

4132

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



15 / 8 85

Mémoire de Fin d'études Master

Filière : Informatique

Option : Ingénierie des Medias

Thème :

**Recherche d'information ubiquitaire basé agent
mobile sensible au contexte**

Encadré Par :

Mr. Ghars Hakim

Présenté par :

Belfil Hanane

Juin 2015

Résumé

L'informatique ubiquitaire est une vision du futur proche, dans laquelle un nombre croissant d'appareils (capteurs, processeurs, actionneurs) inclus dans divers objets physiques participent à un réseau d'information global. La mobilité et la reconfiguration dynamique seront des traits dominants de ces systèmes, imposant une adaptation permanente des applications. Les principes d'architecture applicables aux systèmes d'informatique ubiquitaire restent encore largement à élaborer.

Le web ubiquitaire est vu comme une synthèse de l'internet et le web ubiquitaire qui étend le Web existant, la coordination de dispositifs mobiles, la sensibilité au contexte, l'adaptation à l'hétérogénéité des réseaux et des sources d'information et également, la mise en disponibilité des nouvelles technologies de l'information et de la communication au service des utilisateurs, à n'importe quel moment, à partir de n'importe quel endroit et avec n'importe quel dispositif mobile. Cependant, rechercher de l'information pertinente, dans un volume important du Web ubiquitaire est une tâche à la fois cruciale et très complexe.

Ce mémoire s'intéresse sur la recherche d'information et plus précisément dans le web ubiquitaire et comprendre le principe de contexte pour l'amélioration de la recherche et obtenir les résultats les plus pertinent aux utilisateurs. Nous utilisons les agents mobiles pour faire la recherche d'information contextuelle pour l'amélioration des SRI, Les agents mobiles seront utilisés sur un seul niveau : agent de recherche. Chaque agent mobile de recherche sera utilisé pour visiter le site serveur cible de l'application afin de collecter des informations pour son client, ce qui permet à ce dernier d'interagir localement avec un serveur, et donc de réduire le trafic sur le réseau en ne transmettant que les données utiles.

Mots-clés : web ubiquitaire, recherche d'information, contexte, agent mobile.

DEDICACE

Je dédie ce modeste travail

Tout d'abord à mes chers parents, pour leur sens du devoir et

Soutien moral pendant toute ma vie et mes années d'études

A ma sœur Besma et son mari ilies

mes frères Adnen, Samir et sa femme Sara

Les petits Yacine et Ilina

ma grande mère et a toute ma famille de loin ou de proche

Et à tous mes amis et collègues d'étude

le groupe de Hi-tech

Et à tous ceux qui j'ai oublié de mentionner

REMERCIEMENTS

Je tiens à remercier tout d'abord DIEU le tout puissant qui
m'a donné, durant toutes ces années, la santé,
le courage et la foi pour arriver à ce jour.

Je tiens à remercier vivement mon encadreur Mr *Ghars Hakim*

pour leurs suivi, leurs aides, leurs remarques judicieuses et leurs conseils
précieux qu'ils m'ont prodigués avec gentillesse tout au long de mon travail. je remercie
également Mr *Belfarhi Brahim* pour ses conseils et son aide précieuse. je remercie
chaleureusement ceux qui m'ont soutenus de près ou de loin pour
bien accomplir ce travail. Enfin mes remerciements s'adressent
aussi à tous nos enseignants de l'informatique pour m'avoir
transmis leur savoir durant ces cinq dernières années.

-Sommaire-

Résumé

Sommaire

Liste des figures

Introduction générale

-Chapitre 01-

I. Introduction	10
II. L'informatique ubiquitaire	11
II.1. Historique	11
II.2. Définitions	13
a. L'informatique ubiquitaire	13
b. L'informatique mobile	14
II.3. Evolution de l'informatique ubiquitaire	15
II.4. Objectif de l'informatique ubiquitaire.....	16
II.5. Les systemes informatique ubiquitaires	16
II.6. Caractéristiques d'un système ubiquitaire :	18
II.7. Contraintes d'environnement à vérifier :	19
c. Interne :	19
d. Externe :	19
II.8. Avantage des systèmes ubiquitaires.....	20
II.9. Acquisition automatique du modèle de contexte.....	20
II.10. Le besoin de l'adaptation :	21
III. Le web ubiquitaire	21
III.1 Qu'est ce que le web ubiquitaire?	21
III.2 Objectif du web ubiquitaire.....	22
III.3 Intérêts du web ubiquitaire.....	22
III.4 Technologies nécessaires pour le Web ubiquitaire ?.....	23
III.5 Les particularités de l'informatique ubiquitaire.....	23
a. La mobilité.....	23
b. La miniaturisation.....	23
III.6 Les contraintes technologiques	23

a. Les infrastructures logicielles :	23
b. La sécurité	24
IV. CONCLUSION :	25

-Chapitre 02-

I. Introduction.....	27
II. La recherche d'information classique.....	28
II.1. Système de recherche d'information.....	28
II.2. Architecture générale d'un Système de Recherche d'Information.....	29
II.3. Les modèles de recherche d'information.....	31
a. Modèle booléen.....	31
b. Modèle vectoriel.....	31
c. Modèle probabiliste.....	32
III. Problème des approches classiques de RI.....	32
IV. Recherche contextuelle d'information.....	33
IV.1. Définition du contexte.....	33
IV.2. Définitions du Contexte pour la RI.....	34
IV.3. Utilisation du contexte en recherche d'information.....	34
IV.3.1. Définition du profil.....	36
IV.3.2. Contexte utilisateur.....	36
IV.3.3. Contexte physique.....	36
V. Architecture d'un SRIC.....	36
VI. RI mobile et accès à l'information ubiquitaire.....	38
VII. Modèles de RI contextuel.....	39
• Modèle de Relevance Feedback.....	40
• Modèles de RI interactive.....	40
• Adapter un modèle de RI classique.....	45
○ Modèle vectoriel.....	45
○ Modèle probabiliste.....	45
VIII. RI à base des Systèmes Multi-Agents.....	47
1. Les agents de recherche.....	47
2. Les agents de recommandation.....	48
IX. Recherche Sémantique d'information.....	48
IX.1. Définition.....	48
IX.2. Intérêt du web sémantique.....	48
IX.3. Les Ontologies.....	50
IX.3.1. Rôles des ontologies.....	51
IX.3.2. Recherche d'information guidée par les ontologies.....	51
X. Conclusion.....	52

-Chapitre 03-

I.	Introduction.....	54
II.	Système Multi agents SMA.....	55
	II.1. Définition	55
	II.2. Etude du concept d'agent	55
	II.2.1. Définition	55
III.	Caractéristiques des agents.....	55
IV.	Typologie des agents.....	56
	VI.1. Classification des agents selon la granularité.....	56
	VI.1.1. Les agents cognitifs.....	56
	VI.1.2. Les agents réactifs.....	56
	VI.1.3. les agents pro-actifs.....	57
V.	La communication entre les agents	57
	V.1. Communication par partage d'information.....	57
	V.2. Communication par envoi de messages.....	57
VI.	Les agents mobiles	57
	VI.1. Définition	57
	VI.2. Motivation	57
VII.	Structure d'un agent mobile	58
VIII.	Tolérance aux pannes	59
IX.	Pourquoi utiliser les agents mobiles ?.....	60
X.	Agent de recherche	60
	X.1. Code mobile.....	61
XI.	Conclusion.....	62

-Chapitre 04-

I.	Introduction.....	64
II.	La Conception de l'application	64
	II.1. Diagramme de cas d'utilisation.....	64
	II.2. Les fonctionnalités du système.....	66
	II.2.1. Le diagramme de classes.....	66
	II.2.2. Les diagrammes de séquences.....	67
III.	L'implémentation.....	68
	III.1. Les différents outils utilisés.....	68
	III.1.1. Java.....	68
	III.1.2. Eclipse luna.....	69
	III.1.3. Java EE.....	69
	III.1.4. Jade	70
	III.1.5. WampServer	70

III.1.6. Hibernate	70
III.1.7. Apache Tomcat	70
V. l'implémentation côté client.....	71
VI. Conclusion.....	79
Conclusion géniale.....	80
Bibliographe.....	81

-Liste des figures-

Figure 1.1 : l'évolution de l'informatique	12
Figure 1.2: Un exemple de «bureau ubiquitaire» (le SmartOffice de l'équipe PRIMA) : un monde où l'utilisateur est entouré de divers appareils avec lesquels il interagit et qui communiquent également entre eux.....	13
Figure 1.3: Illustration de l'informatique ubiquitaire intégrant de nombreux appareils hétérogènes dans le but de les faire collaborer.....	14
Figure 1.4 : l'évolution des ordinateurs dans le milieu ambiant.....	16
Figure 1.5 : Fonctionnement d'un système d'informatique ubiquitaire.....	18
Figure 2.1 : Processus général de recherche d'information.....	30
Figure 2.2 : exemple de modèle vectoriel.....	31
Figure 2.3 : scénario de recherche d'information.....	33
Figure 2.4 : la différence entre la recherche avec et sans contexte.....	34
Figure 2.5 : Architecture de base d'un SRI contextuel.....	37
Figure 2.6 : Processus de RI contextuelle.....	38
Figure 2.7 : scénario de RIC	39
Figure 2.8 : Scénario de RI interactive.....	41
Figure 2.9 : Modèle cognitif.....	42
Figure 2.10 : Modèle stratifié.....	43
Figure 2.11 : Modèle stratifié.	44
Figure 2.12: Comparaison entre le web actuel et le web sémantique.....	50
Figure 3.1: Exécution asynchrone et autonome.	58
Figure 3.2: Architecture interne des agents mobiles de recherche.....	61
Figure 3.3: Code mobile.....	61
Figure 4.1 : Diagramme de cas d'utilisation pour les acteurs de système.....	65
Figure 4.2 : Le diagramme de classe pour la recherche des Films	66
Figure 4.3 : Le diagramme de séquence pour la recherche des Films	67

Figure 4.4: fenêtre principale de site internet	71
Figure 4.5 : Interface administrateur : ajout de film.....	72
Figure 4.6 : Interface administrateur : modification des films.....	73
Figure 4.7 : Interface administrateur.....	73
Figure 4.8 : les étapes d'inscription	74
Figure 4.9 : représentation d'espace ou les opérations d'utilisateur	75
Figure 4.10 : étape d'évaluation	76
Figure 4.11 : résultat de recherche	77
Figure 4.12 : l'agent mobile avant la recherche	78
Figure 4.13 : l'agent mobile pendant la recherche se déplace vers la machine 2	79

Introduction générale

Nos environnements et nos vies sont remplis d'une multitude d'appareils informatiques qui se contentent de jouer chacun leur rôle indépendamment les uns des autres et inconscients du monde qui les entoure. L'informatique ubiquitaire vise à les orchestrer pour constituer un tout cohérent, un réseau d'appareils interconnectés et communiquant, jouant désormais leur rôle dans une pièce dont ils sont tous acteurs. Le but final de cette comédie est de tapisser l'environnement de l'utilisateur d'une couche transparente d'intelligence, capable de le comprendre et de lui être utile partout et tout le temps, sans le perturber. Les dispositifs unitaires deviennent alors des parties d'un ordinateur virtuel, intégré à l'environnement et à notre vie quotidienne, interagissant avec l'utilisateur d'une manière naturelle, presque en arrière-plan de son attention. L'informatique ubiquitaire peut être considérée comme une tendance vers l'informatisation et la miniaturisation des dispositifs électroniques favorisant l'accès aux ressources

Les volumes gigantesques des données électroniques, la diversité et l'hétérogénéité des sources d'information, durant les dernières décennies, c'est les problèmes de la recherche d'information classique nécessitent une mise à niveau de la philosophie des traitements de ces données. La recherche d'information contextuelle pour délivrer une information pertinente et appropriée au contexte de l'utilisateur qui a émis la requête. La RI contextuelle traduit précisément l'exploitation des éléments du contexte de la recherche dans l'une des principales phases de l'évaluation de requête : reformulation, calcul du score de pertinence de l'information, présentation des résultats de recherche.

À l'heure actuelle, beaucoup d'approches furent proposées en intelligence artificielle afin de résoudre plusieurs problèmes de la vie quotidienne. Souvent, ces différentes approches utilisent le concept de l'agent, qui est par définition une entité capable de percevoir son environnement et le modifier par le biais de ses interactions et ce pour atteindre ses propres objectifs.

Aujourd'hui, l'intelligence artificielle distribuée se focalise plutôt sur des applications qui organisent un ensemble d'agents en groupes, parfois hétérogènes et qui doivent interagir entre eux, afin de résoudre un problème donné. Un agent peut fournir des informations adaptées au système de RI auquel il accède.

Le document est organisé comme suit :

Le chapitre 1 : l'objectif de ce chapitre est d'apporter une présentation et une étude du domaine du *Web ubiquitaire*

Le chapitre 2 : décrit la notion de la recherche d'information classique et contextuelle avec quelque modèle de recherche.

Le chapitre 3 : l'objectif de ce chapitre et la représentation des systèmes multi agent en détaillant la notion d'agent et l'agent mobile.

Le chapitre 4 : présente deux parties la conception et l'implémentation de notre application. Et en fin une conclusion générale qui résume notre travail.

Chapitre 1

I. Introduction

Dans ces dernières années et avec l'arrivée de phénomènes la miniaturisation, de la mobilité et de l'accès sans fil, l'informatique omniprésente et invisible, appelée également informatique ubiquitaire (Ubiquitous Computing), est l'un des enjeux de demain. [1]

Les technologies de l'informatique et des réseaux sont en perpétuelle (r)évolution, les capacités de traitement et de communication de l'information croissent au même rythme que la miniaturisation des supports.

Aujourd'hui, les objets commencent à dialoguer entre eux, les réseaux de communication se construisent de manière spontanée et s'autorégulent, les interfaces homme-machine deviennent naturelles et intuitives comme la reconnaissance vocale et gestuelle, les capteurs facilitent la localisation et les échanges d'information sans contact.

Aussi bien, les systèmes mobiles ont franchi le seuil de la maturité et ont trouvé un réel usage auprès de nombreux utilisateurs ; citons par exemple le système de géolocalisation ou encore les Smartphones permettant de lire le courrier électronique tout en étant mobile (le système PUSH de Blackberry).

S'appuyant sur cette évolution technologique, l'«informatique ubiquitaire» est un nouveau domaine de recherche, appelé aussi dans la littérature académique : intelligence ambiante, informatique diffuse, informatique invisible...

Ces systèmes dits ubiquitaires permettent la mobilité de l'utilisateur ; En outre le principal intérêt de ces systèmes est d'être sensible à l'environnement de l'homme (géolocalisation, disponibilité des réseaux de communication, identification de l'utilisateur etc.). En d'autres termes c'est le système qui s'adapte à l'homme (l'homme est au centre de l'information), et pour y arriver, le système se doit être présent partout et à temps réel. D'où le nom d'intelligence ambiante.[2]

II. L'informatique Ubiquitaire

Nous allons présenter le domaine de l'informatique ubiquitaire sous forme chronologique, en montrant les racines du domaine et son évolution.

- 1991 : l'informatique ubiquitaire est introduite par Mark Weiser ;
- 1999 : Philips et l'ISTAG développent la notion d'intelligence ambiante, basée sur l'informatique ubiquitaire ;
- 2005 : un constat est fait sur l'avancement de l'informatique ubiquitaire par rapport à la vision initiale de Weiser ;
- 2008 : la vision du futur (notamment par Microsoft. [3])

II.1. Historique

L'informatique ubiquitaire ou informatique ambiante a été développée par Mark Weiser (Weiser, 1991) au cours des années 80 à Xerox Parc. Il revient sur l'évolution de l'informatique depuis les années 60, caractérisée par trois ères :

- Les mini-ordinateurs : une unité centrale partagée par un ensemble d'utilisateurs.
- Les ordinateurs personnels : une unité centrale par utilisateur.
- La mobilité : plusieurs unités centrales par utilisateur, ces unités centrales pouvant le suivre dans ses déplacements.

La miniaturisation des unités centrales, la réduction de la consommation d'énergie et la généralisation des réseaux (Wifi, 3G, CPL etc) conduit à une omniprésence de dispositifs informatiques nous accompagnant dans notre vie courante : Smartphones, console de jeux portables, PDA, ordinateurs portables . . .

La vision de Mark Weiser est que la multiplication des systèmes va changer radicalement notre façon de les utiliser. Les applications ne seront plus associées à une machine physique et surtout un écran associé. Elles vont pouvoir migrer et nous suivre au gré de nos déplacements. Les interactions vont être plus naturelles et l'ordinateur va se fondre dans notre environnement et disparaître. Il doit agir de manière analogue à des lunettes : elles jouent leur rôle sans encombrer notre esprit :

Cette idée est reprise par Norman dans la préface de son livre « The Invisible Computer » (Norman, 1999). L'auteur pense que l'ordinateur de son époque est trop intrusif, trop difficile à utiliser, et inadapté à l'être humain. L'informatique ubiquitaire peut être vue comme l'opposé de la réalité virtuelle. La réalité virtuelle met une personne à l'intérieur d'un monde créé par



Figure 1.2: Un exemple de «bureau ubiquitaire» (le SmartOffice de l'équipe PRIMA) : un monde où l'utilisateur est entouré de divers appareils avec lesquels il interagit et qui communiquent également entre eux.

[3]

II.2. Définitions

L'informatique ubiquitaire

L'informatique ubiquitaire ou omniprésente ou pervasive (ubiquité numérique) est la notion de « l'informatique partout », c'est le modèle qui suit l'ordinateur de bureau au niveau de l'interaction homme machine dans lequel le traitement de l'information a été complètement intégré dans tous les objets des activités journalières, aujourd'hui les ordinateurs se trouvent dans les véhicules, les appareils électroménagers....etc. (par exemple un environnement informatique ubiquitaire domestique pourrait relier tous les contrôles de l'éclairage et de l'environnement avec des capteurs biométriques individuels cousus dans l'habillement pour moduler en conséquence l'éclairage et le chauffage dans une chambre, sans interruption et imperceptiblement) mais ils sont souvent invisibles : les utilisateurs ne sont souvent pas avertis qu'ils sont au cours d'utilisation des services ubiquitaires. On peut donc définir

l'ubiquité numérique (ou informatique diffusé) comme une technologie invisible à des utilisateurs avec lesquels elle entretient des interactions permanentes. « exploiter les capacité d'un ordinateur nécessite aujourd'hui toute l'attention de son utilisateur. Les systèmes nomades actuels, conçu pour être utilisés par des utilisateurs mobiles, ne peuvent s'accommoder de ce type de contrainte ». [4]

L'informatique mobile

Représente la notion de « la mobilité à la fois des personnes et des machines », le problème c'est que seules quelques machines se déplacent avec leurs propriétaires, elles sont souvent très petites et possèdent généralement des ressources très limités (mémoire, batterie ... etc) mais l'avantage est qu'elles sont peu coûteuses et permettent un accès distant à pas mal de services et un échange de donnée, dans des travaux collaboratifs (Figure 1.3). L'un des défis de ce domaine est de prendre en considération l'instabilité de l'environnement, par exemple : le type et les caractéristiques des réseaux ... etc



Figure 1.3: Illustration de l'informatique ubiquitaire intégrant de nombreux appareils hétérogènes dans le but de les faire collaborer.[3]

II.3. Evolution de l'informatique ubiquitaire

Mark weister a inventé l'expression « informatique ubiquitaire » lors de ses fonctions en tant que technologue en chef du centre de recherche de Xerox Palo Alto(PARC) et exactement en 1988. A la foi seul avec Ohn Seely Brown, Weister a écrit certains des premiers écrit sur le sujet, en grande partie le définissant hors de ses points principaux.

L'informatique ubiquitaire et mobile connaît depuis le début des années 1990 un essor important. Deux aspects principaux contribuent à cet essor. D'une part le développement des calculateurs mobiles : ordinateur portable, assistants numériques, téléphones mobiles, outils de navigation GPS, qui sont autant de formes des calculateurs qui peuvent être embarqué par un utilisateur aujourd'hui. D'autre part, les infrastructures de communication pour terminaux mobiles sont également en plein essor, en particulier les réseaux téléphoniques cellulaires (voir *Figure 1.4*).

Parallèlement à ces développements, le web s'est imposé comme le fédérateur des systèmes d'information grande échelle, en offrant une interface universelle. En 2004, le « Massachusetts Institute of Technology » ont créé un système de configuration dynamique des environnements ubiquitaires : « Oxygen ». Aujourd'hui des laboratoires dans plus de 22 pays s'intéressent au développement de tels types de projet.

Technologiquement, l'informatique mobile offre aujourd'hui le potentiel pour permettre un accès omniprésent à des systèmes d'informations grande échelle come le web. Cependant offrir cet accès n'est pas suffisant. [5]

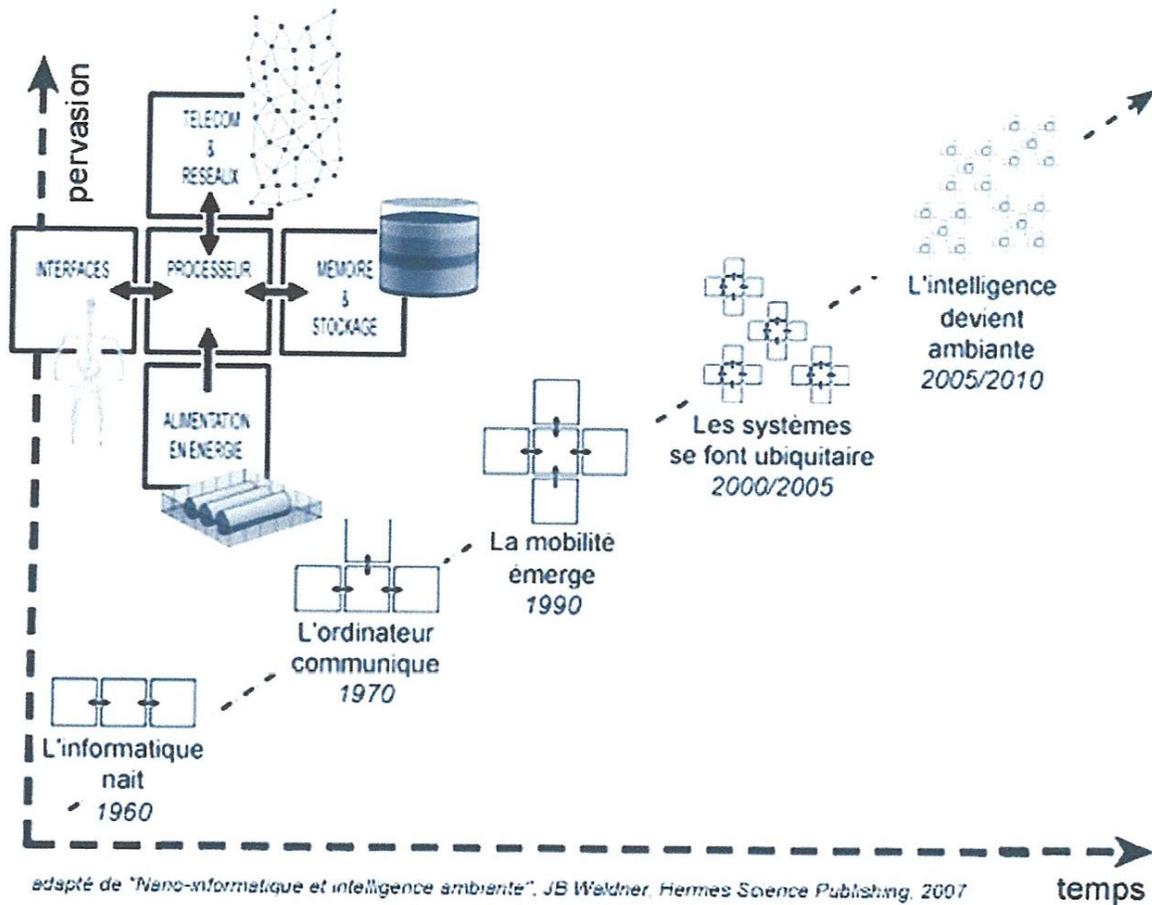


Figure 1.4 : l'évolution des ordinateurs dans le milieu ambiant.

