

Des agents logiciels et mobiles pour la gestion d'un entrepôt de données

Zakaria Maamar¹ et Gouiaa Mohamed²

¹ *Groupe Interfonctionnement*

Section Technologie des systèmes, d'information

Centre de recherches pour la défense Valcartier

2159 bout. Pie-XI Blvd Nord Val-Bélair, QC G5J 1X5, Canada

zakaria.maamar@drev.dnd.ca

&

Département d'informatique et Centre de recherche en géomatique

Université Laval

² *Département d'informatique*

Université Laval

Ste-Foy, QC G1K 6P1, Canada

gouiaa@ift.ulaval.ca

RÉSUMÉ- Cet article présente un projet de recherche qui traite des problèmes d'intégration de source d'informations distribuées et hétérogènes. Un environnement d'intégration a pour objectif de fournir aux usagers des données pertinentes et consolidées. Une approche de solution pour atteindre cette intégration est de construire un entrepôt de données. Ceci est un répertoire de données qui est disponible pour répondre à des requêtes de prise de décision. De plus, afin de gérer de manière efficace un entrepôt de données, des agents logiciels dotés de mécanismes de mobilité sont introduits {un agent logiciel est une entité autonome qui est capable de collaborer avec d'autres agents et de répondre aux besoins des usagers. Pour exécuter leurs opérations, les agents peuvent se déplacer de leurs sites et parcourir le réseau à la recherche de données pertinentes.

MOTS-CLES : Entrepôt de données, Agents logiciels/mobiles, Scénarios..

1. Introduction

Avec l'évolution des technologies de l'information et de communication, fournir aux usagers des données pertinentes et à jour est devenu une opération très complexe. En général, ces données proviennent de plusieurs sources d'informations distribuées à travers les réseaux et hétérogènes à différents niveaux (matériels, logiciels et surtout terminologiques). Par conséquent, il est devenu nécessaire de proposer des approches d'intégration de ces sources d'information.

Deux approches d'intégration de sources d'informations distribuées et hétérogènes peuvent être utilisées [10]: l'approche "paresseuse" (pour lazy en anglais) qui se caractérise par l'extraction des données à la demande des usagers et l'approche *entrepôt de données* (pour data warehouse en anglais) qui se caractérise par le stockage préalable de données consolidées et d'intérêts [4]. Cependant, il n'existe pas d'approches de conception pouvant orienter les concepteurs dans l'application de la technologie des *agents logiciels* au domaine des entrepôts de données. Par conséquent, notre travail vise à proposer une telle approche. Ce travail fait partie d'un projet de grande envergure, intitulé SAW¹, qui traite de l'application de la technologie des agents à la mise en place d'environnements soumis à des contraintes de distribution et d'hétérogénéité.

¹ SHAW est l'acronyme de software Agents for Warehousing.

Actuellement, l'approche entrepôt de données fait partie des intérêts de recherche et de développement de la communauté des bases de données et de l'industrie. Une telle approche suppose les situations suivantes [10]:

- Des usagers ayant besoin de données consolidées servant à des indicateurs de gestion. Nous entendons par usagers des cadres de haut niveau.
- Des usagers ayant besoin d'exécuter leurs requêtes avec une très grande performance, mais sans se soucier de l'état des données au niveau des sources d'informations. Cette performance est due au fait que les sources d'informations ne sont pas impliquées dans l'exécution des requêtes.

Une approche d'intégration basée sur un entrepôt de données doit s'assurer que les sources d'informations restent *autonomes* et *indépendantes*. Pour atteindre cet objectif, et comme solution possible, nous suggérons d'introduire plusieurs composantes spécialisées, appelées *agents logiciels* [5]. Ces agents sont considérés comme les *représentants* (pour front-ends en Anglais) de ces sources et ont par conséquent, la responsabilité d'agir en leurs noms. De plus, en raison de la contrainte de distribution des sources d'informations, nous suggérons de doter certains de ces agents de mécanismes de *mobilité*. Le paradigme du code mobile est une nouvelle tendance de recherche en Intelligence Artificielle Distribuée (IAD) [6]. Le principe fondamental de ce paradigme est de permettre à un agent de se déplacer d'un site hôte à un autre site pour exécuter des opérations spécifiques. Chaque site fait partie d'un réseau de communication, comme l'Internet. D'une manière générale, la mobilité d'un agent dépend de la *disponibilité* des ressources, telles que les données des sources d'informations dans le cas de l'environnement SAW, dont l'agent a besoin pour exécuter ses opérations [8].

