

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE SAAD DAHLAB BLIDA



Faculté des Sciences

Département d'Informatique

MÉMOIRE DE FIN D'ÉTUDES

POUR L'OBTENTION DU DIPLÔME MASTER EN INFORMATIQUE

OPTION : INGÉNIERIE DE LOGICIEL

THÈME

Conception et réalisation des requêtes complexes dans un système de médiation : cas des requêtes décisionnelles du département formation de SONATRACH

RÉALISÉ PAR :

Melle. ABDELLAOUI SOUMIYA

Melle. MORCELI NOUR ELHOUDA

PROMOTRICE : Mme. FAREH

ENCADREUR : Mr. TARBINT

Mme F. Boumaladi



2013/2014



Dédicace

Merci Allah de m'avoir donné la capacité d'écrire et de réfléchir
, la patience d'aller jusqu'au bout du rêve et le bonheur
de lever mes mains vers le ciel et de dire

" Ya Kayoum "



Je dédie ce modeste travail
A celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée
pour mon bonheur et ma réussite, à ma mère ...
A mon père, école de mon enfance, qui a été mon ombre durant
toutes les années des études, et qui a veillé tout au long de ma vie
à m'encourager, à me donner l'aide et à me protéger
Que dieu les garde et les protège, ainsi à mes grands-parents.
Mon cher frère Omar qui a été toujours à mes côtés durant ce mémoire
pour m'aider et m'encourager.



Mes petits frère adorés Youcef & Hamza.
Mes adorables cousines Meriem, Noussaiba, Saliha, Chaimae & sans oublier
ma chouchou Ahlam Mon binôme Soumia ainsi que sa famille
Mes meilleures amies Sonia, Khaoula, Soumia, Noussaiba
Mon chère amie Hiba & sans oublier son frère Raouf & son ami Oussama



Mes amis de la fac
Wassim, Anes, Mohamed & Abd elhamid qui ont été toujours disponibles
pour m'aider
Et enfin à toute ma famille
A tous ceux qui me sont chères.
A tous ceux qui m'aiment.

Nour

DEDICACES

Le parcours d'une vie est jalonné d'opportunités qui dépendent de nous, mais également des personnes qu'il nous a été donné de rencontrer : des personnes qui nous guident, qui nous conseillent et qui nous font confiance, j'ai eu de la chance de rencontrer quelques uns d'entre eux et à qui je tiens à dédier ce mémoire :

Mes chers et adorables Parents : Pour votre amour, Votre affection et Votre sacrifice consentis, Pour votre soutien indéfectible, bienveillance et Vos encouragements, Que dieu bienveillant vous garde à mon côté

Mes chères sœurs : CHERIFA, KHADIDJA ET ZINEB

A mes amies AMINA, NASSIMA, LAITMAS, NESSRINE et SOUMIA.

Sans oublier ma chère binôme NOUR,

A tout mes professeurs pour l'effort et le dévouement pendant tout mon parcours d'études.

A tous ceux que j'aime

Soumiya

Résumé

Le système de médiation doit être capable de gérer l'hétérogénéité pour fournir des vues intégrées homogènes des informations. Il collecte les données et uniformise les langages de requêtes permettant alors à l'utilisateur d'interroger les sources pour obtenir une réponse. La requête est lancée par l'utilisateur pour être décomposée par le médiateur en sous requêtes, selon les sources existantes, puis ces sous requêtes seront transmises aux adaptateurs appropriés pour les traduire, et les transmettre aux sources d'informations locales pour être exécutées. Les résultats de l'exécution des sous requêtes seront composés en un seul résultat final au niveau du médiateur qui sera transmis à l'utilisateur.

Notre projet rentre dans la problématique générale de la médiation qui est le traitement des requêtes complexes dans un environnement hétérogènes, parmi les données hétérogènes on s'intéresse aux: données qui sont représentés par le modèle relationnel, objet relationnel et XML, les métadonnées qui sont représentées par RuleML et RDF et les connaissances par OWL.

Le but de notre étude est de savoir répondre efficacement à des requêtes décisionnelles (requête avec jointure, requête d'agrégation) qui permettes au décideur de prendre des décisions à partir de plusieurs sources de données.

Afin de valider notre travail, nous avons travaillé sur le cas du département formation de la division production de SONATRACH.

Mot clé :

Médiation, hétérogénéité, homogène, connaissance, données, métadonnée, jointure, agrégation, source.

ملخص

يجب أن يكون نظام الوساطة قادر على التعامل مع التجانس لتقديم وجهات معلوماتية متكاملة متجانسة. تجمع وتوحد لغة الاستعلام البيانات و تسمح للمستخدم بالبحث كوسيط للرد. يتم إنشاء الاستعلام من قبل المستخدم و ذلك بكسرها من قبل المنضم في الاستعلامات الفرعية عن طريق المصادر الموجودة، ثم سوف تحال هذه الاستعلامات الفرعية للمحاولات المناسبة لها لترجمتها وإحالتها إلى مصادر محلية من المعلومات ليتم تنفيذها. يتم إعلان عن نتائج تنفيذ الاستعلامات الفرعية إلى نتيجة واحدة و نهائية في المنضم إلى أن تنتقل للمستخدم

يندرج مشروعنا في مشكلة عامة من الوساطة هو علاج الاستعلامات المعقدة في بيئة غير متجانسة، ونحن مهتمون في:

XML, RuleML و RDF, OWL.

الهدف من دراستنا لهذا العمل هو معرفة الاستجابة بفعالية في نظام الوساطة ، للقيام بذلك قمنا بدراسة الاستعلامات المعقدة (الاستعلام مع انضمام الاستعلام بتجميع) التي تساعد اخذ القرار من اتخاذ الحل المناسب من طرف معطيات مختلفة. للتحقق عملنا، قمنا بالبحث في قسم إنتاج إنتاج سوناطراك

الكلمات المفتاحية

الوساطة، غير متجانسة، متجانسة، المعرفة، البيانات، البيانات الفوقية، الانضمام، التجميع، المصدر

Abstract

The mediation system must be able to handle heterogeneity to provide integrated views homogeneous information. It collects and standardizes data query languages while allowing the user to search only the mediator for an answer. The query is generated by the user to be broken by the mediator in sub queries, as existing sources, then these sub queries will be forwarded to the appropriate adapters to translate and transmit them to local sources of information to be executed. The results of the execution of sub queries will be made into one final result at the mediator to be transmitted to the user.

Our project falls into the general problem of heterogeneous data mediation among these heterogeneous data we are interested in: the data that is represented by the relational model (eg MySQL), metadata that are represented by XML or RDF and OWL knowledge.

The aim of our study of this paper is to know to effectively an answer to decision-support queries in a mediation system, incorporating semantic, to do this we treat decision come complex queries (query with join, aggregation query ...). Get the decision maker to make decisions from multiple data sources.

Finally to validate our work, we worked on the case of the formation of the production division of SONATRACH department.

Keywords:

Mediation, heterogeneity, homogeneous, knowledge, data,metadata,join,aggregation, sources.

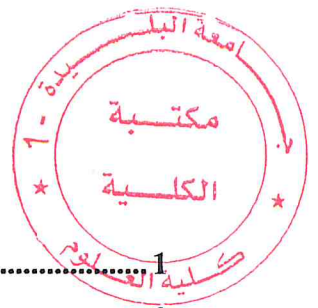
Liste des tableaux

Tableau 1 : Taxonomie de conflits	06
Tableau 2 : Evaluation des approches ontologiques	17
Tableau 3: Etude comparative des différentes approches précédentes.....	36
Tableau 4: description de cycle de vie d'UP.....	45
Tableau 5 : Tableau descriptive des acteurs.....	46
Tableau 6: description de cas d'utilisation (s'authentifier).....	48
Tableau 7: description de cas d'utilisation (lancer la requête globale).....	48
Tableau 8: description des cas d'utilisation (établir le PMTE).....	49
Tableau 09: description de cas d'utilisation (mise à jour du PMTE)	49
Tableau10: description des cas d'utilisation (valider le PMTE).....	50
Tableau 11: description de cas d'utilisation (établir le bilan).....	50
Tableau 12: description de cas d'utilisation (établir le TBRH).....	50
Tableau 13: description de cas d'utilisation (choisir les indicateurs à calculer)	52
Tableau 14: description de cas d'utilisation (consulter le PMTE).....	52
Tableau 15: description de cas d'utilisation (établir le bilan).....	53
Tableau 16: description de cas d'utilisation (établir le TBRH)	54
Tableau 17 : dictionnaire de données.	67
Tableau 18 : liste des méthodes.....	67

Table des figures

Figure 1 : Système d'intégration	07
Figure2 : Intégration par la médiation	10
Figure3 : L'architecture GAV	11
Figure4 : L'architecture LAV	12
Figure 5 : L'architecture GLAV.....	13
Figure 6 : Architecture utilisant une ontologie simple	15
Figure7 : Architecture utilisant plusieurs ontologies.....	16
Figure 8 : Intégration de sources de données par approche hybride.....	16
Figure 9 : Processus de traitement des requêtes.....	21
Figure10 : Architecture générale de MediaGRID.....	23
Figure 11 : Evaluation de requêtes dans MediaGRID.....	26
Figure 12 : Architecture du système SICOM	28
Figure 13 : Architecture d'un système de médiation à base d'ontologie.....	31
Figure 14 : a) formulation d'une requête, b) Traitement d'une requête.....	34
Figure 15 : Système de réécriture de requêtes.....	35
Figure 16 : architecture globale de notre solution.....	42
Figure 17 : Cycle de vie d'UP	44
Figure 18 : diagramme des cas d'utilisation global.	47
Figure 19 : Diagramme de cas d'utilisation pour le choix des indicateurs à calculer.....	51
Figure 20 : diagramme de cas d'utilisation pour la mise à jour du PMTE.....	52
Figure 21 : diagramme de cas d'utilisation pour l'établissement du bilan.....	53
Figure 22 : Diagramme de cas d'utilisation détaillé pour l'établissement du TBRH.	54
Figure 23 : Diagramme de séquence d'authentification	55
Figure 24 : Diagramme de séquence de la suppression des thèmes de formation dans un plan.	56
Figure 25 : Diagramme de séquence de l'ajout des notes et validation du plan.....	57
Figure 26 : Diagramme de séquence de lancement de la requête globale.....	58
Figure 27 :Diagramme de séquence de choix d'indicateurs.....	59

Figure 28 : diagramme d'activité pour le choix d'indicateurs.....	60
Figure 29 : diagramme d'activité pour le lancement de la requête globale.....	61
Figure 30 : Diagramme d'activité pour la traduction de la sous requête métadonnées	62
Figure 31 : Diagramme d'activité pour la traduction de la sous requête de données	63
Figure 32 : diagramme d'activité pour la traduction de la sous requête connaissance.	64
Figure 33 : diagramme de classes de système.	79
Figure 34 : représentation de la table A_Agent.....	79
Figure 35 : représentation de la table A_BESOINSFORM.....	80
Figure 36 : représentation de la table A_CSP.....	80
Figure 37 : représentation de la table A_FORMATEUR.....	80
Figure 38 : représentation de la table A_MODULES.....	80
Figure 39 : représentation de la table A_TBRHFORM.....	80
Figure 40 : représentation de la table C_FORMATIONBILAN	81
Figure 41 : représentation de la table C_FORMATIONPLAN	81
Figure 42 : la fenêtre principale	90
Figure 43 : fenêtre d'exécution de la requête	91
Figure 44 : fenêtre d'établissement de PMTE	92
Figure 45 : fentre d'etablissement du bilan.....	93
Figure 46 : fentre d'etablissement du bilan.....	94
Figure 47 : fenêtre de consultation du PMTE.....	95



Sommaire

1. Contexte	1
2. Problématique	3
3. Objectifs	3

Partie I: Synthèse bibliographique

Chapitre 1: Etat de l'art sur les systèmes de médiation

1. Introduction	4
2. L'intégration	4
2.1 Définition d'un système d'intégration	4
2.2 Caractéristiques des sources de données	4
2.2.1 La Distribution	4
2.2.2 L'Autonomie	5
2.2.3 Hétérogénéité	5
3. Composants des systèmes d'intégration	7
4. Processus d'intégration	8
4.1 La pré-intégration	8
4.2 L'identification des correspondances	8
4.3 L'intégration	8
5. Les approches de mise en œuvre du processus d'intégration	8
5.1 L'approche virtuelle	8
5.2 L'approche matérialisée	9
6. Le système de Médiation	9
6.1 Niveaux d'un système de médiation	9
6.2 Les composants d'un système de médiation	10
6.2.1 Schéma global	11
6.2.2 Le mapping de schéma global avec des sources	13
6.2.3 Les fonctions de transformation	13
6.3 Algorithme de réécriture des requêtes	13
6.4 La composition des résultats	14
7. Rôle des ontologies dans un système des médiations	14
7.1 Définition	14
7.2 Rôle des ontologies	14

7.3	Approche d'une ontologie simple	15
7.4	Approche d'une ontologie multiple	15
7.5	Approche hybride	16
8.	Conclusion	18

Chapitre 2:Etat de l'art sur le traitement des requêtes complexes

1.	Introduction	19
2.	Types de requêtes complexes	19
2.1	Requêtes d'intervalle	19
2.2	Requêtes meilleur- k	19
2.3	Requêtes d'agrégation	19
2.4	Requêtes de jointure	20
2.5	Requêtes imbriquée	20
3.	Evaluation des requêtes	20
4.	Processus de traitement des requêtes de médiation	20
5.	Les projets de recherche existants	22
5.1	Le projet MEDIAGRID	22
5.1.1	Description	22
5.1.2	Objectif	22
5.1.3	Architecture	23
5.1.4	Gestion des métadonnées	24
5.1.5	Génération de requêtes de médiation	24
5.1.6	Interrogation des données XML	25
5.1.7	Evaluation de requêtes	26
5.2	Le projet SICOM	27
5.2.1	Les différentes étapes de l'approche SICOM	28
5.2.2	Les ontologies	29
5.2.3	Traitement des requêtes avec SICOM	30
5.3	L'approche de MAIZ	31
5.3.1	Description	31
5.3.2	L'architecteur de système :	32
5.3.3	Traitement des requêtes	33
5.3.4	Langage de requêtes	33
5.3.5	Réécriture de requête	35

6. Etude comparative des différentes approches précédentes.....	36
7. Conclusion	38

Partie II: Analyse, conception et implémentation

chapitre 3: Analyse et conception

1. Introduction.....	39
2. Terrain d'étude	39
3. Procédure de travail de département formation (SONATRACH)	39
4. Notre solution proposée.....	41
5. Architecture globale de la solution proposée.....	42
6. Présentation de la démarche utilisée	43
6.1 Le Processus Unifié UP	43
6.2 Le langage UML	45
7. Expression des besoins.....	46
7.1 Identification des acteurs	46
7.2 Identification des cas d'utilisation.....	47
8. Analyse.....	54
8.1 Diagramme de séquence.....	55
8.2 Diagramme d'activité	60
9. Conception	64
9.1 diagramme de classe	65
9.2 Architecture détaillée de la solution proposée.....	68
9.3 Description des modules de l'architecture proposée	69
10. Conclusion	78

Chapitre 4: Implémentation et test

1. Introduction.....	79
2. Présentation des sources.....	79
2.1.1 Source relationnel.....	79
2.1.2 Source objet relationnelle	81
2.1.3 Les sources XML	81
2.2 Les sources de métadonnées	84
2.2.1 La source RDF	84

2.2.2	La source RuleML.....	85
2.3	Les sources de connaissance	86
2.3.1	La source OWL	86
3.	Outil et langage de programmation utilisé	87
3.1	Les outils	87
3.2	Langage de programmation	89
4.	Presentation de OMediator	90
4.1	Authentification	90
4.2	La fenêtre principale	91
4.3	Fenêtre d'exécution de la requête.....	91
4.4	Fenêtre d'établissement de PMTE	92
4.5	fenetre d'etablissement de bilan	93
4.6	fenetre de consultation de PMTE.....	94
5.	Conclusion	95
1.	Conclusion générale	96
2.	Perspective	97
ANNEXE A	98
ANNEXE B	104
ANNEXE C	111
ANNEXE D	113

Partie 1

Synthèse bibliographique

1. Contexte

Un système d'information décisionnel ou SID est avant tout un moyen qui a pour but de faciliter la définition et la mise en œuvre de stratégies gagnantes. Il ne s'agit pas de définir une stratégie une fois pour toute, mais de s'adapter continuellement à son environnement, et de le faire plus vite que ses concurrents.

Le système d'information décisionnel traditionnel permet de faire l'analyse des activités déjà réalisées et d'en tirer des enseignements pour les activités futures. Il a pour objectif de fournir des indicateurs sur la vie de l'entreprise. Pour cela, il exploite une richesse déjà mise à sa disposition. Cette richesse est constituée d'un ensemble d'informations qu'une entreprise génère dans le cadre de son activité. Il utilise des données plus ou moins récentes au mieux mises à jour quotidiennement. Le système d'information décisionnel plus avancé gère des données plus fraîches, certaines sont mises à jours en quasi temps réel, automatise des décisions et supporte en temps réel des opérations (un centre d'appels par exemple).

Le nombre croissant de données accessible via des réseaux (intranet, internet ,etc) pose le problème de l'intégration de source d'information préexistantes, souvent distante et hétérogène , afin de faciliter leurs interrogation par un large public, une des première approches proposées en intégration d'informations prône la construction des médiateurs , un médiateur joue le rôle d'interface de requête entre l'utilisateur et des sources de données, il donne à l'utilisateur l'illusion d'interrogés un système homogène et centralisé en lui évitant d'avoir trouver les sources de données pertinentes pour sa requête , de les interroger une à une, et de combiner lui-même les informations obtenues.

Dans une entreprises nous constatons des données dispersées dans une grande variété de sources hétérogènes, nous trouvons des données qui sont *internes* à l'entreprise (protégées) et d'autre qui sont *externes*(chez des fournisseurs, des partenaires ou des clients), pour cela l'intégration de ses données qui sont de différentes types(hétérogènes) comme le modèle relationnel MYSQL, les métadonnée qui sont représenté par : XML,RDF,OWL) elle offre un accès simple,rapide et efficace aux informations désirées.

L'intégration d'informations est un domaine de recherche et d'actualité, elle propose plusieurs approches pour construire un système d'interrogation des informations distribuées et hétérogènes. Pratiquement ces approches permettent d'accéder à travers un

Introduction générale

schéma global unifié à plusieurs sources d'information hétérogènes, avec chacune son propre schéma local. Parmi ces approches, nous distinguons les entrepôts de données, les systèmes de bases de données fédérées, ou encore les systèmes de médiation.

Notre étude s'inscrit principalement dans le contexte des systèmes de médiation qui sont capables de gérer l'hétérogénéité pour fournir des vues intégrées homogènes des informations dispersées.

La médiation repose sur un composant essentiel, appelé médiateur, qui est chargé de répondre à des besoins à partir de connaissances mises à sa disposition.

Le médiateur permet de localiser l'information et de résoudre les conflits schématiques et sémantiques. Un composant secondaire, appelé wrapper(ou adaptateur), sert d'interface avec les SI. Le wrapper résout les conflits syntaxiques en présentant les données dans le modèle de médiation.

La validation de notre travail nécessite des sources d'informations réelles, de données, de métadonnées et de connaissances. C'est pourquoi nous nous sommes dirigées vers l'entreprise SONATRACH. Nous avons effectué un stage pratique au sein de cette entreprise et précisément dans le département formation. Le stage a permis d'avoir une vue sur ce qui se passe réellement dans le département et comprendre les procédures de travail ; pour ensuite analyser les données dont nous avons besoin pour valider notre solution.

2. Problématique

Plusieurs problèmes des conceptions émergent lors de l'utilisation des médiateurs. Les problèmes les plus importants dans la médiation de données ou les travaux de recherche ont été focalisés ces dernières années sont des suivants :

1. Comment intégrer les différents schémas de sources ?
2. Comment répondre efficacement aux requêtes posées en termes de schéma global ?

Ce projet s'intéresse par le deuxième problème, donc la question principale est de savoir comment répondre efficacement à des requêtes décisionnelles, dans un système de médiation, en intégrant leur sémantique.

Il s'agit donc, de :

- ✓ Comment concevoir des requêtes décisionnelles, dans un système de médiation ?
- ✓ Comment produire un ensemble de sous requêtes possible à partir d'un ensemble de sources de données distribuées et hétérogènes.

3. Objectifs

✓ Un outil doit être développé, permettant de traiter, automatiquement, des requêtes décisionnelles.

✓ Notre but est, dans un premier temps est de concevoir les requêtes complexes, dont le but est de les utiliser par des décideurs, puis, dans un deuxième temps de traiter ces requêtes, c'est-à-dire à partir d'une requête décisionnelle globale (écrite en terme du schéma global), comment la décomposer en sous requêtes écrites en modèle des sources. Et cela effectuée d'une manière automatique avec la prise en compte de la sémantique.

1. Introduction

L'augmentation exponentielle des échanges d'informations a mis en exergue les problèmes d'hétérogénéité de représentation et de stockage de l'information.

Dans ce cadre, nous présentons dans ce chapitre qu'est-ce c'est un système d'intégration, ses différentes approches ou nous allons faire une petite comparaison entre un entrepôt de données et la médiation et par la suite nous approfondir dans les systèmes de médiation comme approche d'intégration après avoir présenté rapidement les principales phases du processus d'intégration.

2. L'intégration

2.1 Définition d'un système d'intégration

L'intégration de données a pour but de combiner les données réparties dans différentes sources et de fournir à l'utilisateur une vue unifiée de ces données. Et Réalisée en maintenant les données dans les sources de données (approche virtuelle, ou médiateur) ou en les stockant dans une nouvelle base de données (approche matérialisée).

Un système d'intégration de données doit prendre en compte des différentes formes d'hétérogénéités [GUE10].

2.2 Caractéristiques des sources de données

Les caractéristiques qui rendent l'intégration de données difficile sont :

2.2.1 La Distribution

Dans plusieurs systèmes d'information les données sont réparties dans des sources Différente s'elles-mêmes également réparties dans des machines géographiquement distribuées. Les données peuvent être distribuées au sens relationnel ou bien dupliquées sans que les réplicas soient structurés de manière homogène. Les bénéfices de la distribution sont la disponibilité, la fiabilité et l'amélioration du temps d'accès [SOL02].

Chapitre 1: Etat de l'art sur les systèmes de médiation

2.2.2 L'Autonomie

Les bases de données sont souvent gérées par de systèmes autonomes. Normalement Il est accepté de partager les données mais pas le contrôle. Pour Cela il est important de comprendre l'autonomie et la manière de la gérer lorsqu'une base de données partage ses données avec d'autres bases. Il y a différents niveaux d'autonomie [VOD].

✓ **Conception**

Les bases de données locales ont leurs propres modèles de données, langage d'interrogation, interprétation sémantique des données, contraintes, fonctions, Etc... [VOD].

✓ **Communication**

Les bases de données locales décident quand et comment répondre aux requêtes [VOD].

✓ **Exécution**

Les bases de données locales n'ont pas besoin d'informer D'autres systèmes de l'ordre d'exécution des transactions locales ou des Opérations externes [VOD].

2.2.3 Hétérogénéité

L'hétérogénéité est indépendante de la distribution physique des données. Un système d'information est homogène si le logiciel qui gère les données Est le même sur tous les sites, les données ont le même format et structure (Modèle de données) et appartiennent à un même univers de discours [SOL02].

Un système hétérogène est celui qui n'adhère pas à toutes les caractéristiques d'un système homogène, c'est-à-dire, le système d'information utilise des Langages de programmation et d'interrogation, des modèles, des SGBD différents [SOL02].

Il existe 2 types d'hétérogénéité, syntaxique et sémantique :

✓ **L'hétérogénéité syntaxique**

Se retrouve dans les formats de stockage des données (XML, relationnel, objet, etc.), dans les langages d'interrogation (XQuery, SQL, OQL, etc.), dans les protocoles d'accès (HTTP, etc.), dans les interfaces, etc [BEN06].

Chapitre 1: Etat de l'art sur les systèmes de médiation

✓ L'hétérogénéité sémantique

Nous parlons d'hétérogénéité sémantique dans le cadre d'interprétation du monde réel, Ceci a généralement lieu durant le processus de conception des systèmes d'information. Les conflits sémantiques peuvent survenir au niveau des schémas et des données [BEN06]. Goh a identifié quatre principaux types de conflit : conflits de nom, conflits de graduation, conflits de confusion et conflits de représentation. Nous les commentons dans ce qui suit **tableau 1**.

Conflits	Description
Conflits de nommage	les différents schémas utilisent des noms différents pour représenter le même concept
Conflits de graduation	les concepts semblent avoir la même signification dans deux schémas mais sont différents à cause de leur contexte
Conflits de confusion	liés à la manière de coder la valeur d'un concept du monde réel dans un système de mesure
Conflits de représentation	deux schémas sources décrivent le même concept de manière différente

Tableau 1 : Taxonomie de conflits [GOH 99]

- **Les conflits de nommage** se produisent lors de l'attribution des noms dans des Schémas qui diffèrent de manière significative. Les cas les plus fréquents sont les cas de présence de synonymes et d'homonymes [BOU08].

- **Les conflits de graduation** apparaissent lorsque différents systèmes de Graduation sont utilisés pour mesurer une valeur. On peut citer en exemple la mesure de la température [BOU08].

Exemples : - Degrés Celsius ou Fahrenheit.

- Dollar \$ ou l'Euro €.

- **Les conflits de confusion**

Se produisent lorsque les concepts paraissent avoir la même signification mais diffèrent en réalité. Ce type de confusion peut être cause par des contextes temporels différents. Par exemple le poids d'une personne dépend de la date ou elle se pèse [BOU08].

- **Les conflits de représentation** se produisent quand deux schémas sources décrivent le même concept de manière différente. Par exemple, dans une source l'adresse peut être désignée par une chaîne de caractères tandis que dans une autre, l'adresse est une structure composée du numéro et du nom de la rue, du code postal et de la ville [BOU08].

3. Composants des systèmes d'intégration

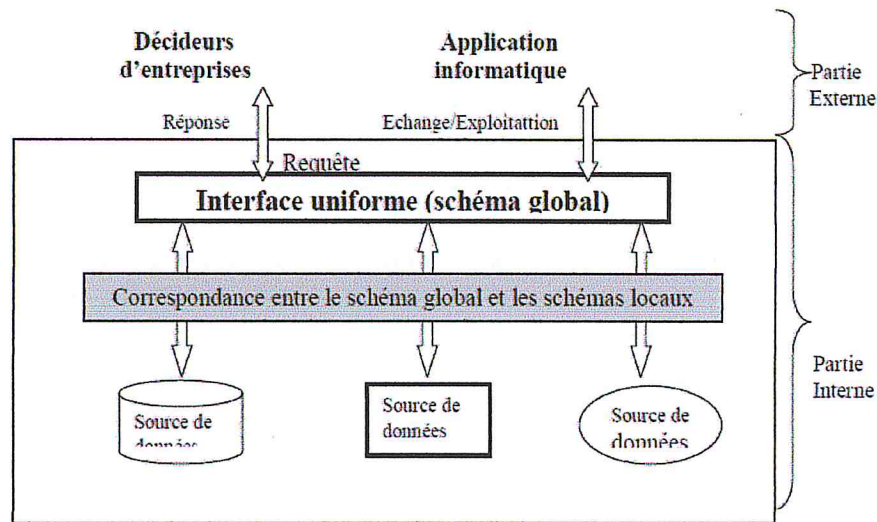


Figure 1 : Système d'intégration [LAH11].

Un système d'intégration se compose de deux parties :

- **La partie externe** Correspond aux utilisateurs du système intégré comme, par exemple, les décideurs d'une entreprise, ou d'autres systèmes [LAH11].

- **La partie interne** Comprend d'une part les sources d'informations participant dans le processus d'intégration et d'autre part une interface (schéma global) permettant aux éléments de la partie externe d'accéder d'une manière transparente aux sources de données. Cette interface est une couche sans données propres (*approche médiateur*) Les éléments externes au système ne voient pas les sources internes. Celles-ci sont

encapsulées, cachées par l'interface. Les éléments externes doivent pouvoir agir sur le système d'intégration d'une manière transparente via un *schéma global* [LAH11].

4. Processus d'intégration

Le processus d'intégration est décomposé en trois phases distinctes :

4.1 La pré-intégration

Cette phase vise à préparer l'intégration des schémas en les rendant plus homogènes. Elle consiste principalement à traduire les schémas initiaux

Dans un modèle de données commun (réduction de l'hétérogénéité syntaxique). Elle s'attache également à enrichir leur sémantique.

4.2 L'identification des correspondances

Durant cette phase, les correspondances entre les éléments des schémas source sont détectées et formalisées de même que les différents conflits.

4.3 L'intégration

Cette phase finale produit le schéma intègre et fournit les règles de traduction permettant de passer des schémas source au schéma intégré et inversement «mapping» [BOU8].

5. Les approches de mise en œuvre du processus d'intégration

Les données intégrées peuvent être matérialisées (elles sont regroupées dans une base unique avec une duplication de données), ou virtuelles (elles restent dans les sources d'origines) [BOU8].

5.1 L'approche virtuelle

Cette approche elle contient trois architectures qui sont :

✓ Les systèmes multi-bases

Les systèmes Multi bases sont des systèmes Qu'ils n'offrent pas une vision unifiée des données. Il n'existe pas de schéma global permettant un accès transparent aux différentes sources de données. La coopération est seulement assurée par l'intermédiaire d'un langage commun : le langage multi base de type SQL notamment.

Chapitre 1: Etat de l'art sur les systèmes de médiation

✓ Les systèmes fédérés

A l'inverse des systèmes Multi bases, les systèmes fédérés se caractérisent par l'existence d'un schéma unifié appelé schéma fédéré qui constitue l'interface d'accès au système intégré. L'intégration se situe au niveau des schémas.

✓ Les systèmes de médiation

L'approche d'intégration par médiation constitue sans doute aujourd'hui la solution la plus courante pour relier différentes sources qui cette fois, ne correspondent pas nécessairement à des bases de données seulement.

Afin de bien comprendre leurs fonctionnalités et déterminer leurs domaines d'application, nous allons en parler avec plus de détail dans les prochains titres.

5.2 L'approche matérialisée

Cette architecture consiste à centraliser physiquement, dans un *entrepôt de données*, l'ensemble de données consolidées à partir de diverses sources (catalogues électroniques, bases de données relationnelles, Web, etc.) [BOU8].

6. Le système de Médiation

Les systèmes de médiations sont des interfaces permettant un accès transparent aux données.

6.1 Niveaux d'un système de médiation

Dans un système de médiation, on trouve quatre principaux niveaux: l'utilisateur, le médiateur, les adaptateurs et les sources [BEN06].

- **L'utilisateur** il accède aux données d'une façon transparente sans se soucier de leurs hétérogénéités.
- **Le médiateur**

Il joue le rôle d'interface entre l'utilisateur et les sources de données en donnant l'impression à celui-ci d'interroger une source centralisée et homogène [DJE07]. Le

Chapitre 1: Etat de l'art sur les systèmes de médiation

médiateur transforme la requête dans un langage pivot et la transmet à différents adaptateurs puis rassemble ou regroupe les résultats [CHA09].

- **L'adaptateur (ou WRAPPER en anglais)**

Il a pour rôle de lancer la requête ou une partie de celle-ci aux différentes sources, puis de récupérer le résultat de la requête et le traduire selon le langage pivot ensuite le transmettre au médiateur [CHA09].

