

# Un modèle sémantique pour l'interopérabilité de systèmes d'information

Fabrice Jouanot

Equipe Ingénierie Informatique et Base de Données - Laboratoire LE2I  
Université de Bourgogne, BP 47870 - 21078 Dijon Cedex France  
fabrice.jouanot@u-bourgogne.fr, Tel : 03 80 39 58 93, Fax : 03 80 39 58 92  
[Catégorie Jeune Chercheur]

---

## Résumé :

Les besoins en matière d'information ont beaucoup évolué ces dernières années. Les nouvelles applications nécessitent des méthodologies flexibles et dynamiques pour partager des informations distribuées sur une multitude de sites. Les solutions d'interopérabilité de systèmes d'information doivent respecter différents critères (autonomie, extensibilité, scalabilité, composabilité et transparence) et résoudre les conflits de données qui apparaissent au niveau de la représentation hétérogène de l'information. La sémantique de l'information est capital pour répondre à ces problèmes. Cet article présente une solution d'interopérabilité, DILEMMA, qui étend les concepts de la médiation de schéma par des outils sémantiques pour faciliter la compréhension des informations partagées, pour aider à l'identification des informations pertinentes et pour adapter ces informations à un contexte d'utilisation. Cet article définit un modèle de médiation pour représenter à la fois la structure des informations, leur sémantique et les règles d'intégration.

## Mots clefs :

Contexte , Interopérabilité, Médiation, Sémantique.

---

## Abstract :

Information requirements have evolved tremendously last years. New applications need flexible and dynamic methodology to allow the sharing of distributed information through many heterogeneous systems. The notion of Interoperability has emerged and solutions have to respect some criteria (autonomy, extensibility, scalability, composability and transparency) to resolve data conflicts and data localization. It appears that the semantic of data is an important knowledge to fulfill these problems. This article presents an interoperability solution, DILEMMA, which extends schema mediation concepts with semantic tools to facilitate the understanding of shared data, to help to identify relevant data and to adapt data to user context. This article defines a mediation model which can represent data structure, information semantic, application domain and integration rules.

## Keywords :

Context, Interoperability, Mediation, Semantic

# 1 Introduction

L'informatisation a conduit au développement d'un nombre considérable de systèmes d'information chargés de stocker, d'organiser, d'interroger et de restituer des données utiles. Une masse d'information considérable est donc disponible et il s'agit d'offrir un moyen simple, efficace et économique pour y accéder. L'évolution des technologies de communication met davantage encore le besoin d'échanger de l'information au premier plan. Partager des données entre SI n'est pas une idée nouvelle mais de nouveaux challenges naissent avec l'arrivée du commerce électronique et du partage de données via Internet.

Les solutions doivent respecter différents critères pour répondre à ces besoins [LIU95a] : **autonomie** des systèmes, **extensibilité** de l'architecture pour maîtriser l'ajout et le retrait de SI, **scalabilité** du système face à l'évolution du nombre de participants, **composabilité** pour combiner les informations de SI différents et **transparence** d'accès à la localisation et au format des données. Les solutions reposent sur la notion d'interopérabilité de SI, c'est-à-dire sur la mise en œuvre d'une collaboration entre systèmes permettant le partage de données et de services pour répondre à une demande d'informations. La résolution d'une requête dans un environnement interopérable passe par l'intégration de données distribuées sur plusieurs systèmes hétérogènes. Trois questions sont soulevées : (1) Quels sont les sites ou les systèmes susceptibles de contenir tout ou partie de l'information désirée ? (2) Quelles sont les données pertinentes en rapport avec les besoins exprimés ? (3) Comment transformer, adapter et appareiller ces données pour obtenir un résultat homogène et compréhensible par l'utilisateur ?

L'intégration des informations est confrontée à de nombreux conflits de données. **Les conflits syntaxiques** résultent de l'utilisation de modèles de données différents d'un système à l'autre. Des concepts différents sont utilisés pour structurer la même information (relation dans le modèle relationnel, classe dans le modèle objet, tag XML, etc.). **Les conflits schématiques** résultent d'une structuration et d'une classification différente des informations. Ils sont étroitement liés aux choix de conception. **Les conflits sémantiques** proviennent des différences d'interprétation des informations partagées entre différents domaines d'application. Plusieurs types de conflits sémantiques apparaissent : les conflits de nom (problèmes taxonomiques et linguistiques), les conflits de valeurs (problèmes d'unités, d'échelles) et les conflits cognitifs (problèmes de signification). Les conflits schématiques et sémantiques sont étroitement liés. La signification des informations doit être clairement définie et les conflits sémantiques résolus avant d'identifier les conflits schématiques.