Cet article présente les principaux concepts nécessaires à l'utilisation de la technologie des agents logiciels et du code mobile dans la mise en place d'un entrepôt de données. La section 2 présente un aperçu général des domaines entrepôt de données et agents logiciels. La section 3 décrit les caractéristiques architecturales et fonctionnelles de l'environnement SAW. Finalement, la section 4 résume les principaux éléments apportés dans notre étude.

2. Entrepôts de données et agents logiciels

Cette section décrit brièvement les technologies suivantes: entrepôt de données, agents logiciels et code mobile.

L'entrepôt de données est une nouvelle technologie dans le domaine de l'*intégration* de multiples sources d'informations distribuées et hétérogènes. Cette technologie a pour objectif de mettre en place un entrepôt de données, destiné essentiellement à des requêtes de *prise de décision*. En général, cet entrepôt est le résultat de *vues consolidées* au dessus des données contenues dans les diverses sources d'informations. Chaque vue répond à des besoins usagers spécifiques et se base sur l'application d'opérations de manipulation de données.

Le mode de fonctionnement d'un entrepôt de données se résume comme suit [10]:

- Les données de chaque source d'informations sont extraites à l'avance, traduites, filtrées, fusionnées avec les données d'autres sources et enfin, stockées dans l'entrepôt de données.
- Lorsqu'une requête est posée, celle-ci est directement exécutée au niveau de l'entrepôt de données, et cela sans avoir recours aux sources d'informations.

La Figure 1 ci-dessous présente l'architecture traditionnelle d'un entrepôt de données. Au niveau le plus bas, plusieurs types de sources d'informations existent, à titre d'exemple des fichiers texte, des bases de connaissance, des bases de données, etc. De plus, trois composantes principales prennent en charge le fonctionnement de l'entrepôt de données: l'*interface* (pour Wrapper en Anglais), le *Surveillant* (pour Monitor en Anglais), et l'*Intégrateur* (pour Integrator en Anglais).

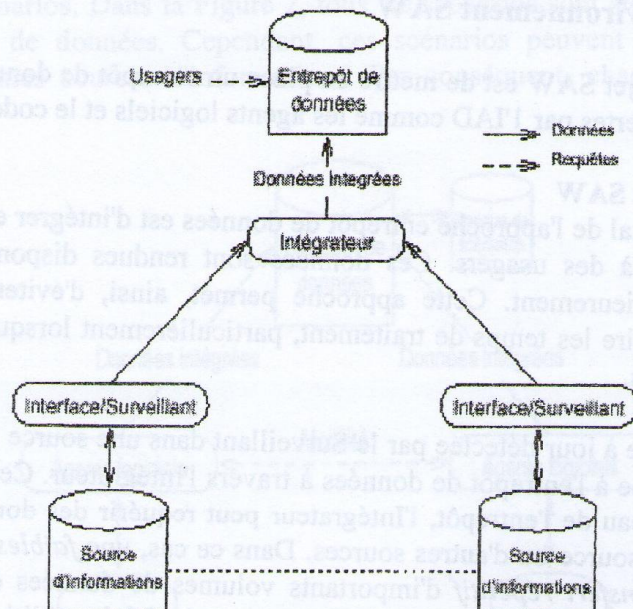


Figure 1. Architecture traditionnelle d'un entrepôt de données

- **Interface:** modifie les données d'une source de leur format initial dans le format utilisé par l'entrepôt de données.
- **Surveillant:** contrôle les mises à jour qui se produisent au niveau d'une source d'informations afin de notifier ces mises à jour à l'Intégrateur.
- **Intégrateur:** gère les données de l'entrepôt. Pour cela, il filtre les données, les résume ou les fusionne avec les données d'autres sources avant de mettre en place les nouvelles données dans l'entrepôt.