II.3. Objectif de l'informatique ubiquitaire

L'objectif de l'informatique ubiquitaire est de réaliser l'intégration transparente des environnements numériques au monde physique. Cette intégration doit offrir aux utilisateurs des modes d'interaction plus naturels avec leur milieu. A terme ces interactions devraient avoir lieu sans que l'utilisateur ait conscience d'utiliser les services de calculateurs avoisinants : on parlera alors d'interactions invisibles. Et comme ces terminaux ubiquitaires sont également informatisés (par exemple liés à un réseau informatique) le but est d'introduire les capacités du traitement informatique invisiblement dans la vie humaine, de façon que les ordinateurs couvrent automatiquement les routines de la vie. [3]

II.4. Les systèmes informatique ubiquitaire

Les systèmes d'informatique ubiquitaire permet d'automatiser certaines tâches quotidiennes grâce aux différents objets communicants disponibles. La figure 1.4 représente

schématiquement le fonctionnement d'un tel système. Le système collecte tout d'abord des informations de l'environnement physique (température ambiante, lumière du soleil, bande passante, présence d'un utilisateur) à partir des objets communicants capables de les capturer. De tels objets communicants sont appelés des capteurs. Ces derniers peuvent être aussi bien physiques que logiciels. Les informations collectées sont ensuite interprétées, filtrées et agrégées par diverses applications, afin de les enrichir de données contextuelles dans le but d'obtenir et de partager des informations de plus haut niveau. À partir de ces dernières, certaines applications peuvent décider des actions à entreprendre (allumer une lumière, déclencher une alarme, modifier un statut, afficher une information) par les objets communicants capables d'agir sur l'environnement physique. De tels objets communicants sont appelés des actionneurs. Un objet communicant peut être à la fois capteur et actionneur. Les exemples de systèmes d'informatique ubiquitaire sont nombreux. Nous pouvons citer le traçage de l'ensemble des produits matériels, la détection précoce par réseaux de capteurs des accidents écologiques (incendies), la gestion intégrée des bâtiments (consommation d'énergie, sécurité) ou encore la surveillance des personnes âgées peu autonomes. Considérons l'exemple d'un système de sécurité pour se protéger contre les intrusions. Ce système est responsable de détecter et de signaler toutes les tentatives d'intrusion d'un individu dans un bâtiment. Pour cela, il peut nécessiter différents types d'objets communicants, tels que des détecteurs de mouvements pour collecter les coordonnées d'une intrusion, un agenda pour connaître les plages horaires de surveillance du bâtiment, diverses applications pour analyser les données collectées et déterminer si une intrusion a bien lieu afin d'y réagir, des alarmes sonores pour alerter les alentours, des caméras vidéo pour filmer l'intrusion, une plate-forme de téléphonie pour envoyer la vidéo de l'intrusion sur le téléphone portable du propriétaire, *etc.* Cet exemple donne une bonne idée de la taille et de la complexité qu'un système d'informatique ubiquitaire peut avoir, coordonnant une multitude d'objets communicants et requérant une expertise étendue dans de nombreux domaines, comme la programmation distribuée ou encore les télécommunications.

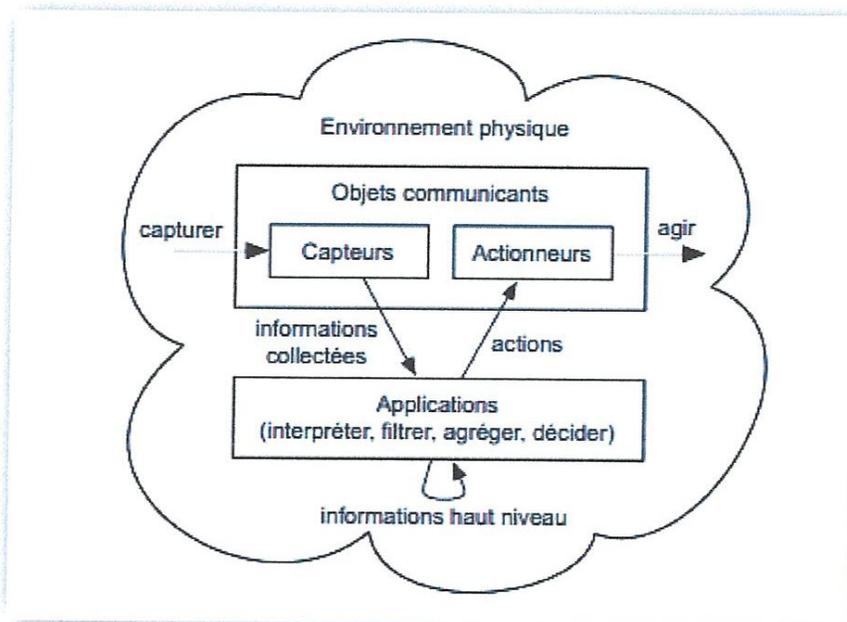


Figure 1.5 : Fonctionnement d'un système d'informatique ubiquitaire

Les systèmes d'informatique ubiquitaire reposent sur deux notions centrales : le contexte à partir duquel ils réagissent et s'adaptent aux différents changements de l'environnement, et la découverte de services pour trouver quels objets communicants sont disponibles dans l'environnement. [5]

II.5. Caractéristiques d'un système ubiquitaire

Voici les caractéristiques principales d'un système ubiquitaire :

1. L'utilisation de dispositifs légers, et facilement manipulable : ce qui causait en effet la création de l'informatique ubiquitaire, car l'ubiquité elle-même nécessite la réduction de la taille d'équipements et leurs invisibilités, ainsi que la facilité de leur manipulation, pour pouvoir généraliser un système informatique pour qu'il soit en fait utilisable par des utilisateurs qu'ils ne sont généralement des experts en informatique ce qui nécessite la normalisation des services pour les rendre auto-manipulables par es actions normales des personnes(exemple :l'entrée a une chambre d'éclanche la lumière)
2. La nature pervasive et sans fil : des dispositifs qui demandes des architectures réseaux supportant une configuration automatique(le terme « pervasif » fait reference à la définition d'informatique pervasive. Celle-ci fournit un accès approprié à l'information et aux applications a travers des dispositifs qui possédant la capacité de fonctionné au moment et l'endroit nécessaire)

3. La haute distribution : a la fois des services et de leurs système de gestion, cette distribution doit être assez puissant et bien contrôlé pour éviter la faiblesse ou la perte des services sur les parties du système ou il y a embouteillage d'accès ou une faiblesse de couverture matérielle.
4. L'autonomie de l'environnement : les environnements ubiquitaires n'obéissent généralement à aucun protocole de gestion ou de contrôle, bien sur car il s'agit de la vie quotidienne des utilisateurs, c'est pour cela que l'environnement et un point axial dans la préparation et le développement des systèmes ubiquitaires ce qui nécessite la prise en charge sérieuse de quelque contrainte d'environnement.

II.6. Contraintes d'environnement à vérifier

Il existe deux classes de contraintes à couvrir ; interne, propre à chaque terminal lui-même, et externe dépendantes des autres terminaux équipements et service qui entourent chaque terminal :

Interne :

- **Les ressources matérielles limitées** : ce qui peut influencer la transportation ou l'exécution normale d'un service sur un terminal, ce problème déclenche généralement un protocole de gestion particulier qui peut retarder l'arrivée d'un service donné à un tel terminal, ou même la défaillance du service.
- **La surcharge de fonctionnalités sur un terminal** : qui cause en effet des problèmes sur les deux cotés en même temps : le serveur et le terminal, provoquant dans le premier cas des problèmes de fonctionnement sur le plan de la performance, et dans le deuxième cas l'impossibilité d'effectuation de quelque fonctionnalité particulièrement surchargées.
- **Le problème de stabilité et de cohérence d'application** : surtout parce qu'elles sont conçus spécialement pour être stable face à des cas divers de problèmes d'incohérence et de chute imprévues des systèmes ubiquitaires. Ces problèmes dépendent généralement de l'architecture du terminal.

Externe :

- **La mobilité des terminaux** : ce qui demande l'assurance de la bonne existence des services tout au tour de la zone de mobilité des équipements.
- **L'hétérogénéité des équipements concurrents** : qui causera des problèmes d'interaction entre eux ainsi que le non standardisation des services au niveau des serveurs (diversité des services selon les terminaux)

- **La sécurité des services** : les environnements ubiquitaires sont souvent facilement accessibles par des terminaux inconnus par les systèmes qui gèrent les services, il est donc indispensable de sécuriser ces derniers contre toute utilisation non désirée et contre tout chevauchement possible.
- **Des utilisateurs ayant des styles d'interaction particuliers** : les styles d'interaction des utilisateurs sont des fois purement rares, par exemple pour le cas d'un aveugle, les services basés sur des interfaces visuelles seront inaccessible par lui et comme la majorité des systèmes ubiquitaires sont conçus pour un fonctionnement générale, ce type de cas cause l'inutilité des services ubiquitaires pour des parties spécifiques d'utilisateurs. Ce qui oblige les développeurs à bien estimer les besoins des futurs utilisateurs d'un système donné.

Ces contraintes affrontées lors du développement des systèmes ubiquitaire et mobiles posaient également des problèmes au niveau de leur utilité, ce qui obligeait les développeurs à penser à créer des applications ubiquitaires adaptables. [7]

II.7. Avantage des systèmes ubiquitaires :

Voici les résultats issus de l'intégration de l'ubiquité numérique dans la vie quotidienne, surtout d'un point de vue conceptuel :

- Une production de logiciels plus complexe,
- Capacité et ressources matérielles non prévisible,
- Caractéristiques réseau non prévisibles,
- Augmentation des coûts,
- Besoin d'experts rares et chers,
- Diminution de la fiabilité des logiciels produits,
- Variation spatiale et temporelle de l'environnement,
- Un bon avantage sur la simplicité des tâches de la vie quotidienne dans des domaines spécifiques : les stations routières et les gares de trains par exemple.

II.8. Acquisition automatique du modèle de contexte

Un contexte exploité par une application d'ubiquité numérique peut comprendre des attributs tels que la présence ou l'absence d'accès réseau, les périphériques accessibles (imprimante, écran...etc), des informations relatives aux personnes environnantes : identité, préférences, localisation...etc.

Des informations relatives à l'environnement physique : conditions climatiques, niveau de bruit, luminosité...etc

Pour illustrer cette notion de contexte, prenons l'exemple d'un musée équipé d'un système de guide virtuel ou chaque visiteur est équipé d'un ordinateur portable de type PDA peut y lire les informations relatives aux œuvres situées à sa proximité. Pour le faire, le système doit être capable de détecter devant qu'elle(s) œuvre(s) se trouve le visiteur. L'ensemble des œuvres à proximité d'utilisateur constitué, pour ce type de service, le contexte utile à l'application.

L'intelligence ambiante propose de se placer dans un système ubiquitaire, de détecter le contexte de l'utilisateur et de fournir des services dépendants de la situation courante des usagers. La question qui se pose alors est comment trouver les associations entre les situations et les services rendus. Ces correspondances constituent ce que nous appelons le modèle de contexte. Le cadre d'application étant l'informatique ubiquitaire, les situations doivent décrire, d'une manière suffisamment précise pour être pertinente, une partie du monde réel, avec toute sa complexité et son dynamisme. Un système véritablement adapté à l'utilisateur choisit un service en fonction du contexte de ce dernier, mais également en fonction de l'utilisateur lui-même. Dans une même situation, différentes personnes ont différentes préférences quant aux actions du système. Comment alors choisir ces associations entre contexte et services de manière personnalisée à chaque utilisateur ?

Faire spécifier le modèle de contexte par un expert ne permet pas de le rendre adapté à chaque utilisateur, ni de le faire évoluer avec l'environnement ou avec les changements dans les préférences de l'utilisateur. Le faire spécifier par l'utilisateur est une tâche trop lourde et va à l'encontre de l'idée d'un système qui sait se faire oublier. Par contre, laisser à l'utilisateur le choix d'en spécifier une partie permet de l'engager et de lui donner un sentiment d'implication dans le système. Il ne faut toutefois pas attendre une granularité fine dans la partie du modèle définie par l'utilisateur. [3]

II.9. Le besoin de l'adaptation

C'est le besoin de créer des applications qui agissent d'une façon autonome face aux problèmes d'insuffisance ou de défaillance causés par des variations des contextes ou des utilisateurs des systèmes ubiquitaires. Le but est d'assurer des procédures d'autoréparation pour garder le système en fonction dans des cas imprévus par son réalisateur. Généralement une assistance d'entretien fréquente est nécessaire pour garder le système (matériel et logiciel) dans les marges normales de l'auto-adaptable. [6]

III. Le web ubiquitaire

III.1. Qu'est ce que le web ubiquitaire?

Le Web ubiquitaire cherche à élargir les capacités des navigateurs Web afin de permettre de nouveaux types d'applications Web, notamment celles nécessitant une coordination avec d'autres terminaux et une adaptation dynamique de l'utilisateur, des capacités des terminaux, et du contexte. Ces applications seront capables d'exploiter les services en réseau pour élargir les capacités des terminaux. Les utilisateurs pourront alors se focaliser sur ce qu'ils font et non sur les terminaux eux-mêmes. La mobilité de ces applications permettra à toute personne de continuer à travailler ou à jouer tout en passant en douceur d'un terminal à l'autre. [1]

III.2. Objectif du web ubiquitaire

Le Web ubiquitaire fournira personnes ayant accès à chaque fois et partout où ils se trouvent, avec des applications qui adaptent dynamiquement aux besoins de l'utilisateur, des capacités des terminaux et des conditions environnementales. La mobilité de l'application permettra aux utilisateurs de basculer de façon transparente entre les périphériques tout en continuant d'accéder aux mêmes applications. Limitations de l'appareil seront contournées en étant couplé à d'autres appareils et en exploitant les services en réseau dans le cadre d'applications distribuées. En tant qu'utilisateurs, nous serons en mesure de choisir la façon dont nous interagissons avec ces applications en fonction de nos besoins actuels et les caractéristiques des appareils que nous utilisons. En un sens, le Web va disparaître, car il devient omniprésente et pris pour acquis, disparaissant ainsi dans l'arrière-plan de l'infrastructure mondiale de l'informatique et réseau. [6]

III.3. Intérêts du web ubiquitaire

Ce qui rend le Web si utile pour les développeurs d'applications est la facilité avec laquelle les applications sont créées par l'utilisation combinée de balises, de graphiques, de feuilles de style et de scripts. Le Web ubiquitaire facilitera le développement d'applications distribuées en présentant des abstractions claires aux développeurs Web souhaitant accéder aux capacités des terminaux et aux services de communication. La découverte et la description des ressources seront essentielles à la création d'applications du Web ubiquitaire. L'utilisation des URI (adresses Web) pour nommer les terminaux, les services et les sessions permettront l'emploi de métadonnées riches (le Web sémantique) pour la découverte de ressources, intervenant à travers les différents réseaux et exploitant la nature distribuée du World Wide Web.

Les applications du Web ubiquitaire permettront d'identifier des ressources et de gérer celles-ci dans le cadre de sessions temporaires ou permanentes. Pour cela, il faudra une infrastructure plus souple que les solutions de rechange actuelles reposant sur les « cookies » et sur les informations faisant partie des URI et indiquant les sessions. Les ressources peuvent être distantes comme dans le cas d'une imprimante ou d'un projecteur réseau, ou bien alors locales, comme dans le cas du niveau de batterie d'un terminal, de l'intensité du signal réseau

et de son niveau sonore. Les ressources ne se limitent pas aux matériels, mais peuvent être également des services, telles que la reconnaissance de la parole, la traduction des langages naturels et l'identification géographique d'un terminal. [1]

III.4. Technologies nécessaires pour le Web ubiquitaire ?

Ce qui rend le Web si utile pour les développeurs d'applications est la facilité avec laquelle les applications sont créées par l'utilisation combinée de balises, de graphiques, de feuilles de style et de scripts. Le Web ubiquitaire facilitera le développement d'applications distribuées en présentant des abstractions claires aux développeurs Web souhaitant accéder aux capacités des terminaux et aux services de communication. La découverte et la description des ressources seront essentielles à la création d'applications du Web ubiquitaire. L'utilisation des URI (adresses Web) pour nommer les terminaux, les services et les sessions permettront l'emploi de métadonnées riches (le Web sémantique) pour la découverte de ressources, intervenant à travers les différents réseaux et exploitant la nature distribuée du World Wide Web.

Les applications du Web ubiquitaire permettront d'identifier des ressources et de gérer celles-ci dans le cadre de sessions temporaires ou permanentes. Pour cela, il faudra une infrastructure plus souple que les solutions de rechange actuelles reposant sur les « cookies » et sur les informations faisant partie des URI et indiquant les sessions. Les ressources peuvent être distantes comme dans le cas d'une imprimante ou d'un projecteur réseau, ou bien alors locales, comme dans le cas du niveau de batterie d'un terminal, de l'intensité du signal réseau et de son niveau sonore. Les ressources ne se limitent pas aux matériels, mais peuvent être également des services, telles que la reconnaissance de la parole, la traduction des langages naturels et l'identification géographique d'un terminal. [8]

III.5. Les particularités de l'informatique ubiquitaire

- a) **La mobilité :** la différence essentielle entre l'informatique actuelle et l'informatique ubiquitaire est le besoin de mobilité, et par conséquent la prise en compte de l'environnement. En effet, une particularité de l'informatique ubiquitaire est le besoin de communication d'un objet avec son environnement, ceci dans le but de pouvoir coopérer avec les objets qui l'entourent et ainsi accéder aisément à l'information.
- b) **La miniaturisation :** avec l'augmentation de l'utilisation de l'utilisation des systèmes ubiquitaire il est préférable de rendre les technologies utilisées aussi invisibles que possible, soit par une invisibilité réelle via la miniaturisation des technologies, soit par une invisibilité mentale via la familiarisation de l'utilisateur avec l'outil.

III.6. Les contraintes technologiques

a) Les infrastructures logicielles

Dans un premier temps, pour permettre la prise en compte des contraintes liées à ces environnements ubiquitaires, il est nécessaire de créer de nouvelles méthodes de conception d'applications ainsi que de nouvelles infrastructures logicielles. Elles doivent assurer le bon fonctionnement et de qualité de service dans leur comportement et dans les services offerts.

Une autre facette de l'informatique ubiquitaire est relative aux ressources des machines. La plupart des petits dispositifs disposent de peu de ressources. C'est le cas aussi bien en terme de puissance de traitement (CPU), de mémoire, d'autonomie énergétique ou encore de connectivité à un réseau. Cette caractéristique fondamentale doit être prise en compte dans les infrastructures logicielles de l'informatique ubiquitaire.

Il faut ainsi penser à des contraintes telles que la mobilité des utilisateurs, les déconnexions fréquentes, l'autonomie limitée des appareils, etc. Cela amène à repenser les infrastructures logicielles pour les rendre adaptables dynamiquement, configurables et auto administrables en fonction de l'environnement dans lequel elles se trouvent. Ces infrastructures devront donc prendre en compte la répartition, l'hétérogénéité, la mobilité mais également les ressources limitées des supports matériels.

b) La sécurité

Pour que les services fournis par ces nouvelles technologies soient acceptés et adoptés par le plus grand nombre, ils doivent être sûrs et de confiance. En rendant l'information plus facilement accessible et en multipliant les moyens souvent transparents de son acquisition, informatique ubiquitaire et intelligence ambiante induisent en effet de nouveaux risques au regard de la confidentialité des données. Plus généralement, préserver la confidentialité de données personnelles distribuées dans une large variété de sources (mobiles, objets intelligents mais aussi bases de données d'entreprises, bases de données publiques, bases de données de sites de e-commerce) se révèle un enjeu majeur pour la communauté bases de données.

Plusieurs contraintes sont donc à prendre en compte dans la conception et la validation de ces nouvelles applications, en particulier vis-à-vis de la sûreté de fonctionnement. De nombreuses recherches portent donc sur ce sujet. Ces dernières permettent de résoudre un problème doublement important :

Tout d'abord, la gestion de données ubiquitaire introduit des problèmes spécifiques vis à vis de la préservation de la vie privée.

D'autre part, apporter une réponse à la gestion de données embarquées dans des calculateurs sécurisés (cartes à puce, tokens électroniques) permet d'envisager de nouveaux modèles de

sécurisation des systèmes d'information. Par exemple, une base de données distante peut être sécurisée en déléguant une partie de la gestion des données à un composant sécurisé. [1]

IV. CONCLUSION

L'informatique ubiquitaire est une vision du futur proche, dans laquelle un nombre croissant d'appareils (capteurs, processeurs, actionneurs) inclus dans divers objets physiques participent à un réseau d'information global. La mobilité et la reconfiguration dynamique seront des traits dominants de ces systèmes, imposant une adaptation permanente des applications. Les principes d'architecture applicables aux systèmes d'informatique ubiquitaire restent encore largement à élaborer.