De nombreuses solutions ont été proposées pour prendre en considération ces conflits de données, elles garantissent toutes au moins l'autonomie, la composabilité et la résolution des conflits syntaxiques.

**Les approches multi-bases** reposent sur l'utilisation d'un langage d'interrogation et d'un modèle de représentation commun [FLOR95]. Chaque SI exporte ses informations sous la forme d'un schéma décrit dans le modèle commun (généralement un modèle orienté objet), le langage multi-bases (extension de SQL ou OQL) permet une interrogation multi-sites. L'approche multi-bases est extensible mais n'offre pas de transparence à la localisation. La résolution des conflits schématiques et sémantiques reste entièrement à la charge de l'utilisateur.

**Les approches fédérées** reposent sur l'intégration. Chaque SI exporte un schéma dans un modèle commun, les différents schémas d'exports sont alors intégrés dans un (ou plusieurs) schéma(s) fédéré(s). Un schéma fédéré permet l'accès aux données partagées d'une manière uniforme et globale. Sheth dans [SHET90] propose une architecture en cinq niveaux. Les conflits

schématiques et sémantiques sont résolus par le processus d'intégration. La transparence est assurée, il suffit de poser une requête sur le schéma fédéré. Néanmoins l'extensibilité et la scalabilité sont des critères qui s'adaptent très mal à l'intégration. La fédération est une approche à préférer dans le cadre d'une coopération de quelques SI et peu évolutive.

**Les approches de type médiation** étendent la fédération en apportant davantage de flexibilité. 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 [WIED92, WIED95]. Le médiateur permet de localiser l'information et de résoudre les conflits schématiques et sémantiques. Un composant secondaire, appelé wrapper, sert d'interface avec les SI. Le wrapper résout les conflits syntaxiques en présentant les données dans le modèle de médiation. On distingue deux types de médiation selon la manière de résoudre les conflits de données : la *médiation de schéma* qui construit au préalable une base d'informations prenant en compte les SI participants pour permettre au médiateur de faire son travail d'intégrateur et la *médiation de contexte* où le médiateur est guidé par des informations à caractère sémantique pour résoudre dynamiquement une requête sans connaissance à priori des SI participants. La médiation de schéma est une extension directe de l'approche fédérée [CODY95, TOMA96, LIU97] avec une meilleure extensibilité et souvent une meilleure scalabilité (interfaces objets, langage à base de règle) [MOLI95, PAPA96]. La médiation de contexte cherche à découvrir les données qui sont sémantiquement proches, elle est capable de localiser et d'adapter l'information pour assurer une transparence complète [KASH94, BISHR97, BRES97, DECK99, FOWL99, OUKS99].

Cet article présente DILEMMA, une médiation de schéma étendue. Elle propose une méthodologie, un modèle et une architecture pour construire une base de connaissance qui intègre les informations correspondant à un profil d'utilisation bien défini. Le modèle de médiation permet de représenter la structure des données dans un modèle commun, d'adapter et de composer les informations pour assurer une transparence complète. Ce modèle associe également une sémantique aux informations partagées dans le but de pouvoir les comparer et d'identifier les données pertinentes. Un mécanisme d'intégration incrémentale guidée par la sémantique permet une adaptation à un environnement dynamique et large.

La section 2 dresse un panorama des solutions basées sur la médiation de contexte et introduit ensuite les points clefs de DILEMMA. La section 3 décrit un exemple type d'environnement de coopération et explique comment DILEMMA peut résoudre les problèmes liés à l'interopérabilité des SI. La section 4 présente les deux aspects schématique et sémantique du modèle de médiation. La section 5 développe le rôle de la sémantique dans le processus d'intégration incrémentale de SI. La section 6 conclut l'article et présente les perspectives de recherches.

## **2 De l'usage de la sémantique**

De nombreuses solutions utilisant la sémantique ont été développées dans le cadre de la médiation de contexte.