- **Les sources** Ce sont les sources de données distribuées et hétérogènes [HAC13].

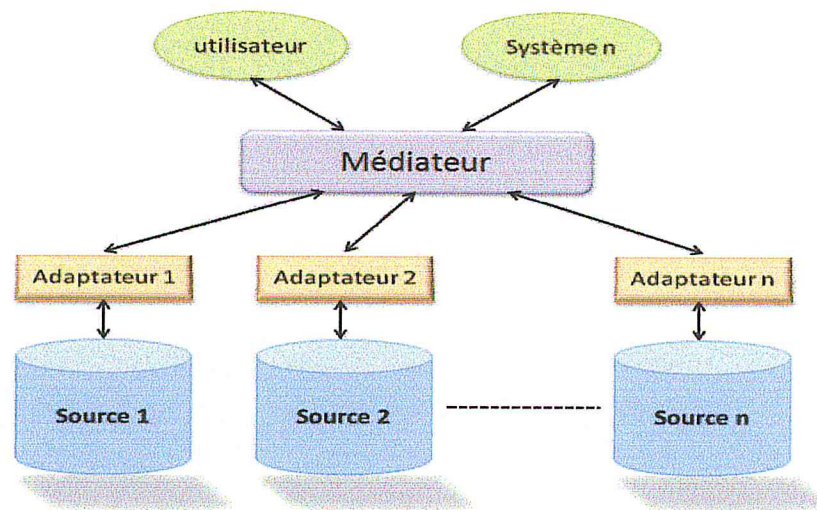


Figure2 : Intégration par la médiation [DJE07].

6.2 Les composants d'un système de médiation

Nous distinguons un ensemble de définitions des composants les plus courants. En effet, ces différents composants se retrouvent dans la plupart des solutions de médiation actuelles.

6.2.1 Schéma global

Il existe plusieurs approches pour définir ce schéma : Global as View et Local as View, Ces approches sont déterminantes pour la réécriture des requêtes, et pour l'évolution du système d'intégration (ajout de sources) et la combinaison des deux approches.

✓ **L'approche GAV**

L'approche GAV a été la première à être proposée pour intégrer des informations. Elle consiste à définir le schéma global en fonction des schémas des sources de données à intégrer puis à le connecter aux différentes sources.

Pour cela, les prédicats du schéma global, aussi appelés relations globales, sont définis comme des vues sur les prédicats des schémas des sources à intégrer [LAH11].

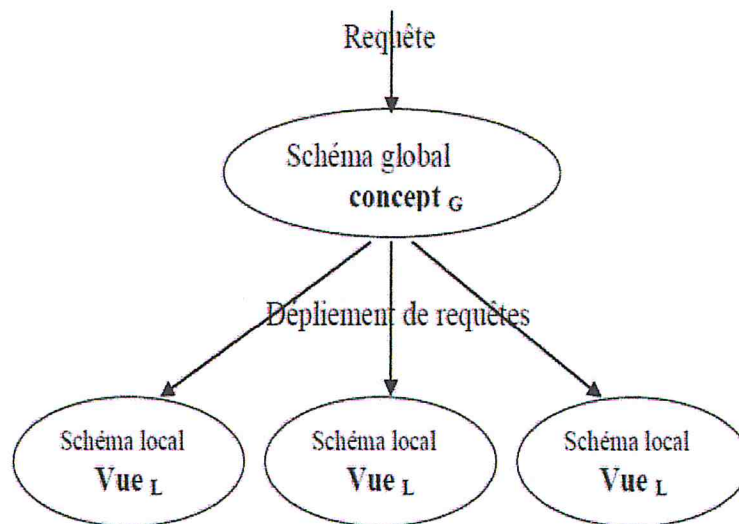


Figure3: L'architecture GAV [LAH11].

✓ **L'approche LAV**

Contrairement à l'approche GAV, LAV suppose l'existence d'un schéma global et consiste à définir les schémas des sources de données à intégrer comme des vues du schéma global. Ces vues définissent comment les informations mappent sur le schéma global en exprimant un mapping entre une relation dans le schéma local en une (un ensemble de) relation(s) dans le schéma global. Le principal avantage de l'approche LAV sur l'approche GAV est qu'il n'y a pas de dépendance sur le schéma global. Dans LAV, chaque schéma de source est mappé sur le schéma global [LAH11].

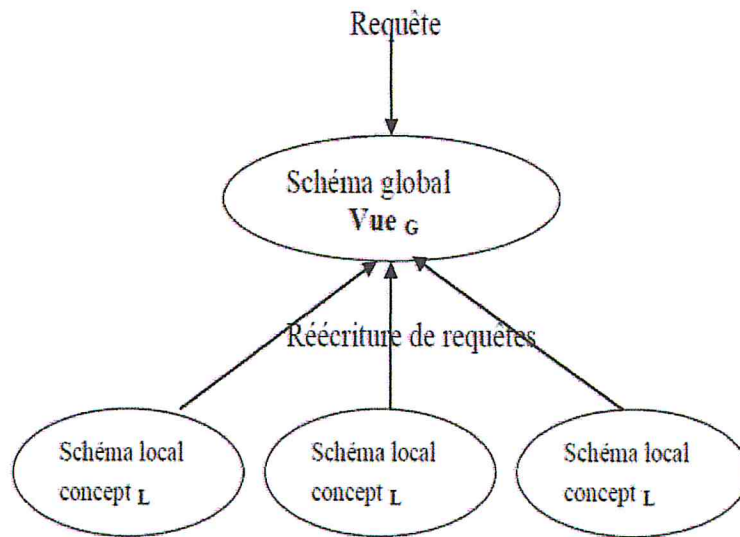


Figure4: L'architecture LAV [LAH11].

✓ **L'approche mixte GLAV (Generalized Local as View)**

Dans GLAV (Generalized-Local-as-View) (figure 4), nous disposons de vues au niveau global et local. Une correspondance entre les vues au niveau global et local est requise. Ces correspondances n'ont pas de direction et s'appliquent dans les deux sens puisque les concepts dans l'ontologie globale sont considérés comme des vues. En fait, l'approche hybride permet d'avoir la correspondance entre les ontologies locales via le vocabulaire partagé. Donc la correspondance peut être calculée via l'ontologie globale. Dans l'approche GLAV, l'indépendance du schéma global (permet l'ajout et la suppression de sources), l'adaptation de nouvelles sources et la complexité pour reformuler des requêtes sont les mêmes que dans l'approche LAV [LAH11].

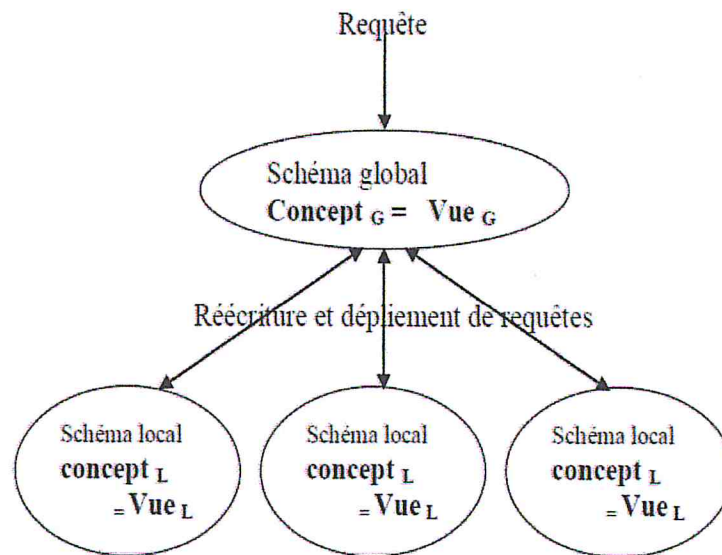


Figure 5 : L'architecture GLAV [LAH11].

6.2.2 Le mapping de schéma global avec des sources

Les mappings du schéma global avec les sources sont des requêtes, appelées requêtes de médiation [BOU8]. C'est explicitement la correspondance schéma global/schémas locaux.

6.2.3 Les fonctions de transformation

Le langage de requête du schéma global n'est pas toujours le même que celui des cibles. Les requêtes globales doivent être traduites dans le langage des sources (ex : Xquery -> SQL)

6.3 Algorithme de réécriture des requêtes

La réécriture des requêtes en termes des schémas locaux doit prendre en compte l'hétérogénéité structurelle et sémantique des schémas. Les requêtes et les sources sont écrites sous forme de requêtes conjonctives. A priori, le nombre de réécritures possibles de la requête en fonction des vues est exponentiel par rapport à la taille de la requête.

Chapitre 1: Etat de l'art sur les systèmes de médiation

Le principe de cet algorithme est de réduire le nombre de possibilités en considérant les sous-requêtes (buts partiels) et en déterminant quelles vues sont pertinentes pour répondre à ces sous requêtes, en trouve par exemple plusieurs algorithmes comme : Bucket, inverse-rules, MiniCon.

6.4 La composition des résultats

Après l'exécution de chaque requête sur la source adéquate par l'adaptateur, les résultats partiels sont transmis au médiateur. Celui-ci procède à composer et combiner les différents résultats issus des différentes sources interrogées en un seul résultat homogène, en vue de les transmettre à l'utilisateur.

7. Rôle des ontologies dans un système des médiations

7.1 Définition

« Ontologie » est un terme emprunté à la philosophie qui implique une branche de la philosophie qui traite la nature et l'organisation de la réalité. Plusieurs définitions d'ontologies sont données dans la dernière décade, mais celle qui caractérise l'essentiel d'une ontologie est fondée sur la définition reliée dans: « Une ontologie est une spécification formelle, et explicite d'une conceptualisation partagée ». Conceptualisation réfère à un modèle abstrait de certains phénomènes dans le monde qui identifie les concepts appropriés de ce phénomène. Explicite signifie que le type de concepts utilisés et les contraintes sur leur utilisation doivent être explicitement définis. Formelle réfère au fait qu'une ontologie doit être compréhensible par la machine, c'est-à-dire cette dernière soit capable d'interpréter la sémantique de l'information fournie. Partagée indique que l'ontologie supporte la connaissance consensuelle, et elle n'est pas restreinte à certains individus mais accepté par un groupe [ABD04].

7.2 Rôle des ontologies

Les ontologies ont initialement été introduites comme une « spécification explicite d'une conceptualisation ». Elles peuvent être utilisées dans le processus d'intégration de données pour décrire la sémantique des sources d'information et rendre ainsi leur contenu explicite. Elles permettent alors l'identification et l'association de concepts sémantiquement correspondants. Plusieurs approches ont été étudiées [ABD04]. Elles sont fondées sur l'utilisation soit d'une seule ontologie, soit de plusieurs nous trouverons :

7.3 Approche d'une ontologie simple

Cette approche utilise une ontologie globale fournissant un vocabulaire partagé pour la spécification de la sémantique. Toutes les sources d'informations sont reliées à une ontologie globale. Une approche importante pour ce type d'intégration des ontologies est .Un modèle indépendant de chaque source d'information doit être décrit pour ce système en reliant les objets de chaque source au modèle du domaine global [ABD04].

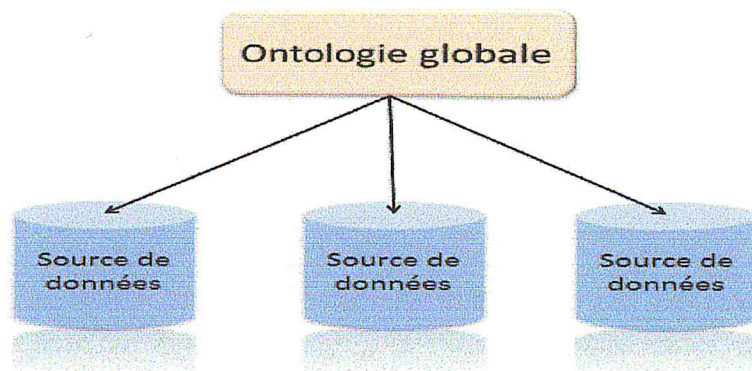


Figure 6 : Architecture utilisant une ontologie simple [WAC].

7.4 Approche d'une ontologie multiple

Chaque source d'information est décrite par sa propre ontologie. L'avantage est que chaque ontologie source peut être définie sans prendre en considération les autres sources ou les autres ontologies. Mais le manque d'un vocabulaire commun conduit à une difficulté extrême pour comparer différentes ontologies sources. Pour surmonter ce problème, un formalisme de représentation supplémentaire définissant le mapping inter-ontologie est fourni. Ce dernier identifie sémantiquement les termes correspondants de différentes ontologies sources. Ce mapping est difficile à définir à cause de divers problèmes d'hétérogénéité sémantique qui peuvent se produire [ABD04].

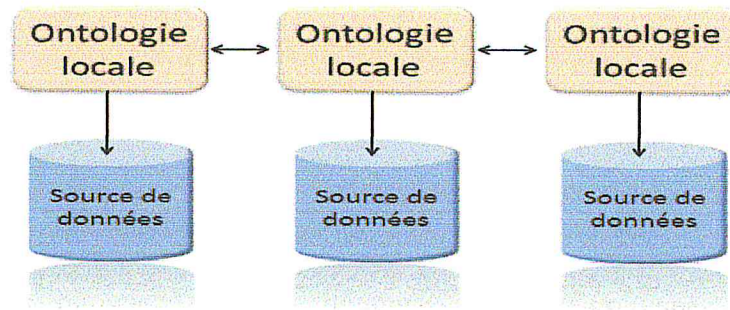


Figure7 : Architecture utilisant plusieurs ontologies [WAC].

7.5 Approche hybride

Dans cette dernière, chaque source possède sa propre ontologie de description mais, afin de permettre de rendre chacune comparable avec une autre, elles se réfèrent toutes à un vocabulaire partagé. Ce vocabulaire comprend des termes fondamentaux d'un domaine qui peuvent être combinés avec ceux présents dans les ontologies locales pour décrire des sémantiques plus complexes. Parfois le vocabulaire partagé est également une ontologie. Le principal avantage de l'approche hybride, illustrée par la Figure 1, est de permettre l'ajout de nouvelles sources aisément, sans besoin de modification. Il faut néanmoins souligner que le principal inconvénient de cette approche est de ne pas permettre une réutilisation simple d'ontologies déjà existant [BER].

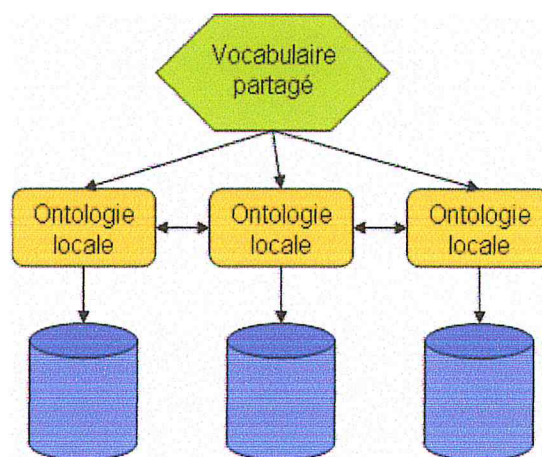


Figure 8 : Intégration de sources de données par approche hybride [BER].

Chapitre 1: Etat de l'art sur les systèmes de médiation

Le tableau suivant récapitule les principales caractéristiques de chacune des trois approches citées précédemment.

	Approche mono-ontologie	Approche multi-ontologies	Approche hybride
Effort d'implémentation	Simple	Élevé	Raisonnable
Hétérogénéité sémantique	Sémantique unifiée d'un domaine	Hétérogénéité sémantique	Hétérogénéité sémantique
Impact lors de l'ajout/mise à jour d'applications	Besoin d'adaptation de l'ontologie globale	Ajouter une autre ontologie locale, mettre à jour l'ontologie locale associée à l'application mise à jour. Liaison avec les autres ontologies locales.	Ajouter une autre ontologie locale, mettre à jour l'ontologie locale associée à l'application mise à jour.
Comparaison d'ontologies	pas de comparaison d'ontologies (car il n'existe qu'une seule ontologie)	Difficile du fait qu'il n'existe pas d'ontologie partagée	Simple du fait qu'il existe une ontologie partagée

Tableau 2 : Evaluation des approches ontologiques [IZZ06].

Il découle de l'analyse de ce tableau que les approches à ontologies multiples et les approches hybrides sont confrontées au problème d'hétérogénéité, du fait qu'elles comportent plus d'une ontologie. Dans ce cas, il devient nécessaire de faire inter opérer plusieurs ontologies.

L'approche hybride peut constituer une approche pertinente pour l'intégration des informations, dans la mesure où elle peut fournir à la fois de la flexibilité et une simplicité de mise en œuvre spécialement la mise à jour.

Les ontologies permettent non seulement de décrire le contenu des sources de données mais offre également la possibilité d'être utilisées comme modèle global d'interrogation dans une architecture distribuée. En effet, le médiateur doit posséder la connaissance du contenu de chaque source pour déterminer de quelle manière il doit effectuer les sous-requêtes. En utilisant les technologies développées pour le « Web sémantique »,

Chapitre 1: Etat de l'art sur les systèmes de médiation

notamment OWL (ou RDF) et son langage de requête associé, SPARQL, il est possible d'indiquer au médiateur la connaissance du contenu de chaque source au moyen d'expressions de chemin associées à leurs sources de rattachement [ABD04].

8. Conclusion

L'intégration des données par médiation est une solution pour les entreprises ayant des sources de données hétérogènes. L'utilisation des ontologies dans la médiation permet de gérer le problème d'hétérogénéité structurelle et sémantique. Une architecture prenant en charge les différentes ontologies consiste à créer des ontologies locales pour chacune des sources et une ontologie globale pour le médiateur.

1. Introduction

Le système de médiation est fondé sur la définition d'un schéma global qui fournit un vocabulaire unique pour l'expression des requêtes des utilisateurs et pour la description des contenus des sources par un ensemble de vues. Il offre à l'utilisateur une vue uniforme des sources de données qu'il exploite et permet de les interroger d'une manière transparente sans que l'utilisateur n'ait souci de la distribution des sources ni de leur format d'origine "hétérogénéité"[BOU11].

Dans ce chapitre nous allons voir comment ça se fait l'évaluation des requêtes, et intéresser spécialement au processus de traitement des requêtes, puis nous allons présenter quelques approches existantes dans ce cadre. Nous allons faire par la fin une étude comparative entre ces approches.

2. Types de requêtes complexes

Avant d'entres dans le détail de traitement de requêtes complexes, il est mieux de citer les types de requêtes complexes;

2.1 Requêtes d'intervalle

Ce sont les requêtes par lesquelles l'utilisateur demande des données ayant des valeurs appartenant à un intervalle précis.

2.2 Requêtes meilleur- k

Dans ce type de requête, l'utilisateur peut être satisfait des meilleur-k réponses trouvées dans le système. Ce type de requêtes attire beaucoup D'attention dans plusieurs domaines de l'informatique comme la recherche d'informations, Monitoring de systèmes et de réseaux, bases de données multimédias, analyse de données spatiales.

2.3 Requêtes d'agrégation

Ce sont des requêtes contenant un opérateur d'agrégation Comme la requête suivante : **SELECT OP_AGR(COL) FROM R WHERE CONDITON-SELECTION ;**

Op-Agr est un opérateur d'agrégation comme : **SUM, COUNT, AVG** etc. Col est un Attribut numérique de la relation R ou une expression contenant plusieurs attributs. La condition-sélection décide quels sont les tuples à être présentes dans l'agrégation.

2.4 Requêtes de jointure

Ce sont des requêtes multi-relations, multi-attributs dans Les quelles il y a un ou plusieurs opérateurs de jointure.

2.5 Requêtes imbriquée

Une sous-requête (aussi appelé « requête imbriquée » ou « requête en cascade ») consiste à exécuter une requête à l'intérieur d'une autre requête. Une requête imbriquée est souvent utilisée au sein d'une clause WHERE ou de HAVING pour remplacer une ou plusieurs constante.

3. Evaluation des requêtes

Pour l'exécution d'une requête, une application lance une requête au médiateur et attend la réponse. Le médiateur reçoit la requête, il l'analyse syntaxiquement et sémantiquement puis la décompose en sous-requêtes. Chaque sous-requête correspond à une source de données et elle utilise seulement les fonctionnalités de la source de données. Le médiateur coordonne l'exécution de la requête : il envoie chaque sous-requête à l'adaptateur qui contrôle la source de données. L'adaptateur la traduit en une requête correspondante dans le langage local de la requête et envoie cette requête traduite à la source de données. La source de données calcule une réponse et la transmet à l'adaptateur qui traduit la réponse dans le format exigé par le médiateur. Le médiateur coordonne les réponses qui arrivent et les combinent en une réponse finale. Pour supporter l'exécution de la requête, le médiateur doit faire face aux différentes fonctionnalités de chaque adaptateur correspondant à chaque source de données [LAH11].

4. Processus de traitement des requêtes de médiation

L'utilisateur pose des requêtes sur le schéma global ou virtuel, qui fournit un vocabulaire unique, et non sur les schémas locaux. Pour pouvoir répondre à ces requêtes et parce que les données sont distribuées dans les sources, le système utilise des WRAPPERS (adaptateurs) pour effectuer la réécriture ou la traduction de la requête utilisateur exprimée dans les termes du vocabulaire du schéma global en des requêtes exprimées selon le vocabulaire des sources locales. Ensuite, les adaptateurs traduisent les réponses aux requêtes sur les sources en des réponses compatibles avec le schéma global du médiateur. Cette traduction se base sur la définition des mappings entre le schéma global et les schémas locaux. On doit s'assurer que les réponses obtenues par la réécriture seront correctes et complètes pour la requête, autrement dit, le résultat de la réécriture de la

Chapitre II : Etat de l'art sur le traitement des requêtes complexes

requête doit contenir toutes les réponses possibles extraites depuis les sources de données. De plus, les sources qui ne participent pas à la requête seront exclues de la réécriture [AHM09].

La **Figure9** représente le processus de traitement des requêtes des utilisateurs. Ce processus est constitué de deux phases [HAD10] :

Phase 1 : le médiateur envoie la requête du demandeur vers les ontologies locales identifiées et de bénéficier le temps de recherche qu'il exploiter les résultats d'un processus de mapping que ce dernier réalise la correspondance entre les différents sources de données et de résoudre le problème de la complexité des structures de données (ontologies de contextes) alors d'augmenter le temps de réponse de la requête demandeur.

Phase 2 : dans le cas où la réponse dans la phase précédente non octroyée alors le médiateur revoie la requête vers l'ontologie globale que cette dernière garantie la réponse demandeur par ce que cette dernière est résultante par les deux processus de mapping et de fusion d'ontologies qu'il implique de fournir une correspondance entre les différents sources de données et assurer le partage de données afin de créer un vocabulaire commun alors d'annuler le problème de perte d'informations.

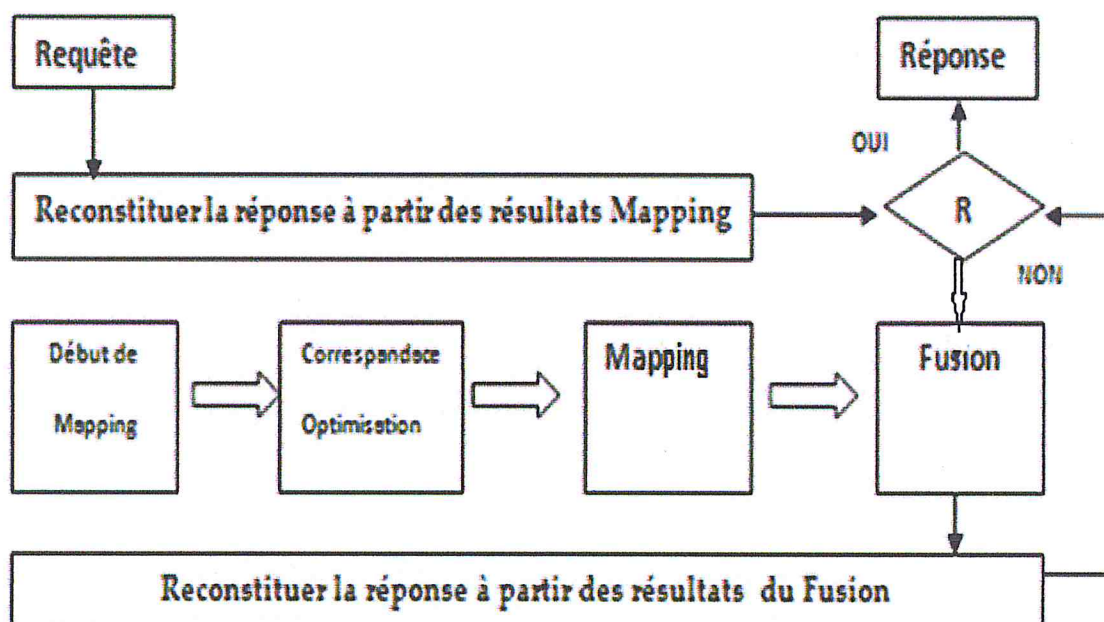


Figure 9 : Processus de traitement des requêtes [HAD10].

5. Les projets de recherche existants

Les recherches dans le domaine des systèmes d'intégration ont été très actives et ont conduit à l'élaboration de prototypes qui, principalement, ont mis l'accent sur les aspects inhérents aux modèles et langages de requêtes, aux algorithmes de traitement de requêtes, à la définition du schéma global et aux problèmes de translation sémantique entre les schémas [LAH11].

Très peu d'approches se sont intéressées au traitement de requêtes et à la prise en compte des conflits liés à l'hétérogénéité des sources durant le processus de traitement. Cette section présente ces approches:

5.1 Le projet MEDIAGRID

5.1.1 Description

Le projet MediaGRID se propose de construire une infrastructure de médiation pour l'accès transparent aux données. Une telle infrastructure offre aux applications des facilités d'intégration, d'interrogation des sources de données tout en cachant leur hétérogénéité et leur complexité, Il met l'accent sur la gestion de métadonnées, la génération de requête de médiation, et l'évaluation adaptative et interactive de requêtes d'utilisateur. Dans ce qui suit nous s'intéresserons qu'à la gestion de métadonnées et la génération de requêtes de médiation [TRI05] et l'interrogation des sources et l'évaluation de requêtes.

5.1.2 Objectif

Définir un canevas de système de médiation ouvert pour l'accès transparent aux sources hétérogènes et distribuées. Ce système de médiation de MediaGRID peut [LIU05] :

- (i) Supporter plus de sources disponibles en considérant des sources contenant des données faiblement structurées.
- (ii) Autoriser des résultats partiels pour des requêtes dans le cas des sources indisponibles et/ou pour satisfaire l'intérêt d'utilisateur.
- (iii) Un générateur de requête de médiation.
- (iv) Un évaluateur de requête adaptatif et interactif.

5.1.3 Architecture

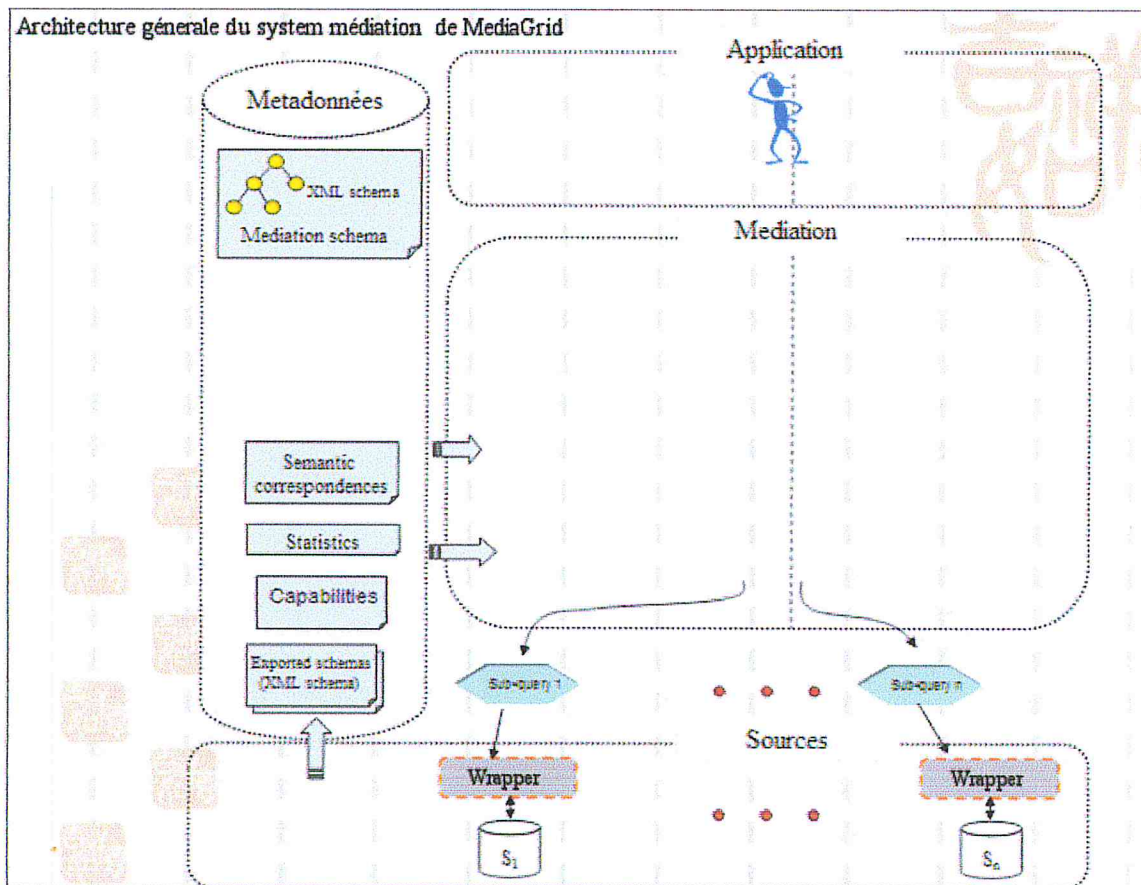


Figure10 : Architecture générale de MediaGRID [LIU05].

La figure 10 présente l'architecture générale de MediaGRID comme un médiateur basé sur XML. Excepté le module de gestion de métadonnées qui est indispensable à tout moment, les autres modules se regroupent autour de deux processus principaux, l'intégration et l'interrogation, qui se déroulent en deux étapes différentes.

La génération de requêtes de médiation repose sur les métadonnées composées de la définition des besoins en interrogation (i.e. le schéma global) de la description des sources (i.e. les schémas exportés des sources) ainsi que des correspondances sémantiques entre le schéma global et les schémas exportés pour générer la (les) requête(s) de médiation.

Les requêtes de médiation décrivent les liens de calcul entre le schéma global et les schémas exportés. Elles peuvent suivre aussi bien l'approche *global-as-view* que l'approche *local-as-view*.

Chapitre II : Etat de l'art sur le traitement des requêtes complexes

Dans MediaGRID, les requêtes de médiation sont générées selon l'approche *global-as-view* et stockées comme partie des métadonnées [TRI05]. Les requêtes d'utilisateur (ou expressions de besoins) peuvent être évaluées par le processus d'interrogation. Puisque le modèle de données pivot de MediaGRID est XML, les requêtes d'entrée sont formulées dans le langage XQuery. Elle est réécrite (par le module Rewriting) en utilisant la (les) requête(s) de médiation générée(s) précédemment. La requête réécrite, exprimée sur les sources sous-jacentes, est évaluée par le module Evaluator. Ce module est susceptible d'envoyer des sous-requêtes aux sources (ou plus précisément aux Wrappers), de récupérer des données et de réaliser des calculs nécessaires pour construire les résultats et les retourner à l'utilisateur [TRI05].