Pour résumer, un entrepôt de données est considéré comme un lieu de stockage qui assure la cohérence et l'intégrité des données en provenance de plusieurs sources d'informations distribuées et hétérogènes [4].

En IAD, les chercheurs ont étudié différents aspects liés à la distribution et coordination des connaissances et actions dans une communauté de plusieurs agents. Ces agents peuvent prendre différentes formes selon la nature de l'environnement dans lequel ils évoluent. Un type particulier, appelé *agent logiciel*, attire actuellement beaucoup d'attention [1]. Les agents logiciels sont des entités autonomes pouvant aider leurs usagers à satisfaire leurs besoins et collaborer entre eux pour résoudre des problèmes complexes.

L'IAD distingue deux domaines de recherche: la Résolution Distribuée de Problèmes (RDP) et le Multi-Agents (MA). Dans la RDP, l'emphasis est mis sur la problématique à résoudre, en faisant intervenir plusieurs agents en même temps. Dans le MA, l'emphasis est mis sur la coordination des activités d'agents autonomes afin d'éviter les situations de conflits. De plus, avec le développement rapide des technologies de communication, comme l'Internet, le paradigme du code mobile attire

beaucoup d'attention pour sa flexibilité et efficacité [6]. Ce paradigme est vu comme une nouvelle alternative au mode de fonctionnement traditionnel *client-serveur*. Un agent doté de mécanismes de mobilité peut, par exemple, quitter son site d'origine et parcourir le réseau afin de recueillir les données appropriées pour son usager. De plus, cet agent peut accéder aux ressources nécessaires de manière très efficace puisqu'il se déplace à la localisation de ces ressources au lieu de communiquer avec celles-ci à travers des connexions, parfois à *faible bande passante*.

3. Présentation de l'environnement SAW

Un des objectifs du projet SAW est de mettre en place un entrepôt de données, en se basant sur les différentes techniques offertes par l'IAD comme les agents logiciels et le code mobile.

3.1. Aperçu sur le projet SAW

Le principe fondamental de l'approche entrepôt de données est d'intégrer et de stocker au préalable les données pertinentes à des usagers. Ces données sont rendues disponibles pour des requêtes pouvant avoir lieu ultérieurement. Cette approche permet, ainsi, d'éviter les accès aux sources d'informations et de réduire les temps de traitement, particulièrement lorsque les sources impliquées sont multiples et distantes.

Évidemment, une mise à jour détectée par le Surveillant dans une source d'informations spécifique (Fig. 1) doit être propagée à l'entrepôt de données à travers l'Intégrateur. Cependant, avant de valider cette mise à jour au niveau de l'entrepôt, l'Intégrateur peut requérir des données supplémentaires en provenance de la même source ou d'autres sources. Dans ce cas, une *faiblesse* de l'approche entrepôt de données serait le *transfert répétitif* d'importants volumes de données du niveau source vers le niveau intégration. Comme solution possible, des agents logiciels additionnés de mécanismes de mobilité et de techniques de *scénarios* sont suggérés. Par exemple, si une mise à jour est détectée dans une source d'informations, les composantes Surveillant et Interface, "*élevés*" au rang d'agents logiciels, demandent au préalable le scénario² de fonctionnement qui correspond à la mise à jour détectée. De plus, si ce scénario indique que d'autres sources d'informations sont nécessaires pour finaliser les opérations de mise à jour de l'entrepôt, alors l'Interface se déplace à ces sources et agit comme étant un Intégrateur (Fig. 2).

Un entrepôt de données résulte de la consolidation de plusieurs vues, qui impliquent une ou plusieurs sources d'informations. Chaque vue est déjà connue par la personne responsable, c'est-à-dire l'administrateur, de cet entrepôt. En fait, cette personne connaît à l'avance les sources d'informations nécessaires à la satisfaction des besoins des usagers. Comme suggéré dans la Figure 2, des agents logiciels sont utilisés dans la gestion de l'environnement SAW. Par contre, tous ces agents n'ont pas besoin d'être mobiles. Un agent statique est nécessaire pour gérer les scénarios qui décrivent comment mettre à jour l'entrepôt de données. De plus, cet agent doit aussi interagir avec l'administrateur de l'entrepôt au moment des mises à jour des scénarios.