Dans la vision ubiquitaire, l'informatique sort de l'ordinateur pour s'intégrer directement dans l'environnement, qui devient donc ubiquitaire. Le but de cette idée est d'élargir les possibilités de l'informatique, de faire en sorte que l'ordinateur profite à l'utilisateur à tout moment lorsqu'il se trouve dans cet environnement. Pour aller plus loin, cet ordinateur ambiant offre des capacités d'interaction plus naturelles, ce qui le rend transparent et utilisable sans effort pour les personnes.

L'informatique ubiquitaire a donc pour objectif d'intégrer les technologies informatiques au quotidien de l'homme le plus simplement possible. Plus précisément, elle devrait rendre familier et instinctif l'outil informatique et en faciliter l'utilisation dans de nombreux domaines tels que l'information, la formation, le transport ou encore la médecine.

Avec l'évolution du domaine informatique, il est devenu nécessaire d'embarquer le programme chargé de l'adaptation dans les applications elles même. C'est-à-dire qu'un système embarqué est un système informatique qui fait partie (intégré) d'un autre système automatique, pas seulement logiciel mais aussi physique, ce qui augmente l'effort de la reconnaissance de l'environnement ou s'exécute un tel système.

Chapitre 2

I. Introduction :

La Recherche d'Information (RI) est un domaine qui s'intéresse à la structure, à l'analyse, à l'organisation, au stockage, à la recherche et à la découverte de l'information. Le défi est de pouvoir, parmi le volume important de documents disponibles, trouver ceux qui correspondent au mieux à l'attente de l'utilisateur. L'opérationnalisation de la RI est réalisée par des outils informatiques appelés Systèmes de Recherche d'Information (SRI), ces systèmes ont pour but de mettre en correspondance une représentation du besoin de l'utilisateur avec une représentation du contenu des documents au moyen d'une fonction de correspondance. L'objectif de la RI contextuelle est de mieux répondre aux besoins en information de l'utilisateur tout en intégrant le contexte de recherche dans la chaîne d'accès à l'information. De même la RI sémantique est généralement guidée par les ontologies qui sont « une spécification explicite et formelle d'une conceptualisation partagée ». L'utilisation d'ontologie en RI a pour finalité de spécifier des connaissances qui seront interprétables à la fois par l'homme et par la machine. Le contexte et la sémantique constituent les deux notions principales représentant les mécanismes que nous utilisons comme support à la modélisation de nos propositions.

II. La recherche d'information classique

De manière générale, la recherche dans un SRI consiste à comparer la représentation interne de la requête aux représentations internes des documents. La requête est formulée, par l'utilisateur, dans un langage de requêtes qui peut être le langage naturel, un langage à base de mots clés ou le langage booléen. Elle sera transformée en une représentation interne équivalente, lors d'un processus d'interprétation. Un processus similaire, dit indexation, permet de construire la représentation interne des documents de la base documentaire. [9]

II.1. Système de recherche d'information

D'après la définition de Salton en 1968, La recherche d'information (RI) est une branche de l'informatique qui s'intéresse à l'acquisition, l'organisation, le stockage, la recherche et la sélection d'information (Boughanem, 2006). Alors, l'objectif de la recherche d'information (RI) est de concevoir des systèmes (nommés désormais SRI pour systèmes de recherche d'information) capables de retrouver parmi un ensemble de documents ceux qui répondent précisément au besoin d'un utilisateur. Ce besoin est généralement formulé par le biais d'une requête en langage naturel (Moreau, 2006). [10]

C'est une démarche faite par un utilisateur pour obtenir, à l'aide du système de recherche d'informations (SRI), les informations (ou les références vers les informations) qui peuvent répondre à son besoin.

Un Système de Recherche d'Information (SRI) : est un système informatique qui facilite l'accès à un ensemble de documents (corpus), pour retrouver ceux dont le contenu correspond le mieux à un besoin d'information d'un utilisateur.

Les SRI et les modèles sous-jacents se basent donc sur trois notions clés : le document, le besoin et la correspondance.

- Document: Un document peut être un texte, une page WEB, une image, une bande vidéo, etc. Dans notre contexte, nous appelons document toute unité qui peut constituer une réponse à une requête d'utilisateur.
- Requête: Une requête exprime le besoin d'information d'un utilisateur.
- Correspondance: Le but de la RI est de retrouver seulement les documents pertinents (qui correspondent le mieux à la requête). [6]

II.2. Architecture générale d'un Système de Recherche d'Information

Un système de recherche d'information intègre trois fonctions principales représentées schématiquement par le processus en U de recherche d'information [Belkin et al., 92]. La Figure 2.1 illustre l'architecture générale d'un système de recherche d'information.

Il s'agit principalement du processus de représentation et du processus de recherche :

1. **Processus de représentation** : un processus de représentation a pour rôle d'extraire d'un document ou d'une requête, une représentation paramétrée qui couvre au mieux son contenu sémantique. Ce processus de conversion est appelé *indexation*.
2. **Processus de recherche** : il représente le processus du noyau d'un SRI. Il comprend la fonction de décision fondamentale qui permet d'associer à une requête, l'ensemble des documents pertinents à restituer. [11]

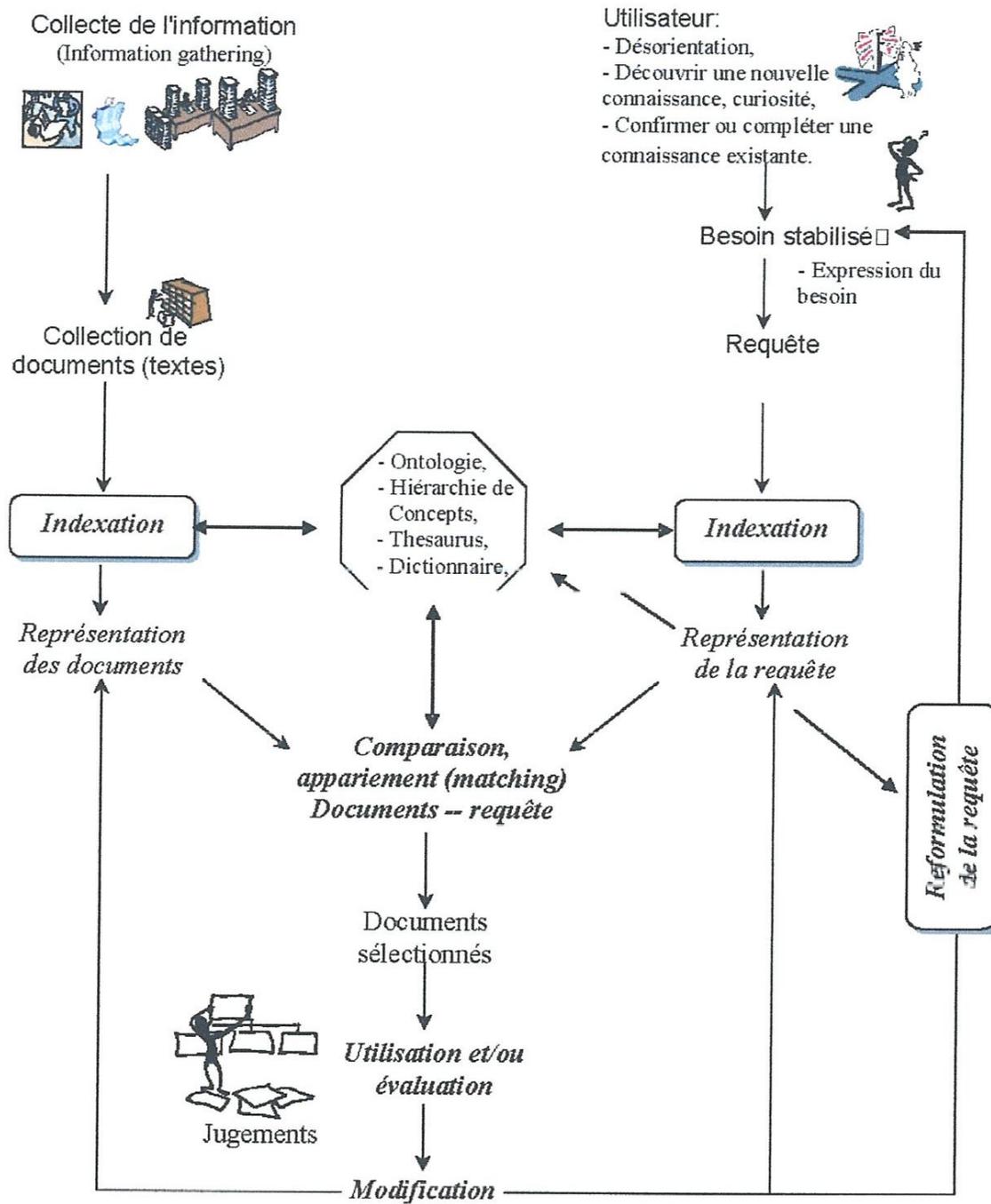


Figure 2.1 : Processus général de recherche d'information.

II.3. Les modèles de recherche d'information

- Modèles fondés sur la théorie des ensembles : Modèle booléen
- Modèles algébriques : Modèle vectoriel
- Modèles probabilistes : Modélisation de la notion de "pertinence"

a. Modèle booléen :

Le modèle booléen est basé sur la théorie des ensembles. Dans ce modèle, le document est représenté par un ensemble de termes. La requête est représentée par un ensemble de mots clés reliés par des opérateurs booléens (AND, OR et NOT)

Exemple :

`(Auteur="Sartre" or Auteur="Camus") and
(Titre~"Mythe" or Titre~"étranger")`

b. Modèle vectoriel :

Documents et requêtes sont exprimés comme des vecteurs de termes

Les documents sont classés en fonction de leur proximité avec la requête (Cosinus).

Exemple : représentation de deux documents (d_1 et d_2) et d'une requête (q) dans un espace vectoriel. La proximité de la requête aux documents est représentée par les angles α et θ entre les vecteurs. d_1 est le document le mieux classé

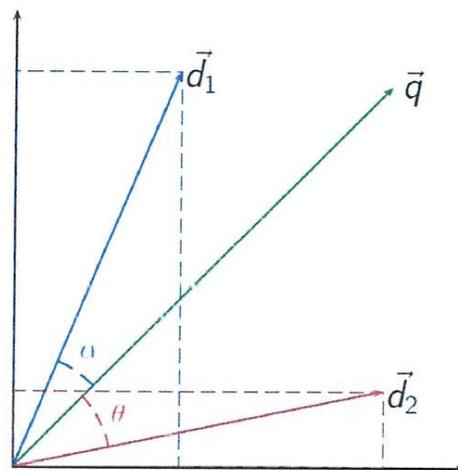


Figure 2.2 : exemple de modèle vectoriel.

c. Modèle probabiliste :

Le premier modèle probabiliste a été proposé par Maron et Kuhns [Maron et al, 1960] le principe de base consiste à présenter les résultats d'un SRI dans un ordre basé sur la probabilité de pertinence d'un document vis-à-vis d'une requête. Une requête utilisateur notée R et un document D , le modèle probabiliste tente d'estimer la probabilité que le document D appartienne à la classe des documents pertinent. [13]

Probability Ranking Principle (Robertson 77)

R : D est pertinent pour Q

$\neg R$: D n'est pertinent pour Q

Le but : estimer ;

- $P(R/D)$: probabilité pour le document .
- D de faire partie des documents pertinents pour Q $P(\neg R/D)$.

$$\text{si } \frac{P(R/D)}{P(\neg R/D)} > 1 \text{ ou si } \log \frac{P(R/D)}{P(\neg R/D)} > 0 \text{ alors } D \text{ est pertinent}$$

III. Problème des approches classiques de RI

"Localiser les termes que d'autres ont utilisé dans le document que vous cherchez est le problème central en RI" [Blair, 90]. [14]

V. Recherche contextuelle d'information :

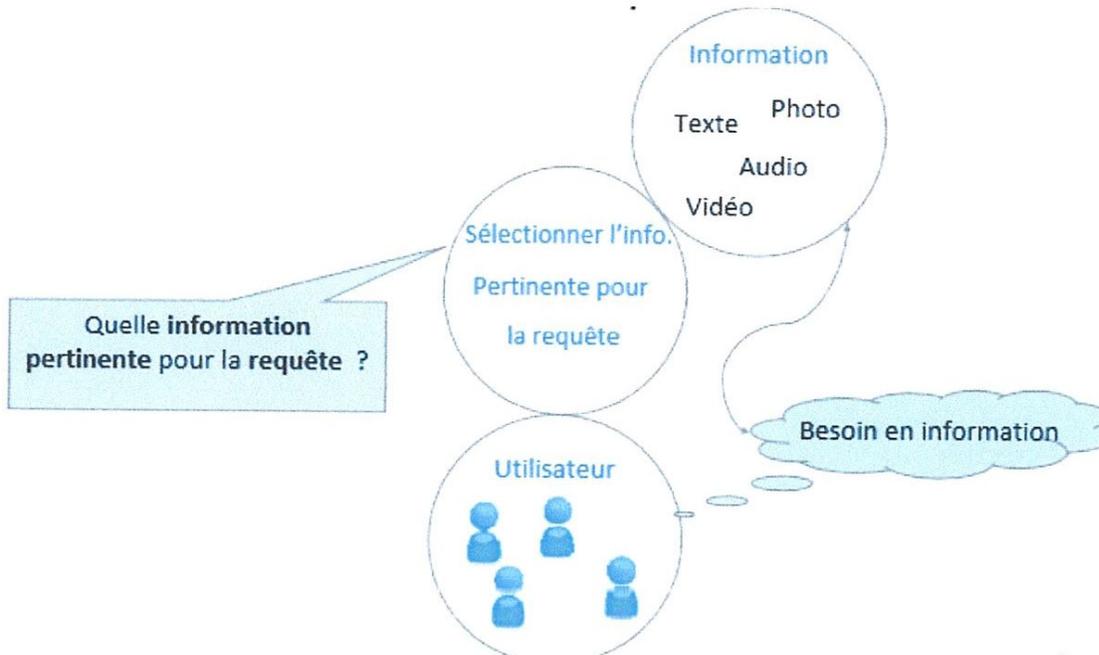


Figure 2.3 : scénario de recherche d'information.

Les approches en RI se sont orientées vers une nouvelle génération de systèmes de recherche basés sur l'accès contextuel et sémantique à l'information. Le domaine de la RI contextuelle est apparu récemment comme une priorité, son objectif est d'exploiter le contexte de l'utilisateur ainsi que les connaissances liées à la requête dans le but de répondre mieux aux besoins en information de l'utilisateur. De même l'émergence du Web sémantique a favorisé les travaux de recherche dans le domaine de la RI sémantique. Le but est d'explicitier la connaissance contenue dans les sites Web et de la formaliser afin que les agents de recherche d'information puissent l'exploiter via des mécanismes d'inférences et fournir de meilleures réponses au besoin de l'utilisateur. [13]

V.1. Définition du contexte

[Allen 97, Sonnenwald 99, Cool 01] précisent que le contexte couvre des aspects larges tels que l'environnement cognitif, social et professionnel dans lesquels s'inscrivent des situations liées à des facteurs tels que le lieu, le temps et l'application en cours.

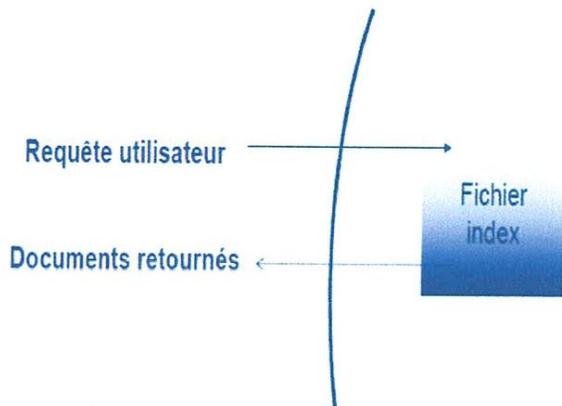
C'est le sens générique du contexte qui a été largement exploré cette dernière décennie en RI contextuelle [Lawrence 00, Quiroga 02, Rulhven 03b, Ingwersen 04, Bottraud 04].

Même si les auteurs ne convergent vers une même définition, on retrouve toutefois des dimensions descriptives communes telles que :

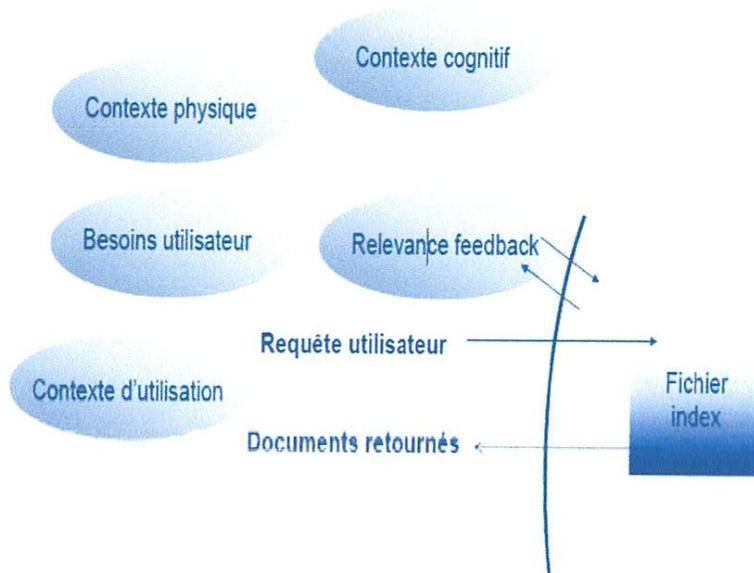
- l'environnement cognitif,
- le besoin mental en information,
- l'interaction liée à la recherche d'information

V.2. Définitions du Contexte pour la RI

- Dimensions du contexte retenues pour ce cours :
 - Contexte utilisateur/Profils utilisateur
 - Contexte cognitif (interaction)
 - Contexte physique mesuré par des capteurs environnementaux
- Comment inclure le contexte en Recherche d'Information ?



Scénario de RI sans contexte



Scénario de RI avec contexte

Figure 2.4 : la différence entre la recherche avec et sans contexte. [12]

V.3. Utilisation du contexte en recherche d'information

En recherche d'information le contexte peut être utilisé à trois stades différents selon l'avancement du processus de recherche lui-même. Il peut donc être considéré au début du processus de recherche, au cours du processus de recherche, ou encore à la fin du processus de recherche.

1. Au début du processus de recherche

Le contexte peut être utilisé dans une étape de pré-recherche pour résoudre le problème de l'ambiguïté des termes dans la requête et améliorer ainsi la qualité des résultats retournés par le système. On peut aider l'utilisateur dans la formulation de sa requête en lui demandant de préciser, selon le contexte de la recherche en cours, le sens d'un terme ambigu en utilisant un thesaurus ou une ontologie. Nous citons dans cette catégorie les travaux de [Navigli, 03] qui utilisent des ontologies avec des relations d'équivalence et de subsomption afin d'extraire les termes à rajouter à la requête initiale.

Une autre façon plus simple d'utiliser le contexte dans une phase de pré-recherche est de l'utiliser dans l'introduction des contraintes de types booléens sur les mécanismes et les algorithmes de recherche d'information existant. Ces algorithmes peuvent considérer également le contexte spatio temporel où les valeurs continues peuvent être décrites de manière non précise à des niveaux de granularité différents [Tao, 2003]. Par exemple, un événement peut avoir lieu à 9h57, à 10h environ ou dans la matinée. Dans ce cas, le contexte peut servir pour le choix de la représentation appropriée.

2. Pendant le processus de recherche

Le contexte peut également être considéré au niveau des interactions avec le système. En effet dans un processus de recherche d'information, c'est l'interaction qui rend possible l'exploitation réelle de l'ensemble des résultats une fois affichés. L'utilisateur est particulièrement habile à extraire des informations d'un environnement qu'il contrôle directement et activement par rapport à un environnement qu'il ne peut qu'observer de manière passive, [Lopes, 2009]. Le contexte à ce niveau dépend de l'action de l'utilisateur dans une situation donnée, du feedback, des jugements de pertinence qui sont relatives aux caractéristiques des différentes situations des utilisateurs, des stratégies de recherches multidimensionnelles et d'autres pratiques informationnelles durant la recherche d'information.

3. A la fin du processus de recherche

Le contexte peut enfin être considéré dans une phase de post-recherche, et cela après avoir obtenu des résultats en utilisant le principe de réinjection de pertinence. L'idée de cette technique est de réaliser une première recherche à l'aide des seuls termes de la requête. L'utilisateur peut alors indiquer quels sont, parmi les meilleurs documents issus de cette première recherche, ceux qui sont pertinents et ceux qui ne le sont pas, et le système utilise cette information pour affiner la recherche en modifiant les poids des termes de la requête par des méthodes d'apprentissage automatique comme dans les travaux de [Lin, 2006]. Une autre façon d'utiliser le contexte avec la réinjection de pertinence a été proposée plus récemment

dans nos travaux [Bouramoul, 09-b] ; où nous proposons une reformulation contextuelle des requêtes à base de profils utilisateur en utilisant d'une manière automatique la notion du contexte statique et contexte dynamique afin de minimiser l'intervention de l'utilisateur dans le processus de reformulation. [13]

V.3.1. Définition du profil

Un profil utilisateur doit représenter l'ensemble des centres d'intérêts, des préférences, des connaissances ou des habitudes de l'utilisateur.

Du point de vue standardisation, il existe à ce jour deux propositions : OPS (Open Profiling Standard) [StCr 98] qui favorise la réutilisation de profils d'une application à l'autre en proposant un modèle de représentation et P3P proposé par le World Wide Web Consortium (W3C) qui est une spécification dédiée à la protection des informations privées.

Le concept de profil est directement lié à l'utilisateur. L'utilisation de ce concept a été introduite par les travaux en filtrage d'information, pour décrire une structure représentative de l'utilisateur, plus particulièrement de ses centres d'intérêts. Cette notion est réutilisée en RI contextuelle pour cibler les éléments du contexte dépendant directement de l'utilisateur : centres d'intérêts, familiarité avec le sujet de la recherche, domaine professionnel, expertise, [12]

V.3.2. Contexte utilisateur

Le but fondamental de la plupart des travaux actuels en RI contextuelle, portant particulièrement sur la modélisation du profil de l'utilisateur, est de représenter, construire puis faire évoluer ses besoins en information à court et moyen terme.