5.1.4 Gestion des métadonnées

Les métadonnées sont définies pour supporter la génération de requête de médiation et l'évaluation de requête globale [LIU05].

La méta-représentation d'un schéma XML est basée sur une graphique modelée comme une série de nœuds, Les métadonnées décrivent:

- ✓ Les schémas de médiation, les schémas exportés et les requêtes de médiation.
- ✓ Les correspondances sémantiques.
- ✓ La capacité des sources.
- ✓ La statistique sur les données (cardinalités, distribution., etc.).

5.1.5 Génération de requêtes de médiation

Étant donné [LIU05] : Un schéma de médiation avec La description des sources de données (schéma exporté), La génération automatique des requêtes de médiation dans cette approche se fait en trois étapes :

Étape 1: Identification des portions pertinentes de sources de données

Un schéma pertinent est composé par des éléments du schéma exporté, impliqués dans des correspondances sémantiques et la clé étrangère qui sont définies dans le schéma exporté.

Chapitre II : Etat de l'art sur le traitement des requêtes complexes

Etape 2: Recherche des opérations candidates

Les opérations candidates sont des opérateurs de jointures "Jointure candidate entre deux schémas pertinents ou Jointure candidate au sein d'un même schéma pertinent".

Etape 3: Génération des requêtes de médiation

Le schéma de médiation est décomposé en parties; Pour chaque partie, on recherche ses mappings partiels; Et enfin de combiner les mappings partiels pour produire les requêtes de médiation.

5.1.6 Interrogation des données XML

Dans MediaGRID, nous avons adopté l'algèbre Xtasy (une algèbre de l'approche *variable binding*) et y avons apporté quelques modifications afin d'enrichir et de faciliter l'expression de requête. L'idée principale de cette algèbre est de considérer la structure intermédiaire comme une *relation* pour le traitement des données. En conséquence, la plupart des opérateurs algébriques ressemblent à des opérateurs de l'algèbre relationnelle et objet (Selection, Projection, Join, etc.) excepté deux opérateurs **Source** et **Return**, appelés opérateurs frontières (*border operators*), le premier pour la transformation des données XML en la représentation intermédiaire (i.e. lecture des données XML) et le second pour la transformation inverse (i.e. production des données XML).

Précisément, la représentation interne des données XML est une *collection* (**Env**) d'un ensemble de variables de type **XTree**, XTree représente la structure d'arbre des éléments XML. Les éléments XML intéressés par le traitement d'une requête sont représentés comme des variables [**TRI05**].

5.1.7 Evaluation de requêtes

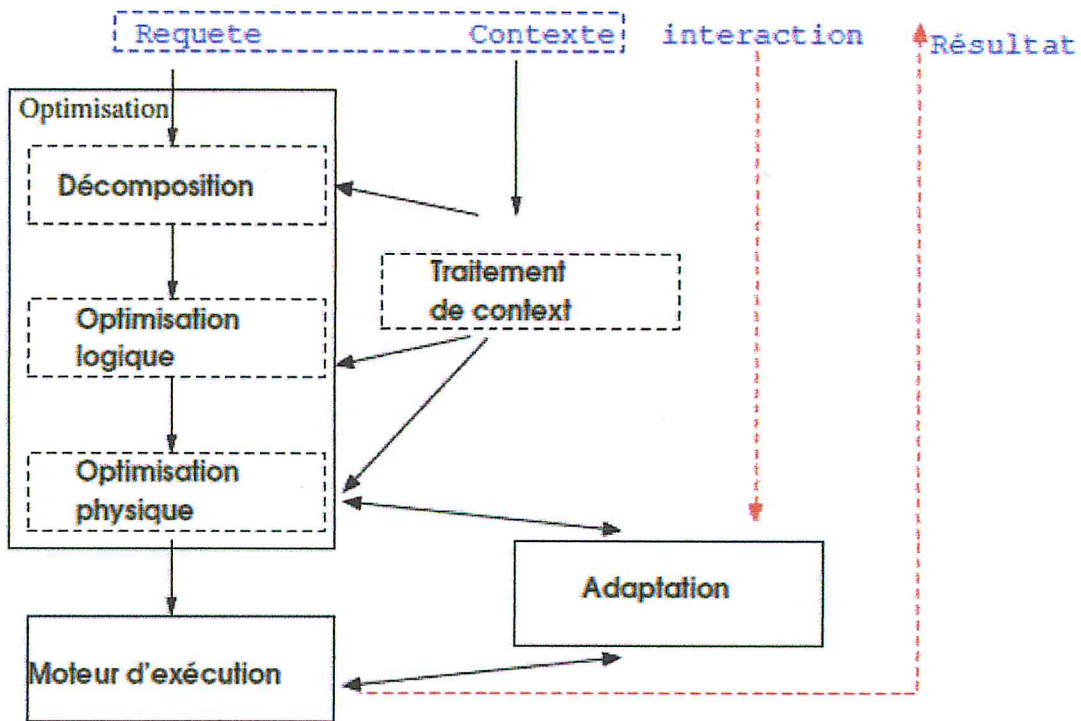


Figure 11: Evaluation de requêtes dans MediaGRID [TRI05].

La figure 11 montre l'évaluateur de requêtes qui reçoit une requête représentée sous forme de plan de requête auxquelles nous associons un contexte composé des paramètres tels que la cardinalité du résultat, la limite de temps d'exécution, le type des résultats partiels, etc.

Le processus d'évaluation de requêtes se compose des phases suivantes [TRI05]:

✓ Décomposition

L'objectif est d'identifier des opérateurs pouvant être évalués par les sources afin d'y déléguer leur exécution. Les informations concernant la capacité des sources sont stockées dans les métadonnées. La décomposition se fait en parcourant le plan de requête en commençant par les feuilles en remontant vers la racine ou selon l'ordre post-fixe afin d'annoter les nœuds. Tous les nœuds pouvant être résolus par une source sont regroupés en un nouvel opérateur appelé `getData`. L'opérateur `getData` a deux paramètres : (i) **url de la source** et (ii) **une requête XQuery** correspondant au sous-plan identifié. Les opérateurs `getData` sont susceptibles d'accéder aux sources distantes pour récupérer les données.

✓ **Optimisation logique**

Elle consiste à déterminer l'ordre des opérateurs à exécuter. Pour cela, les transformations de plan se basant sur les propriétés de l'algèbre sont appliquées. Quelques exemples de transformation sont : pousser les sélections, les projections vers les feuilles, fusionner l'opérateur Project et l'opérateur Source, etc.

✓ **Optimisation physique**

Elle vise à choisir le meilleur algorithme pour chacun des opérateurs du plan. Ce choix peut être guidé par les paramètres de contexte (gérés par la fonction du **traitement de contexte**) si ces derniers existent, (choisir des algorithmes pour pouvoir retourner les résultats partiels).

✓ **Adaptation**

Cette phase se déclenche lors qu'un changement significatif. Dans l'environnement d'exécution se produit. L'objectif est d'adapter l'exécution de requête à de nouvelles conditions (architecture, nombre de ressources, réseau, etc).

Nous considérons les interactions avec l'utilisateur comme une forme d'adaptation. Dans ce dernier cas, l'adaptation peut consister en modification d'intérêts, le Contrôle d'exécution de requêtes et aussi le raffinement des requêtes (changement de la sémantique) durant l'exécution de la requête.

5.2 Le projet SICOM [BOUM05]

Ce projet consiste à construire un système permettant à l'utilisateur de manipuler des informations stockées sur différentes bases de données appartenant à un même domaine, mais hétérogènes aussi bien dans leur contenu que dans leur conception.

La Figure suivante présente l'architecture du système de médiation proposé et qui permet à l'utilisateur de formuler une requête sur une vue unifiée et de récupérer un résultat provenant de l'exécution de la requête sur un ensemble de bases de données hétérogènes.

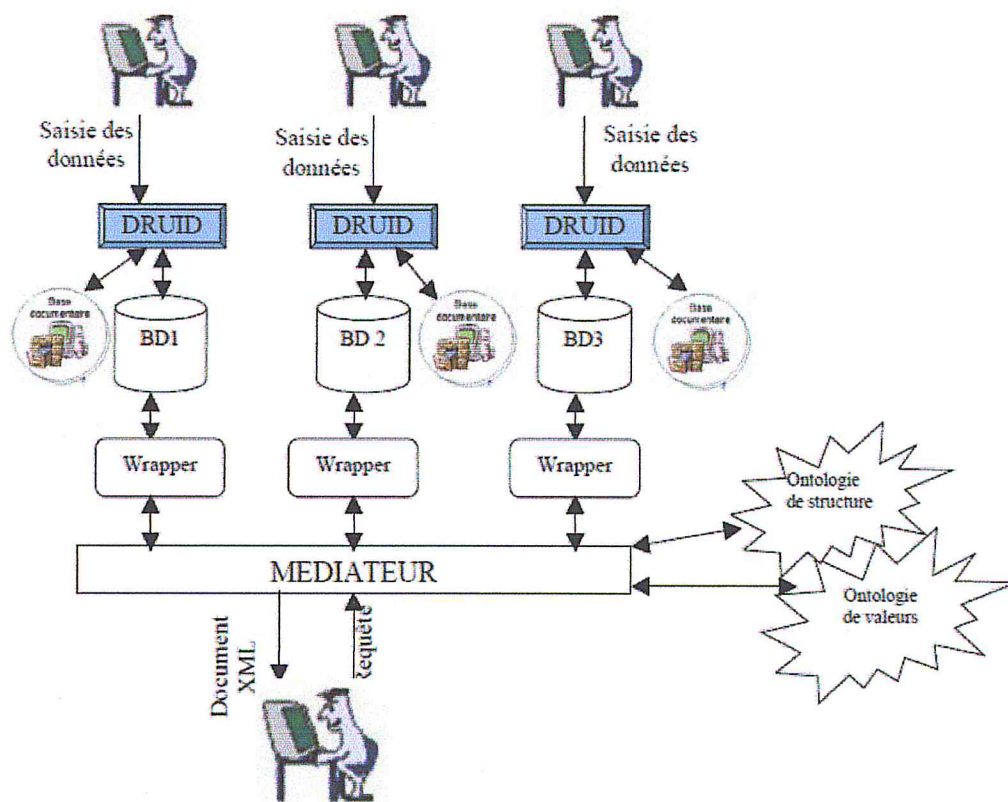


Figure 12 : Architecture du système SICOM.

DRUID permet à l'utilisateur de saisir ses informations dans un document semi-structuré où chaque paragraphe de texte libre est balisé. Ce document est ensuite traité afin d'en extraire les informations et de remplir la base de données. L'utilisateur qui interroge la base de données à travers DRUID récupère les tuples résultats ainsi que le document qui a alimenté la base de données. Cette solution est certes performante pour la saisie et la consultation des données, mais il est difficile de contrôler les informations entrantes et sortantes

5.2.1 Les différentes étapes de l'approche SICOM

Description des bases de données Son but est de créer, pour chaque schéma d'une Base de données source, un document XML le décrivant. Ce document contient la description des relations et attributs, issus des termes de l'ontologie. Pour ce faire, pour chaque relation (ou attribut) nous recherchons dans l'ontologie le concept le décrivant. Une fois terminé, toutes les structures à intégrer sont décrites par des termes unifiés.

Chapitre II : Etat de l'art sur le traitement des requêtes complexes

Recherche du concept Durant cette phase nous recherchons, dans l'ontologie de structure, la liste des concepts ci qui peuvent décrire la relation R. Ces concepts sont recherchés à partir de leur nom et de leurs synonymes

Génération du schéma médiateur Durant cette partie, le schéma médiateur est créé à partir des documents décrivant les bases de données. Nous trouvons aussi un regroupons les concepts équivalents décrivant les relations afin de créer une relation dans le schéma médiateur. Nous créons ensuite les correspondances la liant aux relations des schémas sources. Une fois terminé, le même traitement est appliqué aux attributs des relations.

5.2.2 Les ontologies

Dans la phase de description des bases de données, deux étapes sont clés :

- 1 : L'étude de la structure en recherchant le concept décrivant une relation ou un attribut,
- 2 : L'étude du contenu dont le but est d'aider l'utilisateur-concepteur dans son choix.

Ces deux phases reposent sur l'utilisation d'ontologies. Afin de différencier les termes utilisés dans les schémas des bases de données et ceux utilisés dans leur contenu, nous avons introduit les notions d'ontologie de structure et d'ontologie de valeurs.

✓ L'ontologie de structure

L'ontologie de structure est composée d'un ensemble de concepts qui sont issus des Vocabulaires utilisés dans les modèles des bases de données à intégrer tels que les noms des relations et attributs. Chaque concept est défini par son identifiant, son nom et son type.

✓ L'ontologie de valeurs

L'ontologie de valeurs recense l'ensemble des valeurs que peuvent contenir certains concepts du domaine. Cette ontologie inclue des concepts organisés par domaine, et où chacun est constitué d'un ensemble de valeurs et d'une liste de concepts fils.

5.2.3 Traitement des requêtes avec SICOM

Phase 1. Décomposition des requêtes

L'utilisateur formule une requête sur le schéma médiateur en utilisant le vocabulaire retenu dans le schéma de ce dernier. La requête est envoyée, ensuite, au médiateur afin d'être décomposée en plusieurs requêtes sur différentes sources de données. Le médiateur décompose la requête en trois parties, à savoir les données à afficher qui sont après la clause SELECT, les relations interrogées, se trouvant après la clause FROM, ainsi que les relations et les attributs qui sont contenus dans la condition et qui viennent après la clause WHERE. Une fois terminée, la liste des relations et attributs est extraite de la requête, elle est renvoyée au médiateur. Ce dernier la renvoie au traducteur de requête.

Phase 2. Traduction de la requête

Dans cette partie nous créons pour chaque base de données, ses propres requêtes. Pour ce faire le traducteur remplace les noms des relations et des attributs par leur équivalent dans chaque base de données. Durant la première phase, nous récupérons pour chaque base de données, les relations des requêtes principales et nous recherchons ensuite dans le document XML contenant les différentes correspondances, les relations qui leur sont équivalentes dans cette base de données.

Une fois le traitement des relations terminé, nous traduisons leurs attributs, sur lesquelles le même traitement est appliqué. Les attributs peuvent avoir, en plus des règles de correspondances, des fonctions de conversion. Dans ce cas cette fonction est ajoutée à l'attribut afin de traduire son contenu dans le format exigé par le schéma médiateur.

Phase 3. Exécution des requêtes

Une fois la requête principale traduite, le médiateur envoie les différentes requêtes qui lui sont équivalentes, au processeur de requêtes afin de les exécuter. Ce dernier interroge chaque base de données, récupère les résultats et les renvoie au composant *FormatXML* qui les rassemble dans un document XML, avant d'être présentés à l'utilisateur. Les informations extraites de chaque base de données sont regroupées dans un même élément XML afin de garder trace de la provenance de chacune d'entre elles. Ce document est composé d'un ensemble d'éléments, appelés « BD », contenant les noms des bases de données interrogées.

Ces éléments permettent de connaître d'où a été extraite chaque information. Chaque élément BD est identifié par le nom de la base de données et par un ensemble de tuples. Chaque tuple est composé d'un ensemble d'attributs ainsi que leur valeur.

5.3 L'approche de MAIZ [MAI07]

5.3.1 Description

Cette approche est constatée dans le cadre de la fusion automatique d'ontologies OWL correspondant aux schémas locaux de sources intégrées selon une approche de médiation. La méthode proposée permet de fusionner plusieurs ontologies à la fois. Elle utilise des techniques de classification hiérarchique afin de classer les concepts des ontologies et repérer les concepts équivalents. L'algorithme de classification utilise une mesure de similarité prenant en compte la terminologie, la structure et la sémantique des concepts [MAI07].

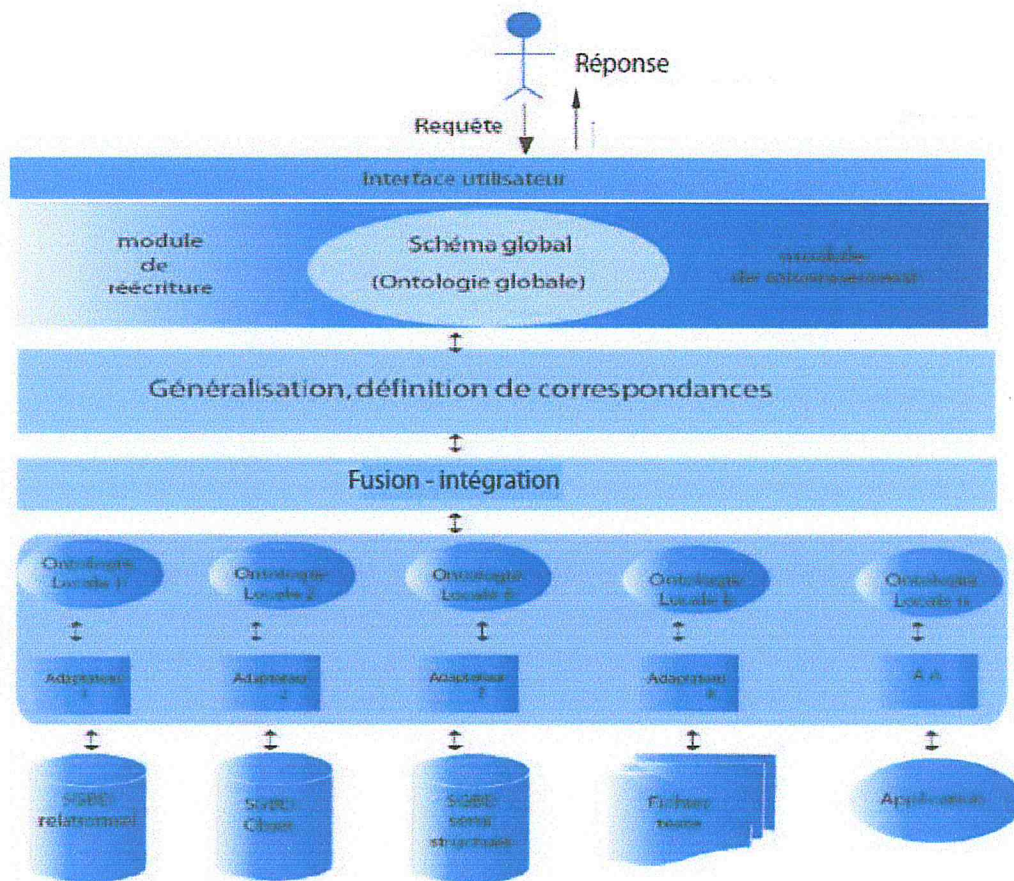


Figure 13: Architecture d'un système de médiation à base d'ontologie.

5.3.2 L'architecteur de système :

Pour la construction de ce système nous passons par trois étapes :

Etape1 : Construction des ontologies locales

Cette étapes consiste à construire une ontologie à partir des sources, c'est la représentation des sources qui ont été choisi comme pertinents dans un langage d'ontologie (par exemple : OWL) .

Etape2 : Construction de l'ontologie globale

Cette étape consiste à construire un vocabulaire partagé, en deux phases principales [MAI07] :

Phase 1: L'analyse des ontologies locales

Implique une analyse complète des ontologies déjà construites indépendamment. Nous argumentons que cette analyse est plus facile que de faire une analyse des sources de données, puisque la sémantique des concepts a déjà été définie dans les ontologies locales.

Phase 2:

La sélection de tous les concepts et la résolution des problèmes d'hétérogénéité sémantique Cette étape est établie pour la localiser des problèmes d'hétérogénéité sémantique(les conflits de confusion, les conflits de graduation et les conflits de nom).

Etape3:

Correspondances entre l'ontologie globale et les ontologies locales Les correspondances entre les deux niveaux d'ontologies sont les règles de passage d'un niveau à un autre.

Chacune de ces règles précise l'emplacement d'origine de l'entité ainsi que sa définition. Les règles de correspondances sont utiles dans la réécriture des requêtes. Elles permettent de réécrire la requête exprimée dans le langage de l'ontologie globale, en un ensemble de sous-requêtes exprimées dans les schémas locaux. L'ontologie globale est construite à partir des ontologies locales.

Pour identifier la source ontologique originelle des concepts dans l'ontologie globale, nous utilisons des annotations.

Chapitre II : Etat de l'art sur le traitement des requêtes complexes

Le langage OWL permet d'annoter des concepts et des propriétés selon un schéma de métadonnées prédéfini. Nous annotons les concepts de l'ontologie globale avec un schéma d'annotation qui contient trois propriétés :

- ✓ Ontologie locale originelle du concept à annoter (*Uri – Local – Ontology*)
- ✓ L'agent responsable de cette ontologie locale (*Agent – Local – Ontology*)
- ✓ Le nom du concept dans l'ontologie locale (*Concept – Local – Ontology*)

5.3.3 Traitement des requêtes

L'utilisateur compose sa requête sous forme de conjonction de concepts et de propriétés du vocabulaire de l'ontologie globale. Ensuite, un mécanisme de réécriture de requêtes doit assurer la décomposition et la recombinaison des résultats élémentaires de la requête décomposée.

5.3.4 Langage de requêtes

Il existe plusieurs manières d'établir la correspondance entre le schéma global et les schémas des sources de données à intégrer. Les approches GAV, LAV et GLAV guident la façon de répondre à une requête.

Dans GAV, on procède à un dépliement de requête dans LAV, on effectue une réécriture de requête et dans GLAV, le traitement de la requête passe par une réécriture puis un dépliement.

Ce système de médiation permet de poser des requêtes en utilisant l'ontologie globale et éventuellement les ontologies locales.

L'ontologie globale est un ensemble de concepts classés en une hiérarchie. Cette ontologie ne contient pas de propriétés reliant les différents concepts. Ces propriétés existent chacune dans leur ontologie locale. L'utilisateur peut donc créer une requête qui contient des concepts de l'ontologie globale et éventuellement des propriétés des ontologies locales. La requête exprimée en termes de concepts et propriétés doit être réécrite de manière à obtenir des résultats qu'on peut agréger. Tout concept qui ne garantit pas la combinaison des données obtenues est exclu. Du point de vue sémantique cette exclusion tend à rendre une requête cohérente. Une requête cohérente est une requête décomposable en sous-requêtes exécutables et dont les résultats sont composables. Cette réécriture peut être vue comme une correspondance entre l'ontologie globale et les ontologies locales, puisqu'elle permet de rendre le traitement de la requête utilisateur direct

de l'ontologie globale vers les ontologies locales. Une requête utilisateur de base est sous la forme :

Concept \wedge Propriété \wedge Concept ou tout simplement Concept.

La requête exprimée en termes de concepts et de propriétés doit être réécrite de manière à obtenir des résultats qu'on peut agréger (joindre). Tout concept qui ne garantit pas la combinaison des données obtenues sera exclu de la requête. Ceci garantit le fait que la requête ne doit contenir que des éléments (concepts) qui sont reliés par des propriétés afin de pouvoir les recomposer par la suite.

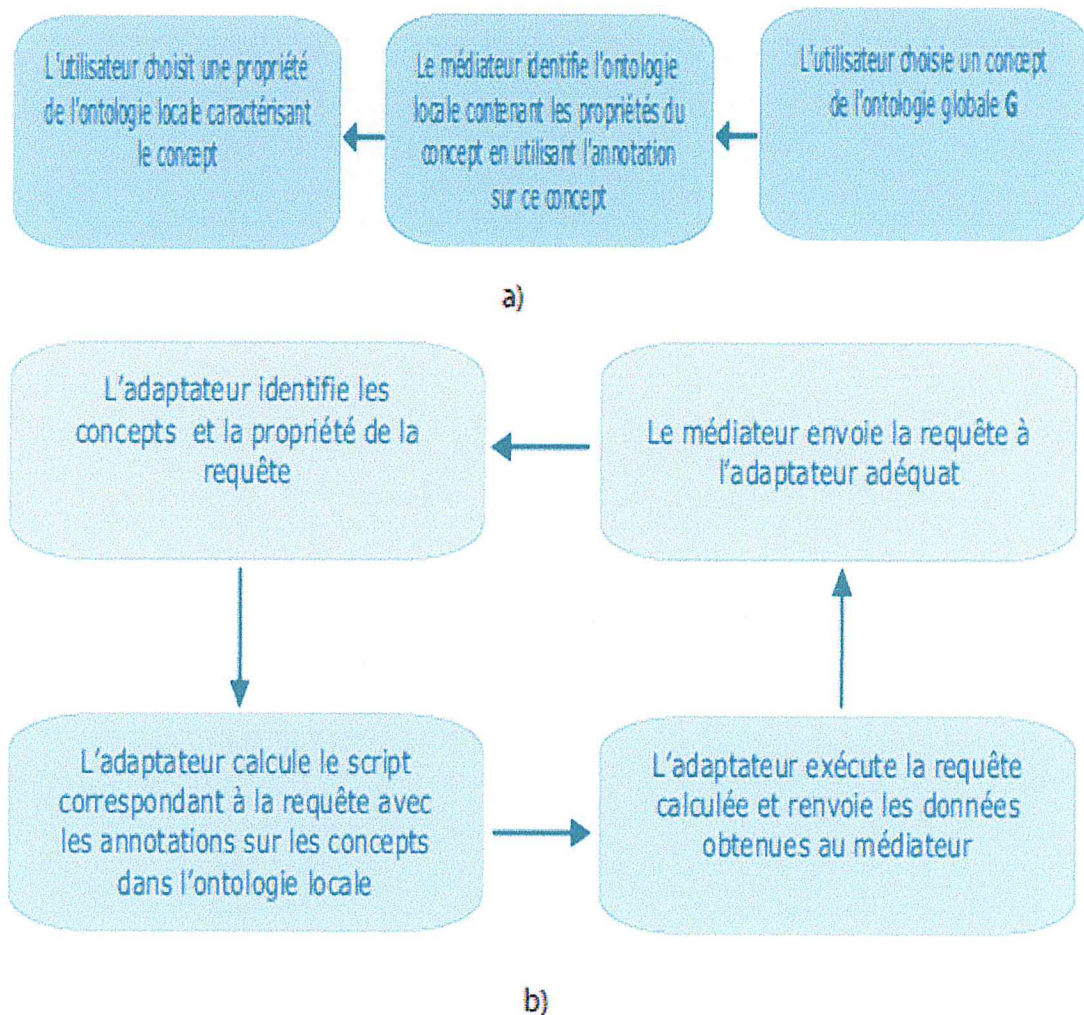


Figure 14: a) formulation d'une requête, b) Traitement d'une requête.

5.3.5 Réécriture de requête

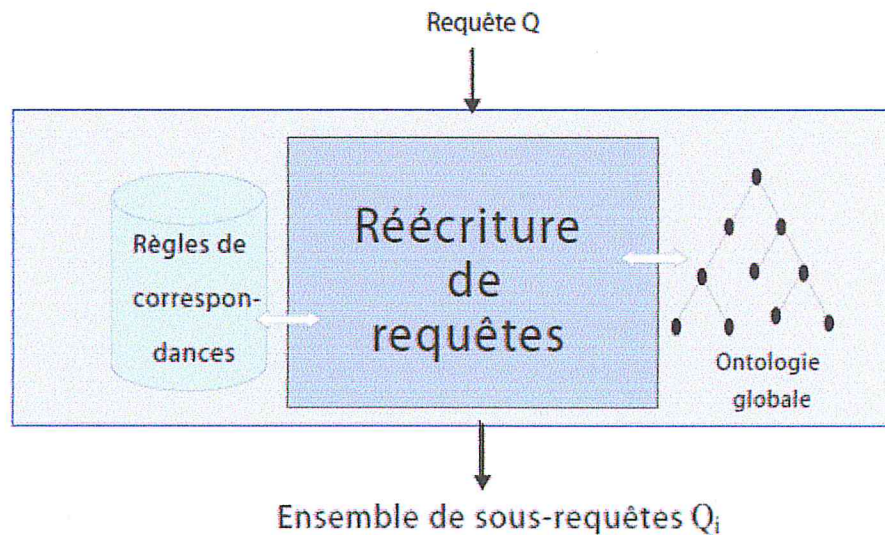


Figure 15: Système de réécriture de requêtes.

L'objectif principal de la réécriture de requêtes est l'obtention d'un ensemble de sous-requêtes exécutables directement sur les sources de données et dont les résultats sont joignables par la suite pour former une réponse finale pour l'utilisateur. Pour cela, il faut d'abord commencer par la recherche des sources (ontologies locales) qui peuvent apporter une réponse à la requête utilisateur en se basant sur la description de ces dernières. L'utilisateur pose une requête Q en utilisant le vocabulaire de l'ontologie globale. Une requête de l'utilisateur est une conjonction de concepts et de propriétés ou tout simplement de concepts de l'ontologie globale. Les requêtes sont de la forme suivante :

$Q : [\text{Concept}(a) \wedge \text{Property}(a, b) \wedge \text{Concept}(b)] \mid \text{Concept}(c).$

Afin de répondre à ces requêtes, le système doit d'abord trouver parmi toutes les sources, celles qui apportent une réponse à la requête utilisateur. Chaque source contient une partie de la réponse. Pour obtenir une réponse finale, il faut faire une jointure des différentes parties de réponse en provenance des différentes sources de données. Pour cela, avant de désigner une source, il faut s'assurer qu'elle soit liée avec celle qui la précède. Après la détection des sources candidates, il faut réécrire les sous-requêtes en fonction du vocabulaire de celles-ci en utilisant les règles de correspondance qui définissent le lien entre l'ontologie globale et les ontologies locales.

6. Etude comparative des différentes approches précédentes

		MEDIAGRID	SICOM	MAIZ
Contexte		Système de médiation - un schéma global -plusieurs schémas locaux	Système de médiation -un schéma global -un schéma local	Système de médiation - un schéma global - plusieurs schémas locaux
Correspondance entre schéma global et schémas sources		1-n	1-1	1-n
Conflit sémantique	Donnée	Oui	Oui	Oui
	Schéma	Oui	Oui	Oui
Langage de source		XML/Relationnel	Relationnel	Relationnel
Langage utilisé pour le schéma global		XMLs	XML	OWL
Requêtes complexes		Oui	Oui	Oui

Tableau 3:Etude comparative des différentes approches précédentes.

✓ Analyse comparative entre les approches précédentes

D'après l'étude de différentes approches existantes dans le domaine d'intégration des données ainsi que la génération des requêtes et le tableau de comparaison entre ces approches, nous avons pu extraire les avantages et les inconvénients principaux pour chaque approche, qui sont comme suit:

7. Conclusion

Ce chapitre conclut les différentes approches existantes dans le domaine d'intégration des données ainsi que le traitement des requêtes, nous avons également présenté un tableau qui résume les similarités et les différences entre ces approches, et une étude sur ces approches en mentionnent les principaux avantages et inconvénients de chaque une d'elle.

Dans le chapitre qui suit nous allons présenter notre solution sur le traitement des requêtes complexes dans un système de médiation.

Partie II
Analyse, conception et
implémentation

1. Introduction

Après avoir donné un aperçu général et comparatif sur les différentes approches sur le traitement automatique des requêtes avec une présentation sur les systèmes de médiation, dans ce chapitre nous allons présenter notre solution à base d'un système de médiation avec l'utilisation des requêtes complexes.

2. Terrain d'étude

Pour améliorer notre travail, il est nécessaire d'avoir un terrain pour faire une étude pratique, pour cela nous avons fait un stage dans le cadre de l'entreprise SONATRACH au niveau de département formation de la division production. Nous allons présenter par la suite les démarches d'informations.