L'amélioration des composantes de base, c'est-à-dire Interface, Surveillant, et Intégrateur, d'un entrepôt de données en utilisant le paradigme des agents logiciels est une perspective de recherche dans le domaine des entrepôts de données. Les technologies des agents logiciels et du code mobile permettent de réduire le volume de données à être transféré des sources d'informations vers l'entrepôt. De plus, toutes les opérations d'intégration sont réalisées au niveau des sources d'informations. Précédemment, ces opérations sont faites par l'intégrateur qui doit, généralement, récupérer les données plusieurs fois de leurs sources respectives (Fig. 1).

Il est à noter qu'avec l'évolution de la technologie des composantes électroniques et de ses diverses retombées (coûts des ordinateurs, capacités des disques durs, vitesses des processeurs, etc.), les scénarios pré-définis de l'environnement SAW peuvent être spécifiés de manière à optimiser les

² Étant donné qu'un entrepôt de données est un ensemble de vues pré-définies, chaque vue peut être décrite à l'avance par son propre scénario.

opérations d'intégration. Pour cela, il s'agit de prendre en compte les caractéristiques de chaque source d'information en termes de type (base de données, fichier, etc.), de langage d'implantation, de performance de matériel de support, de type de réseau (LAN *versus* WAN), etc.

Une autre perspective de recherche dans le projet SAW consiste à utiliser une approche *distribuée* pour la gestion des scénarios. Dans la Figure 2, tous les scénarios sont stockés dans une *Banque* qui appartient à l'entrepôt de données. Cependant, ces scénarios peuvent être délégués de manière appropriée aux différentes sources d'informations. Par conséquent, chaque source gère sa propre banque de scénarios.

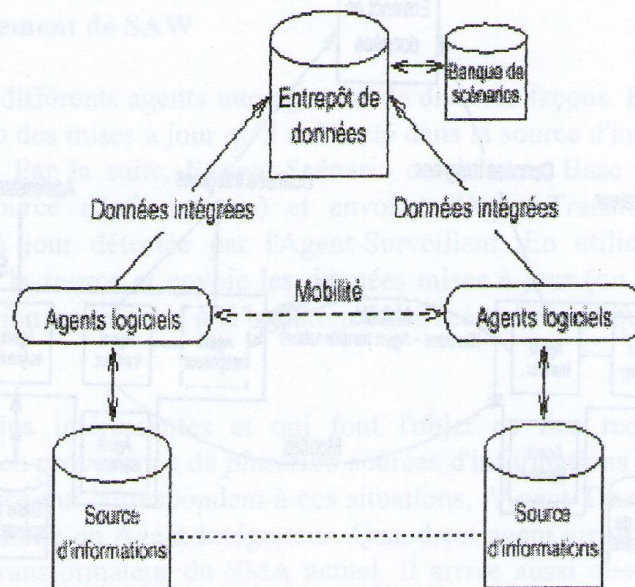


Figure 2. Architecture préliminaire de l'environnement SAW

Pour récapituler, deux perspectives de recherches sont identifiées dans le contexte du projet SAW, savoir:

- Comment utiliser la technologie des agents logiciels pour mettre en place un entrepôt de données?
- Comment utiliser le code mobile et la technique des scénarios pour gérer le fonctionnement de cet entrepôt de données?

Les sections suivantes décrivent l'environnement SAW en termes d'architecture, de mode de fonctionnement et d'implantation.

3.2. Architecture de SAW

Pour la conception de l'architecture de SAW, nous avons appliqué les mêmes principes énoncés dans le domaine de l'intégration de systèmes d'informations. Ces principes se résument comme suit [2]:

- *Evolutivité*: l'environnement d'intégration doit être extensible et permettre ainsi un développement incrémental. L'intégration peut commencer avec des parties réduites de l'organisation, pour s'étendre graduellement au reste de l'organisation.
- *Adaptabilité*: les systèmes utilisant des technologies de l'information standards ou non, peuvent être intégrés à l'environnement d'intégration sans causer de perturbations à l'architecture de cet environnement.