V.3.3. Contexte physique

Travaux portant sur l'implication de ces nouveaux environnements sur la RI : [Rhodes et Maes 00], [Coppola 03], [Brown et Jones 02] Le contexte est focalisé sur le monde physique (contrairement à la RI interactive) Exemple : contextes temps et localisation utilisation d'ontologies géographiques [Makkonen 04] [12]

VI. Architecture d'un SRIC

Les systèmes de recherche sur le web traitent les requêtes isolées, les résultats pour une requête donnée sont indépendants de l'utilisateur, ou du contexte dans lequel l'utilisateur pose sa requête. On dit qu'un système de recherche d'informations est contextuel s'il utilise des données récupérées du contexte dans le but de délivrer l'information pertinente et appropriée. Ainsi la pertinence de l'information dépend de l'adéquation entre la requête et l'ensemble des éléments constituant le contexte qui sont perceptibles lors de la recherche.

Pour pouvoir adapter les résultats de recherche au contexte de l'utilisateur un processus de contextualisation est généralement mis en œuvre, il consiste à construire une représentation de ces éléments contextuels [Kob, 89]. La figure 2.1 représente l'architecture d'un SRI contextuel telle qu'elle a été proposée par [Lec, 09].

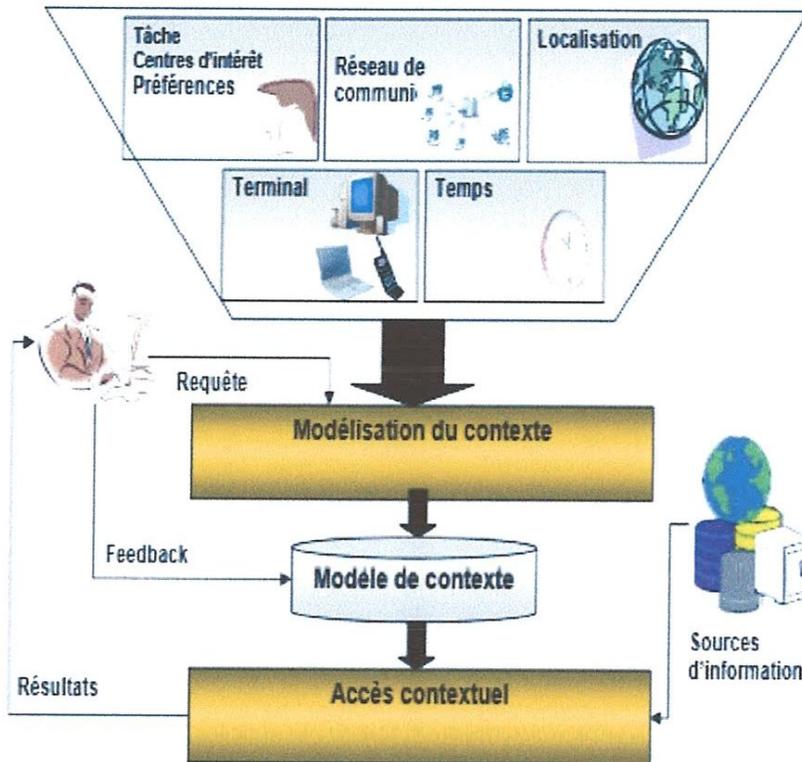


Figure 2.5 : Architecture de base d'un SRI contextuel.

Cette architecture distingue particulièrement deux fonctionnalités fondamentales :

- **La modélisation du contexte** : La RI contextuelle s'appuie sur une source d'évidence additionnelle exprimée à travers le contexte qu'il convient alors de modéliser. La nature et la portée du modèle dépendent des dimensions du contexte considérées. Le contexte utilisateur étant la dimension la plus abordée, la modélisation du contexte est alors qualifiée souvent de modélisation de l'utilisateur (user modeling en anglais). De manière générale, un modèle de contexte est défini par l'instanciation des éléments suivants :

- Les sources d'information : environnement (temps, température etc.), collection de documents, historique des interactions etc.
- Des stratégies de collecte de ces informations : on distingue principalement entre les stratégies implicites et stratégies explicites pour la collecte des données du contexte,
- Des ressources de modélisation : des ressources, généralement sémantiques (ontologies, dictionnaires, ...), sont parfois exploitées pour enrichir les données du modèle,
- Des modèles de représentation et/ou d'évolution : permettent de formaliser la représentation du contexte en qualité de structure unifiée (partie d'une ontologie,

classe de vecteurs de termes, ensemble de concepts ...) ou d'un ensemble d'informations avec des structures différentes et spécifiques, puis de les faire évoluer en cours du temps.

- **L'accès contextuel à l'information** : c'est le processus classique de RI projeté selon une dimension additionnelle liée au contexte de recherche. Principalement, son objectif est de sélectionner l'information pertinente à la requête adressée au SRI, tenant compte de la requête d'une part et du contexte de recherche en cours d'autre part. [16]

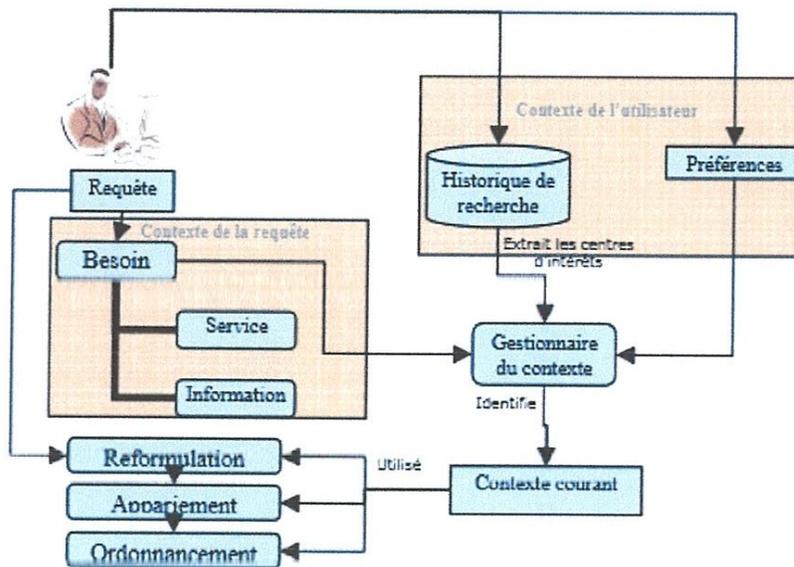


Figure 2.6 : Processus de RI contextuelle. [17]

-Exemples de SRI contextuels

- Okapi,
- InQuery,
- Smart Verity,
- Fulcrum,
- Excalibur,
- www.invisibleweb.com,
- www.completeplanet.com,
- www.amazon.com.

VII. RI mobile et accès à l'information ubiquitaire

Dans les applications de RI mobile, les données du contexte physique sont disponibles via des capteurs personnels et environnementaux

Exemples :

- température ambiante, conditions de trafic, commerces et bureaux proches, ...
- Utilisation du contexte pour choisir le meilleur mode de restitution de l'information : audio si en voiture,
- Notion de prédiction de contexte physique,
- Génération automatique de requêtes en utilisant les données du contexte.

Geo-parsing

Geo-parsing : processus identifiant le contexte géographique

- Pages (inférence à partir de n° de tél, code postal, etc.),
- Requetes (utilise un dictionnaire des noms de lieu),
- Utilisateurs,
- Explicites (donnez-moi votre localisation),
- A partir d'adresses IP,
- Téléphones mobile. [12]

VIII. Modèles de RI contextuel

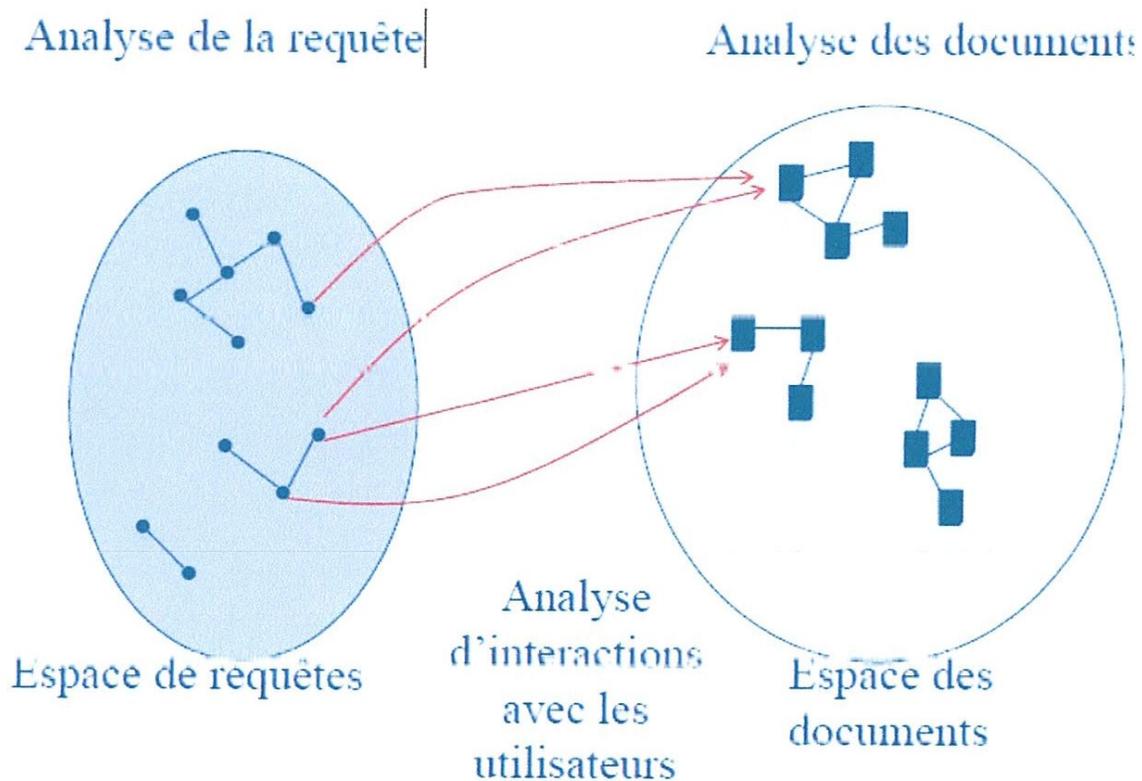


Figure 2.7 : scénario de RIC .

Question : comment modéliser le contexte comme des attributs dans un SRI et comment les intégrer dans les algorithmes de RI ?

Une solution : Utilisation de la reformulation de requêtes par injection de pertinence (relevance feedback)

Autre solution : introduction de contraintes booléennes sur des mécanismes de recherche existants -> vision limitée de l'utilisation du contexte

- Manque de représentation du contexte dans les modèles de RI. Les modèles classiques ont été définis en supposant un utilisateur unique, un besoin d'information pour chaque requête, une localisation, un temps, un historique, un profil
- Des techniques ad-hoc pour capturer le temps, l'espace, l'historique et les profils sont injectées dans les modèles
- Des capteurs pour récupérer la localisation, des fichiers logs pour implémenter l'historique, des métadonnées pour décrire les profils, des horloges et des calendriers pour récupérer le temps
- **Les modèles :**
- Utiliser le Relevance Feedback
- Utiliser les modèles de RI interactive
- Adapter un modèle de RI classique
 - Modèle vectoriel [Melucci 05]
 - Modèle probabiliste [Wen 04]
- **Autres modèles :**
- Modèle fondé sur un langage statistique [Rode et Hiemstra 04]
- Modèle de Finkelstein (2002)
- Modèle de recherche contextuelle sur le Web [Lawrence 00]
- Modèle de recherche personnalisée basé sur les ontologies
- [Pretschner et Gauch 04]
- Utilisation des « web log data » [Cui , Wen, Nie et Ma 02]
- Modèle de SearchPad : capture explicite des contextes de recherches sur le web [Bharat 02]

1. Utiliser le Relevance Feedback

Relevance Feedback (Pertinence rétroaction) est une caractéristique de certains systèmes de recherche d'informations. L'idée derrière le retour de pertinence est de prendre les résultats qui sont initialement renvoyés par une requête donnée et d'utiliser des informations sur si oui ou non ces résultats sont pertinents pour effectuer une nouvelle requête.

2. les modèles de RI interactive

- Modèle cognitif de Ingwersen (1996)
- Modèle épisodique de Belkin et al. (1993/95)
- Modèle stratifié de Saracevic (1996)
- Autres modèles :

- Les modèles de T.D. Wilson (1981... 1996... 1999) et Vakkari (2000)
- Dervin & Nilan : « sense making » (1986)
- Le modèle d'étapes phénoménologique et émotionnelle de Kuhlthau (1991)

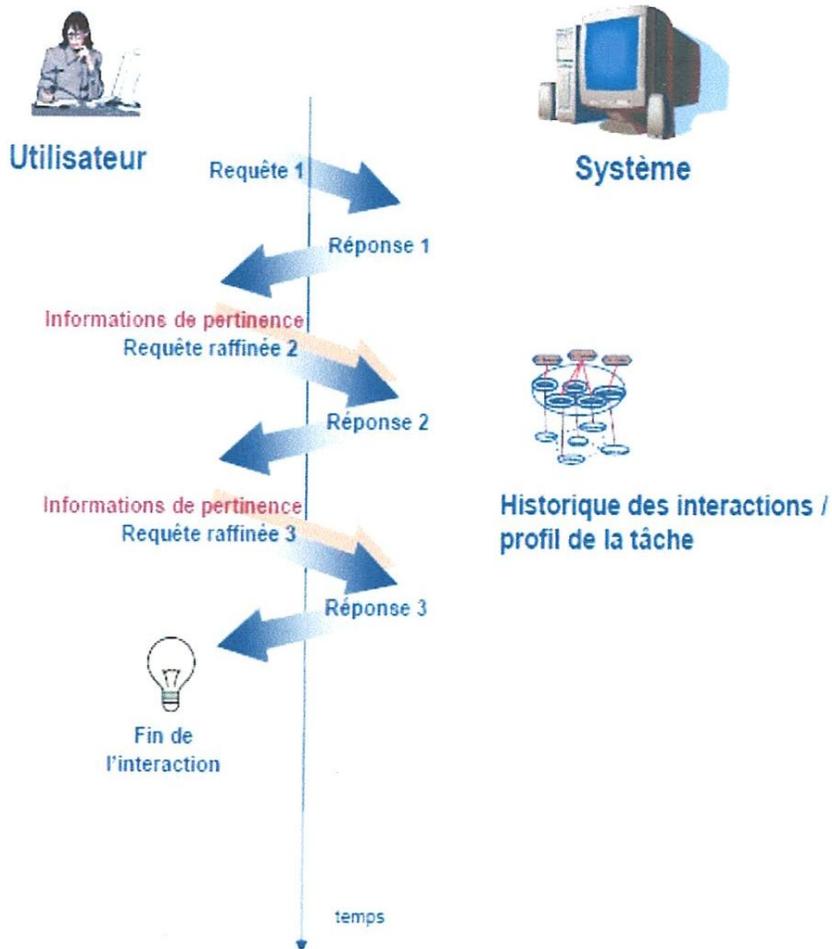


Figure 2.8 : Scénario de RI interactive.

- propose une vision globale du processus de recherche qui identifie :
- la composante situationnelle (informations liées aux contextes d'usages : tâches, définition du problème) ;
- la composante intentionnelle (analyse des croyances et motivations des usagers) ;
- la composante cognitive (représentation des connaissances de l'utilisateur) ;
- la composante requête (caractéristiques des questions de l'utilisateur) ;
- la composante interface homme-machine ;
- la composant système du logiciel de recherche (puissance, mémoire, etc...) ;
- la composante logicielle et algorithmique (composants du SRI) ;
- la composante fonds documentaire (structure et caractéristiques de la base documentaire).

- **Modèle stratifié de Saracevic (1996)**

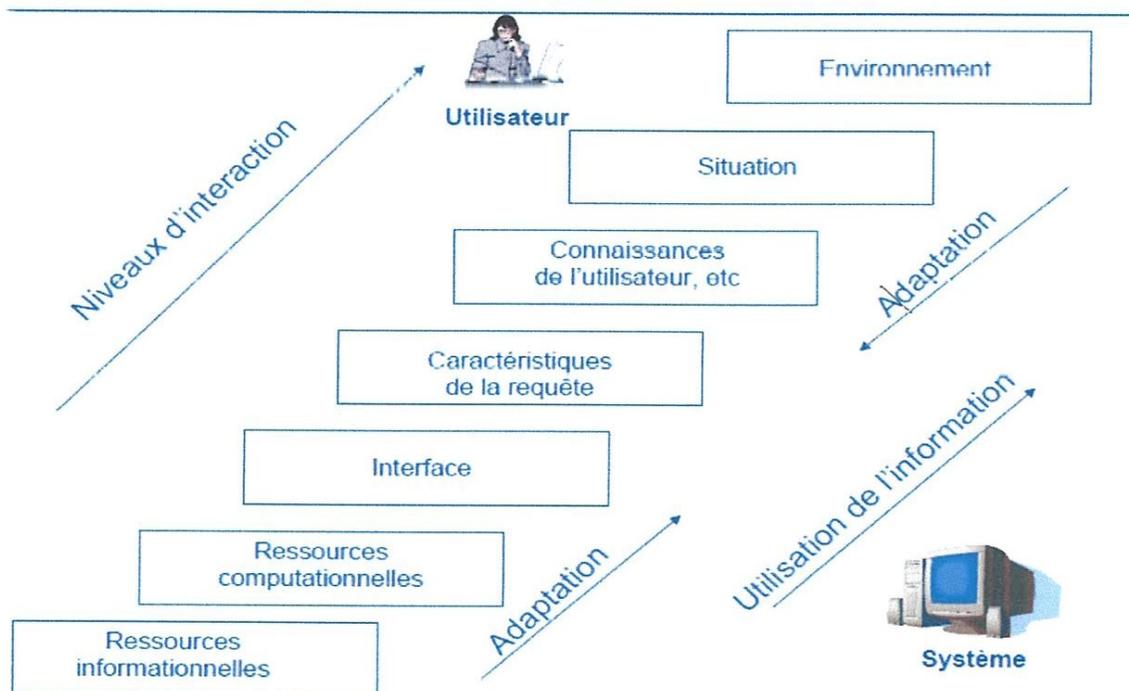


Figure 2.10 : Modèle stratifié.

- **Le modèle de Wilson (1)**

Modèle général du comportement informationnel qui s'appuie sur trois éléments:

- Le besoin informationnel et ses origines, c'est-à-dire les facteurs qui ont conduit à la perception du besoin par l'individu ;
- Les facteurs qui déterminent la réponse de l'individu en réaction à la perception du besoin ;

- Les processus ou les actions qui sont impliqués par cette réponse.
Macro-modèle qui tente d'expliquer les stratégies de recherche individuelles, pourquoi certaines sources d'informations sont utilisées et d'autres non, etc.

- **Le modèle de Vakkari (2000)**

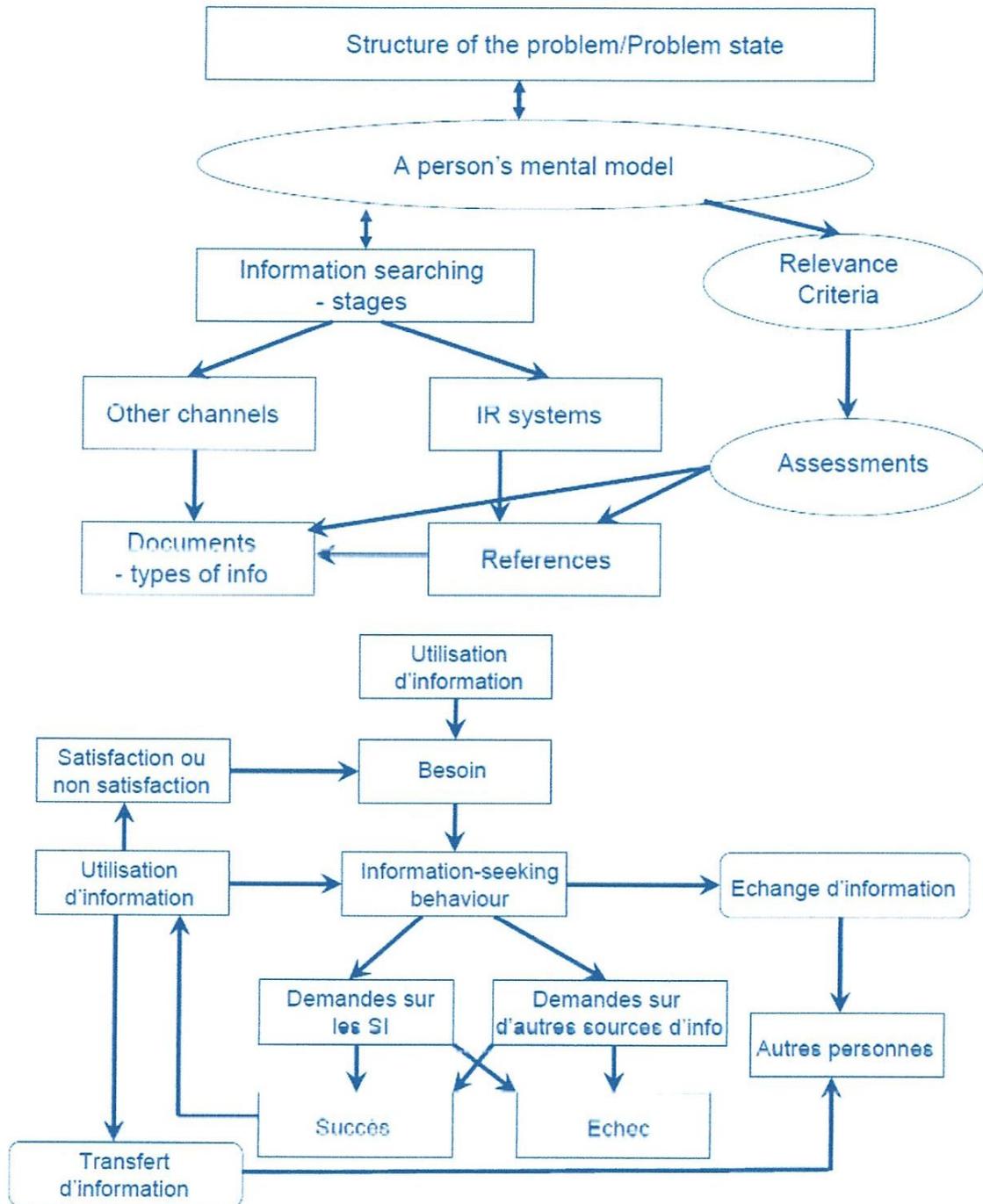


Figure 2.11 : Modèle stratifié.