3. Procédure de travail de département formation (SONATRACH)

Nous sommes dirigé a l'entreprise SONATRACH pour faire une étude sur l'activité ingénierie de département formation.

Le département formation fait partie de la division ingénierie et production SONATRACH de direction ressource humaine, ce département gère la gestion et la planification des formations pour les employés de la division.

La formation a pour but d'améliorer les compétences et les performances des employés de L'entreprise selon leur demande, c'est à dire chaque employé de l'entreprise a le droit de spécifier ses besoins en formation.

Il existe deux types de formations, celles qui sont externe de l'entreprise (en Algérie ou à l'étranger) et d'autres qui sont interne.

La gestion des formations suit la procédure de travail suivante:

- ✓ Durant l'année l'ingénieur de département formation envoie une **fiche de besoin** à chaque employé de structures de l'entreprise, l'employé remplit la fiche et la remettre à l'ingénieur avec un **tableau récapitulative**.
- ✓ L'activité AMONT envoie des **canevas** a l'ingénieur pour les remplir, et établit une synthèse globale et consolide les fiches de besoin pour avoir un **plan des formations prévisionnel à moyen terme(PMTE)** qui définit les besoins en formation pour l'année prochaine, après il envoie ce plan au chef de département pour une validation, le plan

Chapitre III: Analyse et conception

validé sera envoyé avec les canevas a AMONT et la direction générale pour une autre validation.

Pendant l'année il est possible de programmer de nouveaux thèmes de formation qui sont en hors plan.

Une fois les formations sont effectuées, chaque employé concerné envoie à l'ingénieur des informations sur la formation réalisée, L'ingénieur de son côté établit un tableau de bord (TBRH) chaque mois en mentionnant les formations effectuées, le tableau de bord sera envoyé a AMONT pour le consolider et à la direction générale pour le valider.

À la fin de chaque année **un bilan** de formations sera établi à base **des tableaux de bord (TBRH)**.

Les tableaux de bord seront accompagnés avec des synthèses, et les bilans avec des commentaires, synthèses et présentation.

L'ingénieur prépare les commentaires et les synthèses par une demande de directeur pour définir les statistiques durant l'année.

Nous trouvons :

- ✓ Des statistiques qui sont effectuées par rapport aux participants, c'est à dire le nombre d'employés qui ont participé à telle formation.
- ✓ Des statistiques par rapport au volume pédagogique en jours :
- ✓ Des statistiques par rapport au cout, ici il s'agit de savoir quel sont les formations qui ont le cout le plus élevé ou le contraire.
- ✓ Une analyse statistique par rapport au métier, Il existe trois types de métiers :
 - Fonction de soutien (ressource humaine, service informatique...etc.)
 - Fonction de support.
 - Cœur de métier (service technique...etc.).

Le but de cette analyse est de savoir pour chaque formation combien d'employés ont participé, et quelle formation contient un grand nombre de participants.

✓ Nous trouvons aussi des statistiques par rapport au domaine (sécurité, informatique, ressource humaine, exploitation, management, finance,...etc.), donc il s'agit de savoir le nombre de formations réalisés pour chaque domaine ou chaque métiers.

Chapitre III: Analyse et conception

Ex : 20% des formations ont été réalisé pour le domaine informatique.

40% des formations ont été réalisé pour le domaine management.

Il y a aussi des statistiques pour avoir:

- ✓ le cout global de chaque formation et le cout total de toutes les formations.
- ✓ Le taux réel des formations pour chaque organisation.
- ✓ la liste des formations de langue durée "diplomates", et de courte durée
- ✓ L'effectif des formations.
- ✓ Le taux réel de chaque formation pour chaque fonction il est calculé par trois façons :
 - Par rapport au cout.
 - Par rapport à l'effectif.
 - Par rapport à la durée en jours et heures.
- ✓ Le nombre total des employés pour chaque formation.

4. La solution proposée

Après l'étude des travaux déjà réalisés dans le cadre de traitement des requêtes dans un système de médiation, et de qui a été déjà construit dans la première partie de ce projet, nous avons pu voir globalement ce que nous allons proposer comme solution.

Notre travail consiste à construire un système qui répond aux besoins suivants:

- le traitement automatique des requêtes complexes « c'est-à-dire la prise en compte des requêtes avec des **JOINTURES** et des clauses **WHERE** (conditions)»;
- En interrogeant des sources de données hétérogènes telles que les données, les métadonnées et les connaissances;
- Et avec la prise en compte des correspondances 1-n entre le schéma global et les schémas sources;
- Le schéma global qui sera en langage OWL ;
- En prenant compte des confits sémantiques.

5. Architecture globale de la solution proposée

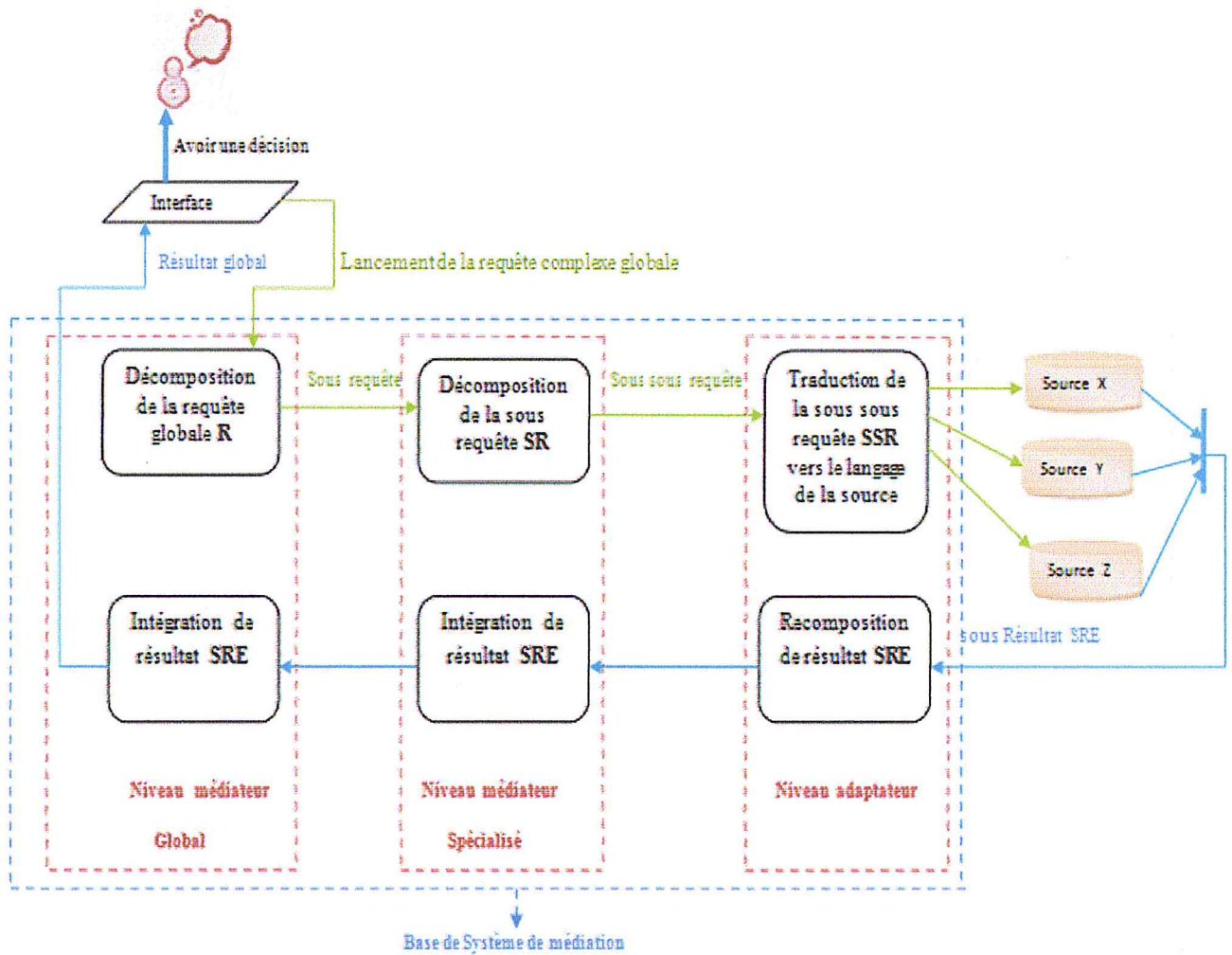


Figure 16: architecture globale de notre solution.

✓ Description du schéma :

Source X : source relationnel ou objet relationnel ou XML;

Source Y : RuleML ou RDF;

Source Z : OWL;

R : requête globale;

SR : Sous requête;

Requête SSR: la sous sous requête qui va interroger une source;

Résultat SRE : le sous résultats de l'exécution de la SSR.

6. Présentation de la démarche utilisée

6.1 Le Processus Unifié UP

UP (Unified Process) est une méthode générique de développement de logiciel : elle regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel [LAS01].

Caractéristiques essentielles du processus unifié [JAC00] :

✓ **Il utilise le langage UML** (ensemble d'outils et de diagramme).

✓ **Il est piloté par les cas d'utilisation**

Le développement est centré sur les besoins d'utilisateurs, il importe de bien comprendre les désirs et les besoins de ses futurs utilisateurs.

✓ **Il est centré sur l'architecture**

L'architecture d'un système logiciel peut être décrite comme les différentes vues du système qui doit être construit.

✓ **Il est itératif et incrémental:**

Le développement d'un produit logiciel est une vaste opération qui peut s'étendre sur plusieurs mois, voire sur une année ou plus. Donc il est très utile de découper le travail en plusieurs parties qui sont autant de mini-projets. Chacun d'eux représente une itération qui donne lieu à un incrément,

Les itérations désignent des étapes de l'enchaînement d'activités, tandis que les incréments correspondent à des stades de développement du produit.

✓ **Cycle de vie du processus unifié**

Le processus unifié répète un certain nombre de fois une série de cycles. Tout cycle se conclut par la livraison d'une version du produit aux clients et s'articule en 4phases : création, élaboration, construction et transition, chacune d'entre elles se subdivisant à son tour en itérations [LAS01].

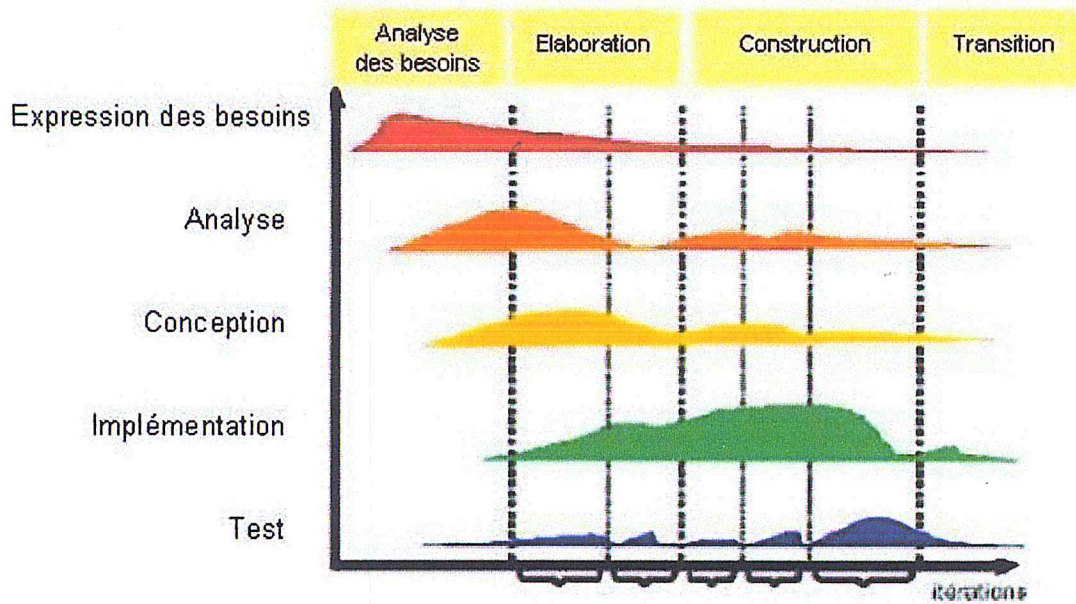


Figure 17: Cycle de vie d'UP [ROQ 07].

UP gère le processus de développement par deux axes :

L'axe vertical représente les principaux enchaînements d'activités, qui regroupent les activités selon leur nature. Cette dimension rend compte l'aspect statique du processus qui s'exprime en termes de composants, de processus, d'activités, d'enchaînements, d'artefacts et de travailleurs.

L'axe horizontal représente le temps et montre le déroulement du cycle de vie du processus; cette dimension rend compte de l'aspect dynamique du processus qui s'exprime en terme de cycles, de phases, d'itérations et de jalons.

Phase	Description et Enchaînement d'activités
Phase de création	Traduit une idée en vision de produit fini et présente une étude de rentabilité pour ce produit - Que va faire le système pour les utilisateurs ? - A quoi peut ressembler l'architecture d'un tel système ? - Quels sont l'organisation et les coûts du développement de ce produit ? On fait apparaître les principaux cas d'utilisation. L'architecture est provisoire, identification des risques majeurs et planification de la phase d'élaboration.
Phase d'élaboration	Permet de préciser la plupart des cas d'utilisation et de concevoir l'architecture du système. L'architecture doit être exprimée sous forme de vue de chacun des modèles. Emergence d'une architecture de référence. A l'issue de cette phase, le chef de projet doit être en mesure de prévoir les activités et d'estimer les ressources nécessaires à l'achèvement du projet.
Phase de construction	Moment où l'on construit le produit. L'architecture de référence se métamorphose en produit complet, elle est maintenant stable. Le produit contient tous les cas d'utilisation que les chefs de projet, en accord avec les utilisateurs ont décidé de mettre au point pour cette version. Celle-ci doit encore avoir des anomalies qui peuvent être en partie résolues lors de la phase de transition.
Phase de transition	Le produit est en version bêta. Un groupe d'utilisateurs essaye le produit et détecte les anomalies et défauts. Cette phase suppose des activités comme la fabrication, la formation des utilisateurs clients, la mise en œuvre d'un service d'assistance et la correction des anomalies constatées. (où le report de leur correction à la version suivante)

Tableau 4: description de cycle de vie d'UP [LAS01].

6.2 Le langage UML

UML ou Unified Modeling Language, est un langage de modélisation qui est né au milieu des années 90 de la fusion de trois méthodes objets: OMT (Object Modeling Technique), BOOCH1 (GRADY BOOCH le concepteur de la méthode) et OOSE (Object Oriented Software Engineering). L'idée de cette fusion est partie du constat qu'à l'époque ils existaient plusieurs méthodes objets liées par un consensus autour d'idées communes: objets, classes, sous-systèmes etc [KHE11].

UML comporte ainsi treize types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information. Ils se répartissent en trois grands groupes [KHEL11] :

- **Diagrammes structurels ou diagrammes statiques (UML Structure)**
 - ✓ diagramme de classes (Class diagram)
 - ✓ diagramme d'objets (Object diagram)
 - ✓ diagramme de composants (Component diagram)
 - ✓ diagramme de déploiement (Deployment diagram)
 - ✓ diagramme de paquetages (Package diagram)
 - ✓ diagramme de structures composites (Composite structure diagram)
- **Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior)**
 - ✓ diagramme de cas d'utilisation (Use case diagram)
 - ✓ diagramme d'activités (Activity diagram)
 - ✓ diagramme d'états-transitions (State machine diagram)
- **Diagrammes d'interaction (Interaction diagram)**
 - ✓ diagramme de séquence (Sequence diagram)
 - ✓ diagramme de communication (Communication diagram)
 - ✓ diagramme global d'interaction (Interaction overview diagram)
 - ✓ diagramme de temps (Timing diagram)

7. Expression des besoins

7.1 Identification des acteurs

Un Acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système), qui interagissent directement avec le système étudié.

Le tableau suivant va nous permettre de savoir les différents utilisateurs de notre système:

Acteurs	Description
Décideur	c'est l'acteur qui peut prendre des décisions à partir du système (chef de département de formation)
AgentBT	Il s'occupe de l'établissement des bilans et des tableaux de bord (TBRH)
Agent P	Il s'occupe de la construction des plans (PMTE)

Tableau 5 : Tableau descriptive des acteurs.

7.2 Identification des cas d'utilisation

Les cas d'utilisation font apparaître les besoins fonctionnels et leur ensemble constitue le modèle des cas d'utilisation qui décrit les fonctionnalités complètes du système.

Dans notre cas d'étude nous avons défini les cas d'utilisation suivants:

- ✓ Le choix des indicateurs à calculer;
- ✓ La mise à jour du PMTE;
- ✓ L'établissement du bilan;
- ✓ L'établissement du TBRH ;
- ✓ Lancement de la requête globale ;
- ✓ Validation de PMTE ;

• Diagrammes des cas d'utilisation

✓ Diagramme des cas d'utilisation global

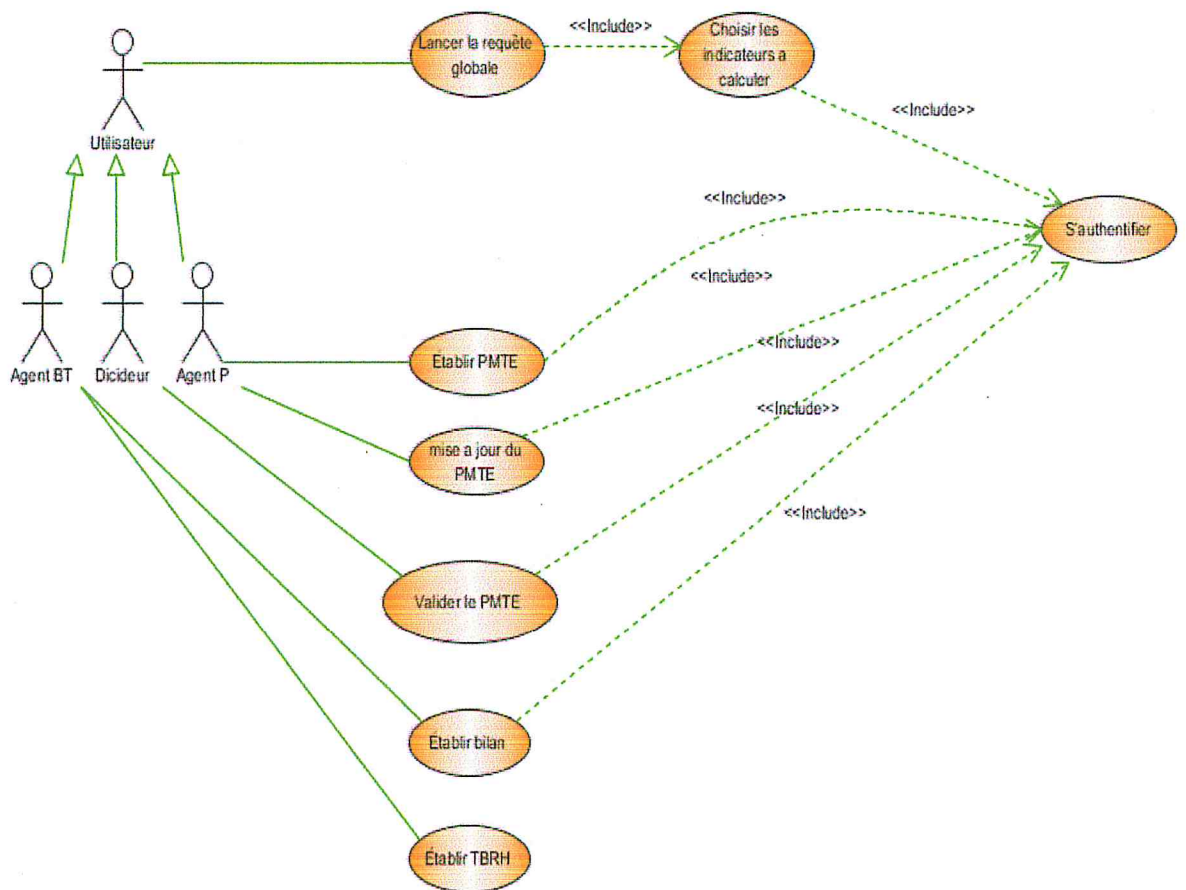


Figure 18 : diagramme des cas d'utilisation global.

Chapitre III: Analyse et conception

➤ Description des cas d'utilisation

Cas d'utilisation N°1: S'authentifier	
Acteur	Utilisateur
But	Permettre à l'utilisateur de se connecter et se déconnecter du système
Description	L'utilisateur demande l'accès au système, ou une interface d'authentification s'apparaît pour entrer le nom d'utilisateur et son mot de passe
Précondition	L'utilisateur doit entrer le nom et le mot de passe correctement
Post condition	L'utilisateur peut effectuer les tâches qui lui correspondent
Exception	Si l'utilisateur entre le nom ou le mot de passe incorrecte, le système va lui demander de s'authentifier à nouveau

Tableau 6: description de cas d'utilisation (s'authentifier).

Cas d'utilisation N°2: Lancer la requête globale	
Acteur	Utilisateur
But	L'utilisateur obtient des informations ou des indicateurs à partir des sources
Description	L'utilisateur pose sa requête et lance le processus d'exécution de la requête et récupère le résultat
Précondition	Le choix des indicateurs à calculer
Post condition	La restitution de résultat
Exception	Si une erreur se produit, le système affiche un message d'erreur
Sous cas d'utilisation N°1: Choisir les indicateurs à calculer	
But	L'utilisateur spécifie les indicateurs à calculer l'exécution de la requête globale
Précondition	L'authentification
Post condition	Le lancement de la requête globale

Tableau 7: description de cas d'utilisation (lancer la requête globale).

Chapitre III: Analyse et conception

Cas d'utilisation N°4: Etablir le PMTE	
Acteur	Agent P
But	L'agent P établit le PMTE, qui contient les formations qui seront réalisées durant l'année prochaine.
Description	L'agent P enregistre les formations demandées par les employées dans le PMTE
Précondition	La création du PMTE
Post condition	La validation du PMTE
Sous cas d'utilisation N°1: Créer le PMTE	
But	L'agent P crée le PMTE pour enregistrer les formations à réaliser
Description	l'agent P crée un plan (qui est un tableau), ou il va enregistrer les formations à réaliser
Précondition	L'authentification
Post condition	L'établissement du PMTE

Tableau 8: description des cas d'utilisation (établir le PMTE).

Cas d'utilisation N°5: Mise à jour du PMTE	
Acteur	Agent P
But	L'agent P fait des modifications au niveau des formations enregistrées.
Description	L'agent P modifie dans le PMTE et enregistre les modifications.
Précondition	L'établissement du plan prévisionnel à moyen terme
Post condition	La validation du PMTE

Tableau 09: description de cas d'utilisation (mise à jour du PMTE).

Chapitre III: Analyse et conception

Cas d'utilisation N°6: Valider le PMTE	
Acteur	Décideur
But	Le décideur valide le PMTE
Description	Le décideur consulte le PMTE et donne son avis à propos des formations demandés, dont le but d'établir les formations.
Précondition	Le lancement de la requête globale

Tableau10: description des cas d'utilisation (valider le PMTE).

Cas d'utilisation N°7: Etablir le bilan	
Acteur	Agent BT
But	L'agent TB établit le bilan qui contient les formations réalisées dans une période passé
Description	L'agent BT enregistre les informations de chaque formation réalisée
Précondition	Validation de PMTE
Post condition	Etablissement des TBRH

Tableau 11: description de cas d'utilisation (établir le bilan).

Cas d'utilisation N°8: Etablir le TBRH	
Acteur	Agent BT
But	L'agent BT établit le tableau de bord qui contient les indicateurs calculés à base du bilan de l'année passé
Description	L'agent BT enregistre les indicateurs dans un tableau de bord
Précondition	le lancement de la requête globale

Tableau 12: description de cas d'utilisation (établir le TBRH).



✓ Diagramme de cas d'utilisation détaillé pour le choix des indicateurs à calculer

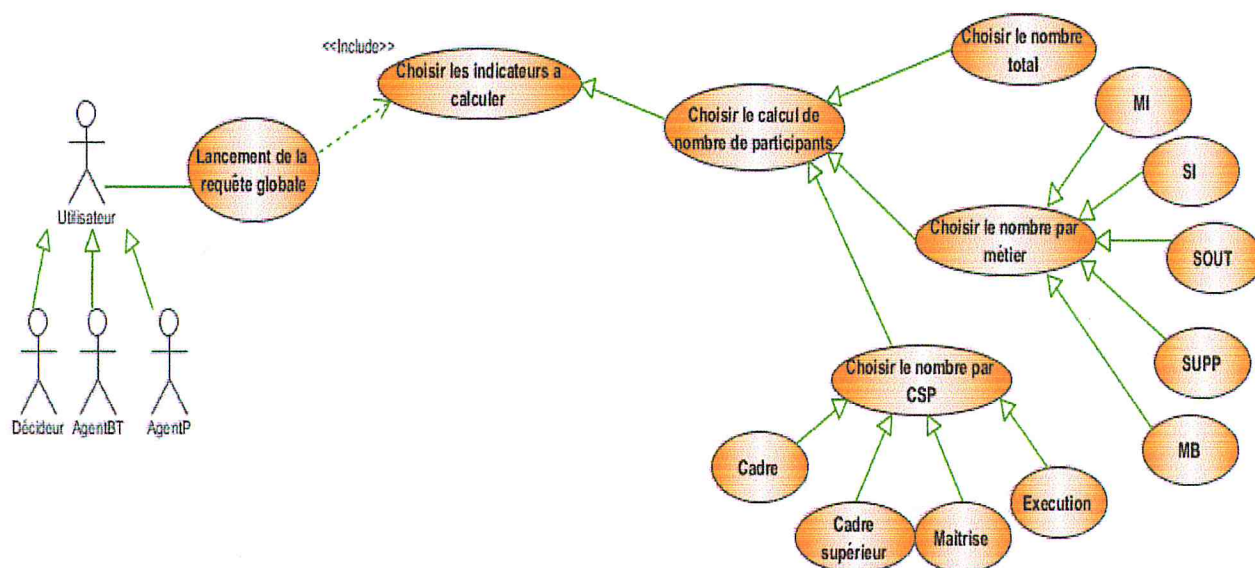


Figure 19 : Diagramme de cas d'utilisation pour le choix des indicateurs à calculer.

✓ Description :

MI : maintenance industrielle. MB : métier de base.

SI : sécurité industrielle. SOUT : fonction de soutien.

SUPP : fonction de support.

➤ Description des cas d'utilisation

Choisir les indicateurs à calculer	
Cas d'utilisation N°1: Choisir le nombre de participants	
But	Avoir la somme des participants aux formations
Sous cas d'utilisation N°1: Choisir le nombre total	
But	Le calcul du nombre total des employés qui ont participé aux formations
Sous cas d'utilisation N°2: Choisir le nombre par CSP	
But	Le calcul du nombre de participants par catégorie socio professionnelle (CADRE,

Chapitre III: Analyse et conception

	CADRE SUPERIEUR, EXECUTION, MAITRISE)
Sous cas d'utilisation N°3: Choisir le nombre par métier	
But	Le calcul du nombre de participants par métier (MI, SI, SOUT, MB, SUPP)

Tableau 13: description de cas d'utilisation (choisir les indicateurs à calculer)

✓ **Diagramme de cas d'utilisation détaillé pour la consultation du PMTE**

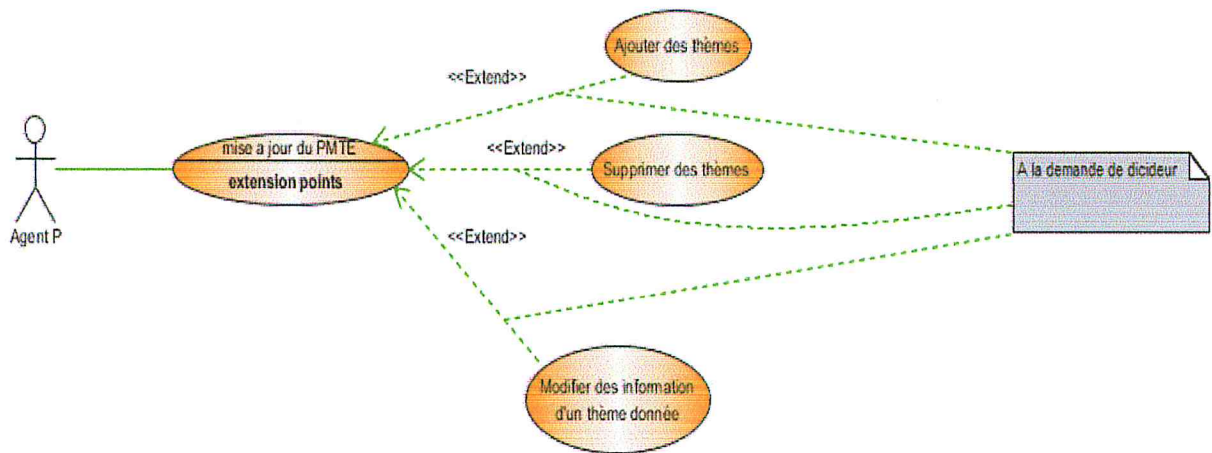


Figure 20 : diagramme de cas d'utilisation pour la mise à jour du PMTE.

➤ **Description des cas d'utilisation**

Mise à jour du PMTE	
Cas d'utilisation N°1: Ajouter des thèmes	
But	Ajouter des thèmes de formations au niveau du PMTE
Cas d'utilisation N°2: Supprimer des thèmes	
But	Supprimer des thèmes de formations au niveau du PMTE
Cas d'utilisation N°3: Modifier des informations d'un thème	
But	Modifier des informations concernant un thème au niveau de PMTE

Tableau 14: description de cas d'utilisation (consulter le PMTE).

✓ Diagramme de cas d'utilisation détaillé pour d'établissement du bilan

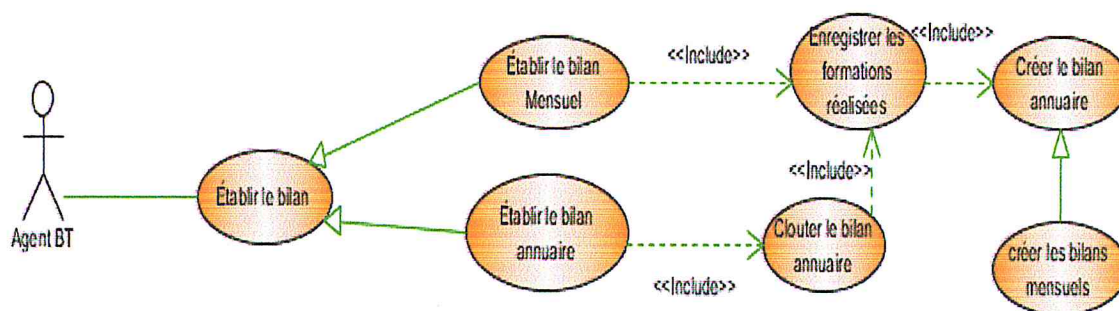


Figure 21: diagramme de cas d'utilisation pour l'établissement du bilan.

➤ Description des cas d'utilisation

Etablir le bilan	
Cas d'utilisation N°1: Etablir le bilan mensuel	
But	L'établissement d'un bilan qui contient les formations réalisées durant un mois
Cas d'utilisation N°2: Etablir le bilan annuel	
But	L'établissement d'un bilan qui contient les formations réalisées durant une année, c'est l'ensemble des bilans mensuels d'une année
Sous cas d'utilisation N°1: Clouter le bilan annuel	
But	L'enregistrement final du bilan (version finale), après la clôture on ne peut pas modifier le bilan
Sous cas d'utilisation N°2: Enregistrer les formations réalisées	
But	A chaque fin de formation on enregistre ses informations nécessaires dans le bilan
Sous cas d'utilisation N° 2.1 : Créer le bilan annuel	
But	La création du bilan annuel qui sera au début de l'année
Sous cas d'utilisation N° 2.1.1 : Créer les bilans mensuels	
But	La création d'un bilan chaque mois qui est inclus dans le bilan annuel

Tableau 15: description de cas d'utilisation (établir le bilan).