- *Autonomie*: les systèmes doivent être flexibles dans leur conception, construction et administration. Les caractéristiques de l'environnement d'intégration ne doivent pas influencer ces systèmes.

La Figure 3 présente l'architecture détaillée de l'environnement SAW. Dans cette architecture, chaque source d'informations est encapsulée dans un Système Multi-Agents (SMA) qui assure l'autonomie et l'indépendance de cette source vis-à-vis de l'entrepôt de données.

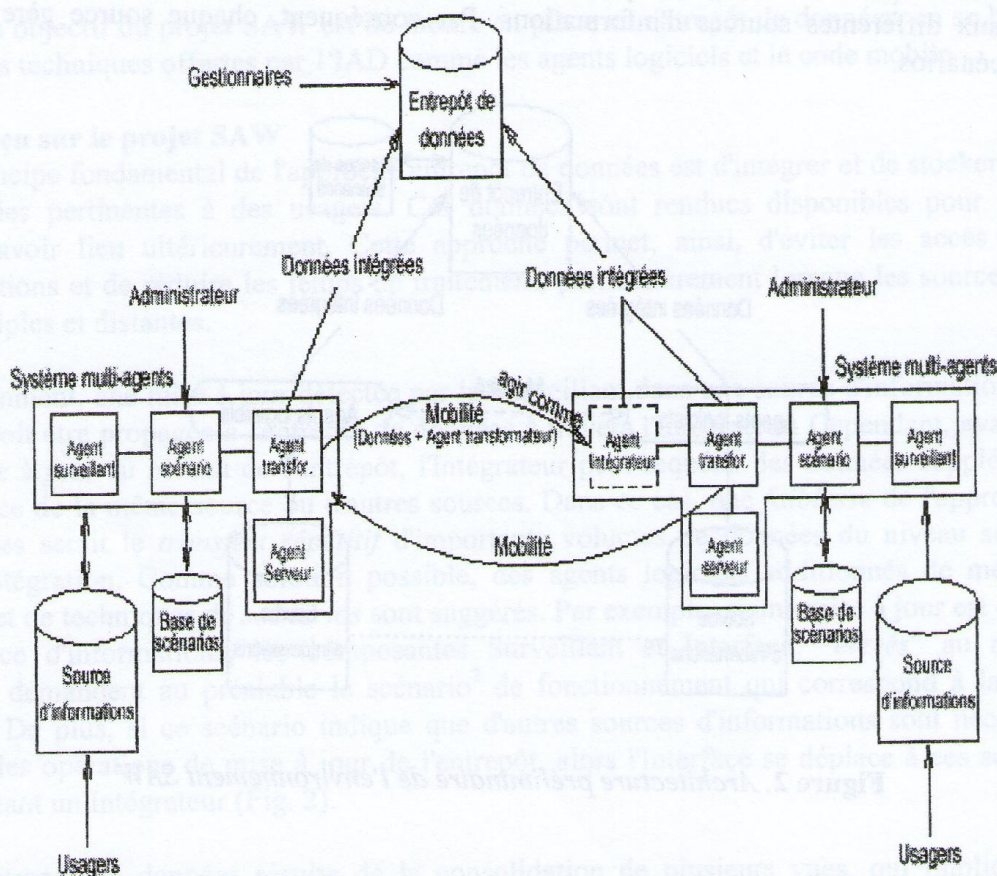


Figure 3. Architecture détaillée de l'environnement SAW

De plus, en se basant sur les principes d'intégration énoncés précédemment et sur les caractéristiques architecturales de SAW, différents types d'agents logiciels sont identifiés (en fait, certains de ces agents s'associent aux composantes de base d'un entrepôt de données):

