicielle interne de ces sous-systèmes est composée d'agents logiciels exploitant, pour certains, des ontologies, soit du Web, soient liées au domaine de recherche considéré, pour réaliser des tâches de classification de filtrage, de recommandation ou d'extraction d'information par raisonnement.

Le système AGATHE est actuellement en cours de développement, un premier prototype fonctionne et des premiers résultats devraient être bientôt produits. Pour l'amélioration de ce prototype, trois axes d'évolution sont actuellement envisagés : (i) l'intégration de techniques de traitement des langues naturelles notamment au niveau des tâches de classification et d'extraction d'information, ceci afin de les rendre plus performantes, (ii) l'intégration de techniques d'apprentissage notamment

au niveau des tâches de recommandation, et enfin (iii) l'usage de Web services (WS), perçus comme composants pouvant être utilisés pour développer certains agents informationnels. Sur ce dernier point, la librairie de WS définie pour l'industrie du voyage dans le cadre du projet Satine (Satine, 2001), pourrait être invoquée dans une prochaine version d'AGATHE.

8. Références

Ambite J., Knoblock C., « Agents for information gathering ». *In Software Agents. Bradshaw*, J. (ed.), MIT Press, Pittsburgh, PA, USA, 1997.

Appelt D. E., Israel D. J., « Introduction to Information Extraction Technology »., IJCAI-99 Turorial. 1999. http://www.ai.sri.com/~appelt/ie-tutorial/

Embley D.W. et al. « A Conceptual-Modeling Approach to Extracting Data from the Web », Proceedings of the 17th *International Conference on Conceptual Modeling* (ER'98),1998.

Eriksson H., « Using JessTab to Integrate Protégé and Jess », *IEEE Intelligent Systems*, 18(2):43-50, 2003.

FIPA, IEEE Foundation for Intelligent Physical Agents, http://www.fipa.org/, 2000.

Freitas F., Bittencourt G., « An Ontology-Based Architecture for Cooperative Information Agents », *IJCAI*, 2003.

Freitas F., Bittencourt G., « Cognitive Multi-agent Systems for Integrated Information Retrieval and Extraction over the Web », *IBERAMIA-SBIA*, 2000

Freitas F., Bittencourt G., Calmet J., « MASTER-Web: An Ontology-Based Internet Data Mining Multi-Agent System », *Second International Conference on Advances in Infrastructure for E-Business, E-Science and E-Education*, 2001.

Gruber, T. R., « Towards Principles for the Design of Ontologies Used for Knowledge Sharing », International Journal of Human and Computer Studies, 43(5/6): 907-928. 1995.

Huhns, M. N., Singh, M. P., « Distributed Artificial Intelligence for Information Systems », *CKBS-94 Tutorial*, June 15, University of Keele, UK. 1994.

Jade, Jade tutorial http://jade.tilab.com, 2006.

Jess, http://hezberg.casandia.gov/Jess, 2006.

Kushmerick N., « Gleaning the Web ». *IEEE Intelligent Systems*. EUA 1999.

Muslea I., « Extraction Patterns for Information Extraction Tasks: A Survey ». http://www.isi.edu/~muslea/RISE/ML4IE/ml4ie.muslea.ps, 1999.

Nwana N.H., « Software Agents: An Overview », *Knowledge Engineering Review*, Vol. 11, No 3, pp.1-40, Sept 1996. Cambridge University Press, 1996.

Oates T. et al, « Cooperative Information Gathering: A Distributed Problem Solving Approach ». *Computer Science Technical Report, University of Massachusetts*, 1994.

Protégé, http://Protégé.stanford.edu/, 2006

R. Gaizauskas,, Robertson A., « Coupling Information Retrieval and Information Extraction: A New Text Technology for Gathering Information from the Web ». Proceedings of *RIAO*, Computer-Assisted Information Search on the Internet. Montreal, Canada. 1997.

Satine project, http://www.srdc.metu.edu.tr/webpage/projects/satine/, 2001.

Steele R., « Techniques for Specialized Search Engines ». http://www-staff.it.uts.edu.au/~rsteele/SpecSearch3.pdf