✓ Diagramme de cas d'utilisation détaillé pour l'établissement du TBRH

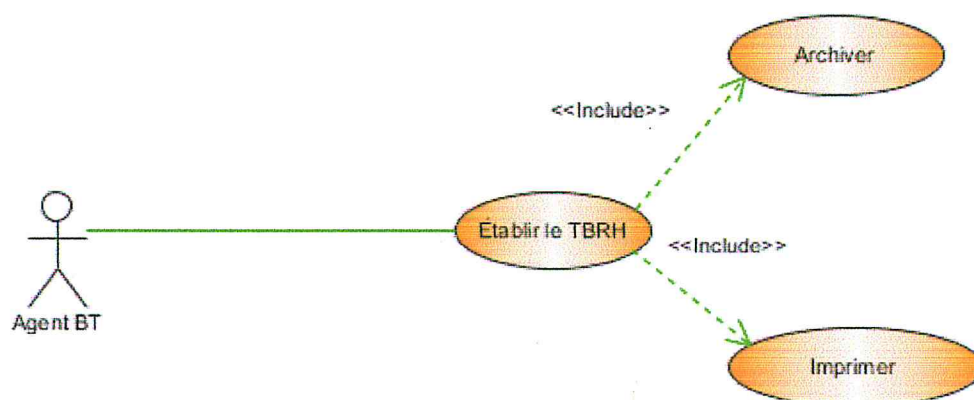


Figure 22: Diagramme de cas d'utilisation détaillé pour l'établissement du TBRH.

➤ Description des cas d'utilisation

Etablir le TBRH	
Cas d'utilisation N°1: Etablir le TBRH annuel	
But	l'enregistrement des indicateurs calculés concernant les formations réalisé durant l'année
Cas d'utilisation N°2: Archiver	
But	La sauvegarde du TBRH sous un fichier Excel
Cas d'utilisation N°3: Imprimer	
But	L'impression du TBRH

Tableau 16: description de cas d'utilisation (établir le TBRH).

8. Analyse

L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences du client. Il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution.

Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structure sous une forme qui facilite la compréhension (scénarios), la préparation (définition de l'architecture), la modification et la maintenance du futur système.

Il s'écrit dans le langage des développeurs et peut être considéré comme une première ébauche du modèle de conception.

8.1 Diagramme de séquence

Un diagramme de séquence est une représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et /ou de ses acteurs

✓ Diagramme de séquence d'authentification

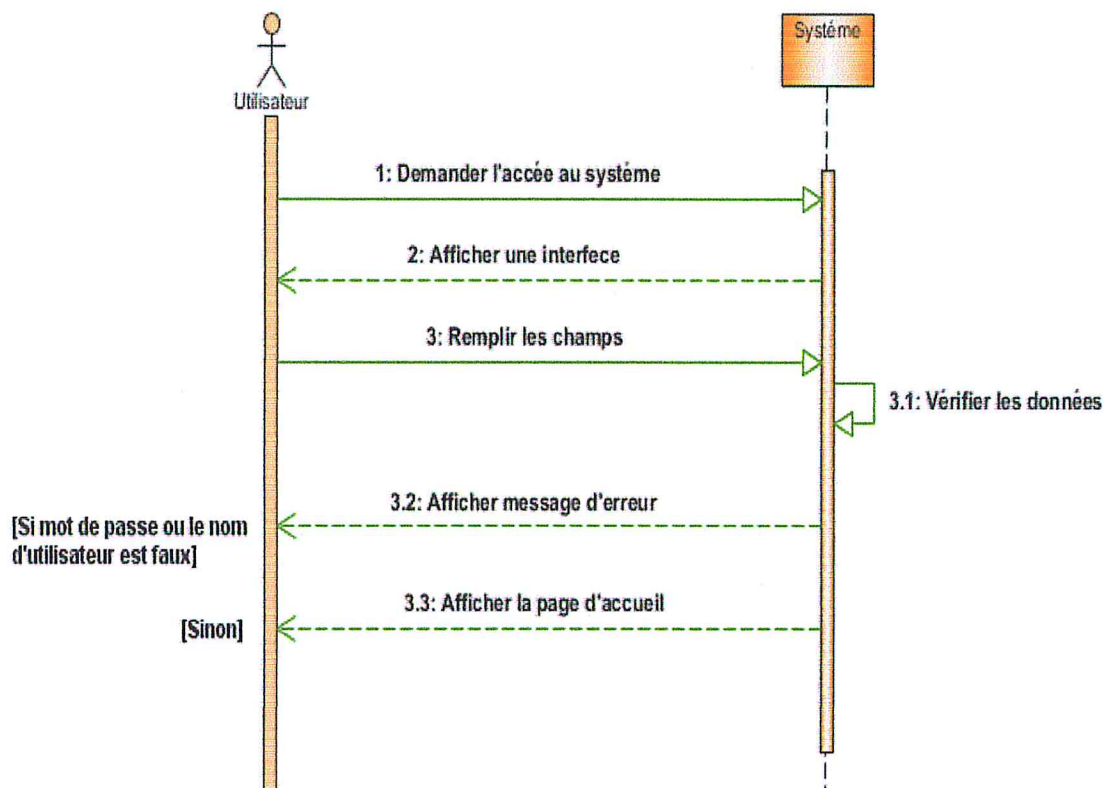


Figure 23 : Diagramme de séquence d'authentification.

✓ Diagramme de séquence de suppression des thèmes de formation dans un plan

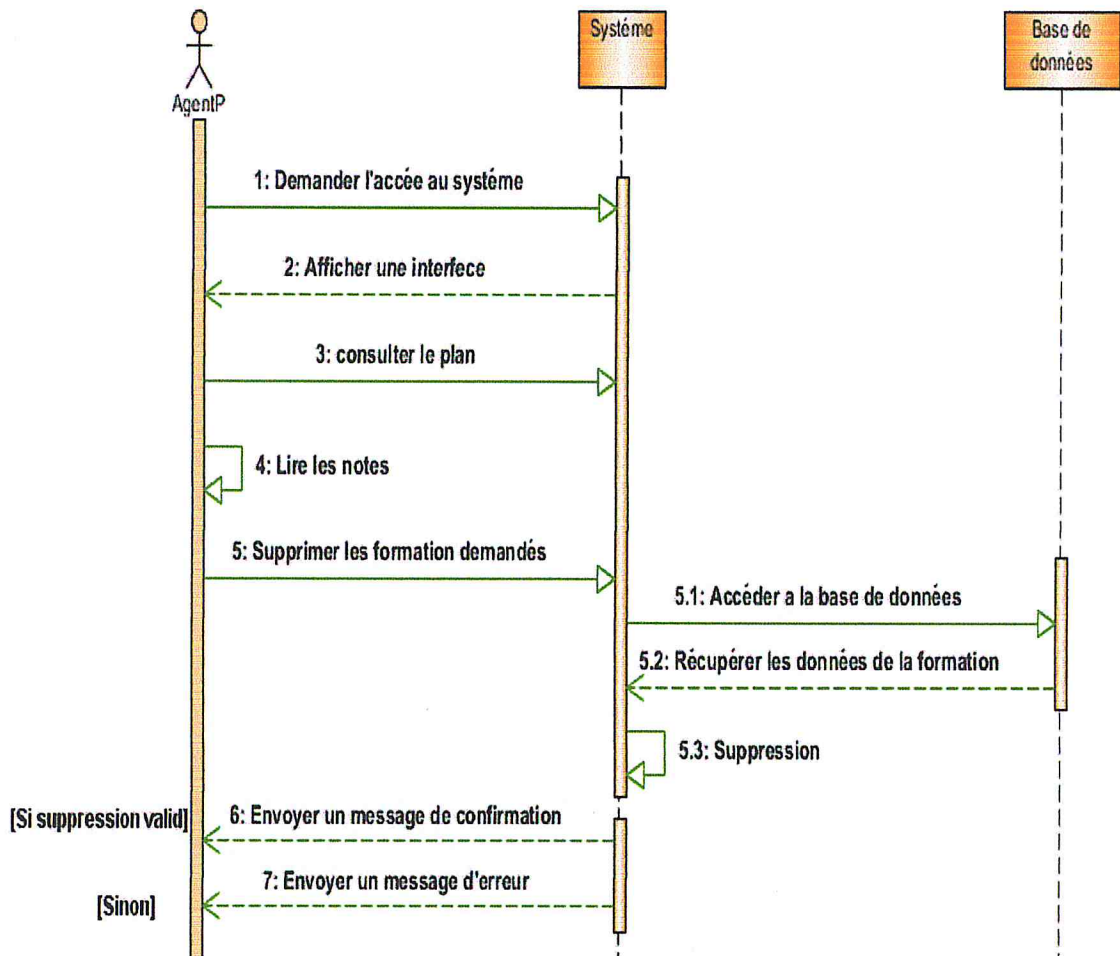


Figure 24: Diagramme de séquence de la suppression des thèmes de formation dans un plan.

✓ Diagramme de séquence validation du plan

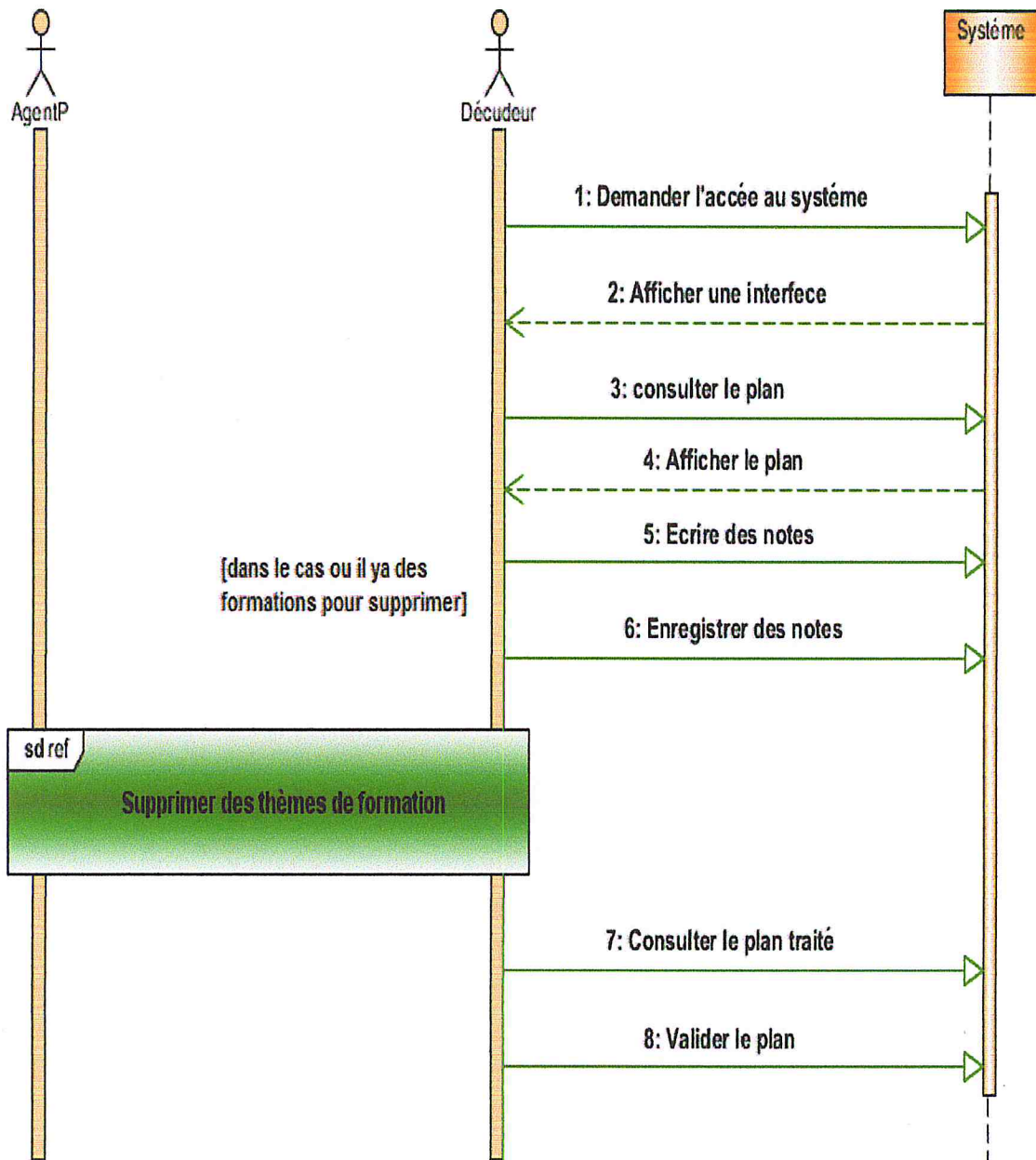


Figure 25 : Diagramme de séquence de l'ajout des notes et validation du plan.

✓ Diagramme de séquence de lancement de la requête globale

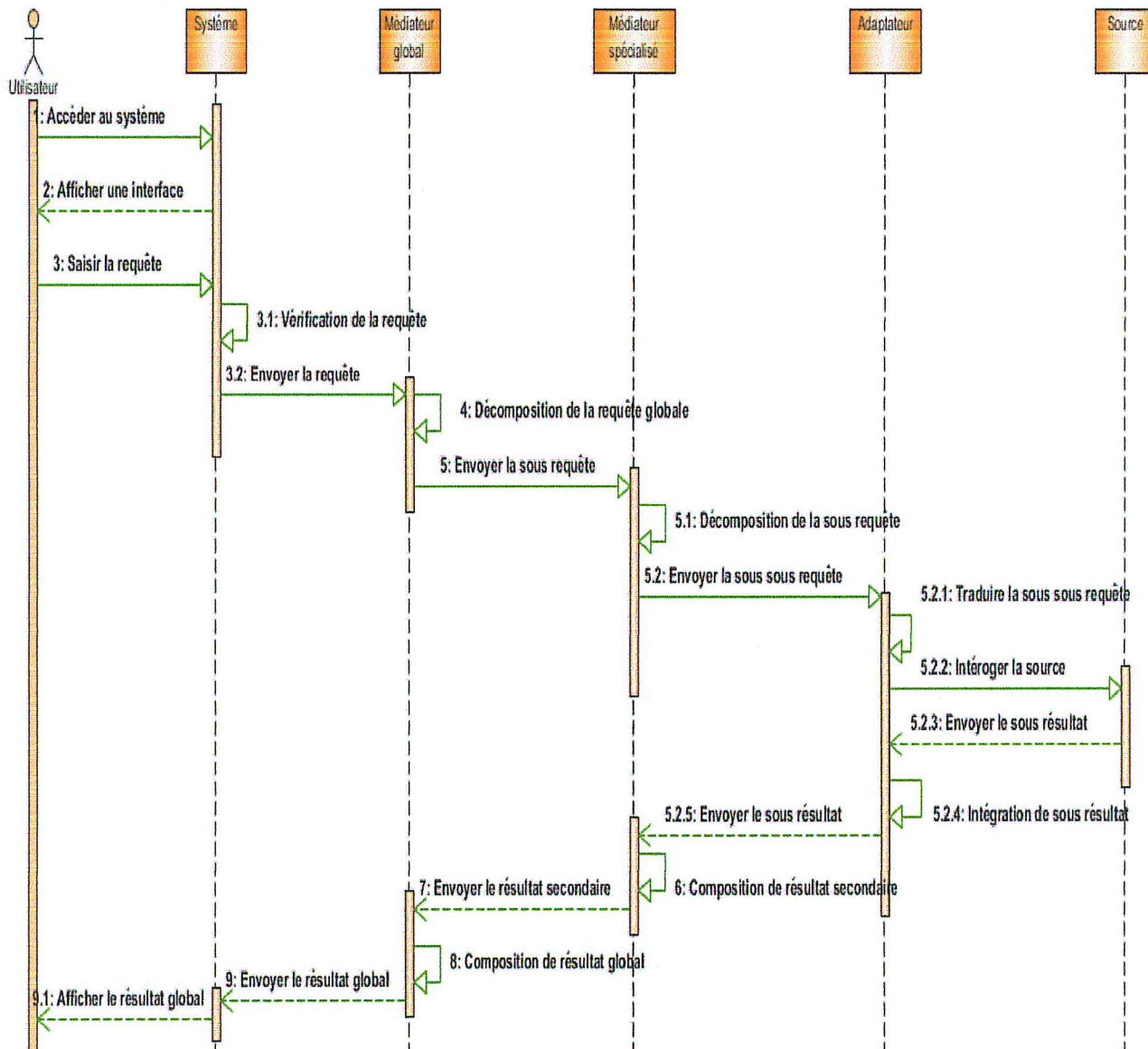


Figure 26: Diagramme de séquence de lancement de la requête globale.

✓ Diagramme de séquence de choix d'indicateurs

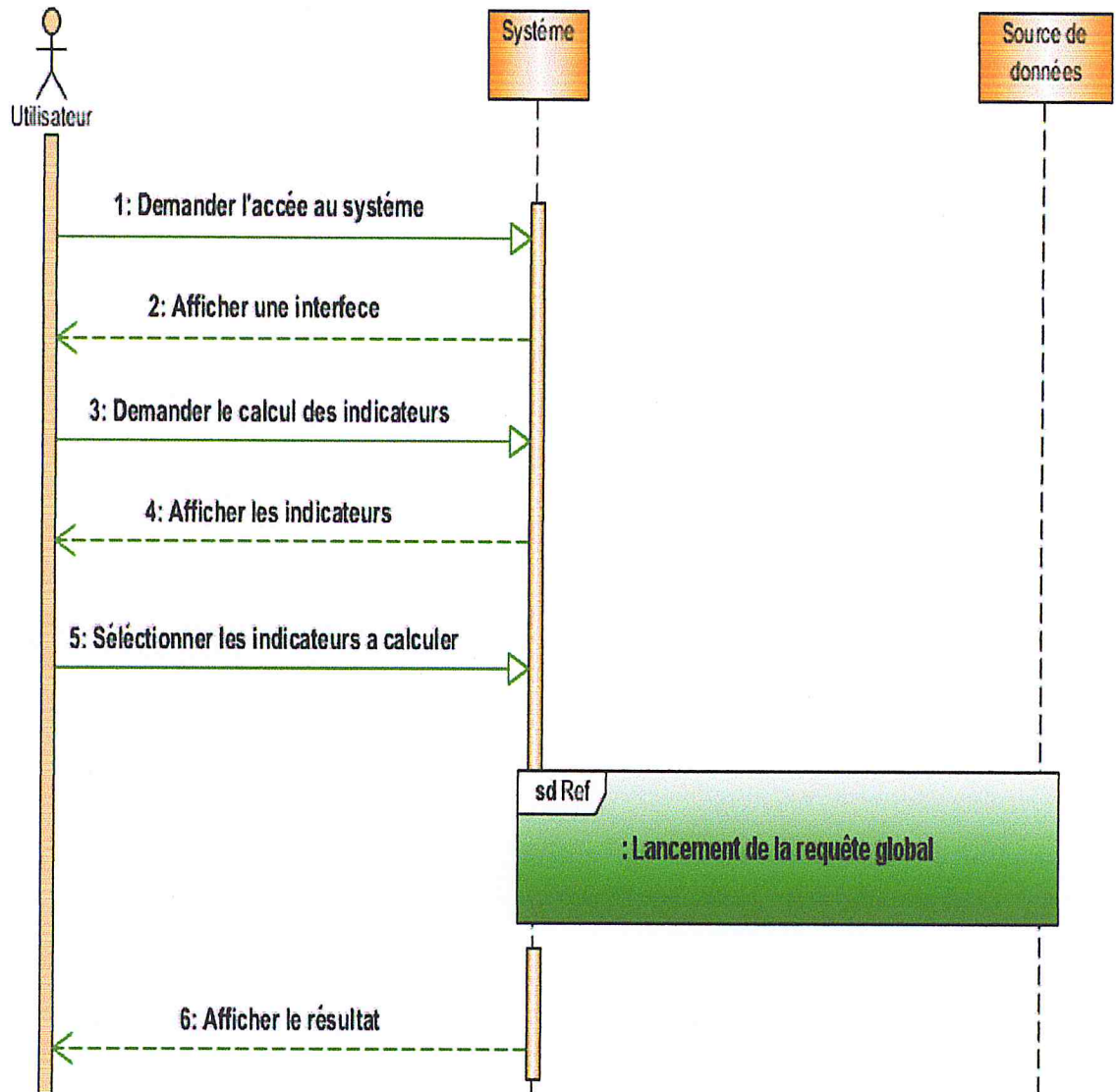


Figure 27:Diagramme de séquence de choix d'indicateurs.

8.2 Diagramme d'activité

Les diagrammes d'activité sont particulièrement adaptés à La modélisation du cheminement de flots de contrôle et de flots de données.

✓ Diagramme d'activité pour le choix d'indicateurs

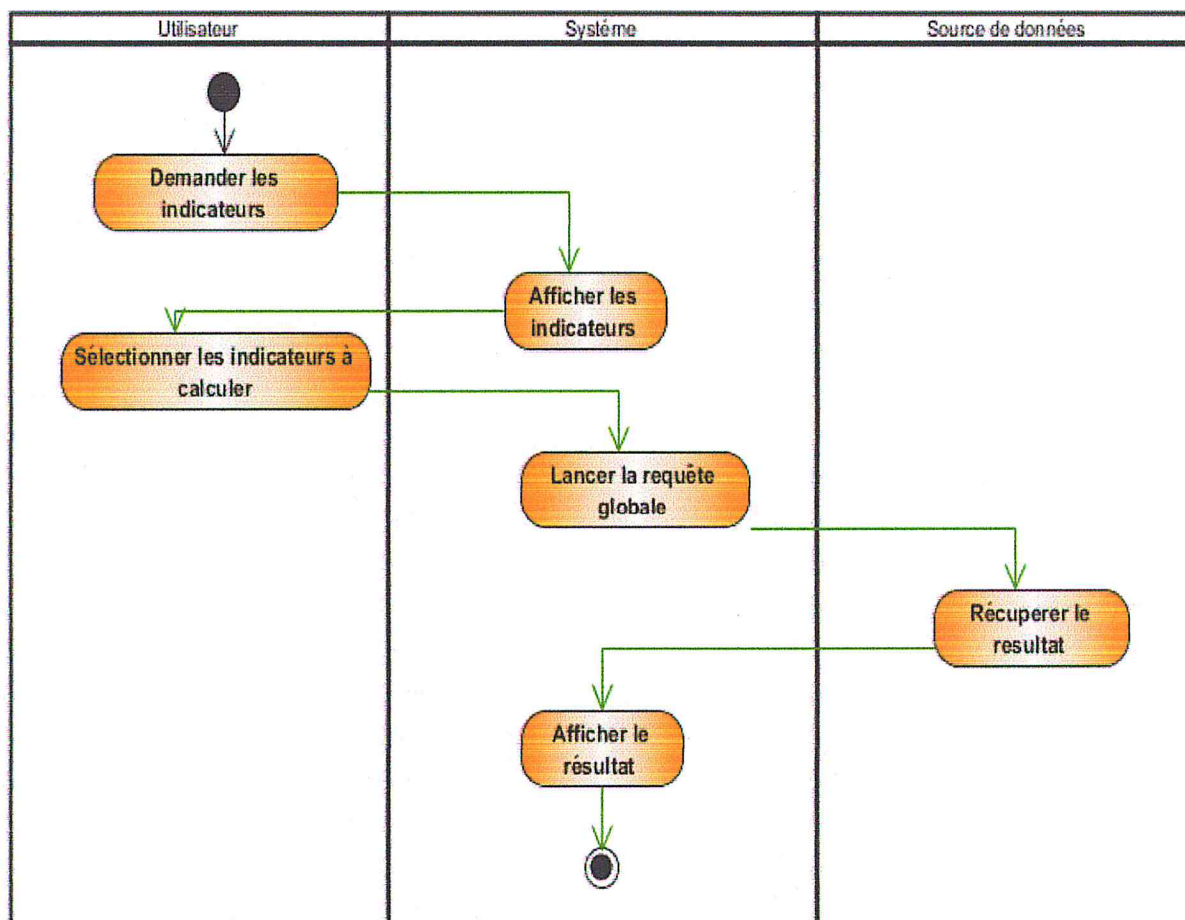


Figure 28 : diagramme d'activité pour le choix d'indicateurs.

✓ Diagramme d'activité pour le lancement de la requête globale

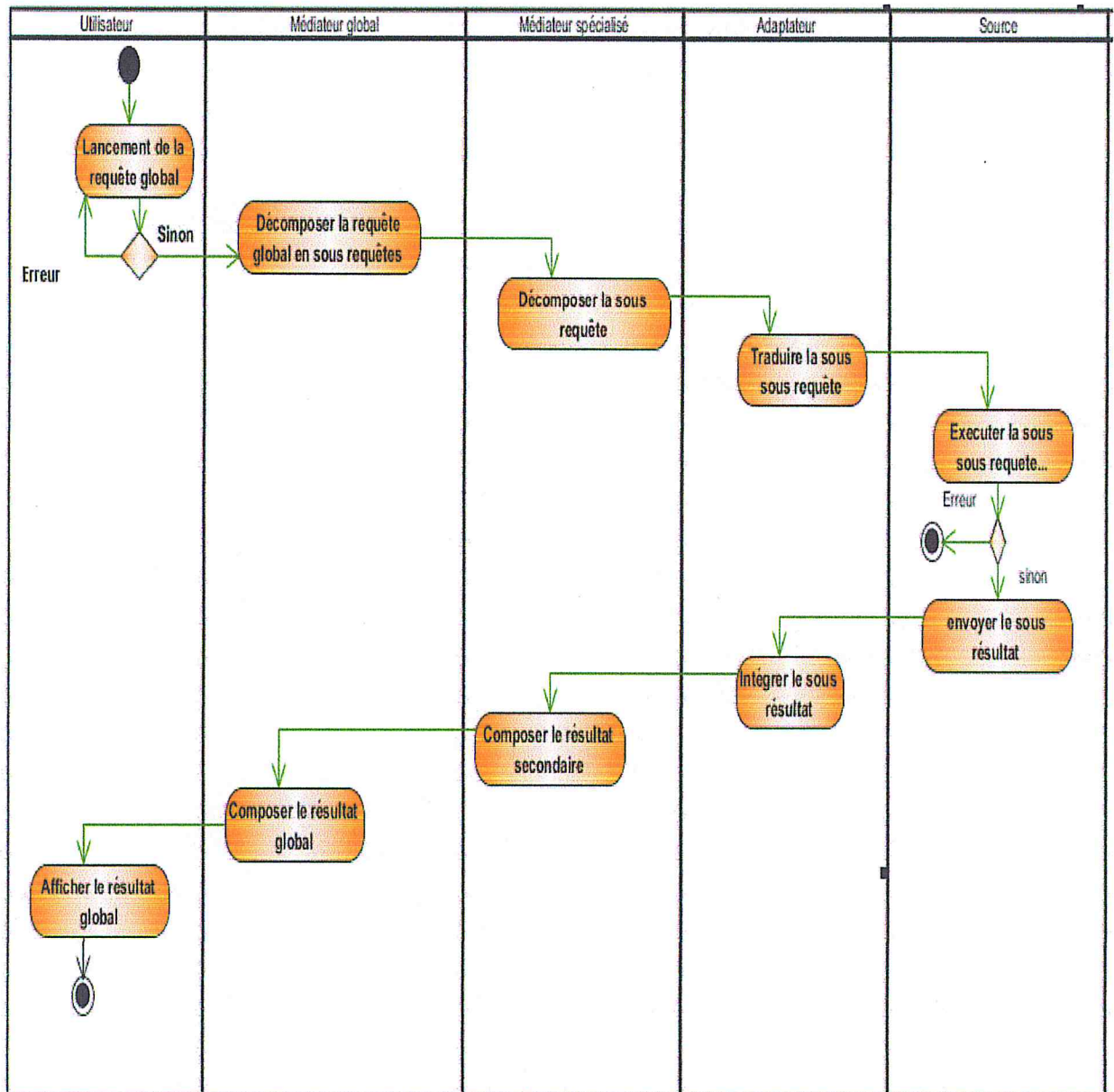


Figure 29 : diagramme d'activité pour le lancement de la requête globale.