- *Agent-Surveillant*: surveille les changements qui interviennent au niveau d'une source d'informations lorsque celle-ci est mise à jour. Les mises à jour correspondent aux opérations d'instantiation exécutées par des usagers.
- *Agent-Scénario*: dispose à son niveau des scénarios qui décrivent les opérations requises pour mettre à jour le contenu de l'entrepôt de données. Les multiples scénarios sont disponibles dans une *Base de Scénarios*, qui est accessible par l'Agent-Scénario. De plus, cet agent interagit avec l'administrateur de l'environnement entrepôt de données pendant la mise à jour de la Base de Scénarios.
- *Agent-Transformateur*: connaît les protocoles à travers lesquels une source d'informations accepte des requêtes et retourne des réponses. De plus, cet agent traduit les informations du format initial de la source dans le format de données de l'entrepôt. La caractéristique majeure de l'Agent-Transformateur est sa capacité d'agir comme un Agent-Intégrateur lorsqu'il se déplace vers d'autres SMAs distants afin de rencontrer leurs Agent-Transformateurs (cf. Section 3.3).

Dans un environnement où les agents sont dotés de mécanismes de mobilité, les opérations de mobilité consistent à transférer les agents à travers le réseau vers d'autres sites distants, à authentifier ces agents une fois arrivés à destination et finalement, à installer ces agents afin qu'ils reprennent leurs opérations. Pour prendre en charge toutes ces fonctionnalités dans l'environnement SAW, nous introduisons un nouveau type d'agent, appelé *Agent-Serveur* dans l'architecture de la Figure 3. Par exemple, lorsque l'Agent-Transformateur décide de se déplacer vers un SMA distant et d'agir comme un Agent-Intégrateur dans ce SMA, selon un scénario donné, cet agent interagit en premier lieu avec son Agent-Serveur.

3.3. Mode de fonctionnement de SAW

Dans la Figure 3, les différents agents interagissent de diverses façons. En fait, l'Agent-Surveillant informe l'Agent-Scénario des mises à jour qu'il a détecté dans la source d'informations qui correspond à cet Agent-Surveillant. Par la suite, l'Agent-Scénario consulte sa Base de Scénarios (celle-ci est spécifique à chaque source d'informations) et envoie à l'Agent-Transformateur le scénario qui correspond à la mise à jour détectée par l'Agent-Surveillant. En utilisant ce scénario, l'Agent-transformateur accède à la source et envoie les données mises à jour (ou modifiées) à l'entrepôt de données. Une telle situation correspond à la situation où le scénario impliqué fait intervenir une seule source d'informations.

Les situations les plus intéressantes et qui font l'objet de nos recherches correspondent à l'intégration de données en provenance de *plusieurs* sources d'informations hétérogènes et distribuées. En fonction des scénarios qui correspondent à ces situations, l'Agent-Transformateur se déplace vers d'autres SMAs et agit comme un Agent-Intégrateur. Quand cet agent arrive à destination, il demande les données à l'Agent-Transformateur du SMA actuel. Il arrive aussi que pour avoir des scénarios optimisés, en terme de volumes de données à transférer entre les sources, que l'Agent-Transformateur transmette une demande de déplacement pour un autre Agent-Transformateur afin que ce dernier vienne le rejoindre dans son propre SMA. Cette situation s'appelle une *invitation*.

Le mode de fonctionnement de l'environnement SAW est spécifié par des scénarios qui dépendent des types de mises à jour des sources d'informations. Dans notre travail, nous considérons un scénario comme un *réseau* dont les nœuds correspondent au couple (SMA, Source-Information) et les *liens* correspondent aux opérations de déplacement entre les nœuds. Par exemple, la Figure 4 illustre un scénario qui implique trois sources d'informations (SI-1, SI-2, SI-3) et leurs SMAs respectifs (SMA-1, SMA-2, SMA-3). Les nombres dans la même figure indiquent la chronologie d'exécution des opérations de traitement. Une fois que l'Agent-Surveillant du SMA-1 a détecté (0) une mise à jour au niveau de la SI-1, cet agent utilise le scénario qui lui est fourni par l'Agent-Scénario. Ce scénario mentionne que l'Agent-Transformateur du SMA-1 doit se déplacer vers le SMA-2. Quand cet Agent-Transformateur arrive au SMA-2 (1), il agit comme étant un Agent-Intégrateur et collabore avec l'Agent-Transformateur du SMA-2. Chacun des agents, Intégrateur et Transformateur, fournit ses propres données afin d'accomplir les opérations indiquées dans le scénario en cours d'utilisation (2). De plus, ce scénario mentionne que d'autres sources d'informations, à savoir la SI-3, sont nécessaires avant que la mise à jour de l'entrepôt de données ne soit finalisée. Par conséquent, l'Agent-Intégrateur, qui est actuellement dans le SMA-2, invite (3) l'Agent-Transformateur du SMA-3 à se déplacer (4) avec les données appropriées de la SI-3. Les résultats partiels fournis au cours de l'étape 2 sont intégrés avec les données de la SI-3 (5). Finalement, l'entrepôt de données est mis à jour (6).