3. Adapter un modèle de RI classique

- **Modèle vectoriel et contexte [Melucci 05]**

Le modèle vectoriel

-Donne une vue intuitive

-Prouvé efficace pour différents médias et langages Soit $\{t_1, \dots, t_m\}$ un ensemble de descripteurs uniques

-L'ensemble des descripteurs est modélisé comme une base $T = \{t_1, \dots, t_m\}$

-Les coefficients a_1, \dots, a_m combinent la base de vecteurs pour générer des vecteurs documents ou des vecteur requête

Une base d'un espace vecteur est la construction pour modéliser le contexte

-La signification des descripteurs est donnée par l'actuelle base de vecteur composants et les corrélations

-Le contexte est modélisé par la base de vecteur composants et corrélations

-Les transformations linéaires entre les bases sont des matrices qui modélisent les changements de contexte

- Modélisation du contexte - exemple

$$T = \begin{vmatrix} 2 & 1 \\ 1 & 2 \end{vmatrix}$$

$$a = \begin{vmatrix} \frac{1}{2} \\ \frac{1}{2} \end{vmatrix}$$

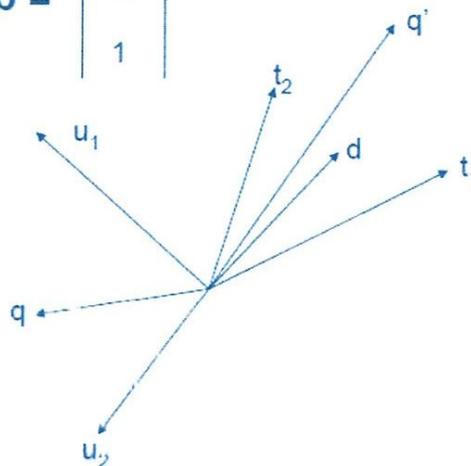
$$d = T \cdot a$$

$$q = U \cdot b$$

$$q' = T \cdot b$$

$$U = \begin{vmatrix} -1 & -1 \\ 2 & -1 \end{vmatrix}$$

$$b = \begin{vmatrix} \frac{1}{2} \\ 1 \end{vmatrix}$$



- **Modèle probabiliste et contexte [Wen 2004]**

-Utilisation de logs requêtes

-Une session requête en RI contextuelle enregistre une séquence requête-contexte-document

-Session requête := <requête,contexte>

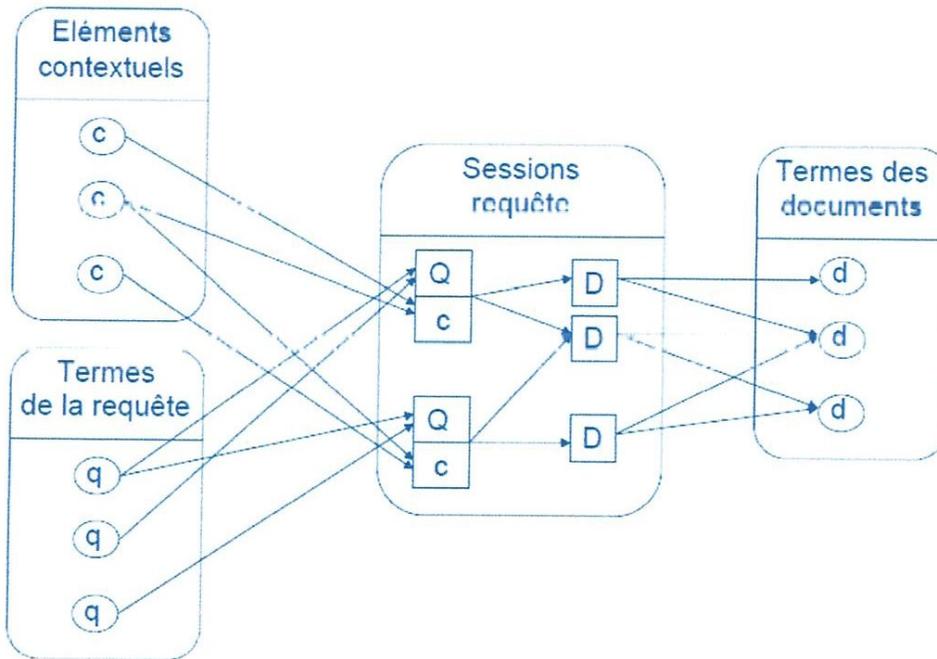
-Chaque session contient une requête, son contexte et un ensemble de documents que l'utilisateur sélectionne (par un click) ou balise (clicked documents)

-Idée générale : si un ensemble de documents est souvent sélectionné pour des requêtes similaires dans des contextes similaires, les termes dans ces documents sont strictement liés aux termes des requêtes et aux éléments du contexte

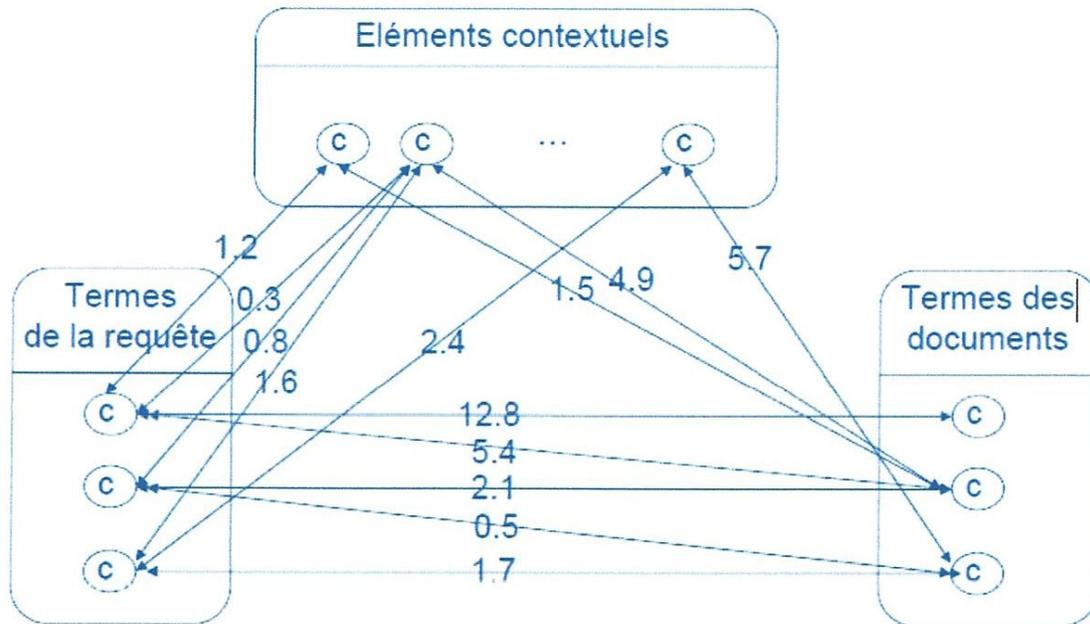
-De même, si des requêtes similaires et des contextes similaires sont fréquemment co-occurentes dans les logs, les termes de la requête sont bien corrélés aux éléments du contexte.

-Ainsi des corrélations probabilistes entre les termes de la requête, les éléments du contexte et les termes des documents peuvent être établies sur la base des logs de requêtes.

▪ Sessions requêtes :



▪ **Information mutuelle entre requêtes, contextes et documents**



Utilisation de l'information mutuelle pour déterminer les degrés de corrélation entre les termes de la requête, les éléments contextuels et les termes des documents. [12]

IX. RI à base des Systèmes Multi-Agents

D'après Koch *et al.* [88], le *paradigme agent* offre des méthodologies et des mécanismes pour la création d'applications distribuées, intelligentes, intégrées et coopératives. Un agent peut fournir des informations adaptées au système de RI auquel il accède. Dans le Web, le contexte de la recherche de l'utilisateur est dynamique puisque les utilisateurs peuvent se déplacer d'un site à un autre, d'un endroit à un autre. Les sources d'informations sont multiples et hétérogènes, le volume d'information est en évolution considérable. Ces changements de l'environnement du Web provoquent des changements dans les tâches et les besoins d'information de l'utilisateur. En conséquence, le paradigme agent peut fournir un apport important à la RI dans le Web.

1. **Les agents de recherche** La catégorie des agents de recherche comprend les métamoteurs de recherche d'information de dernière génération jusqu'aux outils de recherche off-line (Copernic par exemple). Ces derniers permettent à l'utilisateur

d'initier des recherches d'informations qui s'exécuteront même lorsque l'utilisateur ne sera plus connecté à Internet.

2. **Les agents de recommandation** Les agents de recommandation, visent à optimiser la recherche d'information de l'utilisateur en lui proposant automatiquement de nouveaux documents au regard de ses besoins ou de ses actions. Ils reposent essentiellement sur une approche Push proposant des informations à l'utilisateur et une caractérisation des besoins au moyen d'un profil utilisateur. [6]

X. Recherche Sémantique d'information

Le web est constitué par un ensemble de documents formatés dans le langage HTML, qui fournit particulièrement des liens hypertextes. Ils sont exploités par des navigateurs ou robots de recherche, tout ce qui est conçu pour être lu et compris par les humains, mais les dispositifs logiciels n'avaient aucune idée sur le sens.

L'idée du web sémantique, qui est une extension du web actuel, est de concevoir un web compréhensible par la machine pour une meilleure coopération homme machine.

La notion de web sémantique fait référence à la vision du web de demain dans lequel les utilisateurs devraient être déchargés d'une bonne partie de leurs tâches de recherche et d'exploitation des résultats, grâce aux capacités accrues des machines à accéder aux contenus des ressources et à effectuer des raisonnements sur ceux-ci. [10]

X.1. Définition

Le web sémantique est une extension du web courant, dans laquelle on donne à une information un sens bien défini pour permettre aux ordinateurs et aux personnes de travailler en coopération « *Tim Berners-Lee* »

X.2. Intérêt du web sémantique

Tim Berners-Lee, a proclamé que le Web sémantique est la prochaine évolution du Web.

C'est-à-dire que l'on va arriver à un Web intelligent où les informations sont stockées de façon compréhensible par les ordinateurs afin d'apporter à l'utilisateur ce qui cherche vraiment (Phan, 2005).

Parmi les points forts du web sémantique que :

- Il munit les langages du Web d'une sémantique formelle à l'aide d'une interprétation en terme d'un modèle. Elle permet une caractérisation précise des opérations applicables, par exemple de pouvoir affirmer la correction des algorithmes comme des algorithmes de recherche (Laublet et al., 2002).

- Les services seront mieux rendus sans engendrer de surcharge pour les utilisateurs (Baget et al., 2003).

- La recherche sur l'internet sera affine. Pour le faire, il va ajouter aux informations existantes une couche de métadonnées pour que les ordinateurs puissent l'exploiter (Phan, 2005). Cet ajout est dû à l'insuffisance majeure du HTML : il ne sépare pas le contenu de la présentation, ce qui pose un problème d'interopérabilité (Mestiri, 2007).

- Il fait coopérer étroitement des acteurs d'origines très différentes: depuis les protocoles de communication jusqu'aux relations entre ordinateur et sens. On peut noter une double partition dans les développements du web sémantique:

D'une part entre théoriciens et praticiens, d'autre part entre industriels et militants (Euzenat, 2003).

- Il définit des ontologies qui permettent aux humains et aux machines de partager les connaissances du domaine et de collaborer ensemble. On les utilisant en général pour permettre aux machines de raisonner et d'interpréter les informations ainsi que d'améliorer la pertinence des recherches. Les agents auxquels les utilisateurs délègueront des tâches, devront communiquer entre eux et interpréter le contenu échangé de la même manière, c'est-à-dire en interprétant les termes décrivant le contenu de la même manière. D'où l'intérêt de cette ontologie. « Avec la notion de Web sémantique, vous définissez un espace virtuel où les hyperliens pointeront non plus sur des documents (textes ou images), mais sur des concepts » explique Pierre Lévy (Mestiri, 2007). [13]

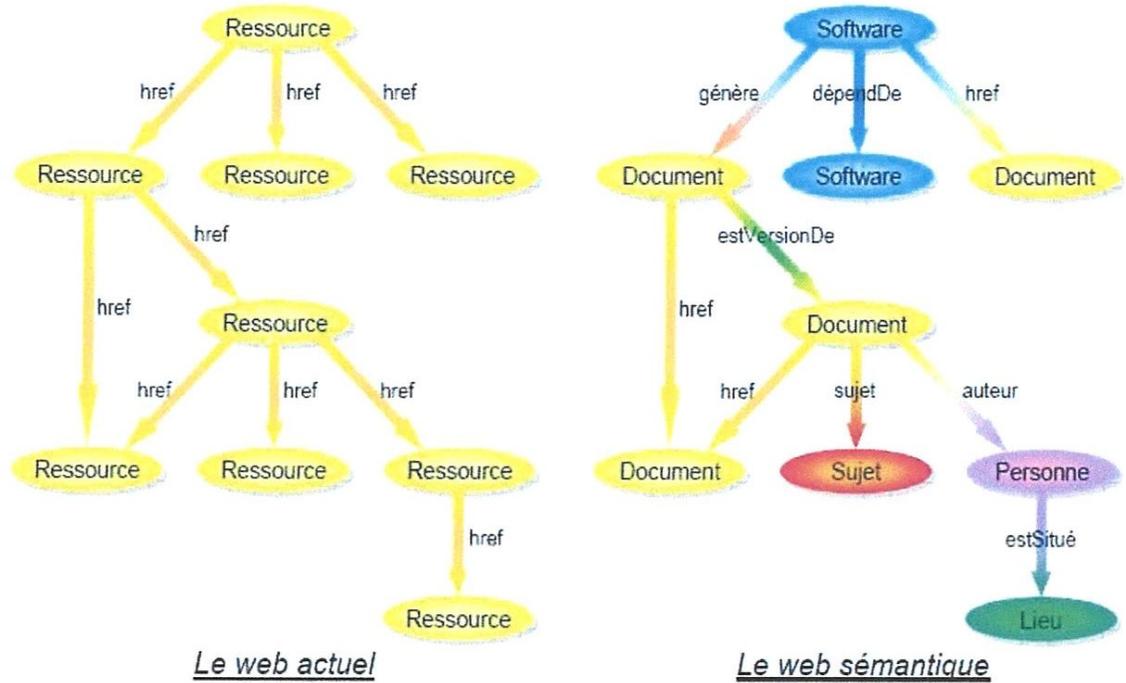


Figure 2.12: Comparaison entre le web actuel et le web sémantique.

X.3. Les Ontologies

Le terme « ontologie » a été emprunté au domaine de la philosophie dans lequel il signifie « l'essence de l'essentiel ». Dans le domaine de la gestion de connaissance, le sens de ce mot est différent. La notion d'ontologie a d'abord été introduite comme « une spécification explicite d'une conceptualisation ». Cette définition a été légèrement modifiée par la suite. Une combinaison des deux définitions peut être résumée ainsi : « une spécification explicite et formelle d'une conceptualisation partagée » (Cardoner, 2004).

Cette définition s'explique ainsi :

- explicite signifie que le « type des concepts et les contraintes sur leurs utilisations sont explicitement définies ».
- formelle se réfère au fait que la spécification doit être lisible par une machine.
- partagée se rapporte à la notion selon laquelle une ontologie « capture la connaissance consensuelle, qui n'est pas propre à un individu mais validée par un groupe ».

- conceptualisation se réfère à « un modèle abstrait d'un certain phénomène du monde basé sur l'identification des concepts pertinents de ce phénomène ». [10]

X.3.1. Rôles des ontologies

- Résoudre des problèmes de compréhension et faciliter le partage des connaissances entre personnes de spécialités différentes ;
- Assurer l'interopérabilité entre applications à base de connaissances
- Accéder à des ressources hétérogènes.
- Permettre la réutilisation de modèles de connaissances;
- Faciliter la communication entre agents logiciels.
- Annoter des ressources à l'aide de méta-données.
- Améliorer les processus de recherche d'informations.

X.3.2. Recherche d'information guidée par les ontologies

La réflexion sur le web sémantique a été essentiellement fondée sur ce problème de la recherche d'informations. En effet, les ontologies peuvent améliorer la pertinence d'une recherche et ce, en recherchant des documents faisant référence à un concept précis au lieu de se baser sur des mot-clés qui peuvent être ambigus.

Prenons l'exemple d'une personne anglo-saxonne qui cherche à trouver l'adresse d'un installateur de fenêtres ; en tapant la requête « Windows installation » dans n'importe quel moteur de recherche, elle obtiendra des milliers de pages traitant l'installation du système d'exploitation de Microsoft et les problèmes qui en résultent, mais elle aura beaucoup de mal à trouver l'information qu'elle recherchait [Khelif, 06b].

Avec l'utilisation d'une ontologie, un moteur de recherche fera la différence entre un site sur lequel 'Windows' désigne un logiciel et un autre sur lequel il désigne une fenêtre.

Cette recherche basée sur les ontologies se présente comme une recherche intelligente qui repose sur la sémantique des ressources et sur les concepts contenus dans les documents qui leur sont associés. Ces ontologies peuvent ainsi, d'une part, guider la création d'annotations sous la forme de métadonnées sur les ressources, et d'autre part, décrire leurs contenus de manière à la fois formelle et signifiante pour être exploitable aussi bien par les humains que par les machines.

Dans cette optique, plusieurs systèmes de recherche d'informations à base d'ontologies ont été proposés, parmi lesquels on peut citer : Ontobroker, Sesame et Corese. La différence entre ces systèmes réside essentiellement dans le langage de représentation et le moteur d'inférence sur les connaissances imbriquées dans les annotations : Ontobroker utilise F-Logic [Kiffer, 95], Sesame utilise SQL92SAIL (du SQL adapté à RDF), et Corese utilise les graphes conceptuels [Sowa, 84]. [13]

X.3.3. Exemple de moteurs de recherche sémantique

- Exalead
- Pertimm
- Swisscows
- Verticrawl
- WolframAlpha
- Yatedo

XI. Conclusion :

Dans ce chapitre nous avons présenté les principales notions et concepts de la recherche d'information classique et contextuelle, des systèmes de recherche d'information et ceux des outils de recherche sur le web.

La RI contextuelle constitue actuellement un domaine de recherche très actif qui vise l'amélioration de la RI sur le Web ubiquitaire, Il s'agit d'un domaine de recherche récent dont l'émergence provient d'une part de l'évolution des supports physiques (PDA, téléphonie mobile) et d'autre part de l'accroissement du volume d'information sur le Web.

Chapitre 3

I. Introduction

Les systèmes multi-agents (SMA) sont devenus un paradigme dominant dans le domaine de développement des systèmes distribués complexe. Leurs avantages consistent notamment en leur capacité d'aborder les problèmes complexe d'une manière distribuée et de proposer, en outre, des solutions réactives robustes. Les concepts qui y sont liés font les SMA une approche prometteuse pour la conception des systèmes complexe. La mobilité est une caractéristique essentielle des stations mobiles qui peuvent utiliser une connexion sans-fil pour communiquer pendant un déplacement et un point d'accès fixe au réseau traditionnel pour se reconnecter après un déplacement.

D'autre part le paradigme agent mobile est devenu aussi un formalisme très puissant pour le développement des applications réparties. Les agents mobiles sont des programmes autonomes qui peuvent se déplacer à travers un réseau hétérogène sous leur propre contrôle. Il se déplace avec son code son état d'exécution et ses données propres. Le but du déplacement est d'accéder localement à des données ou à des ressources initialement distantes, d'effectuer le traitement en local et de ne déplacer que les données utiles.

des réactions réflexes aux changements de leurs environnements. Généralement un système réactif comprend un grand nombre d'agent de faible granularité.

VI.1.3. les agents pro-actifs

Ce sont des agents dynamiques qui entreprennent car ils possèdent, en plus de leurs attributs et méthodes, des processus internes qui leur permettent de prendre des initiatives pour réaliser leurs buts. Un agent pro-actif est donc un agent dirigé buts.

V. La communication entre les agents

La communication est l'un des concepts pertinents dans les systèmes multi-agents. On distingue essentiellement deux modèles de communication :

- Communication par partage d'information.
- Communication par envoi de messages.

V.1. Communication par partage d'information

La communication entre les différents agents du système est réalisée par partage d'information lorsque ceux-ci disposent d'une zone de données commune dans laquelle ils rangent les conclusions qu'ils ont pu tirer. Outre ces résultats partiels, cette zone renferme les données du problème initial. Les agents peuvent ainsi y puiser les informations dont ils ont besoin pour résoudre une partie du problème globale. Ce type de communication correspond à ce que la littérature désigne communément sous le nom de modèle du blackboard (tableau noir).

V.2. Communication par envoi de messages

Les systèmes multi-agents fondés sur la communication par envoi de messages se caractérisent par le fait que chaque agent possède une représentation propre et locale de l'environnement qui l'entoure. Chaque agent va alors interroger les autres agents sur cet environnement ou leur envoyer des informations sur sa propre perception des choses. La communication se fait soit en mode point à point, soit en mode par diffusion.

VI. Les agents mobiles

VI.1. Définition

Un agent statique est un agent qui s'exécute seulement dans les systèmes où il commence son exécution. Il utilise un mécanisme de communication tel que RPC. Par contre, un agent mobile n'est pas lié au système dans lequel il débute son exécution.

L'agent mobile est capable de se déplacer d'un hôte à un autre hôte dans le réseau avec son code, ses données et son état d'exécution.. Il peut transporter son état et son code d'un environnement vers un autre dans le réseau où il poursuit son exécution [Far 03].

Selon [Che 99], un agent mobile est un programme informatique capable de s'exécuter sur un ordinateur distant, il peut se déplacer d'un site à un autre en cours ses données propres, mais aussi avec son état d'exécution.