✓ Diagramme d'activité pour la traduction de la sous requête de métadonnées

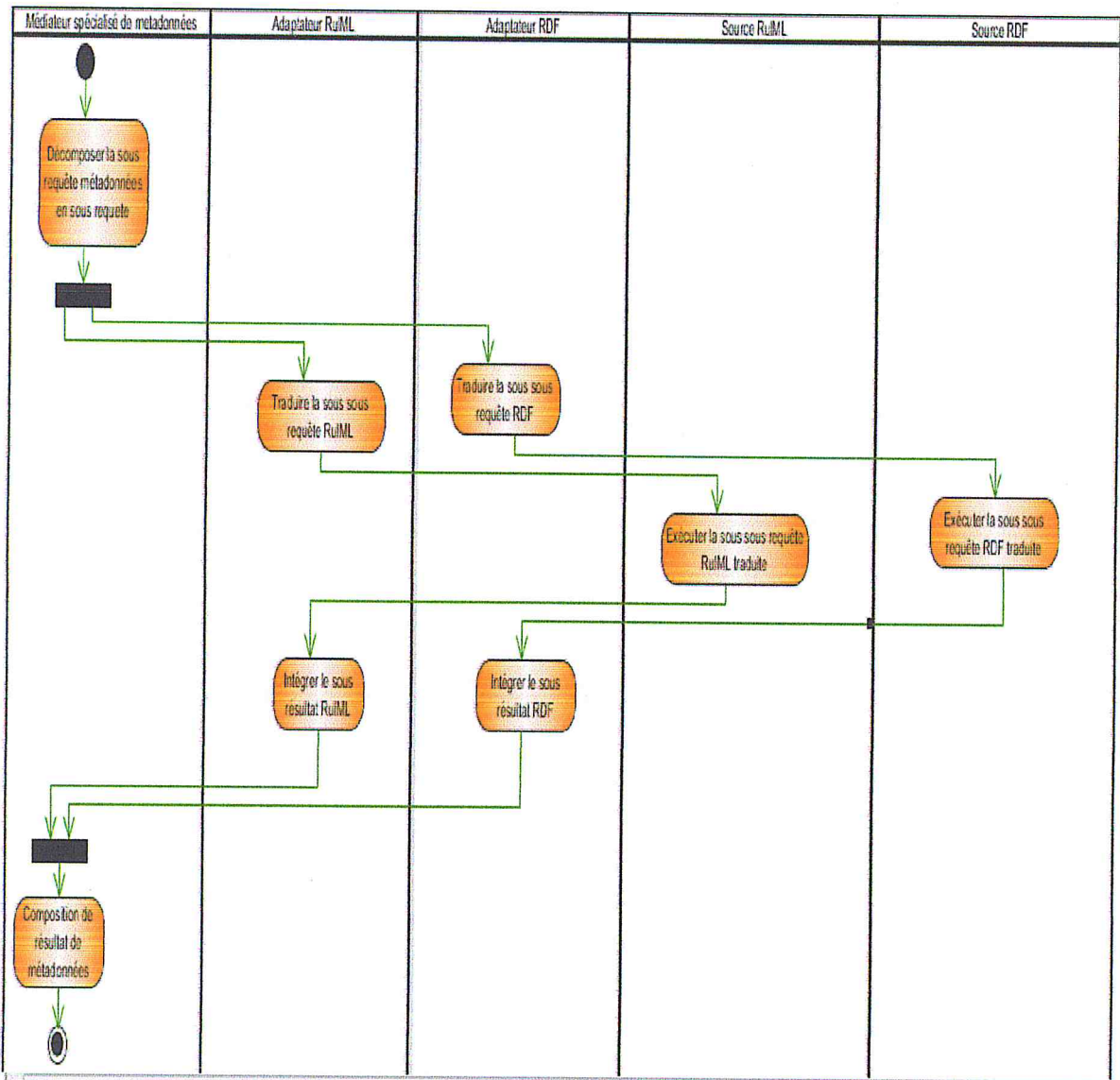


Figure 30: Diagramme d'activité pour la traduction de la sous requête métadonnées

✓ Diagramme d'activité pour la traduction de la sous requête données

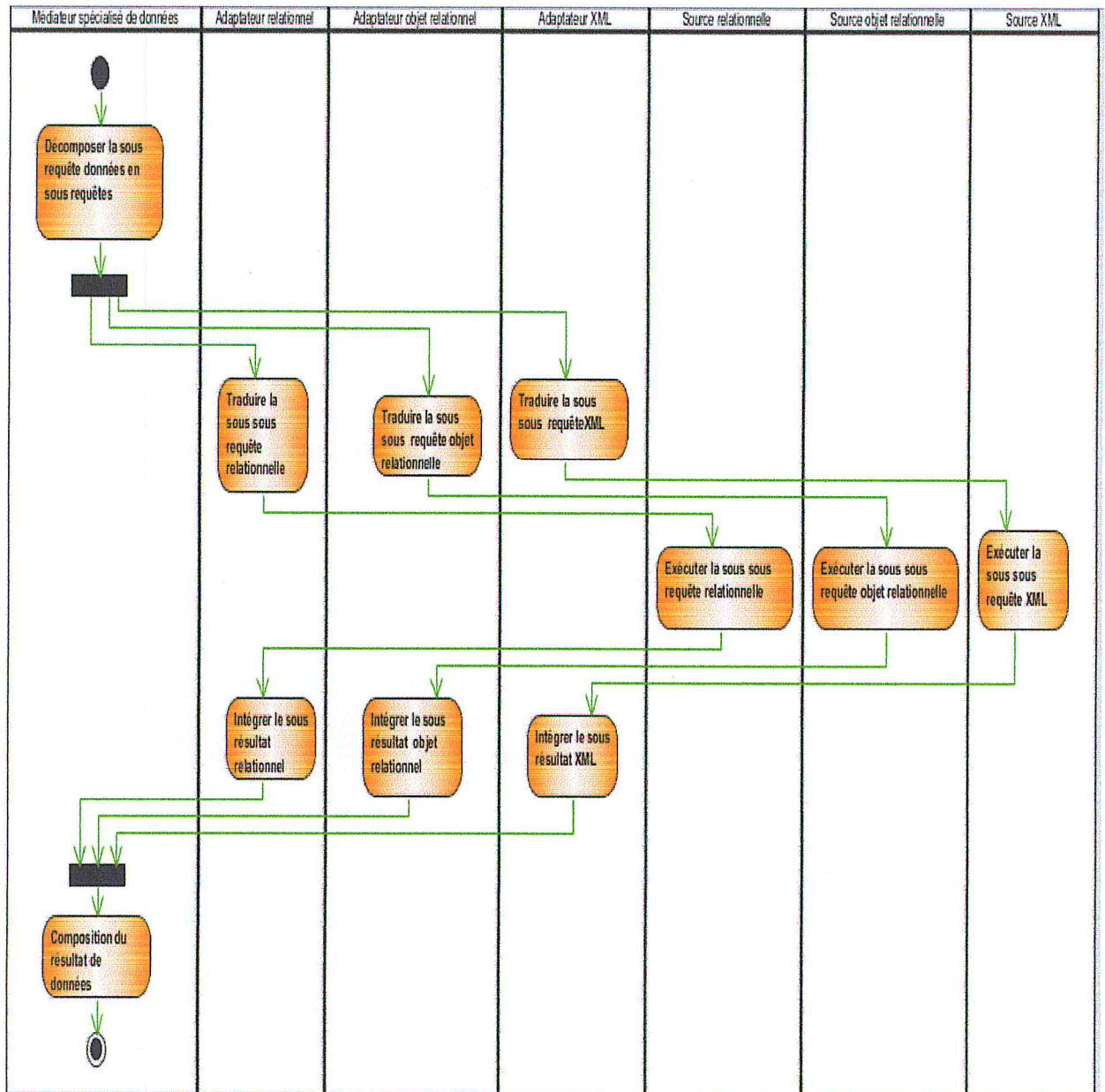


Figure 31: Diagramme d'activité pour la traduction de la sous requête de données

✓ Diagramme d'activité traduction de la sous requête connaissance

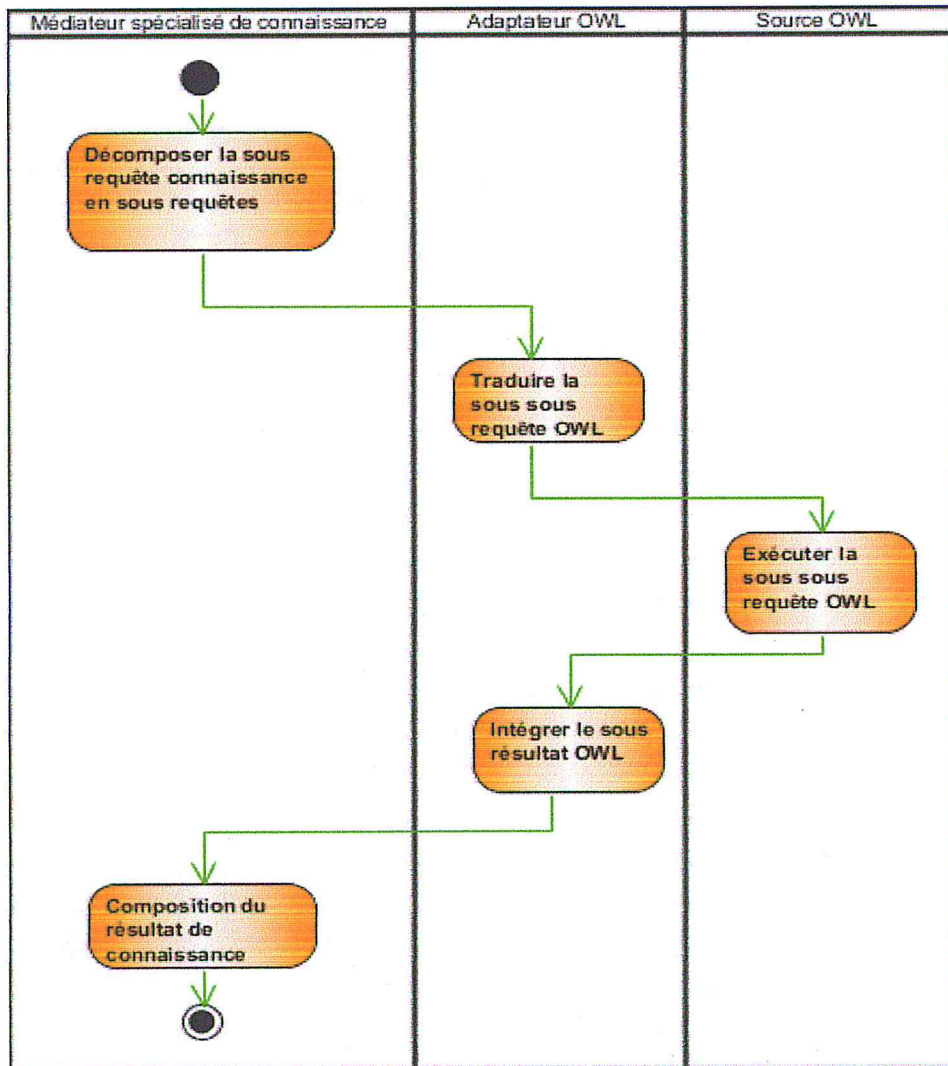


Figure 32: diagramme d'activité pour la traduction de la sous requête connaissance.

9. Conception

La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et au système d'exploitation. Elle détermine les principales interfaces et les transcrit à l'aide d'une notation commune. Elle constitue un point de départ à l'implémentation :

- ✓ elle décompose le travail d'implémentation en sous-système;
- ✓ elle crée une abstraction transparente de l'implémentation.

9.1 diagramme de classe

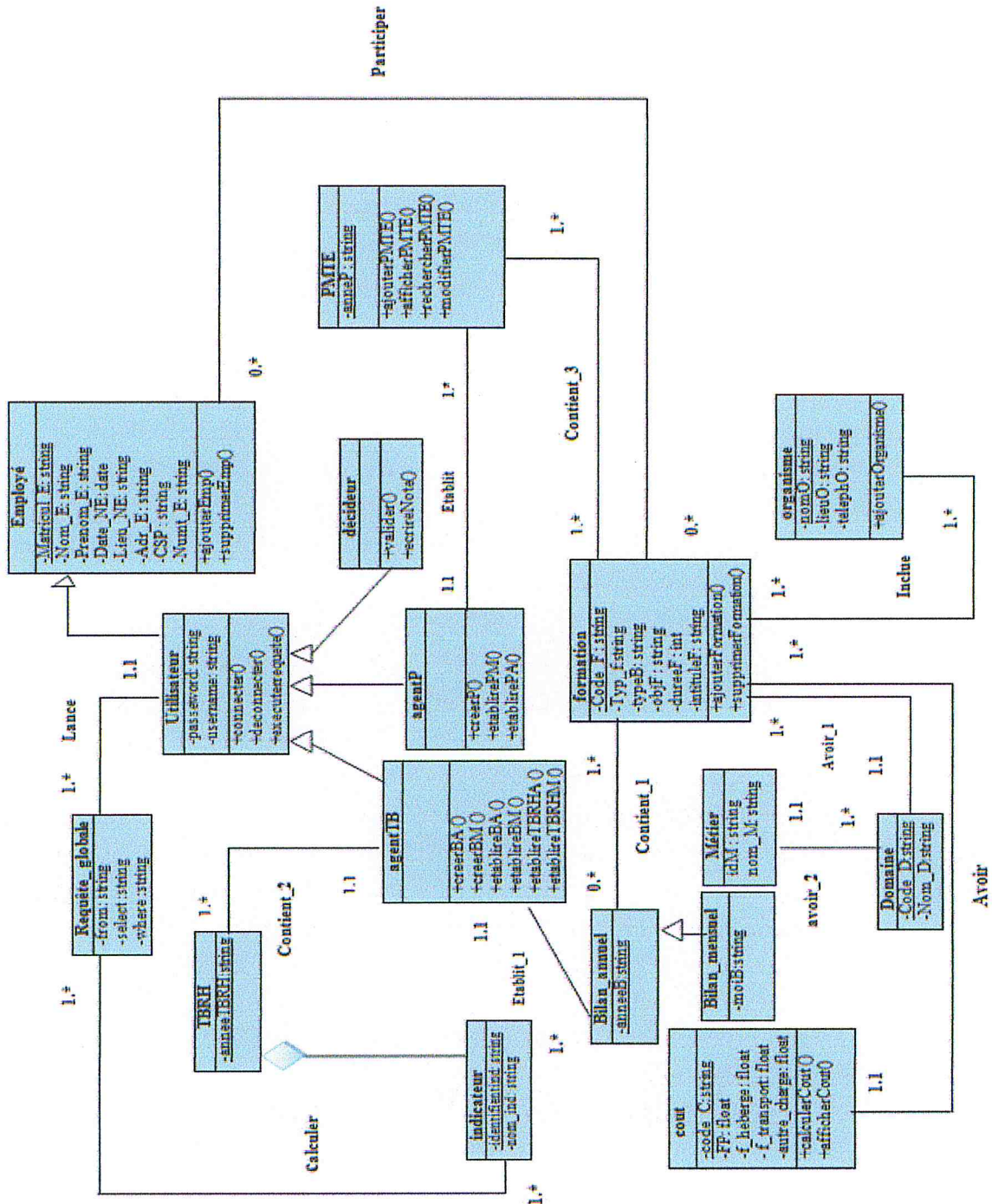


Figure 33: diagramme de classes de système.

Chapitre III: Analyse et conception

➤ Dictionnaire de données

CLASSE	ATTRIBUT	TYPE	DESCRIPTION
Formation	Code_F	VARCHAR	Code de la formation (identifiant)
	Type_F	VARCHAR	Type de formation
	TypeB	VARCHAR	Type de besoin de formation
	Obj_F	VARCHAR	Objectif de la formation
	Intitule_F	VARCHAR	Intitulé de la formation
	Dure_F	VARCHAR	Duré de la formation
PMTE	Annee_P	VARCHAR	Année du plan
Organisme	Nom_O	VARCHAR	Nom de l'organisme
	Lieu_O	VARCHAR	Lieu de l'organisme
	Teleph_O	VARCHAR	Numéro de téléphone de l'organisme
Cout	Code_C	VARCHAR	Identifiant de Cout
	FP	FLOAT	Frais pédagogique pour chaque formation
	F_heberge	FLOAT	Frais hébergement pour chaque formation
	F_transport	FLOAT	Frais de transport pour chaque formation
	Autre_charge	FLOAT	Autre charge (autre frais personnalisé)
Employé	Matricule_E	VARCHAR	Matricule de l'employé
	Nom_E	VARCHAR	Nom de l'employé
	Prenom_E	VARCHAR	Prénom de l'employé
	Adr_E	VARCHAR	Adresse de l'employé
	CSP	VARCHAR	Catégorie sociaux professionnel
	Numt_E	VARCHAR	Numéro de téléphone de l'employé
TBRH	AnneeTBRH	VARCHAR	Année de tableau de bord
Métier	IdM	VARCHAR	Code de métier (identifiant)
	NomM	VARCHAR	Nom de métier
Utilisateur	Password	VARCHAR	Mot de passe utilisateur
	Username	VARCHAR	Nom d'utilisateur
Indicateur	Identifiantind	VARCHAR	identifiant de l'indicateur
	Nom_ind	VARCHAR	Nom d'indicateur
Bilan annuel	AnneeB	VARCHAR	Année de bilan
Bilan mensuel	MoiB	VARCHAR	Moi de bilan
Domaine	Code_D	VARCHAR	Code d'identification du domaine

Chapitre III: Analyse et conception

	Nom_D	VARCHAR	Nom de domaine
Requete_globale	From	VARCHAR	La clause concept
	Select	VARCHAR	La clause attribut
	Where	VARCHAR	La clause condition

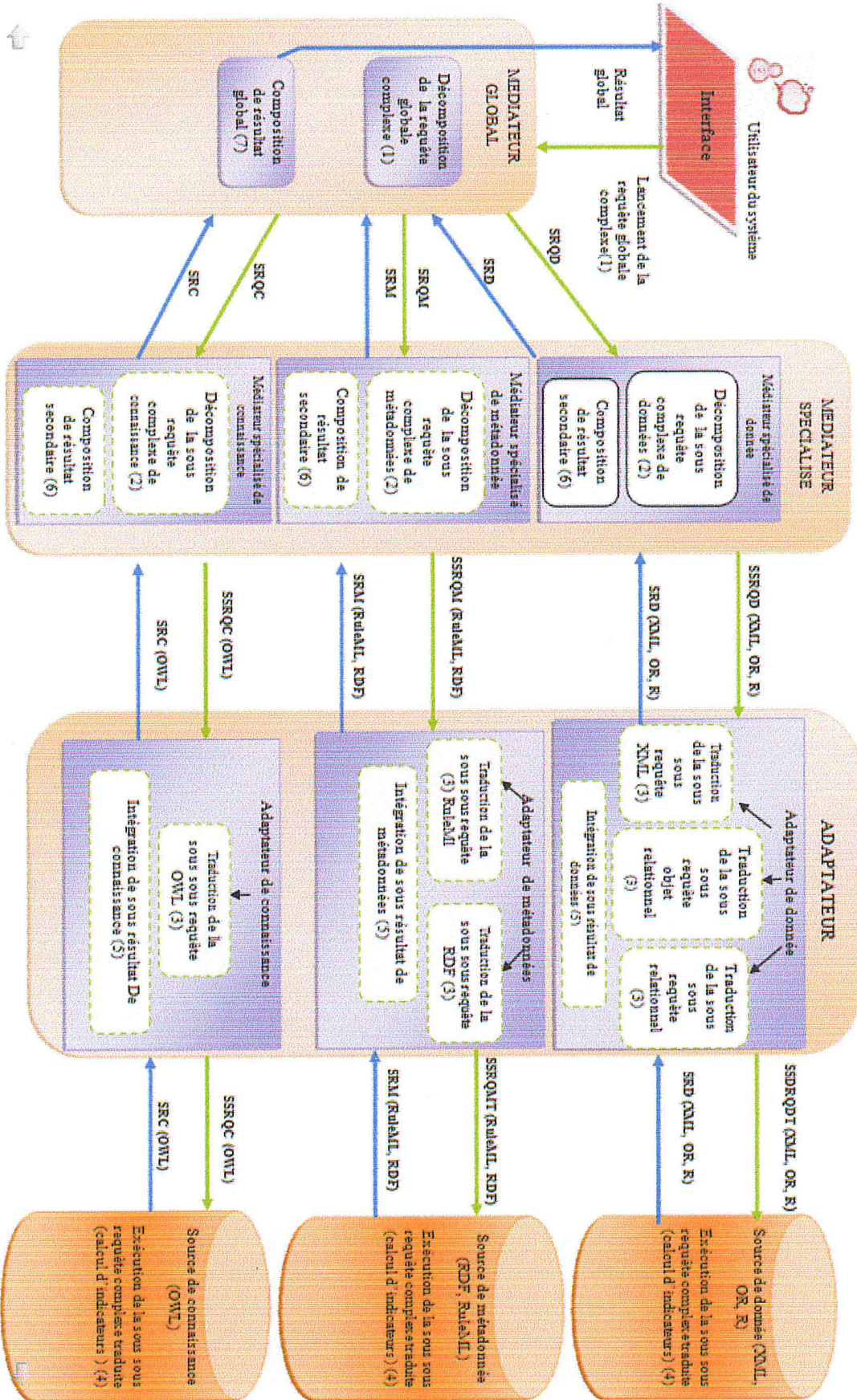
Tableau 17 : dictionnaire de données.

➤ Liste des méthodes

CLASSES	METHODE	DESCRIPTION
Formation	Ajouter Formation ()	Ajouter une formation donnée
	supprimerFormation ()	Supprimer une formation donnée
PMTE (Plan a moyen terme)	AjouterPMTE ()	Ajouter un plan a moyen terme
	afficherPlan ()	Afficher un plan
	rechercherPlan ()	Rechercher un plan
	modifierPlan ()	Modifier un plan
Organisme	AjouterOrganisme ()	Ajouter un organisme
Cout	CalculerCout ()	Calculer r un cout
	afficherCout ()	Afficher un cout
Employer	ajouterEMP ()	Ajouter un employeur
	supprimerEMP ()	Supprimer un employeur
Utilisateur	Connecter ()	Se connecter au système
	Deconnecter ()	Se déconnecter au système
	Executerrequete ()	Composer et exécuter la requête globale
AgentBT	CreerBA ()	créer un bilan annuel
	CreerBM ()	créer un bilan mensuel
	etablireBA ()	Etablir un bilan annuel
	etablireBM ()	Etablir un bilan mensuel
	etablireTBRHA ()	Etablir un tableau de bord annuel
	etablireTBRHM ()	Etablir un tableau de bord mensuel
AgentP	creerP ()	Créer un plan
	etablirePM ()	Etablir un plan mensuel
	etablirePA ()	Etablir un plan annuel
Décideur	Valider ()	Valider plan, bilan
	EcrireNote ()	Ecrire des notes pour modifier ou supprimer formation

Tableau18 : liste des méthodes.

9.2 Architecture détaillée de la solution proposée



9.3 Description des modules de l'architecture proposée

Au niveau de l'interface, l'utilisateur choisit les indicateurs à calculer qui est en langage naturel, et lance la requête globale

Module 1: Décomposition de la requête globale

La requête globale est une liste d'indicateurs à calculer qui sera envoyée au médiateur global pour une traduction en langage de schéma global qui est l'OWL, en construisant cinq listes (attributsG, conceptsG, conditionsG, conceptsConditionsG, valConditions), ensuite elle sera décomposée en sous requêtes selon les types de médiateurs spécialisées et pour cela en récupère pour chaque attribut de la liste attributsG son nom d'ontologie qui sera (OSD pour données, OSM pour métadonnées et OSC pour connaissances) et par rapport à ce résultat en extrait trois listes d'attributs (attributsD, attributsM et attributsC), et le même travail pour conceptsG, conditionsG, conceptsConditionsG.

✓ Pseudo algorithme de la décomposition de la requête globale

Soit :

En entrée

- RQG: la requête globale

En sortie

- {attributsD, conceptsD, conditionsD, conceptsConditionsD, valConditionsD} : sous requête données
- {attributsM, conceptsM, conditionsM, conceptsConditionsM, valConditionsM} : sous requête métadonnées
- {attributsC, conceptsC, conditionsC, conceptsConditionsC, valConditionsC} : sous requête connaissances

Début

{attributsG, conceptsG, conditionsG, conceptsConditionsG, valConditionsG} = traduire(RQG);

Tant que attributsG.taille != 0 début

Si nomOnto = OSD

Alors début

attributsD, conceptsD, conditionsD, conceptsConditionD, valConditionsD} = DecomposerReqG(
attributsG, conceptsG, conditionsG, conceptsConditionG, valConditionsG);

```
Si nomOnto=OSM
attributsM,conceptsM,conditionsM,conceptsConditionM,valConditionsM}=
DecomposerReqG(
attributsG,conceptsG,conditionsG, conceptsConditionG,valConditionsG);
Si nomOnto=OSC
attributsC, conceptsC, conditionsC, conceptsConditionC, valConditionsC}=
DecomposerReqG( attributsG,conceptsG,conditionsG,
conceptsConditionG,valConditionsG);
Fin
Fin
Fin
```

Module 2: décomposition de la sous requête

Après la décomposition de la requête globale, les sous requêtes (les listes) seront envoyées aux médiateurs spécialisés (données, métadonnées, connaissances) pour une autre décomposition en sous sous requêtes selon les types des sources existantes, et pour cela en récupère le nom d'ontologie locale pour chaque attribut, concept, condition, conceptCondition, qui sera ;

- ✓ R, Or, et XML pour les données;
- ✓ RDF, RML pour les métadonnées;
- ✓ OWL pour les connaissances.

A ce niveau en vérifier si un attribut qui est dans plus qu'une source en garde un seul et les autres seront supprimé pour éviter les doubles

- ✓ Pseudo algorithme de la décomposition de la sous requêtes

Soit:

En entrée

- {attributsD,conceptsD,conditionsD, conceptsConditionD,valConditionsD }:
sous requêtes données
- {attributsM,conceptsM,conditionsM, conceptsConditionM ,valConditionsM}:
sous requêtes métadonnées
- attributsC,conceptsC,conditionsC, conceptsConditionC,valConditionsC: sous
requêtes connaissances début

- {attributsR,conceptsR,conditionsR, conceptsConditionR,valConditionsR}: sous requête Relationnelle
- {attributsOR,conceptsOR,conditionsR,, conceptsConditionOR ,valConditionsR}: sous requête Objet relationnelle
- {attributsXML,conceptsXML,conditionsXML, conceptsConditionXML,valConditionsXML}: sous requête XML

- {attributsRDF,conceptsRDF,conditionsRDF, conceptsConditionRDF,valConditionsRDF } : sous requête RDF
- {attributsRML,conceptsRML,conditionsRML, conceptsConditionRML,valConditionsRML}: sous requête RuleML
- {attributsOWL,conceptsOWL,conditionsOWL,conceptsConditionOWL, valConditionsOWL}: sous requête OWL

Début

Tant que attributsD.taille != 0 début

Si ontoLoc=XML

Alors

{attributsXML,conceptsXML,conditionsXML, conceptsConditionXML,valConditionsXML}=DecomposerReqD(attributsD,conceptsD,conditionsD,valConditionsD);

Si ontoLoc=R

{attributsR,conceptsR,conditionsR, conceptsConditionR,valConditionsR}=DecomposerReqD(attributsD,conceptsD,conditionsD, conceptsConditionD,valConditionsD);

Si ontoLoc=OR

{attributsOR,conceptsOR,conditionsOR,conceptsConditionOR,valConditionsOR}=DecomposerReqG(attributsD,conceptsD,conditionsD, conceptsConditionD,valConditionsD); fin

EliminerDouble(attributsR ,conditionsR, attributsXML ,conditionsXML, attributsOR ,conditionsOR);

Tant que attributsM.taille != 0 debut

Si ontoLoc=RML

Alors

{attributsRML,conceptsRML,conditionsRML,conceptsConditionRML,valConditionsRML}=DecomposerReqM(attributsM,conceptsM,conditionsM, conceptsConditionsM,valConditionsM);

Si ontoLoc=RDF

Alors

{attributsRDF,conceptsRDF,conditionsRDF,conceptsConditionRDF,valConditionsRDF}=DecomposerReqM(attributsM,conceptsM,conditionsM,valConditionsM);

Fin

```
EliminerDouble(attributsRML ,conditionsRML, attributsRDF ,conditionsRDF);
```

```
Tant que attributsC.taille  $\neq$  0 début
```

```
  Si ontoLoc=OWL
```

```
  Alors {attributsOWL, conceptsOWL, conditionsOWL,  
  conceptsConditionOWL,valConditionsOWL}=DecomposerReqC(attributsC,concepts  
  C,  
  conditionsC, conceptsConditionC,valConditionsC);
```

```
  Fin
```

```
Fin
```

Module 3: la traduction de la sous sous requête

Les sous sous requêtes résultantes de la décomposition seront envoyées chaque une au adaptateur adéquat, par la suite, l'adaptateur traduit la sous sous requête au langage d'interrogation qui correspond au type de la source, ou pour chaque type de sources a un langage d'interrogation;

- ✓ SQL pour la source relationnelle;
 - ✓ SQL99 pour la source objet relationnelle;
 - ✓ XQuery pour XML et RuleML;
 - ✓ SPARQL pour RDF et OWL.
-
- ✓ Pseudo algorithme de la traduction de la sous sous requête

Soit:

En entrée

- {attributsXML,conceptsXML,conditionsXML,conceptsConditionXML,
valConditionsXML}: sous sous requête XML
- {attributsR,conceptsR,conditionsR, conceptsConditionR,valConditionsR}: sous
sous requête Relationnelle
- {attributsOR,conceptsOR,conditionsR,, conceptsConditionOR
,valConditionsR}: sous sous requête Objet relationnelle
- {attributsRDF,conceptsRDF,conditionsRDF,
conceptsConditionRDF,valConditionsRDF } : sous sous requête RDF

- {attributsRML,conceptsRML,conditionsRML, conceptsConditionRML,valConditionsRML}: sous sous requête RuleML
- {attributsOWL,conceptsOWL,conditionsOWL,conceptsConditionOWL, valConditionsOWL}: sous sous requete OWL

En sortie

- {attributsXML,conceptsXML,conditionsXML, conceptsConditionXML,valConditionsXML}: sous sous requête XML traduite
- {attributsR,conceptsR,conditionsR, conceptsConditionR,valConditionsR}: sous sous requête Relationnelle traduite
- {attributsOR,conceptsOR,conditionsR,, conceptsConditionOR ,valConditionsR}: sous sous requête Objet relationnelle traduite
- {attributsRDF,conceptsRDF,conditionsRDF, conceptsConditionRDF,valConditionsRDF } : sous sous requête RDF traduite
- {attributsRML,conceptsRML,conditionsRML, conceptsConditionRML,valConditionsRML}: sous sous requête RuleML traduite
- {attributsOWL,conceptsOWL,conditionsOWL,conceptsConditionOWL, valConditionsOWL}: sous sous requete OWL traduite

Début

{attributsXML,conceptsXML,conditionsXML,conceptsConditionXML,valConditionsL}=TraduireXML (attributsXML,conceptsXML,conditionsXML,conceptsConditionXML, valConditionsXML);

{attributsR,conceptsR,conditionsR,conceptsConditionR,valConditionsR}=TraduireR (attributsR,conceptsR,conditionsR,conceptsConditionR,valConditionsR);

{attributsOR,conceptsOR,conditionsOR,conceptsConditionOR,valConditionsOR}=TraduireOR (attributsOR,conceptsOR,conditionsOR,conceptsConditionOR, valConditionsOR);

{attributsRML,conceptsRML,conditionsRML,conceptsConditionRML, valConditionsRML}=TraduireRML(attributsRML,conceptsRML,conditionsRML, conceptsConditionRML, valConditionsRML);

```
{attributsRDF,conceptsRDF,conditionsRDF,conceptsConditionRDF,valConditionsRDF}=
TraduireRDF (attributsRDF,conceptsRDF,conditionsRDF,conceptsConditionRDF,
valConditionsRDF);

{attributsOWL,conceptsOWL,conditionsOWL,conceptsConditionOWL,valConditionsOWL}
=TraduireOWL (attributsOWL,conceptsOWL,conditionsOWL,conceptsConditionOWL,
valConditionsOWL);

Fin
```

Module 4 : exécution de la sous sous requête traduite

Chaque sous sous requête traduite sera exécutée au niveau de la source on prend en compte des conditions qui existe, le sous résultat retourné de l'exécution qui est l'indicateur F calculée va être dans le model de la source, et sera envoyé au adaptateur adéquat.

✓ Pseudo algorithme de l'exécution de la sous sous requête traduite

Soit:

En entrée

- {attributsXML,conceptsXML,conditionsXML,conceptsConditionXML,valConditionsXML}: sous sous requête XML traduite
- {attributsR,conceptsR,conditionsR, conceptsConditionR,valConditionsR}: sous sous requête Relationnelle traduite
- {attributsOR,conceptsOR,conditionsR,, conceptsConditionOR ,valConditionsR}: sous sous requête Objet relationnelle traduite
- {attributsRDF,conceptsRDF,conditionsRDF, conceptsConditionRDF,valConditionsRDF } : sous sous requête RDF traduite
- {attributsRML,conceptsRML,conditionsRML, conceptsConditionRML,valConditionsRML}: sous sous requête RuleML traduite
- {attributsOWL,conceptsOWL,conditionsOWL,conceptsConditionOWL, valConditionsOWL}: sous sous requête OWL traduite

En sortie

- SRXML: sous résultat XML
- SRR: sous résultat Relationnel
- SROR: sous résultat Objet relationnel
- SRRML: sous résultat RDF
- SRRDF: sous résultat RuleML

- **SROWL**: sous résultat OWL

Debut

**SRXML=ExecuterReXML (attributsXML,conceptsXML,conditionsXML,
conceptsConditionXML,valConditionsXML);**

SRR=ExecuterReq(attributsR,conceptsR,conditionsR,conceptsConditionR,valConditionsR);

**SROR= ExecuterReqOR (attributsOR,conceptsOR,conditionsR,, conceptsConditionOR
,valConditionsR);**

**SRRML=ExecuterReqRML(attributsRML,conceptsRML,conditionsRML,
conceptsConditionRML,valConditionsRML);**

**SRRDF= ExecuterReqRDF(attributsRDF,conceptsRDF,conditionsRDF,
conceptsConditionRDF,valConditionsRDF);**

**SROWL= ExecuterReqOWL(attributsOWL,conceptsOWL,conditionsOWL,
conceptsConditionOWL,valConditionsOWL);**

Fin

Module 5: intégration des sous résultats

Les sous résultats seront envoyés aux adaptateurs pour une traduction dans un même format de représentation qui est dans ce cas un tableau, ensuite ils vont être envoyés aux médiateurs spécialisés.