Spécification des scénarios

Le tableau 1 contient le langage proposé dans la spécification des scénarios de l'environnement SAW. Nous utilisons une représentation à base de prédicats et le symbole suivant $*[<...>]^*$ pour signifier 0...n.

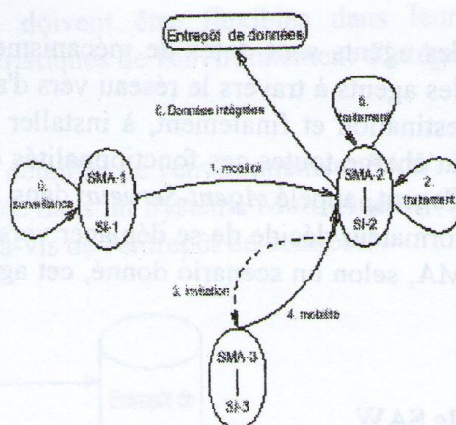


Figure 4. Temple du mode de fonctionnement de l'environnement SAW

Tableau 1. langage de Pacification de scénarios dans l'environnement SAW

1	<scénario>	::=	<tête_scén.> <corps_scén.>
2	<tête_scén.>	::=	<id_scénario> <id_source> <id_agent>
3	<corps_scén.>	::=	Si <condition> Alors <opérations>
4	<condition>	::=	insertion modification suppression "données"
5	<opérations>	::=	traiter "données" *(<opération>)* mettre-à-jour <id_entrepôt_de_données>
6	<opération>	::=	<op._migra.> <op._invita.>
7	<op._migra.>	::=	migrer <id_agent> "données" vers <id_source> traiter <données>
8	<op._invita.>	::=	inviter <id_agent> "données" vers <id_source> <op._migra.>

Dans le paragraphe suivant, les numéros entre parenthèses correspondent aux numéros du Tableau 1.

Un scénario (1) se compose de deux parties: la *tête* et le *corps*. La partie tête identifie le scénario, la source d'informations et l'agent responsable de ce scénario (2). La partie corps consiste en une règle à valider avant de pouvoir déclencher le scénario (3). Si la partie condition de cette règle est vraie, alors une mise à jour (c'est-à-dire, soit une insertion, soit une modification, soit une suppression) se produit au niveau des données de la source d'informations (4) et des opérations spécifiques doivent être réalisées (5). Ces opérations consistent à traiter les données mises à jour, à se déplacer ou non vers d'autres sources d'informations (6,7,8) et à mettre à jour l'entrepôt de données.

3.4. Implantation de SAW

Plusieurs aspects en relation avec l'implantation de l'environnement SAW doivent être pris en compte. n s'agit de: Comment implanter les agents de SAW? Comment permettre aux agents de SAW de se déplacer? Et, comment permettre aux agents de SAW d'accéder aux sources d'informations?