VI.2. Motivation

L'idée de la mobilité du code et des agents mobiles trouve plusieurs motivations dans les applications modernes de l'informatique. Nous décrivons ci-dessous certains de ces motivations :

- **Plus d'efficace:** au lieu d'avoir des interactions à distance avec un serveur (en générale coûteuse) c'est-à-dire appel de procédure à distance (RPC : Remote procedure call), le mieux est de déplacer le code vers le serveur et avoir des interactions locales moins coûteuses.
- **Robustesse et tolérance aux fautes:** lorsqu'un système/machine hôte est en difficulté (rupture de communication), les agents mobiles visiteurs prévenus ont la possibilité de se dispatcher ailleurs dans le réseau. En plus, l'exécution vasynchrone et autonome des agents mobiles rend l'application tolérante en vers les ruptures possible entre hôte source et hôte destination.

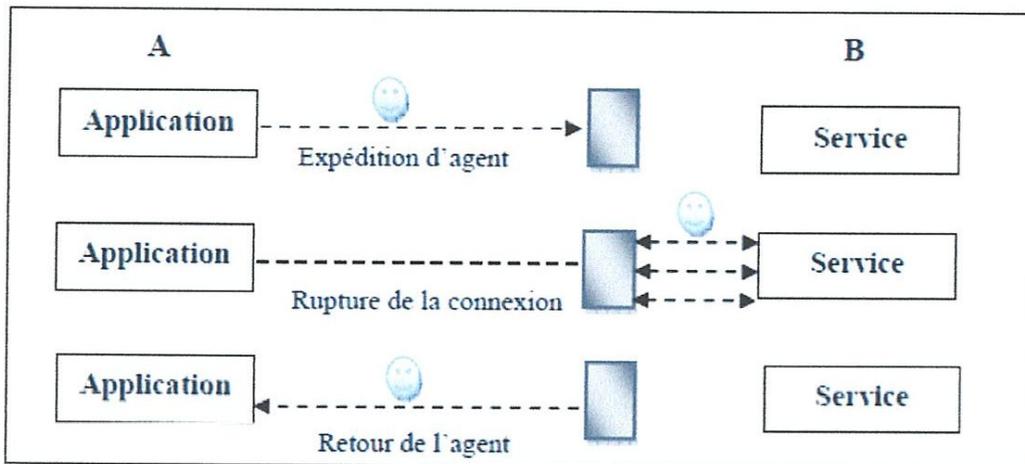


Figure 3.1: Exécution asynchrone et autonome.

- **Réduction de la charge dans le réseau :** les agents mobiles réduisent le trafic du réseau, en déplaçant le traitement vers les données brutes plutôt que de déplacer les données vers le traitement.
- **Adaptation dynamique à l'environnement :** les agents mobiles sont autonomes, ils peuvent « découvrir » le réseau, c'est-à-dire qu'ils pourront visiter des hôtes qui leurs étaient inconnus. à chaque nouvel hôte parcouru, ils pourront prendre des décisions basées sur leurs propres connaissances ainsi que sur les connaissances acquises des hôtes visités. Des agents mobiles ont la propriété de se répartir uniformément entre les machines d'un réseau pour résoudre un problème complexe de manière optimale.

VII. Structure d'un agent mobile

Un agent mobile est une entité qui possède cinq attributs : son état, son implémentation, son interface, son identifiant et son autorité. Quand un agent se déplace à travers le réseau, il transporte ses attributs [Far 03] :

- **L'état** : l'état d'un agent peut être considéré comme une photo instantanée de son exécution. Quand un agent voyage, il transporte avec lui son état, ceci lui permet de reprendre son exécution quand il arrive à destination.
- **L'implémentation** : comme n'importe quel autre programme, l'agent mobile a besoin d'un code pour pouvoir s'exécuter. Quand il se déplace à travers le réseau, l'agent peut soit emporter son code soit aller à destination, voir quel code est disponible sur la machine distante et récupérer le code manquant à partir du réseau (c'est la technique du « code-t-on demande »).
- **L'interface** : un agent fournit une interface qui permet aux autres agents et autres systèmes d'interagir avec lui. Cette interface peut être un ensemble de méthode qui permet aux autres agents et applications d'accéder aux méthodes de l'agent par un système de messagerie.
- **L'identifiant** : chaque agent possède un identifiant unique durant son cycle de vie, qui lui permet d'être identifié et localisé. Puisque l'identifiant est unique, il peut être utilisé comme clé dans les opérations qui exigent un moyen pour référencer une instance particulière d'agents.
- **L'autorité** : une autorité est une entité dont l'identité peut être authentifiée par n'importe quel système auquel elle essaye d'accéder. Une autorité peut être soit une personne privée, soit une organisation. L'identité est constituée d'un nom et d'autres attributs.

VIII. Tolérance aux pannes

Le modèle d'exécution de l'agent mobile implique son interaction avec plusieurs sites ce qui expose l'agent à une éventualité de disparition à cause de la défaillance ou de la déconnexion soudaine et imprévue d'un site sur lequel il s'exécute. La disparition d'un agent entraîne un dysfonctionnement de l'application basée sur ce dernier. Une application distribuée sûre doit pouvoir continuer de fonctionner en cas de défaillance d'une partie du système. Pour certains types d'applications, il est essentiel que les environnements d'exécution d'agents mobiles offrent des mécanismes de tolérance aux fautes.

Dans une architecture de services, les défaillances de sites peuvent conduire à un comportement défaillant d'un service et donc le rendre inutilisable par le client.

Plusieurs types de défaillances sont à considérer :

- Une défaillance par arrêt (crash, panne) quand un serveur ne rend pas de résultats suite à des invocations répétées.
- Une défaillance par omission quand un serveur omet de répondre à son client.
- Une défaillance temporelle quand la réponse du serveur est fonctionnellement correcte mais n'est pas arrivée dans un intervalle de temps donné.
- Une défaillance de valeur quand le serveur rend des résultats incorrects.

Pour une application distribuée basée sur le concept d'agents mobiles, la migration des agents engendre d'autres types de défaillance :

– Un agent peut disparaître après avoir visité plusieurs sites. Si aucune précaution n'est prise, les résultats de son exécution sur ces sites peuvent être perdus [Shao05].

– Il est important de détecter la disparition d'un agent pour en informer la place qui l'a lancé.

– Un problème d'atomicité pour l'exécution globale d'un agent qui se déroule successivement sur plusieurs sites. Ce qui implique qu'il faut garantir que l'agent reprend son exécution exactement là où elle était avant son déplacement et ceci avec le même contexte d'exécution. C'est au programmeur de garantir cette atomicité dans une migration faible.

Pour faire face à la défaillance, les environnements d'agents mobiles offrent un mécanisme de point de reprise [Grasshopper98]. Les points de reprise sont conservés sur disque, support suppose fiable. Ils peuvent ainsi être utilisés ultérieurement pour restaurer l'agent en cas de défaillance. Cependant, lors de la restauration de l'agent il faut veiller à ne pas avoir deux agents actifs en même temps (problème rencontré en cas de défaillance lié à la perte de la connexion avec le site d'accueil de l'agent par exemple).

IX. Pourquoi utiliser les agents mobiles ?

Le placement dynamique de tâches nécessite la migration de ces tâches entre les différentes machines du réseau. Cette migration devient opérationnelle grâce à l'utilisation de la technologie des agents mobiles. Dans notre approche, une tâche sera représentée par un agent. Ainsi, la terminaison d'une tâche correspond à la fin de la vie de l'agent. Les échanges d'informations entre les tâches correspondent à la communication entre les agents. Quant au déplacement d'une tâche, il correspond à la migration d'un agent à travers le réseau. Il s'agit ainsi de représenter une tâche par un agent mobile.

Lorsque l'on parle de la technologie d'agents mobiles, deux types de migration sont à considérer : (1) la migration forte qui permet à un agent de se déplacer quelque soit l'état d'exécution et de communiquer avec l'extérieur dans lequel il se trouve et de reprendre son exécution après la migration exactement là où elle en était avant ; (2) la migration faible ne fait que transférer avec l'agent son code et ses données. Elle nécessite des moments privilégiés dans le code de l'agent pour pouvoir être lancée

(point d'arrêt) ; le programmeur doit donc explicitement préserver dans les données les informations d'état permettant la poursuite de l'exécution au point d'avancement atteint.

X. Agent de recherche :

Ces agents de recherche sont des entités qui se déplacent d'une machine à l'autre sur le réseau afin de satisfaire les besoins de leurs clients. Sur chaque machine, ils effectuent des échanges et traitent les informations collectées. Ce traitement permet de réduire la quantité d'informations transportées avec l'agent et par conséquent le trafic sur le réseau. L'agent transporte avec lui les données demandées par Maître Aglets.

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 récupérées sur le Web à ces classes et d'en extraire l'information pour alimenter ces bases de données.

- **Les savoirs** : ils connaissent :
 - Maître Aglet
 - Agent fournisseur
- **Les savoir-faire** :
 - Ils se déplacent vers les sites fournisseurs.
 - Ils soumission la demande à des agents fournisseurs.
 - Ils interagissent avec les agents fournisseurs.
 - Ils reçoivent les résultats à partir des agents fournisseurs.
 - Ils envoient les résultats à Maître Aglet.

Rôle : livraison les demandes à fournisseurs, et recueil les résultats.

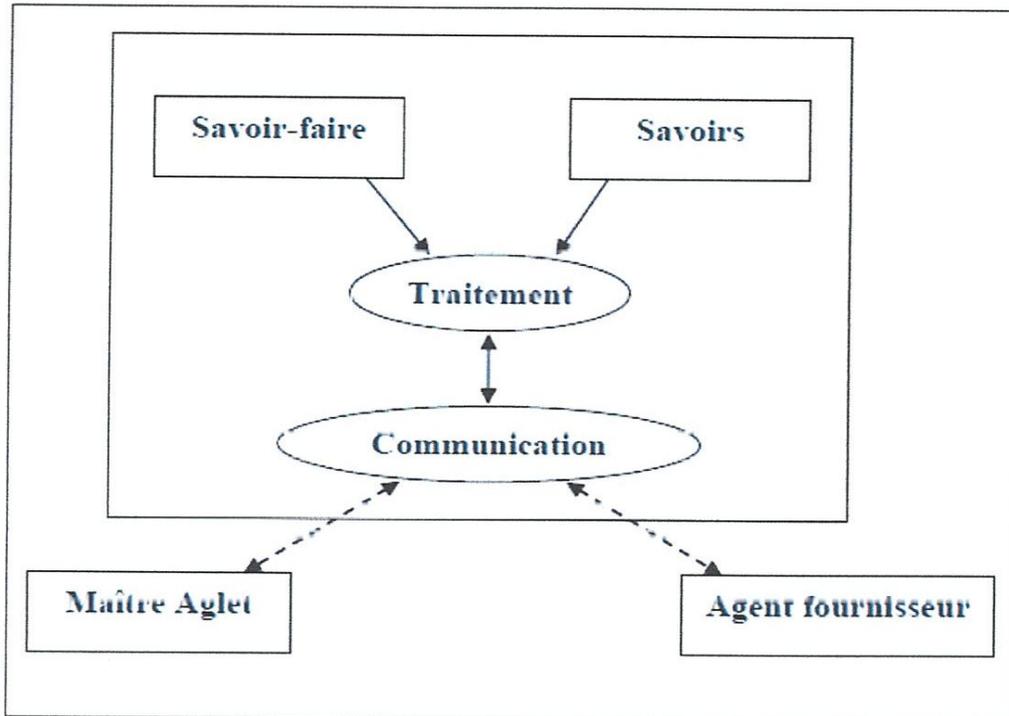


Figure 3.2: Architecture interne des agents mobiles de recherche.

X.1. Code mobile Lors d'un déplacement, le code d'un agent est téléchargé sur la machine cible et s'exécute au sein de la plateforme pour effectuer une tâche requise (recherche d'information locale, négociation, etc.), avant de se télécharger de nouveau vers une autre machine ou vers la machine de l'utilisateur afin de rapatrier les résultats. [6]

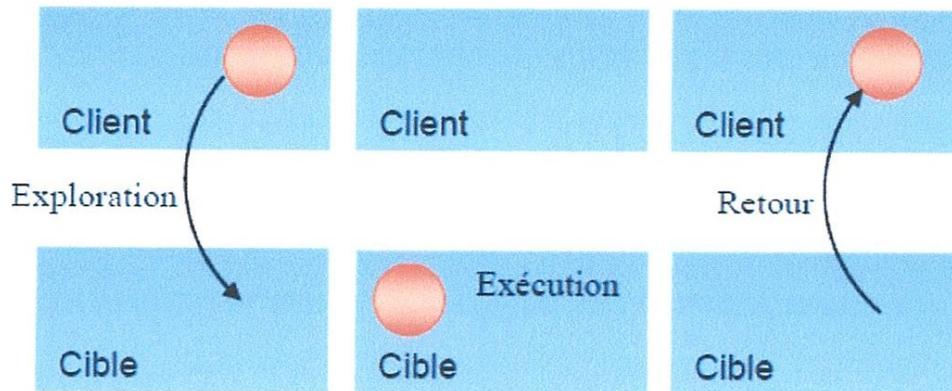


Figure 3.3: Code mobile

XI. Conclusion

Dans ce chapitre, nous avons présenté les différentes notions de système multi agent, ainsi nous avons situé la technologie d'agents mobiles par rapport aux différentes techniques présentées et l'efficacité des agents mobiles dans le processus de recherche d'information.

Un agent mobile est une entité autonome qui se déplace d'une machine à l'autre sur le réseau, sans perdre son code ni son état. C'est l'environnement d'exécution qui se charge d'assurer cette fonctionnalité. Il permet la création et la migration d'un agent, la communication et l'échange de messages entre les agents mobiles et il assure la sécurité de l'agent et de son site d'accueil.

Chapitre 4

I. Introduction

Dans ce chapitre, nous allons décrire l'application que nous avons développée. En fait, cette description est divisée en deux grandes parties, à savoir : la conception et l'implémentation, la première présente quelques aspects relatifs à la conception, et la modélisation de l'application, en revanche, la deuxième traite la phase d'implémentation, en exposant les outils que nous avons utilisés pour le développement de notre application, ainsi que les différentes interfaces graphiques de celle-ci.

En fait, l'objectif principal de notre conception et implémentation est justement le développement d'un site Internet dédié au visionnement de films sur des dispositifs mobiles, possédant des caractéristiques plus au moins forts. En fait, notre site Internet, permettra à son utilisateur d'effectuer une recherche ubiquitaire de films, et de visionner uniquement les films qui correspondent aux caractéristiques techniques du dispositif qu'il utilise pour le visionnement, en plus cette recherche prendra en considération des informations contextuelles sujettes à l'utilisateur, telles que le pays, l'âge, les préférences, en termes de genres, etc.

II. La Conception de l'application

Dans cette partie nous allons exposer l'architecture générale de notre système, en utilisant des diagrammes UML (le diagramme de cas d'utilisation, le diagramme de classes, et le diagramme de séquences).

Le langage de modélisation unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes, conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

II.1. Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. De ce fait, le diagramme représenté dans la (figure 4.1) expose les vues utilisateur de notre application, et il décrit les cas d'utilisation de notre système implémenté. Ainsi, les différents utilisateurs de notre système sont :

- L'administrateur : c'est le gestionnaire du site Internet, son rôle principal est d'effectuer les différentes opérations de mise à jour de la table des films, telles que l'ajout, la modification, et la suppression. Ces films qui seront par la suite accessibles aux utilisateurs de notre site.

- Le client : est bien entendu l'élément principal de notre site Internet, car c'est lui qui va mettre en œuvre les différentes techniques de recherche ubiquitaire que nous avons implémentées sur notre site

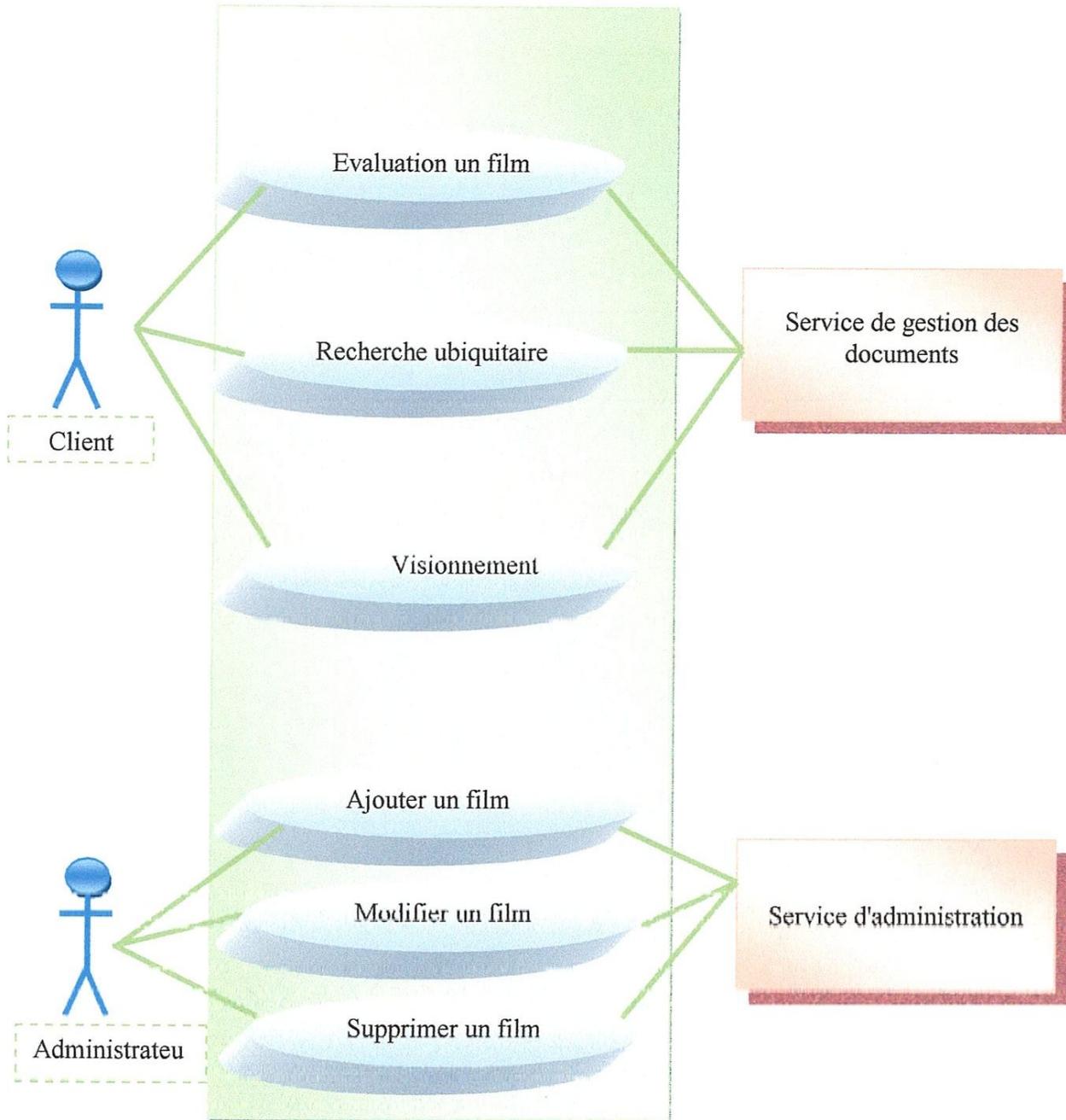


Figure 4.1 : Diagramme de cas d'utilisation pour les acteurs de système .

II.2. Les fonctionnalités du système

II.2.1. Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Le diagramme suivant (figure 4.2) décrit l'ensemble des classes de notre application. Nous avons utilisé des simples classes de client et administrateur pour faciliter les choses. La relation entre les différentes classes est une simple relation dépendance.

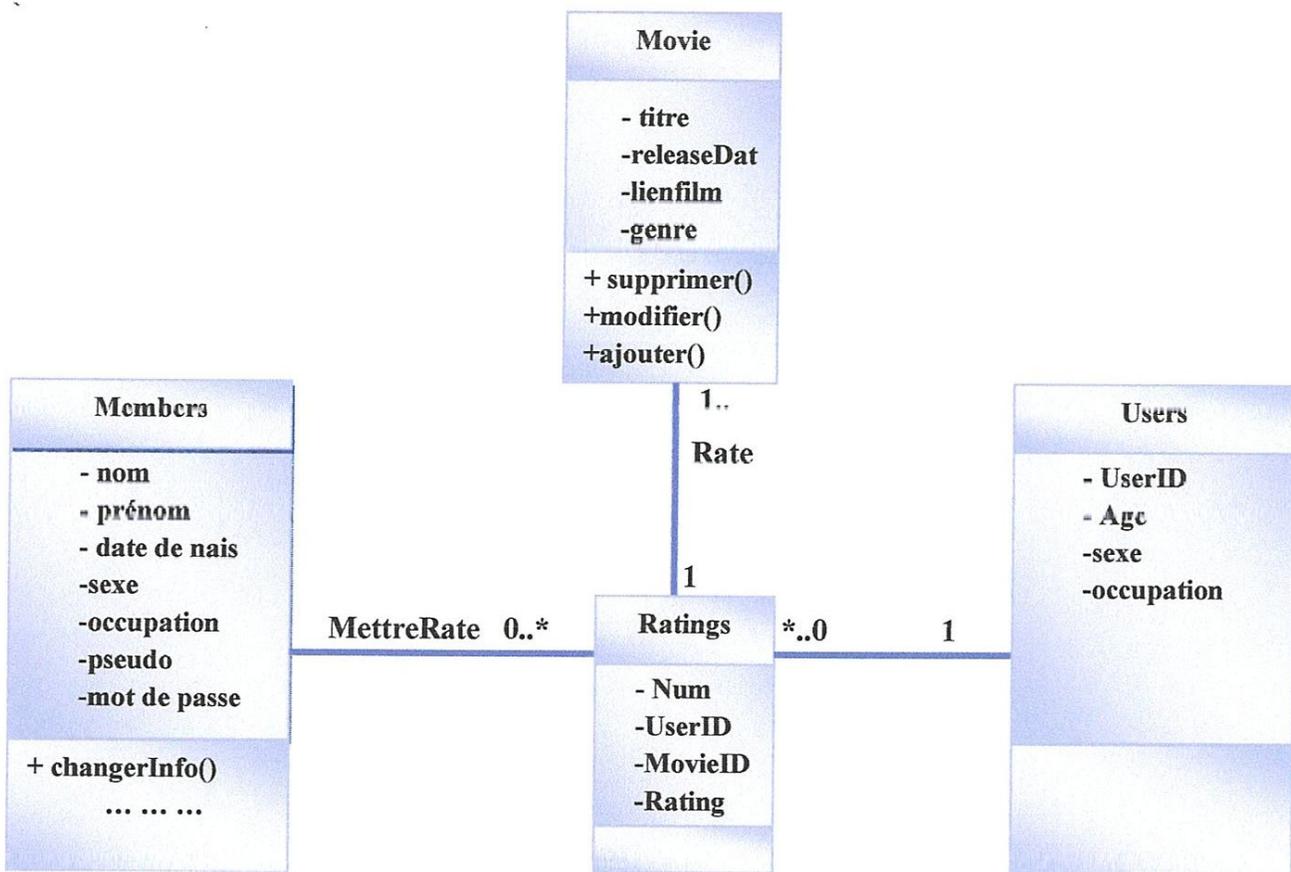


Figure 4.2 : Le diagramme de classe pour la recherche des Films .