✓ **Pseudo algorithme de l'intégration des sous résultats**

Soit

En entrée

- **SRXML**: sous résultat XML
- **SRR**: sous résultat Relationnel
- **SROR**: sous résultat Objet relationnel
- **SRRML**: sous résultat RDF
- **SRRDF**: sous résultat RuleML
- **SROWL**: sous résultat OWL

En sortie

- **TabSRXML**: tableau qui contient le sous résultat XML
- **TabSRR**: tableau qui contient le sous résultat relationnel
- **TabSROR**: tableau qui contient le sous résultat objet relationnel
- **TabSRRML**: tableau qui contient le sous résultat RuleML
- **TabSRRDF**: tableau qui contient le sous résultat RDF
- **TabSROWL**: tableau qui contient le sous résultat OWL

Début

TabSRXML= Transformer (**SRXML**);
TabSRR= Transformer (**SRR**);
TabSROR= Transformer (**SROR**);
TabSRRML= Transformer (**SRRML**);
TabSRRDF= Transformer (**SRRDF**);
TabSROWL= Transformer (**SROWL**);

Fin

Module 6: composition des résultats secondaire

Après l'envoi des tableaux de sous résultats chaque une a son médiateur spécialisé, ils vont être composés dans un seul tableau dans chaque médiateur (résultat donnée, résultat métadonnées, résultat connaissances).

✓ **Pseudo algorithme de la composition secondaire**

Soit:

En entrée

- **TabSRXML**: tableau qui contient le sous résultat XML
- **TabSRR**: tableau qui contient le sous résultat relationnel
- **TabSROR**: tableau qui contient le sous résultat objet relationnel
- **TabSRRML**: tableau qui contient le sous résultat RuleML
- **TabSRRDF**: tableau qui contient le sous résultat RDF
- **TabSROWL**: tableau qui contient le sous résultat OWL

En sortie

- **TabSRD**: tableau qui contient le sous résultat données
- **TabSRM**: tableau qui contient le sous résultat métadonnées
- **TabSRC**: tableau qui contient le sous résultat connaissances

Début

TabSRD=ComposerD (**TabSRXML**, **TabSRR**, **TabSROR**);

TabSRM=ComposerM (**TabSRRML**, **TabSRRDF**);

TabSRC= ComposerC (**TabSROWL**);

Fin

Module 7: composition du résultat global

A ce niveau, nous avons trois tableaux de résultat (données, métadonnées, connaissances), le médiateur global va retourner le résultat global qui est la composition des trois tableaux dans un seul tableau.

✓ **Pseudo algorithme de composition du résultat global**

Soit:

En entrée

- **TabSRD:** tableau qui contient le sous résultat données
- **TabSRM:** tableau qui contient le sous résultat métadonnées
- **TabSRC:** tableau qui contient le sous résultat connaissances

En sortie

- **TabRG:** tableau qui contient le résultat global

Début

TabRG=CompositionG (TabSRD, TabSRM, TabSRC);

Fin

Le résultat global sera affiché sur l'interface qui est l'ensemble d'indicateurs que l'utilisateur a choisi, organisés dans un tableau.

12. Conclusion

A l'issue de ce chapitre, nous avons pu, construire l'architecture générale sur laquelle repose notre solution, nous avons utilisé également le langage UML pour définir la partie conceptuelle de notre système, ainsi que la démarche d'information de l'organisme SONATRACH.

Dans le prochain chapitre nous allons définir la partie de réalisation de système avec une vue globale des différentes sources de données existés.

1. Introduction

Dans ce chapitre nous allons définir les différentes sources de donnée ainsi que les outils de développement choisis pour l'implémentation de notre système et le langage de programmation

2. Présentation des sources Il existe trois types de sources :

2.1 Source de données

2.1.1 Source relationnel

✓ BDDFORMATION

Il existe une base de données relationnelle non exploitable au niveau de département formation de la division production de SONATRACH implémentée dans le SGBDOR Oracle 10g, dont nous avons eu une partie de son schéma. A base de cette partie nous avons conçu une source relationnelle les figure si dessous représentent les différentes table de cette base de données relationnel BDDFORMATION

Column Name	Data Type	Nullable	Data Default	COLUMN ID
MATRICULE	VARCHAR2(8 BYTE)	Yes	(null)	1
NOM	VARCHAR2(60 BYTE)	Yes	(null)	2
SEXE	CHAR(1 BYTE)	Yes	'M'	3
CSP	VARCHAR2(20 BYTE)	Yes	(null)	4

Figure 34: representation de la table A_Agent

Column Name	Data Type	Nullable	Data Default	COLUMN ID
BES_REFBESOIN	VARCHAR2(20 BYTE)	Yes	(null)	1
BES_CORGANISM	VARCHAR2(60 BYTE)	Yes	(null)	2
BES_CMODULE	VARCHAR2(200 BY...	Yes	(null)	3
BES_DUREE	NUMBER	Yes	(null)	4
BES_DDEBSOUH	DATE	Yes	(null)	5
BES_DFINSOUH	DATE	Yes	(null)	6
BES_OBJECTIF	VARCHAR2(200 BY...	Yes	(null)	7
BES_CSTRUCT	VARCHAR2(25 BYTE)	Yes	(null)	8
BES_PLAN	VARCHAR2(1 BYTE)	Yes	0	9
BES_OBS	VARCHAR2(55 BYTE)	Yes	(null)	10
BES_MATRICULE	VARCHAR2(8 BYTE)	Yes	(null)	11
BES_DOMAIN	VARCHAR2(4 BYTE)	Yes	(null)	12

Figure 35: representation de la table A_BESOINSFORM

Chapitre IV : implémentation

Column Name	Data Type	Nullable	Data Default	COLUMN ID
CSP	VARCHAR2(20 BYTE)	Yes	(null)	1

Figure 36 : representation de la table A_CSP

Column Name	Data Type	Nullable	Data Default	COLUMN ID
FTR_CFORMAT...	VARCHAR2(8 BYTE)	Yes	(null)	1
FTR_CORGANISM	VARCHAR2(30 BYTE)	Yes	(null)	2
FTR_NOMPREN	VARCHAR2(60 BYTE)	Yes	(null)	3
FTR_FONCFOR...	VARCHAR2(60 BYTE)	Yes	(null)	4
FTR_MATRICUL...	VARCHAR2(8 BYTE)	Yes	(null)	5

Figure 37: representation de la table A_FORMATEUR

Column Name	Data Type	Nullable	Data Default	COLUMN ID
MOD_CODE	VARCHAR2(15 BYTE)	Yes	(null)	1
MOD_INTITULE	VARCHAR2(100 BY...)	Yes	(null)	2
MOD_DOMAINE	VARCHAR2(8 BYTE)	Yes	(null)	3

Figure 38: representation de la table A_MODULES

Column Name	Data Type	Nullable	Data Default	COLUMN ID
FEN_MATRICULE	VARCHAR2(8 BYTE)	Yes	(null)	1
FEN_RFORM	VARCHAR2(20 BYTE)	Yes	(null)	2
FEN_TYFFORE	VARCHAR2(8 BYTE)	Yes	(null)	3
FEN_DDEBREALFE	DATE	Yes	(null)	4
FEN_DFINREALFE	DATE	Yes	(null)	5
FEN_LIEUFORM	VARCHAR2(45 BYTE)	Yes	(null)	6
FEN_DIPLOME	VARCHAR2(20 BYTE)	Yes	'non'	7
FEN_OBSERV	VARCHAR2(100 BY...)	Yes	(null)	8
FEN_DUREE	NUMBER	Yes	(null)	9
FEN_CLANGUE	VARCHAR2(8 BYTE)	Yes	(null)	10
FEN_ALTERNEE	VARCHAR2(1 BYTE)	Yes	0	11
FEN_CEMPLOYE...	VARCHAR2(8 BYTE)	Yes	'SHDP'	12
FEN_PREVISION	VARCHAR2(1 BYTE)	Yes	0	13
FEN_MODULE	VARCHAR2(15 BYTE)	Yes	(null)	14

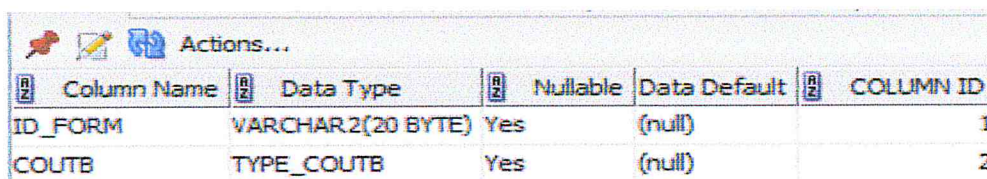
Figure 39: representation de la table A_TBRHFORM

Chapitre IV : implémentation

2.1.2 Source objet relationnelle

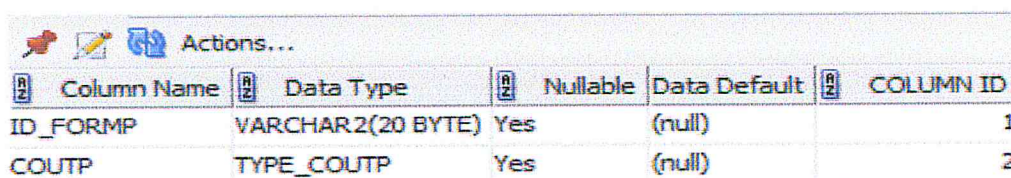
Il existe une base de données objet- relationnelle COSTFORMAT est implémentée dans le SGBDOR Oracle 10g, avec une définition de deux types complexe TYPE_COUTB et TYPE_COUTP.

Nous allons présenter par la suite les tables de type objet :



Column Name	Data Type	Nullable	Data Default	COLUMN ID
ID_FORM	VARCHAR2(20 BYTE)	Yes	(null)	1
COUTB	TYPE_COUTB	Yes	(null)	2

Figure 40 : representation de la table C_FORMATIONBILAN



Column Name	Data Type	Nullable	Data Default	COLUMN ID
ID_FORMP	VARCHAR2(20 BYTE)	Yes	(null)	1
COUTP	TYPE_COUTP	Yes	(null)	2

Figure 41 : representation de la table C_FORMATIONPLAN

2.1.3 Les sources XML

Dans notre travail nous somme intéressé aux différentes sources de données (**bilan** et **plan**) de département formation, nous allons présenter par la suite ces sources sous format XML.

✓ Présentation du plan

```
<?xmlversion="1.0"encoding="UTF-8"?>
<xs:schemaxmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
<xs:complexTypename="type_formé">
<xs:sequence>
<xs:elementname="matricule_P"type="xs:string"/>
<xs:elementname="structure_P"nillable="true"type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexTypename="type_formation_P">
<xs:sequence>
<xs:elementminOccurs="1"name="forme_P"type="type_forme"/>
<xs:elementname="type_besoin_P"type="xs:string"/>
<xs:elementname="organisme_P"type="xs:string"/>
<xs:elementname="type_formation_P"type="xs:string"/>
</xs:sequence>
<xs:attributename="identifiant_P"type="xs:string"/>
<xs:attributename="intitule_action_P"type="xs:string"/>
<xs:attributename="objectif_formation_P"type="xs:string"/>
<xs:attributename="metier_P"type="xs:string"/>
<xs:attributename="duree_P"type="xs:integer"/>
</xs:complexType>
<xs:complexTypename="type_plan">
<xs:sequence>
<xs:elementname="formation_P"type="type_formation_P"maxOccurs="unbounded"/>
</xs:sequence>
<xs:attributename="annee_P"type="xs:integer"/>
</xs:complexType>
<xs:elementabstract="false"name="Plan"type="type_plan"/>
</xs:schema>
```


✓ **Presentation du bilan**

```
<?xmlversion="1.0encoding="UTF-8"?>
<xs:schemaxmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
<xs:complexTypename="type_forme_B">
<xs:sequence>
<xs:elementname="matricule_B"type="xs:string"/>
<xs:elementname="structure_B"type="xs:string"nillable="true"/>
</xs:sequence>
</xs:complexType>
<xs:complexTypename="type_formation_B">
<xs:sequence>
<xs:elementname="forme_B"type="type_forme_B"/>
<xs:elementname="type_besoin_B"type="xs:string"nillable="false"/>
<xs:elementname="organisme_B"type="xs:string"/>
<xs:elementname="type_formation_B"type="xs:string"/>
</xs:sequence>
<xs:attributename="id_formation_B"type="xs:string"use="optional"/>
<xs:attributename="intitule_action_B"type="xs:string"/>
<xs:attributename="metier_B"type="xs:string"/>
<xs:attributename="mois_B"type="xs:integer"/>
<xs:attributename="durée_B"type="xs:integer"/>
</xs:complexType>
<xs:complexTypename="type_bilan">
<xs:sequence>
<xs:elementname="formation_B"type="type_formation_B"
maxOccurs="unbounded"/>
</xs:sequence>
<xs:attributename="annee_B"type="xs:string"use="optional"/>
</xs:complexType>
<xs:elementblock=""name="Bilan"type="type_bilan"/>
</xs:schema>
```

2.2 Les sources de métadonnées

2.2.1 La source RDF

La source RDF contient 3 classes DOMAINE, METIER et REF_BDD.

```
<?xml version='1.0' encoding='UTF-8'?><!DOCTYPE rdf:RDF [  
<!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>  
<!ENTITY kb 'http://protege.stanford.edu/kb#'><!ENTITY rdfs  
'http://www.w3.org/2000/01/rdf-schema#'>]>  
<rdf:RDF xmlns:rdf="&rdf;" xmlns:kb="&kb;" xmlns:rdfs="&rdfs;">  
<kb:Domaine rdf:about="&kb;metadataRDF_Instance_0"  
kb:code="1" kb:designation_domaine="Exploration" rdfs:label="1"><kb:ref_bdd  
rdf:resource="&kb;metadataRDF_Instance_10000"/><kb:ref_bdd  
rdf:resource="&kb;metadataRDF_Instance_10006"/><kb:ref_metier  
rdf:resource="&kb;metadataRDF_Instance_7"/></kb:Domaine>  
<kb:Metier rdf:about="&kb;metadataRDF_Instance_10" kb:designation="Fonction de  
soutien" kb:reference="SOUT" rdfs:label="SOUT">  
<kb:ref_src rdf:resource="&kb;metadataRDF_Instance_10002"/><kb:ref_srce  
rdf:resource="&kb;metadataRDF_Instance_10003"/></kb:Metier>  
<kb:Ref_bdd rdf:about="&kb;metadataRDF_Instance_10000" kb:BDD="bddformation"  
kb:table="A_BESOINSFORM" rdfs:label="A_BESOINSFORM"/>
```

2.2.2 La source RuleML

Le schéma suivant montre une représentation de RuleML schéma

```
<Assert mapClosure="universal">
<Rulebase mapClosure="universal">
<Atom mapClosure="universal">
  <Rel>designer</Rel>
  <Var>t_formation_d</Var>
  <Var>designation_t_formation</Var>
</Atom> <Atom mapClosure="universal">
  <Rel>localiserXML</Rel> <Var>t_formation_1</Var>
  <Var>bdd_XML</Var></Atom>
<Atom mapClosure="universal"> <Rel>localiser</Rel>
  <Var>type_besoin</Var><Var>bdd</Var>
  <Var>tableBdd</Var></Atom> <Atom mapClosure="universal">
  <Rel>localiserXMLBES</Rel> <Var>REF_BES</Var>
  <Var>srceXML</Var> </Atom> <Atom mapClosure="universal">
```

2.3 Les sources de connaissance

2.3.1 La source OWL

Le programme ci-dessous il représente un fichier sou format OWL

```
<?xmlversion="1.0"?><rdf:RDFxmlns="http://www.owl-
ontologies.com/OntologyFormation.owl#"xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"xmlns:xsd="http://www.w3.org/2001/XMLSchema#
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"xml:base="http://www.owl-
ontologies.com/OntologyFormation.owl"><owl:Ontologyrdf:about=""/><owl:Class
rdf:ID="Formé"/><owl:Class rdf:ID="Plan"/> <owl:Class rdf:ID="TBRH">

<rdfs:subClassOf><owl:Class rdf:ID="Bilan"/></rdfs:subClassOf> </owl:Class><owl:Class
rdf:ID="Organisme"/><owl:ObjectPropertyrdf:ID="Participe_dans_bilan"><rdfs:range
rdf:resource="#Bilan"/><rdfs:domain
rdf:resource="#Formé"/></owl:ObjectProperty><owl:DatatypePropertyrdf:ID="mois">
<rdfs:domainrdf:resource="#TBRH"/><rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/></owl:DatatypeProperty><owl
:DatatypeProperty rdf:ID="MATRICULE"><rdfs:domain rdf:resource="#Formé"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty> <owl:DatatypeProperty rdf:ID="année_plan">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:domain rdf:resource="#Plan"/></owl:DatatypeProperty> <owl:DatatypeProperty
rdf:ID="nom_Org"><rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/><rdfs:domain
rdf:resource="#Organisme"/></owl:DatatypeProperty>
```

3. Outil et langage de programmation utilisé

Nous allons présenter par la suite les outils de développement choisis pour l'implémentation de notre système ainsi le langage de programmation utilisé. Ensuite, nous présentons notre application.

3.1 Les outils

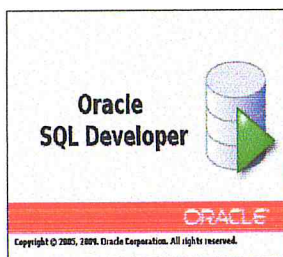
✓ **Eclipse :**



Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la Fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.

La spécificité d'Eclipse IDE vient du fait que son architecture est totalement développée autour de la notion de plug-in. Cela signifie que toutes les fonctionnalités de celui-ci sont développées en tant que plug-in.

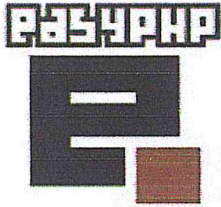
✓ **Oracle SQL Developer :**



Oracle SQL Developer permet le développement de A à Z d'applications en PL/SQL, la mise à disposition de feuilles de travail pour exécuter les requêtes et les scripts, une console pour l'administration de bases de données (DBA), un interface pour la génération de rapports (reporting), un solution complète de conception du modèle de données et un interface de migration permettant de migrer les bases de données d'éditeurs tiers vers Oracle.

Oracle SQL Developer supporte les produits Oracle ainsi que des plugins qui permettent de se connecter à des bases de données non Oracle.

✓ EasyPHP :



EasyPHP est un environnement comprenant deux serveurs : un serveur web Apache, et un serveur de bases de données MySQL.

Dans notre travail, EasyPHP nous a servi comme un serveur de bases de données MySQL, cet outil dispos d'une interface permettant la gestion des aléas « PhpMyAdmin»

✓ SPARQL

SPARQL est un langage de requête et un protocole qui permet l'accès en RDF. il a été créé par le W3C RDF Data Access Working Group. Le langage de requête SPARQL peut être utilisé pour avoir des informations à partir d'un RDFs. Il fournit certaines facilités l'extraction des informations sous forme d'URIs.

✓ API Jena :

Apache Jena est un Framework java dont l'objectif est de fournir un environnement facilitant le développement d'applications dédiées au web sémantique. Jena permet de manipuler des documents RDF, RDFS et OWL, et fournit en plus, un moteur d'inférences permettant des raisonnements sur les ontologies

✓ Xom

XOM est un nouveau modèle d'objet XML. Il s'agit d'une solution open source, une API basée sur une arborescence pour le traitement XML avec Java qui s'efforce pour l'exactitude, la simplicité et la performance¹⁰. Elle nous a servi pour la manipulation des schémas XML (.XSD) en java.

✓ OOjdrew

OO jDREW est une bibliothèque open source contenant un ensemble de classes servant à parser un document RuleML sous le langage java, écrit dans le langage de programmation Java. il nous a servi pour la manipulation des documents RuleML.

Chapitre IV : implémentation

✓ Saxon

Saxon est un processeur XSLT et XQuery. Il est disponible en solution open source et également en versions commerciales. Il existe des versions pour Java, JavaScript et .NET . La ligne de développement actuel, Saxon 9, implémente les spécifications XQuery 1.0 et XSLT 2.0. Saxon permet d'exécuter les requêtes Xquery sous le langage JAVA.

3.2 Langage de programmation

✓ Java :

Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton de Sun Microsystems.

Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris).

Le langage Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut-être utilisé sur internet pour des petites applications intégrées aux pages web (applet) ou encore comme langage serveur (JSP).

✓ MySQL



MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde¹, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL

Son nom vient du prénom de la fille du cocréateur Michael Widenius, My. SQL fait allusion au *Structured Query Language*, le langage de requête utilisé.

4. Présentation de OMediator



OMediator « Ontology Médiator » est un système de médiation à base ontologique pour les données, les métadonnées et les Connaissances, destiné au département formation de la DP de l'entreprise SONATRACH.

qui traite des requêtes simple qui sont faite pour rechercher des informations, et aussi des requêtes complexes qui sont faite pour le calcul des indicateurs pour aider le décideur de prendre des décisions, ou sauvegarder autant d'un tableau de bord chaque fin d'année ,et sans oublie que OMediator prend en considération l'établissement des bilans et des TBRH .

4.1 Authentification

Pour accéder a la fenêtre principale d'OMediator, il faut s'authentifier en renseignant le nom d'utilisateur et son mot de passe.



Figure 42: Authentification.

4.2 La fenêtre principale

La figure ci-dessous montre la fenêtre principale d'OMediator où se trouvent toutes ses fonctionnalités. Ce que nous intéresse le plus c'est le calcul d'indicateurs et l'établissement du plan et bilan

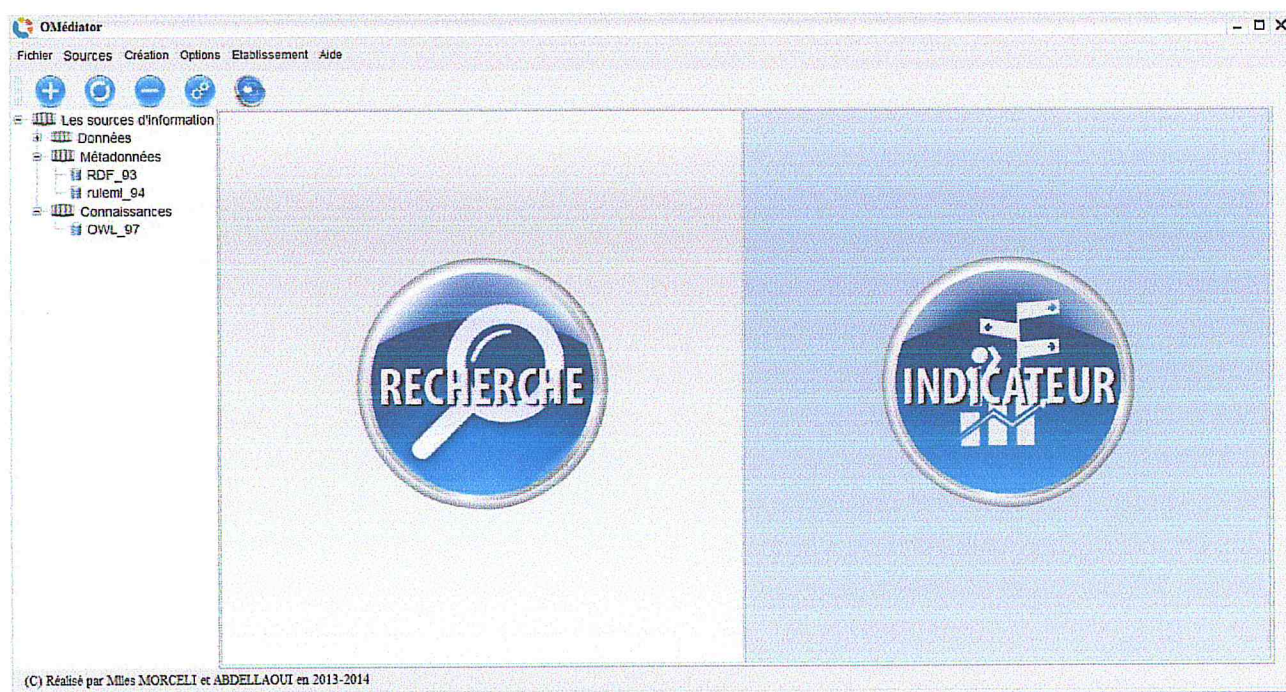


Figure 43: la fenêtre principale

4.3 Fenêtre d'exécution de la requête

Cette fenêtre est utilisée pour l'exécution de la requête complexe (le calcul d'indicateurs), la fenêtre est composée de:

- ✓ Une liste d'indicateurs;
- ✓ Les indicateurs sélectionnés;
- ✓ Les étapes de l'exécution de la requête dans le système de médiation;
- ✓ Un tableau pour le résultat de l'exécution.
- ✓ Un Bouton pour le sauvegarde de résultat sous format de fichier Excel, un Bouton pour l'impression, et un autre qui actualise la fenêtre pour pouvoir lancer une autre requête.

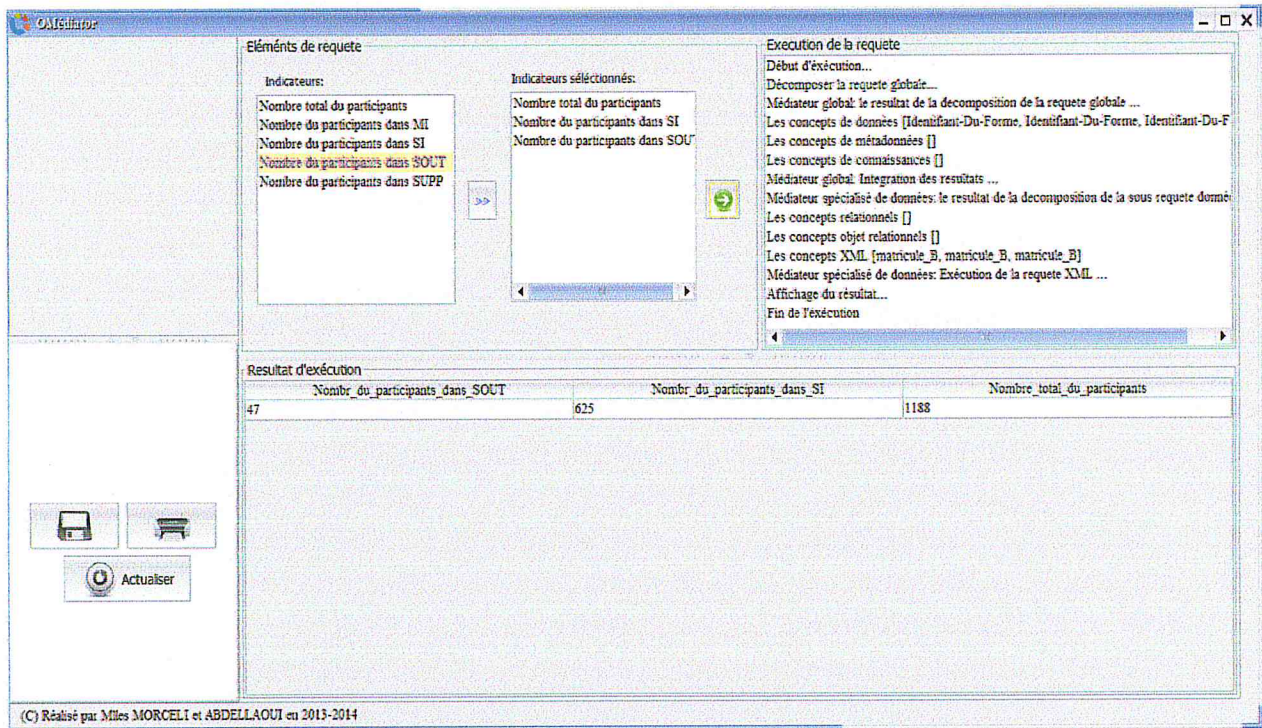


Figure 44: fenêtre d'exécution de la requête

4.4 Fenêtre d'établissement de PMTE

C'est une fenêtre qui permis à l'agentP d'établir le plan à moyen terme, qui sera enregistré dans la base de données formation pour le consulté dans le besoin.

Fichier Help

سونا تراش

sonatrach

Enregistrer

Ajouter Formation

Année de Plan 2016 Valider la date

Formation Code 2 Objectif Durée 5 Intitulé e de contrôle d'accès ALCATEL

Organisme EGZI Lieu ARZEW

Charges

Transport 145 Frais Pédagogiques 120 Hébergement 160 Autres charges 10

Employé

Matricule 04707v Nom abdellaoui Prenom mohamed

Figure 45 : fenêtre d'établissement de PMTE

4.5 fenetre d'etablissement de bilan

C'est une fenêtre qui permis à l'agentBT d'établir le bilan qui sera enregistré dans la base de données formation pour l'archivé.

Fichier Help

Système

sonatrach

Enregistrer

Ajouter Formation

Année de bilan 2014 Valider la date

Formation

Code 2 Objectif Durée 5 Inibulé e de contrôle d'accès ALCATEL

Organisme EGZI Lieu ARZEW

Charges

Transport 145 Frais Pédagogiques 120 Hébergement 160 Autres charges 10

Employé

Matricule 04707y Nom abdellaoui Prénom mohamed

Figure 46: fenêtre d'établissement du bilan.

4.6 fenetre de consultation de PMTE

Cette fenêtre affiche le plan enregistré dans a base de données pour faire des remarques sur les formations ou pour les mises à jour.

Code	Besoin	Méber	Domaine	Objectif de...	Matricule	Nom et Pré...	Intitulé	Durée (jours)	Organisme	Lieu	Frais Péda...	Héberger
2	4	MB	Réservoir In...		590377	BENDELLALI...	Système de ...	5	EGZI	ARZEW	120	160
2	4	MB	Réservoir In...		590377	BENDELLALI...	Système de ...	5	EGZI	ARZEW	458	45

Figure 47: fenêtre de consultation du PMTE.

5. Conclusion

Dans ce chapitre, nous avons présenté notre application qui nous a permis de mettre en œuvre notre conception qui est le traitement des requêtes complexes décisionnelle dans un système de médiation.

L'étape de l'implémentation nous a permis de mettre en pratique notre connaissance théorique tout en améliorant nos compétences en programmation.

1. Conclusion générale

L'intégration d'informations est un domaine de recherche qui a pour but de proposer Des architectures pour la construction de systèmes de questions/réponses sur des informations distribuées et hétérogènes provenant de multiples sources de données pouvant être disparates. Ces architectures de données permettent aux utilisateurs d'accéder à travers un schéma global unifié à plusieurs sources de données ayant chacune un schéma local. Ces sources de données sont le plus souvent réparties, autonomes et hétérogènes.

La médiation est l'une des approches d'intégration existantes. Elle est fondée sur la construction de systèmes jouant le rôle de médiateurs entre les utilisateurs en fournissant une interface d'interrogation centralisée et unique au-dessus de sources d'informations réparties et hétérogènes.

Le thème traité par ce mémoire a pour but de réaliser des requête complexe décisionnelles a base de système de médiation pour accéder au plusieurs source de données (XML, RDF, RuleML, des données relationnelle et objet relationnelle) pour répondre aux besoins des utilisateurs.