Actuellement, plusieurs langages de programmation d'agents mobiles existent, tels que java, Tcl, Telescript, etc. [3]. Dans le cadre du projet SAW, nous projetons de développer notre prototype avec le langage Java. Par contre, malgré sa caractéristique de portabilité et son leadership comme langage de programmation du Web, Java est vu comme le langage qui assure la plus "faible" mobilité dans un environnement d'agent mobiles. En effet, Java supporte le déplacement des agents qu'une seule fois [7]. Par contre, nous pensons que l'utilisation de l'Object Request Broker³ (ORB) Voyager [9] permet

³ Un ORB est une plate-forme logicielle qui respecte les conventions proposées. par l'Object Management Group (OMG) dans la mise en place d'applications client/serveur à base d'objets distribués.

d'éliminer cette faiblesse. Voyager permet de rester en accord avec les tendances actuelles de standardisation entreprises par l'OMG au niveau des systèmes d'objets distribués et offre des services de support pour les agents mobiles..

Avec Voyager, un agent se déplace en utilisant la commande *move()* avec comme paramètres l'adresse du site de destination et le nom de la fonction à exécuter une fois arrivé à ce site. Avant de se déplacer, cet agent attend jusqu'à ce que toutes les opérations en attente soient finalisées, laisse derrière lui une "secrétaire" afin de lui transmettre les messages qui lui arrivent et enfin, se déplace.

Les agents de l'environnement SAW, en particulier l'Agent-Surveillant et l'Agent-Transformateur, doivent accéder à des sources d'informations pour exécuter leurs requêtes. Généralement, pour accéder à une source d'informations, par exemple une base de données, le protocole le plus utilisé est le *Open DataBase Connectivity* (ODBC) de Microsoft. Cependant, en raison du développement rapide des technologies d'Internet et d'Intranet, un autre protocole a fait son apparition. Il s'agit du *Java DataBase Connectivity* (JDBC). JDBC est une interface de programmation d'application Java qui permet des accès transparents à des bases de données locales ou distantes.

4. Conclusion

Dans cet article, nous avons présenté les caractéristiques principales de l'environnement SAW qui traite de l'application des agents logiciels et du code mobile dans la gestion d'entrepôts de données. Les approches de solution discutées dans cet article font partie d'un projet de grande envergure dont l'objectif est d'utiliser des agents dans des environnements soumis à des contraintes de distribution et d'hétérogénéité. Les systèmes entrepôt de données sont devenus une composante clé dans la stratégie de développement des technologies de l'information des organisations.

Le projet SAW fait l'objet de recherches continues. Actuellement, nous entamons la phase de développement du prototype qui valide l'architecture de SAW. SAW utilise JAVA comme langage de programmation des agents, le protocole JDBC pour accéder aux sources d'informations disponibles et enfin, l'ORB Voyager pour spécifier le comportement des agents pendant les opérations de distribution et de mobilité.

BIBLIOGRAPHIE

- [1] Special issue on intelligent services. *Communication of the ACM*, 37(7), 1994.
- [2] G. Rabin, C. Hsu, and Z. Maamar. A distributed metadatabase architecture for an enterprise scalable, adaptive integration environment. Technical Report 37-94-424, DSES, Rensselaer Polytechnic Institute, Troy, N.Y., USA, 1994.
- [3] G. Cugola, C. Ghezzi, G. Picco, and G. Vigna. A characterization of mobility and state distribution in mobile code language. In *Proceedings of the 2nd ECOOP Workshop on Mobile Object Systems*, pages 10-19, Linz, Austria, July 8-9 1996.
- [4] B. Devlin. *Data Warehouse from Architecture to Implementation*. Addison-Wesley, 1997.
- [5] H.-S. Nwana. Software agents: An overview. *Knowledge Engineering Review*, 11(3):1-40, September 1996.
- [6] Dartmouth Workshop on Transportable Agents '97 (DWTA'97). Dartmouth College, Hanover, NH, USA, 1997.
- [7] J. Ordille. When agents roar, who can you trust? In *Proc. First. Conf. On Emerging Technologies and Applications in Communications (etaCOM)*, May 1996.
- [8] B. Venners. Solve real problems with aglets, a type of mobile agent. Technical report, Netscape World Magazine, May 1997.
- [9] Voyager. ObjectSpace, <http://www.objectspace.com>.
- [10] J. Widom. Research problems in data warehousing. In *Proceedings of the 4th International Conference on Information and Knowledge Management, (CIKM'95)*, November 1995.