II.2.2. Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique.

Les diagrammes de séquences montrent les interactions entre les objets, la représentation se concentre sur la séquence des interactions selon un point de vue temporel. Les diagrammes suivants représentent les diagrammes de séquence pour les cas d'utilisation précédemment décrits

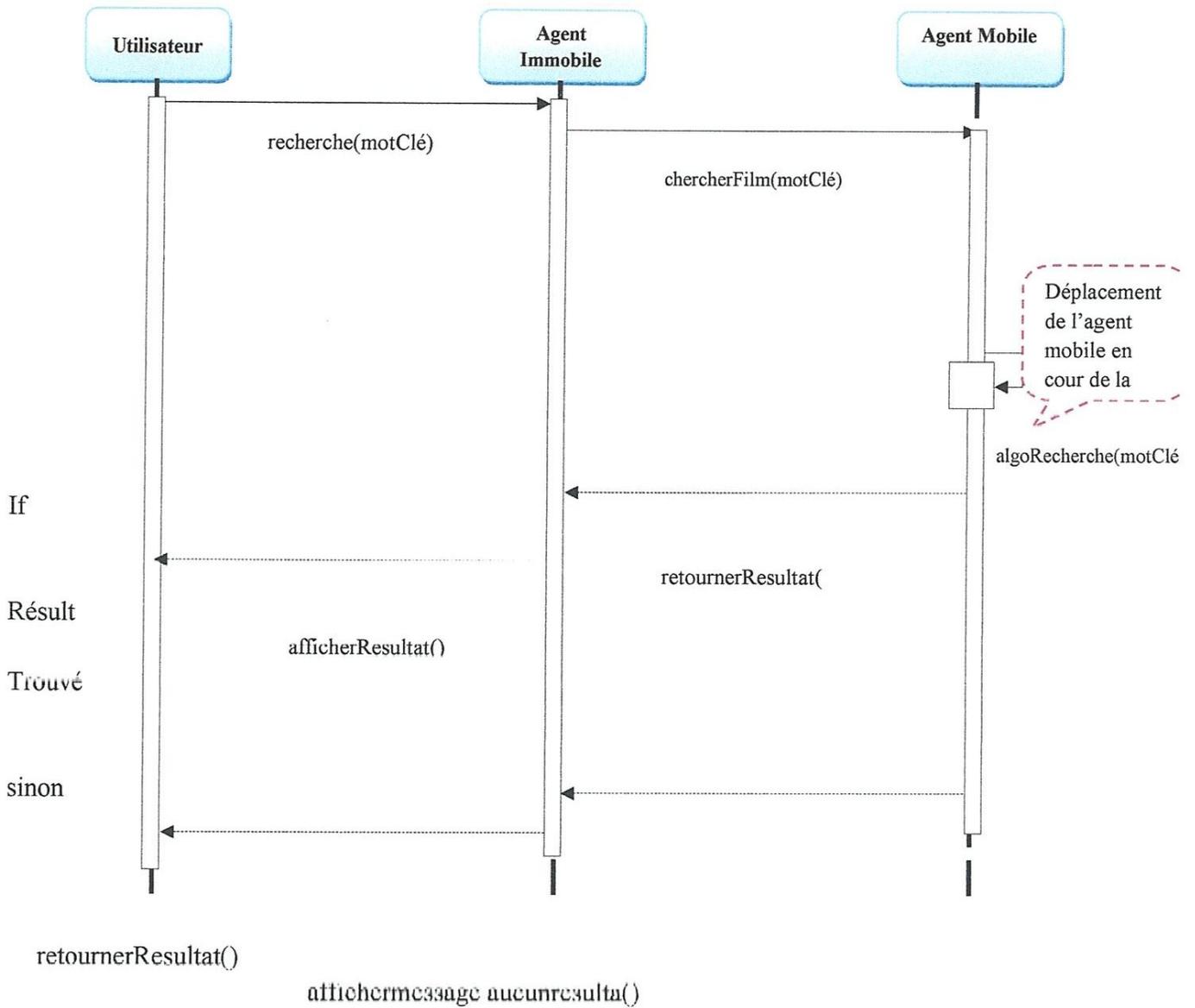


Figure 4.3 : Le diagramme de séquence pour la recherche des Films .

- 2) Support des fonctions lambda. En termes simples, cela est une fonction anonyme (on n'a pas besoin de lui donner un nom) qui prend un certain nombre de paramètres et renvoie une valeur. Une valeur retournée sera transmis le pipeline à l'opération suivante.
- 3) Support pour passer des fonctions dans d'autres fonctions.[19]

III.1.2. Eclipse luna :

Eclipse est un IDE, *Integrated Development Environment* (EDI environnement de développement intégré en français), c'est-à-dire un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation. Il est développé par IBM, est gratuit et disponible pour la plupart des systèmes d'exploitation.[20]

Eclipse Luna comprend un soutien officiel pour Java 8 dans les outils de développement Java, Plug-in Development Tools, objet équipes, cadre Eclipse communication, l'intégration Maven, Xtext, Xtend, Web Tools Platform, et Memory Analyzer. Le compilateur Eclipse comprend des améliorations linguistiques, de recherche et de refactoring, Assistant rapide et nettoyer à migrer classes anonymes aux expressions lambda et le dos, et de nouvelles options de formatage pour lambdas.[21]

III.1.3. Java EE :

Java Enterprise Edition, ou Java EE (anciennement J2EE), est une spécification pour la technique Java d'Oracle plus particulièrement destinée aux applications d'entreprise. Ces applications sont considérées dans une approche multi-niveaux¹. Dans ce but, toute implémentation de cette spécification contient un ensemble d'extensions au *framework* Java standard (JSE, *Java Standard Edition*) afin de faciliter notamment la création d'applications réparties.

Pour ce faire, Java EE définit les éléments suivants :

- Une plate-forme (*Java EE Platform*), pour héberger et exécuter les applications, incluant outre Java SE des bibliothèques logicielles additionnelles du Java Development Kit (JDK) ;
- Une suite de tests (*Java EE Compatibility Test Suite*) pour vérifier la compatibilité ;
- Une réalisation de référence (*Java EE Reference Implementation*), dénommée GlassFish ;
- Un catalogue de bonnes pratiques (*Java EE BluePrints*) [22]

III. L'implémentation

Dans cette partie nous avons présenté l'objectif de notre thème de recherche (la recherche d'information dans le web ubiquitaire basé agent mobile sensible au contexte) qui le montre sur un site d'internet.

C'est un site de visionnement adaptatif de film, la méthode de recherche utilisé c'est la recherche sémantique en utilisant le dictionnaire de wordnet, à base d'agent mobile qui effectue cette recherche le résultat selon des critères d'adaptation contextuelle.

Nous avons utilisé quelque critère dans notre application :

- Critères de performance du dispositif visionnant le film (la fréquence de processur, RAM, disque dur, la résolution de l'écran) ;
- Critères de sujet d'utilisateur et les préférences (évaluation des films, age, pays) on peut rajouter d'autres critères

Pour réaliser cette application nous avons utilisé quelque outils de programmation

III.1. Les différents outils utilisés

III.1.1. Java

Java est un langage de programmation orienté objet à usage multiple, évolué, simple, portable et assurant un usage sécuritaire. Le langage java offre au développeur la possibilité d'appliquer des concepts liés aux quatre paradigmes : impératif, orienté objet, distribué et parallèle.

L'objectif de cette approche est de faciliter la production de programmes possédant de bonnes qualités de modularité : on veut développer une partie d'un programme sans qu'il soit nécessaire de connaître les détails d'implémentation des autres parties (on pourra modifier un module sans que cela n'affecte le reste du programme) tout en favorisant la réutilisabilité.[18]

Java 8 est peut-être l'une des éditions les plus passionnantes du langage Java dans la période récente. Une des caractéristiques de titre est un soutien pour la programmation fonctionnelle qui est l'objet de ce blog. Le soutien vient principalement dans trois caractéristiques:

- 1) Soutien aux pipelines de travail ruisseaux. Streams nous permettent de traiter des données à travers un certain nombre d'étapes dans un pipeline d'une manière fonctionnelle. Nous pouvons créer de tels flux de cas de conteneurs tels que List, ou les créer en utilisant des classes de flux spécifiques tels que IntStream.

III.1.4. Jade

JADE (Java Agent DEvelopment Framework) est un framework implémenté en Java. Il est open-source et distribué sous la licence LGPL (GNU Lesser General Public License). Son but est de permettre le développement de système multi-agents en mettant à disposition un middleware et des outils graphiques Sur des plateformes fixes, téléphones mobiles. JADE est conforme aux spécifications FIPA (Foundations of Intelligent Physical Agents) pour l'interopérabilité des systèmes multi-agents intelligents. Parmi les caractéristiques de cet plateforme : Support à l'envoi de messages, transparent et multi protocoles et la Mobilité des agents.[23]

III.1.5. WampServer

WampServer (anciennement WAMP5) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL. [24]

III.1.6. Hibernate

Hibernate est un framework open source gérant la persistance des objets en base de données relationnelles.

Hibernate est adaptable en termes d'architecture, il peut donc être utilisé aussi bien dans un développement client lourd, que dans un environnement web léger de type Apache Tomcat ou dans un environnement Java EE complet : WebSphere, JBoss Application Server et Oracle WebLogic Server.

Hibernate apporte une solution aux problèmes d'adaptation entre le paradigme objet et les SGBD en remplaçant les accès à la base de données par des appels à des méthodes objet de haut niveau. [25]

III.1.7. Apache Tomcat

Apache Tomcat est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta, c'est un des nombreux projets de l'*Apache Software Foundation*. Il implémente les spécifications des servlets et des JSP du Java Community Process¹, est paramétrable par des fichiers XML et de propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP. [26]

Dans cette partie nous avons présenté les outilles utilisée pour obtenir notre objectif ensuite c'est la représentation détaillé avec les figures de notre site

V. l'implémentation côté client :

L'interface client est développée avec la technologie JEE(pages JSP), en utilisant le serveur tomcat pour le déploiement des pages JSP. Dans notre application on a deux types d'utilisateurs : administrateur et client. L'administrateur qui fait la mise à jour de site (modification, suppression ou l'ajout d'un ou plusieurs films) et l'utilisateur qui effectue la recherche. Dans le premier cas(administrateur) pour entrer dans l'application, il faut connecté avec son pseudo et le mot de passe, si l'utilisateur a un compte donc il va connecté et faire la recherche, sinon il faut faire les inscriptions pour obtenir un compte Ce qui lui permet de faire l'étape suivante c'est l'étape d'évaluation des films proposé par le système pour que ce dernier obtient des informations

Les figures suivantes présentent respectivement les interfaces graphiques relatives à la partie client :

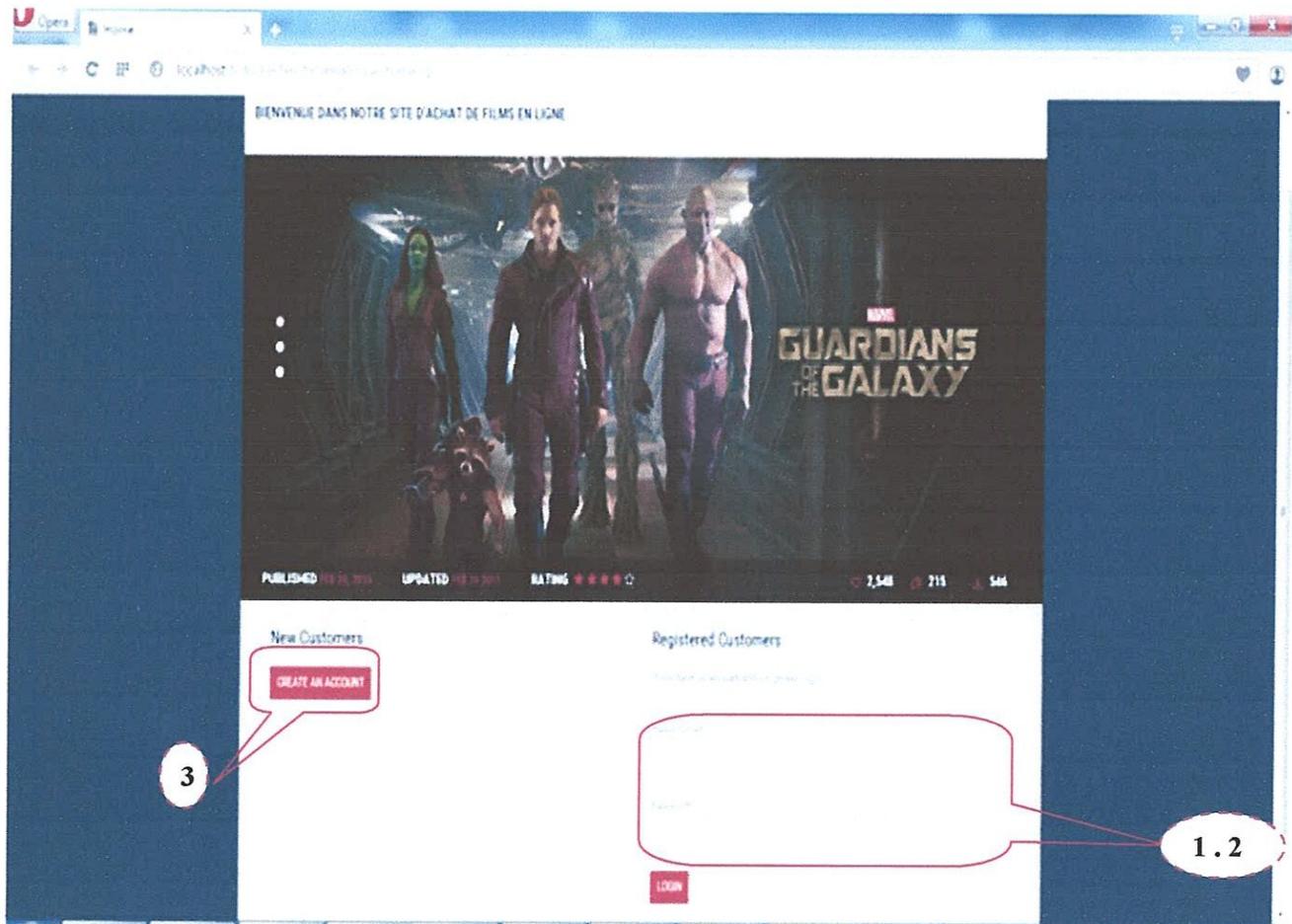


Figure 4.4: fenêtre principale de site internet

Voilà c'est la fenêtre principale de notre site, il y a plusieurs opération que l'on peut faire :

- 1) L'admin de la page : connecté avec le pseudo 'admin' et le mot de passe 'admin' pour faire la mise à jour ;
- 2) Utilisateur a un compte : il va se connecter.
- 3) Nouveau utilisateur : il va crée un compte ;

Ensuite en va présenter la première étape les opérations faites par l'admin.

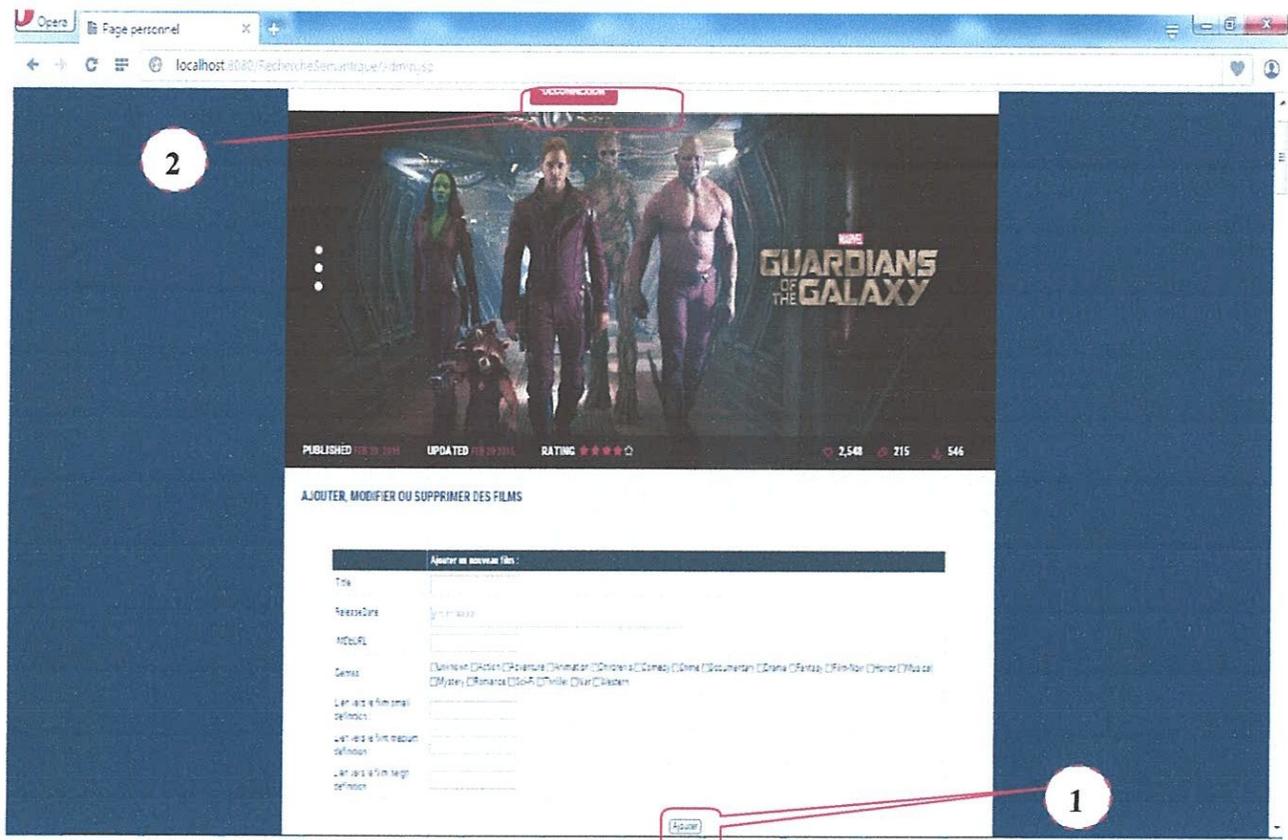


Figure 4.5 : Interface administrateur : ajout de film.

- 1) Si l'administrateur veut ajouter un film il doit remplir le formulaire (le titre de film, la date de réalisation.....) ;
- 2) Si l'administrateur veut se déconnecté.



Figure 4.6 : Interface administrateur : modification des films.

Si l'administrateur veut faire des modifications à un film, il va choisir le film à modifier ensuite faire les modifications sur le nom ou la date... etc après faire la validation pour les enregistrés.

La figure suivante représente les différentes qualités des films à visionner selon les critères d'adaptation de chaque dispositif ;

Numéro du film	Movie Title	Release Date	IMDb URL	Genres	Small definition	Medium definition	High definition	Modifier	Supprimer
1	Toy Story (1995)	01-Jan-1995	Page Internet du film	(Animation Children's Comedy)	Regarder	Regarder	Regarder	Modifier	Supprimer
2	GoldenEye (1995)	01-Jan-1995	Page Internet du film	(Action Adventure Thriller)	Regarder	Regarder	Regarder	Modifier	Supprimer
3	Four Rooms (1995)	01-Jan-1995	Page Internet du film	(Thriller)	Regarder	Regarder	Regarder	Modifier	Supprimer
4	Get Shorty (1995)	01-Jan-1995	Page Internet du film	(Action Comedy Drama)	Regarder	Regarder	Regarder	Modifier	Supprimer
5	Copycat (1995)	01-Jan-1995	Page Internet du film	(Crime Drama Thriller)	Regarder	Regarder	Regarder	Modifier	Supprimer
6	Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)	01-Jan-1995	Page Internet du film	(Drama)	Regarder	Regarder	Regarder	Modifier	Supprimer
7	Twelve Monkeys (1995)	01-Jan-1995	Page Internet du film	(Drama Sci-Fi)	Regarder	Regarder	Regarder	Modifier	Supprimer

Figure 4.7 : Interface administrateur

1, 2 et 3 pour le choix de la qualité de visionnement de film selon les caractéristiques des films et dispositif (l'adaptation à l'environnement) ,

4 et 5 pour la modification des films ou la suppression.

Après on a le deuxième cas ou l'utilisateur est un client mais il n'est pas un compte, il va inscrire pour obtenir le droit de visionnement. La figure suivante montre les actions à faire pour obtenir un compte.

The screenshot shows a web browser window with the URL 'localhost:8080/localhost/Security/register.jsp'. The page content is a registration form for 'MOVIE STORE'. The form is titled 'Formulaire d'inscription' and is divided into two main sections: 'Informations Personnelles' and 'Login Information'.
 Under 'Informations Personnelles', there are the following fields:
 - 'Nom' (Name): text input field.
 - 'Prénom' (Surname): text input field.
 - 'Date de naissance' (Date of Birth): date input field with the value '20/03/2000'.
 - 'Sexe' (Gender): dropdown menu with 'M' selected.
 - 'Occupation' (Occupation): dropdown menu with 'Technicien' selected.
 - 'Pays' (Country): dropdown menu with 'Algerie' selected.
 Under 'Login Information', there are the following fields:
 - 'Pseudo' (Pseudo): text input field.
 - 'Mot de passe' (Password): text input field.
 - 'Confirmez le mot de passe' (Confirm Password): text input field.
 At the bottom left of the form, there is a button labeled 'S'.