En ce qui concerne le traitement des requêtes, notre système prend en considération des requêtes décisionnelles qui visent à interroger soit des données, des métadonnées, des connaissances, ou une combinaison de deux ou trois types d'information. Ces requêtes sont de type complexe formées sous un langage unifié selon le schéma global du système, leur décomposition et exécution est automatique.

Les travaux menés dans ce mémoire nous ont permis d'approfondir nos connaissances dans plusieurs domaines :

- ✓ Notre projet de fin d'études nous a été d'un grand apport sur plusieurs plans principalement le plan conception, Nous avons appris l'aspect essentiel dans les applications d'intégration.
- ✓ Sur le plan technique nous avons acquis une certaine expérience dans le domaine de développement par les technologies suivantes : Java, XQuery, Jena et plusieurs API.

Conclusion générale

2. Perspective

La réalisation de notre projet de fin d'études nous a permis à atteindre les objectifs qui nous étaient fixés au départ en y ajoutant d'autres idées :

1. Améliorer notre étude en utilisant d'autre type de requête complexe décisionnelle : (Requête d'intervalle, Requête de meilleur-k et Requête imbriqué)
2. Doter d'un module d'écriture, de vérification et d'optimisation de requêtes formées par l'utilisateur.
3. Doter d'un module qui gère le profil utilisateur en vue d'optimiser la qualité du résultat.

ANNEXE

A

Présentation de l'organisme d'accueil

SONATRACH



SONATRACH est une entreprise publique économique à caractère industriel et commercial, placée sous la tutelle du ministère de l'énergie et des mines. La Société Nationale pour le TRANsport, la transformation et la Commercialisation des Hydrocarbures plus connue par son abréviation SONATRACH, a été fondée le 31 décembre 1963 par le décret N°63-491. Elle contribue largement au développement de l'Algérie en lui assurant un approvisionnement énergétique constant et à long terme. Sa mission se présente actuellement sous l'aspect d'une entreprise intégrée intervenant directement dans l'ensemble des activités du secteur d'hydrocarbures.

Les documents EXEL de SONATRACH



(Source de données sous EXCEL)

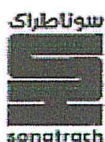
Plan 2011

Code	Type de Besoin	Niveau	Domaine	Objectif de formation	Matricule	Nom & Prénom	Intitulé	Durée (jour)	
4	CDA	2	SUPP	25	Perfectionnement	04707V	ABDELLAOUI MOHAMMED	Supervision Full Catering	10
5	CDA	2	SUPP	25	Perfectionnement	59037J	BENDELLALU SALAH	Supervision Full Catering	10
6	CDA	2	SUPP	25	Perfectionnement	72979G	BOUCETTA SD AHMED	Supervision Full Catering	10
7	CDA	2	SUPP	25	Perfectionnement	77098D	GOUBI MOHAMED	Supervision Full Catering	10
8	CDA	2	SUPP	25	Perfectionnement	62911H	MEKHARBECH AMINE	Supervision Full Catering	10
9	CDA	2	SUPP	25	Perfectionnement	09231W	ZENAGUI Ahmed	Supervision Full Catering	10
10	CDI	2	MB	4	Perfectionnement	65370C	CHALAL KAMEL	Rqp, Debits D'injection Des Inhibitions	5
11	CDI	2	MB	4	Perfectionnement	65674M	BOUNNAH ATHMANE	Rqp, Déshyd Du Gaz Regener Et Message Du Glycol	5
12	CDI	2	MB	4	Perfectionnement	60926J	ZAIZ FAYCAL	Rqp, Equipements De Surface Pour Test	5
13	CDI	2	MB	4	Perfectionnement	05144W	HAMADOU CHEKH	Rqp, Equipements De Test De Puits	5
14	CDI	2	MB	4	Perfectionnement	54923U	ARBOU KAMEL	Rqp, Exploitation Et Suivi Des Puits Prod Et Injecteur	5
15	CDI	2	MB	4	Perfectionnement	73728H	ABDEDAIM SMAIL	Rqp, Installat De Surface Puit/Gez Lift & inject S Pecke	5
16	CDI	2	MB	4	Perfectionnement	67197T	BOUSSAID LAO	Rqp, interv Installations Surface Puits Et Collecteur	5
17	CDI	2	MB	4	Perfectionnement	57823F	DJEROUNI ABDERRAHM	Rqp, Niton Sur Les Gisements Des Hydrocarbures	5
18	CDI	2	MB	4	Perfectionnement	05359K	LAYADI LARBI	Rqp, Optimisation Des Puits En Gaz Lift	5
19	CDI	2	MB	4	Perfectionnement	57949X	BEDOU MOHAMED LAO	Rqp, Pressurisation Et Dépressurisation Des Installation	5
20	CDI	2	MB	4	Perfectionnement	90510L	ARHAB MADJID	Rqp, Réseau De Collecte Et Equipement De Surface	5
21	CDI	2	MB	4	Perfectionnement	65041K	BAHEDA AMOR	Rqp, Sécurité De Surface	5
								Rqp, Stockage Des Produits Détruite	

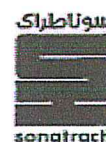
Bilan 2011

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Type de bes	Métier (2)	Domaine (3)	Matricule de	Nom & Prén	SEXE	CSP	Intitulés Act	Durée (en jo	Organismes	Lieu	Diplomante/non diplomante	Frais pédago	Frais Déplac	Frais d'Hebe Fr
2	2 SI		12 01091J	ABABDI BRAI	M	CADRE	Conduite Dé	4	Centre de Fc	CF/HRM	non	13.5	0	0	
3	4 Sout		16 01091J	ABABDI BRAI	M	CADRE	For. Ipbx Alc	10	Alcatel-Luce	HRM	non	0	0	0	
4	4 Sout		16 01091J	ABABDI BRAI	M	CADRE	For. Ipbx Alc	10	Alcatel-Luce	HRM	non	0	9.47	0	
5	2 SI		12 01091J	ABABDI BRAI	M	CADRE	Habilitation	1	formateur in	CF/HMD	non	0.67	0	0	
6	2 Supp		17 01091J	ABABDI BRAI	M	CADRE	Microsoft Ou	5	Naftogaz	STAH	non	16.25	0	0	
7	4 Sout		16 79596V	BENABED LA	M	CADRE	AIX Adminis	5	IBM	France	non	0	54.4	0	
8	2 Supp		17 69466P	ABABOU LAK	M	MAITRISE	Informatiqu	3	NAFTOGAZ	CF/RNS	non	1.25	0	0	
9	2 SI		12 57237R	ABABSA AM	M	MAITRISE	Habilitation	2	formateur in	CF/HRM	non	5.6	0	0	
10	2 Supp		17 57237R	ABABSA AM	M	MAITRISE	Informatiqu	10	CPE	CF/HRM	non	55	0	0	
11	2 SI		12 92452J	ABABSA BOL	M	MAITRISE	Permis De Tr	5	formateur in	CF/HRM	non	13.5	1.1	0	
12	2 SI		12 90507W	ABABSA NAS	M	CADRE	Habilitation	3	formateur in	STAH	non	1.35	0	0	
13	2 SI		12 90507W	ABABSA NAS	M	CADRE	Habilitation	2	formateur in	CF/HRM	non	5.6	0	0	
14	4 Sout		16 91543X	ABAD BOUAL	M	MAITRISE	Théorie Sate	12	EPNET Sat	HMD	non	0	11.88	0	
15	2 MI		11 92458W	ABADA YAH	M	MAITRISE	Usure dans l	3	IAP	Boumerdes	non	33.35	10.7	29.84	
16	2 SI		12 55573R	ABADI NABIL	M	MAITRISE	Habilitation	3	formateur in	CF/HMD	non	0.69	0	0	
17	2 Sout		24 03531L	ABADJ HAK	M	CADRE	Habilitation	3	formateur in	HBK	non	2.16	0	0	
18	2 MB		4 03531L	ABADJ HAK	M	CADRE	PROCEDEE E	5	IAP/SKIKDA	CF/HMD	non	49.73	0.91	35	
19	2 SI		12 57875G	ABADJ KAM	M	MAITRISE	Habilitation	2	formateur in	CF/HRM	non	5.6	0	0	
20	2 Sout		9 65363E	ABADOU MC	M	MAITRISE	Gestion des	4	IAP	Boumerdes	non	44.46	20.89	19.66	
21	2 SI		12 85056E	ABAI DJAME	M	MAITRISE	Les Risques l	5	IAP	Skikda	non	49.73	0.5	38.03	
22	2 Sout		22 91051J	ABAI DI LAR	M	CADRE	Passation de	4	IAP	Boumerdes	non	44.46	0	0	
23	2 SI		12 88932N	ABAS ABDEL	M	MAITRISE	L'incendie: r	5	IAP	Skikda	non	49.73	1.16	38.03	
24	2 SI		12 73982N	ABASSI NEE	F	MAITRISE	Les Gestes q	5	Naftogaz	STAH	non	48.75	0	0	
25	4 Sout		16 09706F	ABAZ DRIS	M	EXECUTION	For. Ipbx Alc	10	Alcatel-Luce	HRM	non	0	9.47	0	
26	2 SI		12 87144K	ABBAD ABDE	M	MAITRISE	Habilitation	3	formateur in	OHT	non	0.99	0	0	
27	2 SI		12 02237F	ABBAD BACH	M	CADRE	Conduite Dé	3	Centre de Fc	CF/HRM	non	10.13	0	0	
28	2 SI		12 05656Y	ABBAD MOH	M	CADRE	Habilitation	3	formateur in	CF/HMD	non	1.04	0	0	
29	2 SI		12 72805J	ABBAS BRAH	M	MAITRISE	Habilitation	3	formateur in	CF/HMD	non	0.73	0	0	
30	2 Supp		26 80733J	ABBAS MOU	M	MAITRISE	ANGLAIS INT	24	Centre de Fc	CF/HMD	non	6.92	0	0	
31	2 MI		11 80733J	ABBAS MOU	M	MAITRISE	Lubrification	5	IAP	Boumerdes	non	55.58	0.19	24.57	
32	2 SI		12 56878V	ABBAS TAYE	M	MAITRISE	Habilitation	3	formateur in	CF/HMD	non	0.55	0	0	
33	2 MI		11 72138S	ABBASSENE	M	CADRE	COMPRESSE	5	IAP/ARZEW	CF/HMD	non	49.73	1.33	6.53	
34	2 SI		12 73138E	ABBASSENE	M	CADRE	Habilitation	2	formateur in	CF/HRM	non	5.6	1.14	0	

Fiche des besoins (besoin collectifs)



SONATRACH
Activité Amont
Division Production
Direction Gestion du Personnel
Département Formation



FICHE D'IDENTIFICATION DES BESOINS

Besoins Collectifs

Direction:

1 - Intitulé de la formation ou Spécialité :

2 - Type de formation :
(Donner la bonne case)

Courte durée	Intra Entreprise	<input type="checkbox"/>
	Algérie	<input type="checkbox"/>
	Etranger	<input type="checkbox"/>

3 - Organisme (si identifié) :

4 - Durée (A spécifier en jours) :

5 - Que doit contenir le programme pédagogique ? :

-
-
-
-
-
-
-

6 - Objectifs opérationnels de la formation (A l'issue de la formation, l'agent doit être capable de) :

-
-
-



FICHE D'IDENTIFICATION DES BESOINS

Besoins Collectifs

(Suite)

7 - Population Concernée :

	Noms & Prénoms	Fonctions	Matricules
01 -
02 -
03 -
04 -
05 -
06 -
07 -
08 -
09 -
10 -
11 -

• Avis du Directeur de Structure

• Avis du Directeur Gestion du Personnel

ANNEXE

B

Descriptions des sources utilisées

✓ Définition des attributs de la base de données

BDDFORMATION

Table	Désignation
A_AGENT	L'agent formé
A_CSP	Catégorie socio professionnelle
A_MODULES	Module des thèmes de formation
A_TBRHFORM	Bilan de formation

Tables	Attributs	Désignation
A_Agent	MATRICULE	Matricule de l'agent formé
	NOM	Nom de l'agent formé
	SEXE	Sexe de l'agent formé
	CSP	Catégorie socio professionnelle (Clé étrangère de CSP)
A_CSP	CSP	Catégorie socio professionnelle
A_MODULES	MOD_CODE	Code Module
	MOD_INTITULE	Intitulé module
	MOD_DOMAINE	Code domaine de formation
A_TBRHFORM	FEN_RFORM	Référence de la Formation Entreprise
	FEN_MODULE	code module
	FEN_MATRICULE	Matricule de l'agent formé
	FEN_TYPFOR	Type de la Formation Entreprise (Formation, perfectionnement, séminaire)
	FEN_LIEUFORM	Adresse du Lieu de la formation Entreprise
	FEN_DIPLOME	Diplômant ou non diplômant
	FEN_OBSER	Observation Formation Entreprise
	FEN_CLANGUE	Code langue
	FEN_ALTERNEE	Formation alternée Oui ou Non
	FEN_PREVISION	Prévue ou non dans le plan

✓ Définition des attributs de la base de données

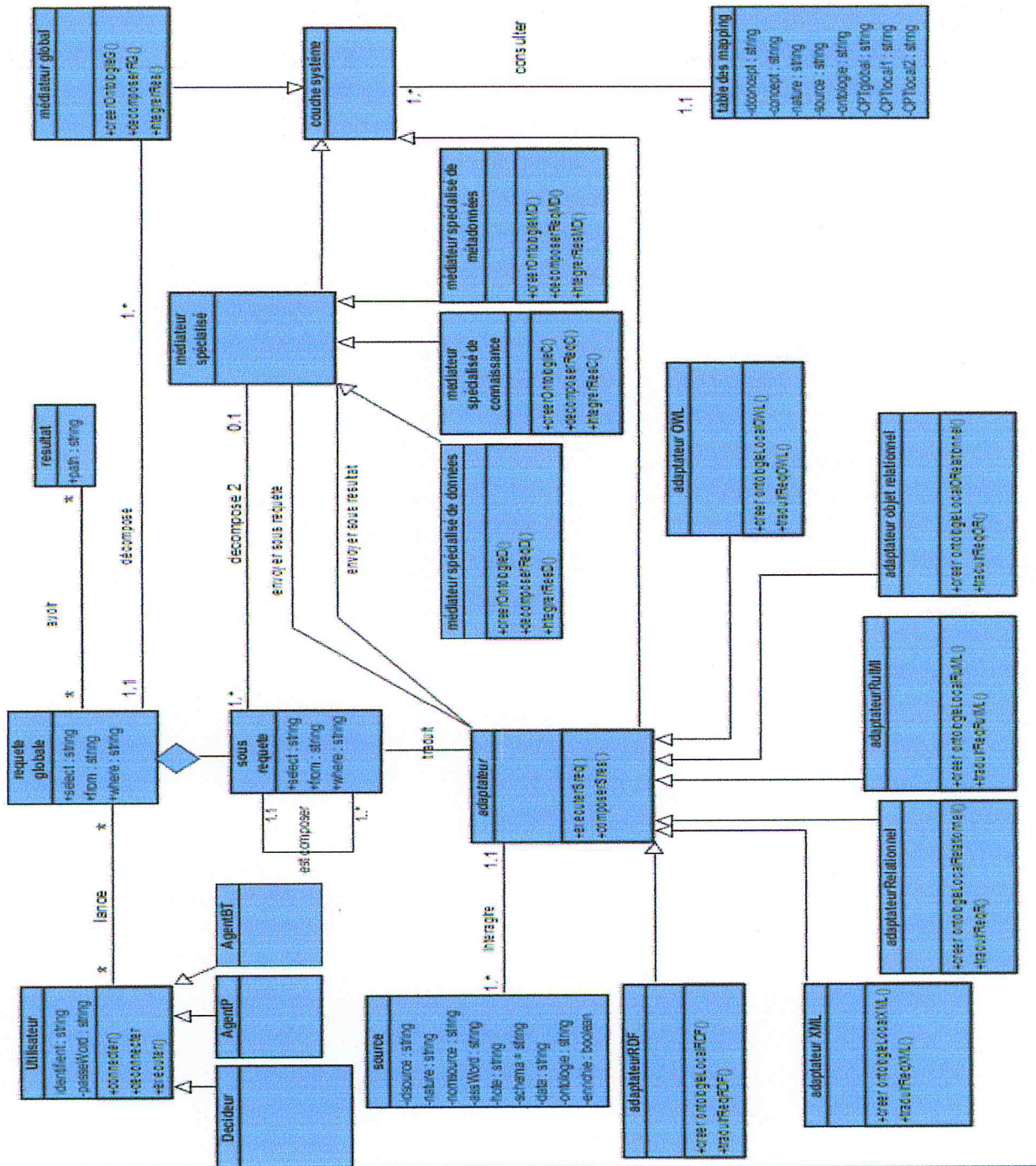
COSTFORMAT

Type complexe	Description
TYPE_COUTB	Les couts du bilan
TYPE_COUTP	Les couts du plan

Table	Attribut	Description
C_FORMATIONPLAN	ID_FORMP	Référence de la formation du plan
	COUPT	Les couts du plan
C_FORMATIONBILAN	ID_FORMB	Référence de la formation du bilan
	COUTB	Les couts du bilan

Table	Description
C_FORMATIONBILAN	Bilan de formations
C_FORMATIONPLAN	Plan des formations

Diagramme de classe



Dictionnaire de données

CLASSE	ATTRIBUT	TYPE	DESCRIPTION
Utilisateur	Identifiant	Varchar	Identifiant de l'exploitant du système.
	PassWord	Varchar	Mot de passe de l'exploitant pour se connecter au système.
Requête	Select	Varchar	La clause attributs.
	From	Varchar	La clause concepts.
	Where	Varchar	La clause des conditions.
Sous Requête	Select	Varchar	La clause attributs.
	From	Varchar	La clause concepts.
	Where	Varchar	La clause des conditions..
Résultat	Path	Varchar	Le chemin de stockage du résultat.
Source	id_srce	Varchar	Identifiant de la source.
	Nature	Varchar	Nature de la source (relationnelle, XML, ...)
	NomSrce	Varchar	Nom de la source.
	PassWord	Varchar	Mot de passe de la source.
	Hote	Varchar	L'hôte de la source.
	Schema	Varchar	Le chemin d'accès au fichier du schéma.
	Data	Varchar	Le chemin d'accès au fichier des données.
	Ontologie	Varchar	L'ontologie locale de la source.
	Enrichie	Char	enrichie =1 Si l'ontologie locale est enrichie. enrichie =0 Si l'ontologie locale n'est pas encore enrichie.

Table Des Mappings			
	id_concept	Varchar	Identifiant concept.
	Concept	Varchar	Le terme du concept dans le schéma d'origine.
	Nature	Varchar	La nature du concept(table, attribut, ...)
	Source	Varchar	La source où se trouve le concept.
	Ontologie	Varchar	L'ontologie locale de la source du concept.
	cpt_locale	Varchar	Le terme du concept dans l'ontologie locale.
	cpt_global2	Varchar	Le terme du concept dans l'ontologie secondaire.
	cpt_global1	Varchar	Le terme du concept dans l'ontologie globale.

Liste des méthodes

CLASSE	METHODE	DESCRIPTION
Exploitant	connecter ()	Se connecter au système.
	deconnecter ()	Se déconnecter du système.
	ExecuterReq ()	Composer et exécuter la requête globale et restituer le résultat.
Administrateur	AjouteSrce ()	Ajouter une source au système de médiation.
	modifierSce ()	Modifier les informations d'une source dans le système de médiation.
	supprimerSce ()	Supprimer une source du système de médiation.
	enrichirSce ()	Enrichir les concepts de l'ontologie locale de la source.
MédiateurSpécialisé Données	créerOntoDonnées ()	Créer l'ontologie secondaire de données.
	décomposerReqD ()	Décomposer la sous requête données en sous requêtes selon les sources de données.
	integreResD ()	Intégrer les résultats de la sous requête données.

MédiateurSpéciali- séMétadonnées	créerOntoMétaonnées ()	Créer l'ontologie secondaire de métadonnées.
	décomposerReqM()	Décomposer la sous requête métadonnées en sous requêtes selon les sources de métadonnées.
	integreResM ()	Intégrer les résultats de la sous requête métadonnées.
MédiateurSpéciali- séConnaissances	créerOntoConnaissances ()	Créer l'ontologie secondaire de connaissances.
	décomposerReqC()	Décomposer la sous requête connaissance selon les sources de connaissances.
	integreResC ()	Intégrer les résultats de la sous requête connaissances.
Adaptateur	executerSReq ()	Envoyer la sous requête traduite à la source.
	composerRes ()	Faire correspondre le résultat avec les concepts globaux.
Adaptateur Relationnel	créerOntoLocaleR ()	Créer l'ontologie locale de la source relationnelle
	traduireReqR ()	Traduire la sous requête en SQL.
Adaptateur ObjetRelationnel	créerOntoLocaleOR ()	Créer l'ontologie locale de la source Objet- relationnelle
	traduireReqOR ()	Traduire la sous requête en SQL3.
AdaptateurXML	créerOntoLocaleXML ()	Créer l'ontologie locale de la source XML.
	traduireReqXML ()	Traduire la sous requête en XQuery.
AdaptateurRDF	créerOntoLocaleRDF ()	Créer l'ontologie locale de la source RDF.
	traduireReqRDF ()	Traduire la sous requête en SPARQL.
AdaptateurRuleML	créerOntoLocaleRuleML ()	Créer l'ontologie locale de la source RuleML.
	traduireReqRuleML ()	Traduire la sous requête en XQuery.
AdaptateurOWL	créerOntoLocaleOWL ()	Créer l'ontologie locale de la source OWL.
	traduireReqOWL ()	Traduire la sous requête en SPARQL.

ANNEXE

C

Calcul des indicateurs

Liste des indicateurs

Indicateurs	Règlements
Calcul de Volume pédagogique	Volume pédagogique = (Nombre de personne * le nombre de jours)/22
Calcul de cout total pour chaque formation	$C_{Total} = \sum$ cout de chaque formation.
Calcul de taux réel par rapport au cout	Txréel (cout) = total de cout de fonction/cout prévisionnel.
Calcul de taux réel Par rapport à l'effectif	Txréel(effectif) = total effectif de fonction/cout prévisionnel
Calcul de taux réel Par rapport a la durée (en jour/en heure)	Txréel(heure/jour) = totaleh/j de fonction/cout prévisionnel

ANNEXE

D

Définitions des langages du web

1. XML

Le XML est un langage de balises [Markup Language]. Mais au contraire du Html où les balises sont définies, vous devez inventer vos balises. Rappelez-vous, le XML est eXtensible. Il faut donc écrire soi-même le nom des balises utilisées. Il y a quelques règles pour la composition des noms (mais elles ne déroutent pas les habitués du Javascript) :

- ✓ Les noms peuvent contenir des lettres, des chiffres ou d'autres caractères.
- ✓ Les noms ne peuvent débuter par un nombre ou un signe de ponctuation.
- ✓ Les noms ne peuvent commencer par les lettres xml (ou XML ou Xml...).
- ✓ Les noms ne peuvent contenir des espaces.
- ✓ La longueur des noms est libre mais on conseille de rester raisonnable.
- ✓ On évitera certains signes qui pourraient selon les logiciels, prêter à confusion comme "-", ";", ". ", "<", ">", etc.
- ✓ Les caractères spéciaux pour nous francophones comme é, à, ê, ï, ù sont à priori permis mais pourraient être mal interprétés par certains programmes.

2. RDF

Le format RDF permet de combiner deux documents XML en un seul, en décrivant les relations entre les données, et est utilisé par XUL pour indiquer dans le fichier lui-même comment le programme affichera les données. Ce format convient lorsque l'information doit avoir une structure correspondant à celle de l'outil qui l'utilise, une interface graphique le plus souvent. Lorsqu'également l'on veut agréger des sources de données séparées.

Par exemple, locales et distantes. L'exemple qui nous intéresse le plus ici est la combinaison des données définissant l'interface graphique et les données qu'il affiche.

Il peut être utilisé pour la localisation en plusieurs langues, les textes étant séparés des éléments logiciels qui les affichent et sélectionnés selon le pays.

3. OWL

Le langage OWL offre trois sous-langages d'expression croissante conçus pour des communautés de développeurs et d'utilisateurs spécifiques :

Le langage *OWL Lite* concerne les utilisateurs ayant principalement besoin d'une hiérarchie de classifications et de mécanismes de contraintes simples. Par exemple, quoiqu'OWL Lite gère des contraintes de cardinalité, il ne permet que des valeurs de cardinalité de 0 ou 1. Mettre en œuvre des outils pour OWL Lite devrait être plus simple que pour ses parents d'expression plus grande, tout comme tracer un chemin de migration rapide pour des thésaurus et autres taxonomies.

Le langage *OWL DL* concerne les utilisateurs souhaitant une expressivité maximum sans sacrifier la complétude de calcul (toutes les inférences sont sûres d'être prises en compte) et la décidabilité (tous les calculs seront terminés dans un intervalle de temps fini) des systèmes de raisonnement. Le langage OWL DL comprend toutes les structures de langage de OWL avec des restrictions comme la séparation des types (une classe ne peut pas être en même temps un individu ou une propriété, une propriété être un individu ou une classe). OWL DL se nomme ainsi pour sa correspondance avec la *logique de description* [Description Logics], un champs de la recherche portant sur un fragment décidable particulier de la logique de premier ordre. Le langage OWL DL, conçu pour gérer le secteur existant de la logique de description, offre les propriétés de calcul souhaitées pour les systèmes de raisonnement.

Le langage *OWL Full* est destiné aux utilisateurs souhaitant une expressivité maximum et la liberté syntaxique de RDF sans garantie de calcul. Par exemple, dans OWL Full, on peut simultanément traiter une classe comme une collection d'individus et comme un individu à part entière.

Références bibliographique

[ABD04] S. ABDUL GHAFOR. Méthodes et outils Pour l'intégration des ontologies. Thèse de Doctorat. Laboratoire d'Informatique en Images et Systèmes. Lyon. 2003-2004.

[AHM09] H. AHMAD. Une approche matérialisée basée sur les vues pour l'intégration de documents XML. Thèse de Doctorat. Université Joseph Fourier – Grenoble. 2009.

[BEN06] L. BENHLIMA, D. CHIADMI. Vers l'interopérabilité des systèmes d'information hétérogènes, Maroc. 2006.

[BER06] F. BERTRAND Utilisation d'ontologies pour l'intégration de données, 2006.

[BOU05] M. BOUMEDIENE. Définition d'un système générique de partage de données entre systèmes existants. Thèse doctorat École doctorale Informatique et Information pour la société (EDIIS). Université de La Rochelle Avenue Michel Crépeau. 2005.

[BOU08] A. BOUSSIS. Intégration de sources de données à base ontologique dans un environnement P2P. Magister en informatique. Institut national d'informatique INI. Algérie. 2007-2008.

[BOU10] F. BOUARAB DAHMANI. Modélisation basée ontologies pour l'apprentissage interactif- Application à l'évaluation des connaissances de l'apprenant. Thèse de Doctorat Algérie: Université Mouloud Mammeri de Tizi-Ouzou. 2010.

[BOU11] J. BOUHASSOUN, L. BOUHASSOUN. Interrogation de sources de données médicales à base de Services Web DaaS. Master en informatique. Université Abou Bakr Belkaid Tlemcen. 2011

[BRU 11] M. BRULEY. Propos Sur les SI Décisionnels. France. 2011.

[CHA09] G. CHANDESRIS. Systèmes d'intégration de données en biologie: Entrepôts de données et systèmes de médiations - Étude sur BioWareHouse et BioMediator. Cours

systèmes d'intégration du Master 2 Génie Biologique et Informatique. Université d'Evry Val d'Essonne, 2008-2009.

[DJE07] L. DJEMA, F.O. BOUMGHAR, S. DEBIANE. L'imagerie médicale dans une base de données distribuée multimédia sous Oracle 9i. In: SETIT. Tunis. 2007.

[GOH 99] C. GOH. Context interchange: new features and formalisms for the intelligent intégration of information. ACM Transactions on Information Systems. 1999

[GUE10] C. GUEYDAN. XeuTL: un outil ETL pour l'intégration de données. Ingénieur C.N.A.M. en informatique. Conservatoire national des arts et métiers centre régional Rhône-Alpes. Centre d'enseignement de Grenoble. 2010.

[HAD10] A.HADJI, M. KHOLLADI. Traitement et Fusion de Données dans le Cadre de l'Interopérabilité Sémantique des Systèmes d'Information Géographiques .Université d'Ouargla, Algérie. 2010.

[INM94] B. INMON W.H, J.WILEY. Using the Data Warehouse. 1994.

[IZZ06] S. IZZA. Intégration des systèmes d'information industriels: Une approche flexible basée sur les services sémantiques. Thèse informatique. Saint-Etienne: Ecole Nationale Supérieure des Mines. 2006.

[LAH11] F. LAHMAR. Une approche Hybride d'intégration de sources de données hétérogènes dans les datawarehouses. Thèse de Doctorat en informatique. Université Mentouri de Constantine. 2011.

[LAS01] A. LASSUS. Méthodologie des systèmes d'information – UML. Cour disponible sur: <fdigallo.online.fr/cours/uml.pdf>. 2000-2001

[MAI07] N. MAIZ. Intégration de données par médiation basée sur les ontologies pour l'analyse en ligne (OLAP) à la demande. Thèse de doctorat en informatique. Université de Lyon. 2007

[PAR 96] C. PARENT, S SPACCAPIETRA. Intégration de bases de données : Panorama des problèmes et des approches. Ingénierie des systèmes d'information, Volume 4, N°3, 1996.

[LIU05] S. LIU Présentation au sujet: "Le Projet MediaGrid: Un canevas de système de médiation pour l'accès transparent aux sources largement distribuées Towards a Mediation.". 2005.

[JAC00] I. Jacobson, G. Booch, J. Rumbaugh. L'ouvrage: The Unified Software Development Process. Eyrolles. 2000.

[ROQ 07] P. ROQUES. UML 2 modéliser une application web. 3rd Ed. EYROLLES. PARIS : EYROLLES. 2007.

[SOL02] G. SOLAR, A. DOUCET. Médiation de données : solutions et Problèmes ouverts. Laboratoire Logiciels, Systèmes, Réseaux, 681 rue de la Passerelle, BP 72, 38402, Saint Martin d'Hères, Laboratoire d'Informatique de Paris VI, 4, Place Jussieu, 75232 PARIS CEDEX 05. 2002.

[SOU05] A. SOUKAE. Génération automatique des requêtes de médiation dans un environnement hétérogène. Thèse de Doctorat en informatique. Université de Versailles Saint-Quentin en Yvelines. 2005.

[TRI05] T. Trinh VU. Une approche pour la construction d'évaluateurs adaptables de requêtes. Thèse de doctorat. Laboratoire Logiciels, Systèmes, Réseaux (LSR) dans le cadre de l'Ecole Doctorale « *Mathématiques, Sciences et Technologie de l'Information* ». 2005.

[VOD14] D. VODISLAV. Architectures d'intégration de données. Master Informatique M1. Cours IED. Université de Cergy-Pontoise. Disponible sur: <<http://depinfo.u-cergy.fr/~vodislav/Master/IED/fichiers/integration.pdf>>. (Consulté le : 19.02.2014).

[WAC14] H. WACHE et al. Ontology-Based Integration of Information A Survey of Existing Approaches.

Disponiblesur: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.142.4390&rep=rep1&type=pdf>>. (Consulté le : 20.02.2014).