Figure 4.8 : les étapes d'inscription

L'utilisateur remplit tous les champs après faire la validation si tous les champs remplis correctement donc validation avec succès sinon il faut refaire tous les étapes correctement.

Après la validation le client connecte avec son pseudo la première chose obligatoire avant la recherche c'est l'évaluation d'un ensemble de film proposé par le système ou moins un film pour que le système prend ces évaluations en compte lors de l'utilisation des critères d'adaptation dans la recherche.

La figure suivante montre les étapes mentionnées précédemment :

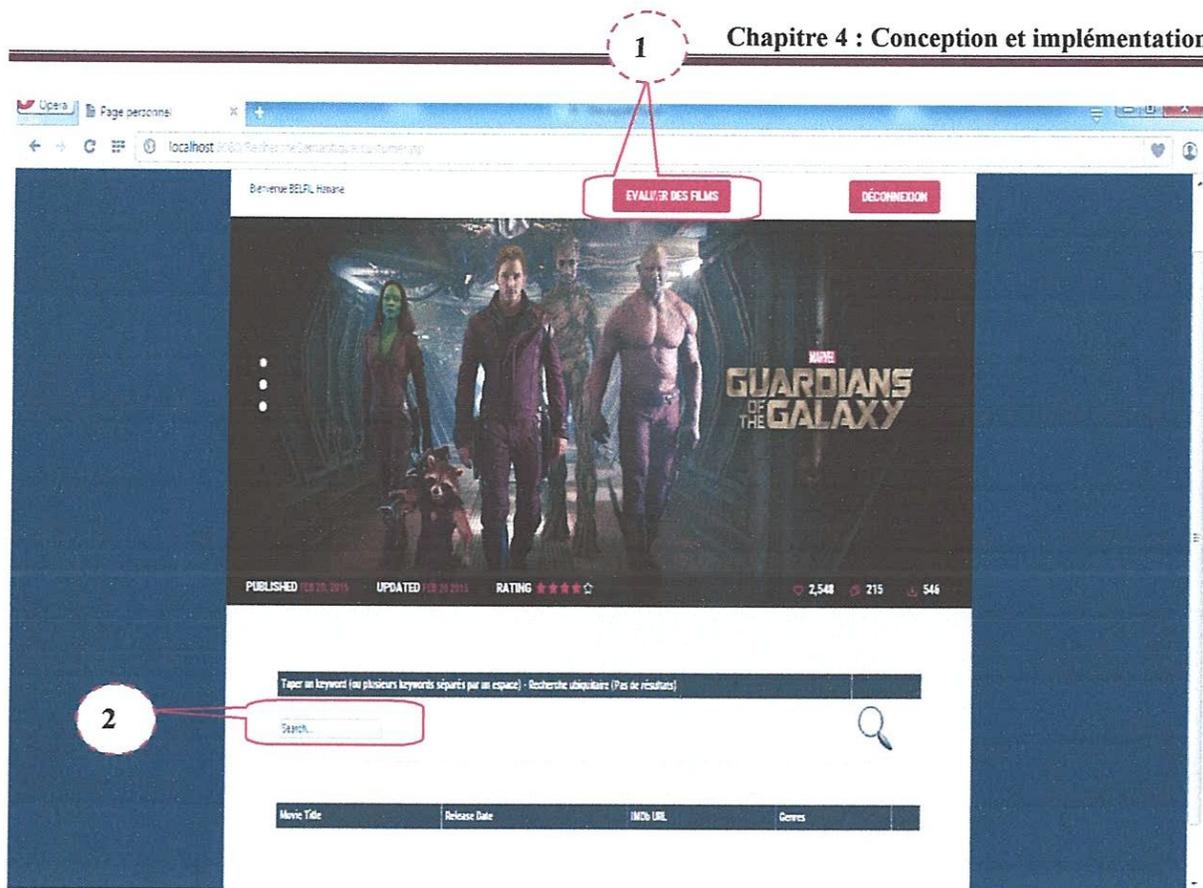


Figure 4.9 : représentation d'espace ou les opérations d'utilisateur

- 1) L'étape d'évaluation de l'utilisateur ;
- 2) L'étape de recherche.

La figure suivante représente l'étape d'évaluation :

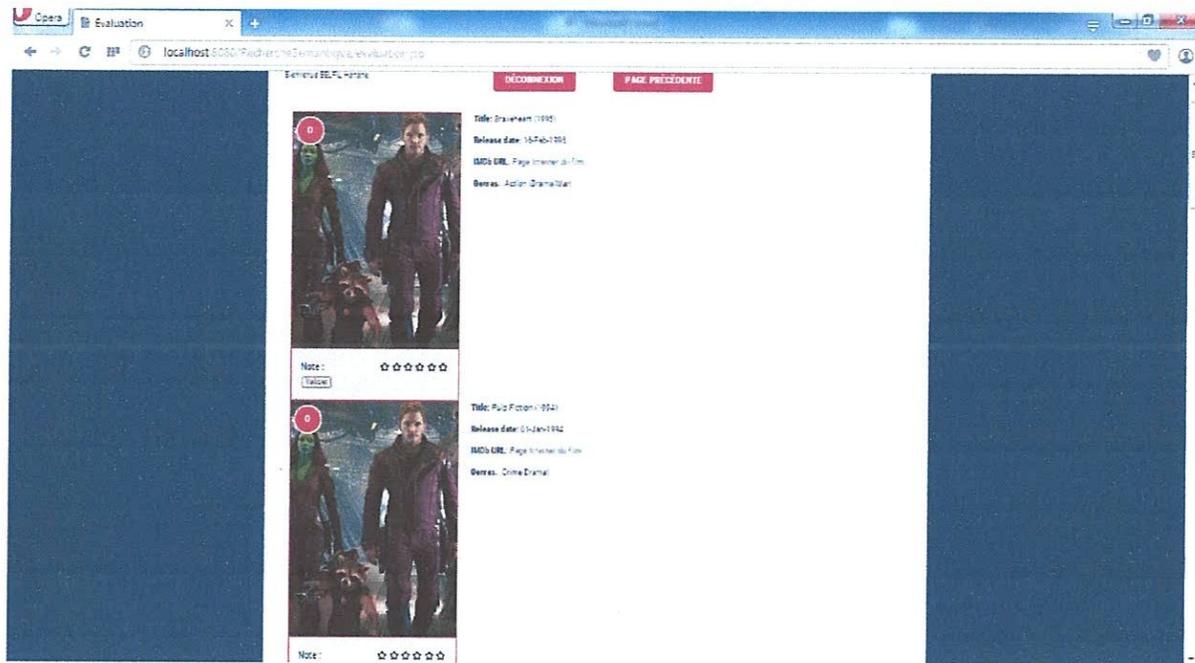


Figure 4.10 : étape d'évaluation

Ensuite pour la recherche il faut que le client écrive le titre de film ou au moins un seul mot ou une lettre pour que le système déclenche la recherche ubiquitaire.

Si le film a cherché existe dans la base de données, le système renvoi un ensemble de films classé selon les critères d'adaptation ; sinon le résultat est introuvable.

La figure suivante représente l'étape de recherche :

Movie Title	Release Date	IMDb URL	Genres	
Toy Story (1995)	01-Jan-1995	Page Internet du film	(Animation Children's Comedy)	Medium definition Regarder le film
From Dusk Till Dawn (1996)	05-Feb-1996	Page Internet du film	(Action Comedy Crime Horror Thriller)	Medium definition Regarder le film
Muppet Treasure Island (1996)	16-Feb-1996	Page Internet du film	(Action Adventure Comedy Musical Thriller)	Medium definition Regarder le film
Aladdin (1992)	01-Jan-1992	Page Internet du film	(Animation Children's Comedy Musical)	High definition Regarder le film
Toy Story (1995)	01-Jan-1995	Page Internet du film	(Animation Children's Comedy)	High definition Regarder le film
From Dusk Till Dawn (1996)	05-Feb-1996	Page Internet du film	(Action Comedy Crime Horror Thriller)	Medium definition Regarder le film

Figure 4.11 : résultat de recherche

Si le client a un compte il va se connecter avec son pseudo et le mot de passe et faire directement l'étape de recherche.

Pendant cour de la recherche l'agent mobile se déplace de premier site vers les autres un par un pour récupérer les résultats, la conversation avec chaque site se fait avec un agent immobile.

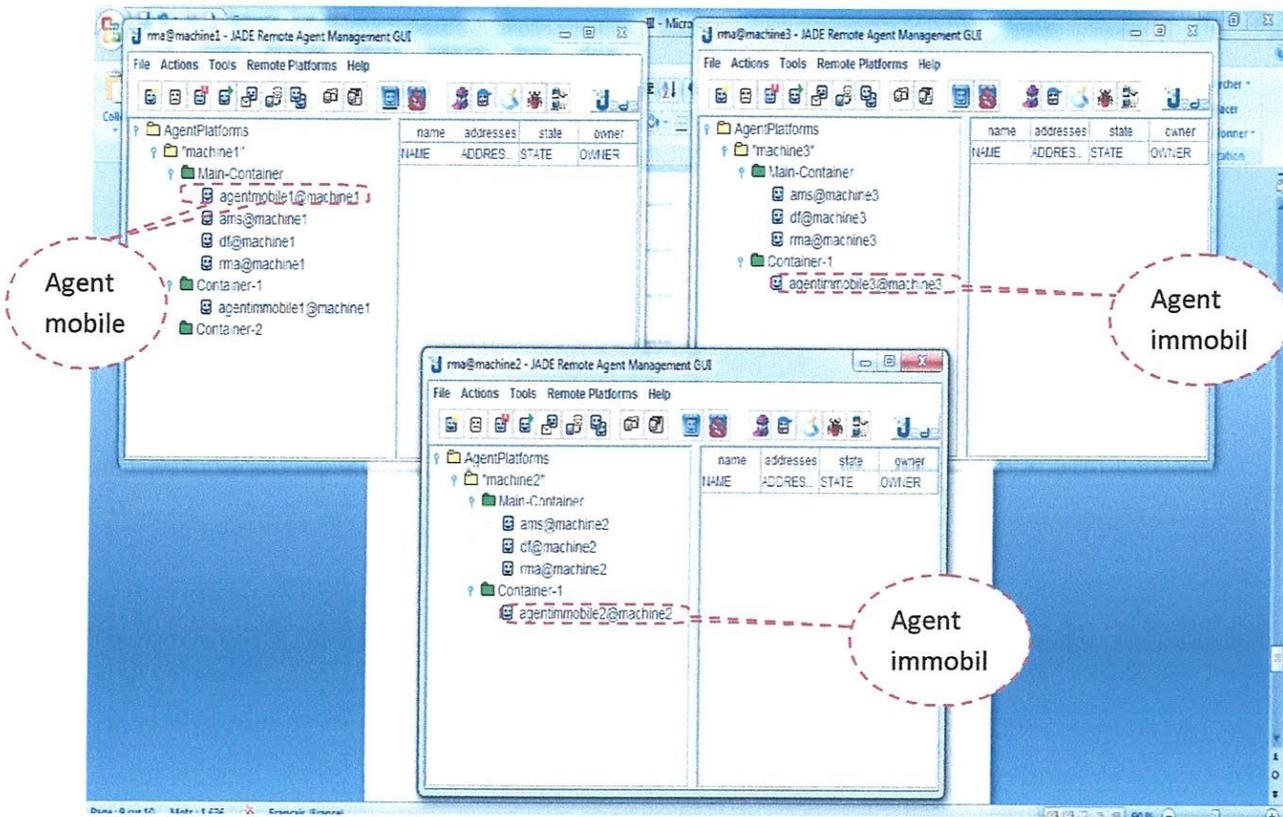


Figure 4.12 : l'agent mobile avant la recherche

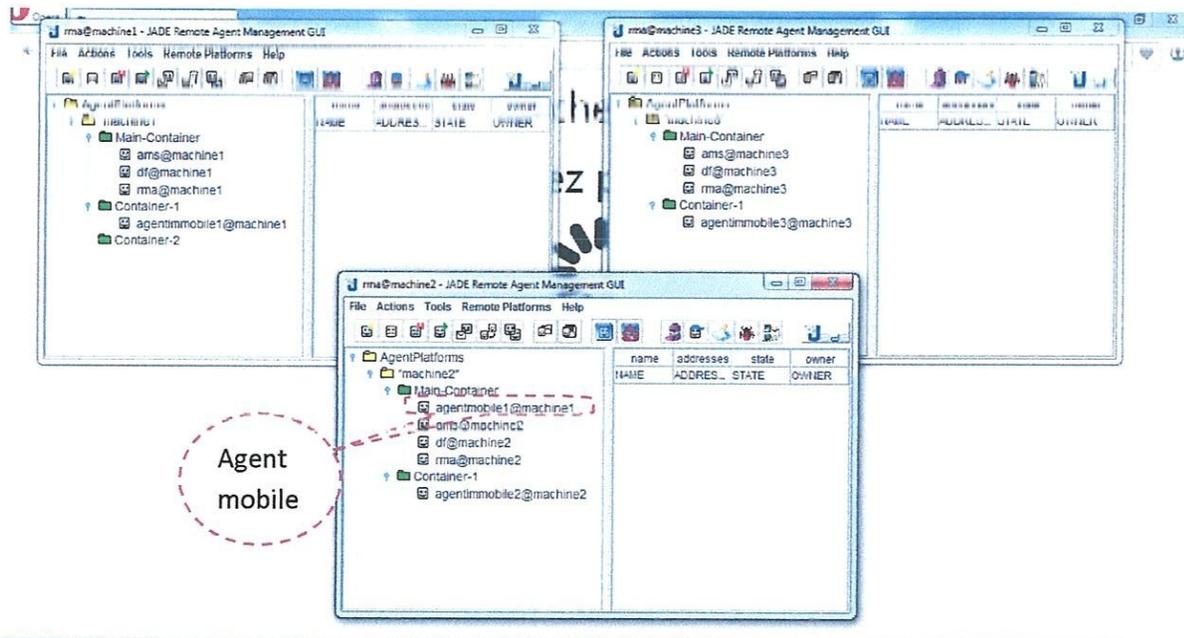


Figure 4.13 : l'agent mobile pendant la recherche se déplace vers la machine 2

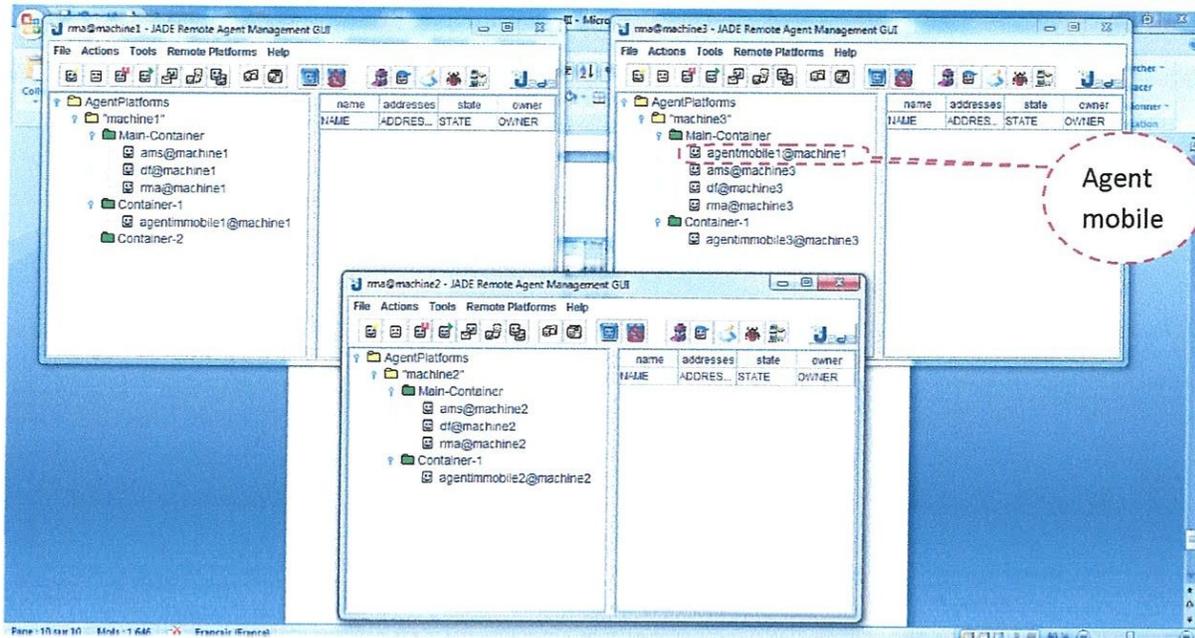


Figure 4.14 : l'agent mobile pendant la recherche se déplace vers la machine 3

VI. Conclusion

Dans ce chapitre nous avons présenté les principes de conception et d'implémentation des différentes parties de notre application.

L'exécution de la partie client et la partie admin est bien déroulée, nous avons montré notre objectif c'est la recherche ubiquitaire dans le web avec un agent mobile Et nous avons obtenu des bonnes résultats

Conclusion générale

Le Web ubiquitaire est un phénomène qui offre beaucoup d'avantage dans la vis quotidienne qui présente en même temps des défis pour la recherche. En effet, les environnements ubiquitaires combinent des aspects complexes tels que la mobilité, les données contextuelles, les interactions multimodales, l'hétérogénéité des dispositifs, les communications sans fil, etc. Le potentiel de ce type d'environnements n'a pas encore été totalement exploité.

La caractéristique principale du web ubiquitaire est l'hétérogénéité de sources d'information, la recherche contextuelle permet d'adapter le contenu et la structure du document au contexte de l'utilisateur, et de diminuer ainsi l'hétérogénéité des sources d'information.

Le grand problème de la recherche d'information dans le web ubiquitaire est le volume et la variation d'information.

Un SRI dans le Web ubiquitaire doit répondre aux requêtes des utilisateurs en utilisant un système distribué, par lequel ils expédient ces requêtes à de nombreux processus de recherche, chacun inspecte à son tour mais le problème c'est comment trouvé l'information pertinente aux utilisateurs pour répondre aux besoins des utilisateurs et évité le problème de la structuration des systèmes de recherche d'information distribuée dans le Web ubiquitaire

Nous avons précisé l'utilisation de l'agent mobile comme outil de génie logiciel et plus précisément dans notre mémoire avec les avantages des agents mobile pour la recherche.

Enfin la meilleur façon pour trouvé l'information pertinente dans le web et plus précisément dans le web ubiquitaire c'est la recherche sémantique c'est la façon ou le moteur de recherche comprend le besoin de l'utilisateur et le sens de la requête pour des bonnes résultats.

Bibliographie

[1]

<http://wapiti.telecomlille1.eu/commun/ens/peda/options/st/rio/pub/exposes/exposesrio2006-ttnfa2007/Rea-Maerens/Informatique%20Ubiquitaire/Introduction.html>

[2] Kevin DZALI et Blaise VESSAH « Systèmes embarqués : réseaux de capteurs et informatique ubiquitaire » ENSET - Master 2 2013

[3] Sofia ZAIDENBERG « Apprentissage par renforcement de modèles de contexte pour l'informatique ambiante » Thèse de doctorat délivrée par INSTITUT POLYTECHNIQUE DE GRENOBLE le 15 septembre 2009

[4] http://www.comoria.com/498988/Informatique_ubiquitaire

[5] Julien Mercadal. « Approche langage au développement logiciel : application au domaine des systèmes d'informatique ubiquitaire. Programming Languages.» Université Sciences et Technologies - Bordeaux I, 2011. French.

[6] Hakim Ghers. « Recherche d'information dans le Web ubiquitaire » Faculté des sciences de l'ingénieur. 2010

[7] Djaafri Abdennaser et Aounie chafik. « Ubi-fractal :une implémentation du modèle de composant Fractal pour informatique ubiquitaire et mobile » université Guelma 2008.

[8] <http://www.w3.org/2006/01/ubiweb-pressrelease.html.fr>

[9] Fatiha Boubekour. « Contribution à la définition de modèles de recherche d'information flexibles basés sur les CP-Nets. Computer Science ». Université Paul Sabatier - Toulouse III, 2008. French.

[10] GASMI Mounira. « Utilisation des ontologies pour l'indexation automatique des sites Web en Arabe ». UNIVERSITE KASDI MERBAH OUARGLA. 2009

[11] Mustapha BAZIZ. « Indexation conceptuelle guidée par ontologie pour la recherche d'information ». Institut de recherche en informatique de toulouse. 2005 .

[12] Sylvie Calabretto & Előd Egyed-Zsigmond. « Recherche d'Information en Contexte ». École d'Automne RIA. 2006.

[13] ABDELKRIM BOURAMOUL. « recherche d'information contextuelle et sémantique sur le web »

- [14] <https://www.yumpu.com/fr/document/view/23511905/indexation-conceptuelle-guidace-par-ontologie-pour-la-irit/17>
- [15] Lynda Tamine-Lechan. « Recherche d'information contextuelle ». Ecole d'Automne en Recherche d'Information: Fondements et Applications EARI -FA,Hammamet, 2014.
- [16] Lynda Tamine-Lechani. « De la recherche d'information orientée système à la recherche d'information orientée contexte : Verrous, contributions et perspectives ». Université Paul Sabatier - Toulouse III, 2008.
- [17] Mariam Daoud. « Recherche d'Information Contextuelle ». IRIT- Equipe SIG/RI 118 route de Narbonne 31062 Toulouse cedex 9 .
- [18] <http://elearning.univ-batna.dz/course/info.php?id=243>
- [19] <http://www.codeproject.com/Articles/792038/An-Introduction-to-Functional-Programming-with-Jav.>
- [20] <http://www.enseignement.polytechnique.fr/informatique/profs/Julien.Cervelle/eclipse/>
- [21] <https://projects.eclipse.org/releases/luna>
- [22] http://fr.wikipedia.org/wiki/Java_EE
- [23] <http://jade.tilab.com/documentation/tutorials-guides/introduction-to-jade/>
- [24] <http://fr.wikipedia.org/wiki/WampServer>
- [25] <http://fr.wikipedia.org/wiki/Hibernate>
- [26] http://fr.wikipedia.org/wiki/Apache_Tomcat
- [27] Mohamed Rida Abdessemed . « Conception par Emergence Inversée d'Agents Autonomes dans le Cadre de Systèmes Complexes Adaptatifs ». Université de Batna Faculté des Sciences. septembre 2013
- [28] ALOUI Ahmed. « Une approche basée agent mobile pour le M-Business ». Université HADJ LAKHDAR – BATNA