### **2.1 Un panorama de la médiation de contexte**

Les solutions de type médiation de contexte se basent sur des modèles de représentation de connaissance capables de décrire, dans une certaine mesure, la sémantique véhiculée par une information et sur des outils permettant de comparer et d'unifier la sémantique des informations indépendamment des structures sous-jacentes. Les solutions seront présentées selon trois critères de comparaison : la représentation de la sémantique, les outils utilisés pour manipuler cette

sémantique et le type d'architecture employée pour assurer la communication entre les composants de médiation.

La notion de sémantique d'une entité ne peut être représentée de manière absolue. Une entité doit être mise en relation avec d'autres entités pour qu'une signification lui soit associée. La description sémantique d'une entité n'a de sens que par rapport à un contexte particulier. Un contexte peut être défini comme l'ensemble fini de concepts, de relations, de contraintes et de règles qui décrivent un domaine d'application. Chaque solution définit des contextes sur lesquels s'appuie l'aspect sémantique des informations partagées. Les informations associées à un même contexte peuvent alors être aisément classifiées et comparées puisqu'elles partagent la même sémantique : c'est la phase de rapprochement sémantique. Cette phase s'avère plus délicate lorsqu'il s'agit de rapprocher deux informations définies respectivement sur deux contextes a priori disjoints. Il s'agit alors de réconcilier les contextes pour estimer s'il est possible de rapprocher ces informations et déduire la manière de les composer.

Un contexte peut être représenté par un schéma conceptuel qui décrit un domaine particulier d'application. Il est composé de concepts en relation les uns avec les autres (relations d'héritage, relations de compositions, relations logiques). Un concept est généralement défini à partir du contenu d'une ontologie. L'ontologie [GRUB93] est la spécification d'une conceptualisation, c'est-à-dire l'expression dans un langage de termes et de concepts. Une ontologie peut être représentée à l'aide d'un réseau sémantique, d'un graphe terminologique, d'un graphe conceptuel, d'un ensemble de règles logiques ou d'un schéma orienté objets. Le contenu d'une ontologie reste proche d'un domaine d'application précis même si quelques projets s'orientent vers la définition d'ontologies génériques. On distingue deux types d'ontologies [GUAR97] : (1) les **ontologies minimales** proposent des concepts, des propriétés et des règles de base à partir desquels il est théoriquement possible de fabriquer tout contexte par spécialisation des concepts de l'ontologie, (2) les **ontologies maximales** sont supposées contenir la description de tout l'univers d'un domaine, un contexte est alors défini comme une vue réduite d'une ontologie.

Les solutions sont classées en deux groupes selon que la médiation utilise une ontologie unique ou des relations inter-ontologiques. Dans le premier cas, ontologie unique, les contextes des informations sont construits à partir du même univers conceptuel et la réconciliation contextuelle passe par une comparaison directe des concepts qui possèdent alors une unité sémantique de base commune.

**Le projet Coin** [GOH94, BRES97] repose sur la description de la sémantique des valeurs des données échangées. Chaque donnée précise la sémantique de ses valeurs en indiquant par exemple le type d'unité et sa signification. Une ontologie fournit un vocabulaire pour décrire la signification de ces valeurs. Chaque SI possède un contexte local qui décrit la sémantique des valeurs manipulées localement. Un contexte de référence décrit le domaine d'utilisation des valeurs et les règles de transformation permettant de réconcilier les sémantiques de valeurs. Ces règles utilisent la logique de frame [] pour déduire un chemin de traduction d'un contexte source vers un contexte cible et adapter la représentation des informations. L'utilisateur peut interroger le système sans se soucier du format des données qui lui sont présentées dans son format local. L'identification des conflits sémantiques est restreinte aux différences d'interprétation de valeurs. La localisation et la composabilité restent encore du ressort de l'utilisateur.

**On2broker** [DECK99] s'intéresse plus particulièrement à l'extraction de données à partir du contenu de pages Web. L'interopération consiste ici à sortir les données pertinentes du contenu des pages web, découvrir les liens avec des informations situées sur d'autres pages, combiner ces données. Un modèle sémantique ajoute de l'information au contenu des pages web sous la forme de balises HTML propriétaires à la manière d'XML [XML]. L'ontologie est définie sous la forme

de classes d'objets et de règles de déduction qui étendent le langage F-logic [KIFE95]. Ce sont des références à ces classes qui se retrouvent liées au contenu des pages web. Un processus de recherche et d'analyse scrute continuellement les sites web participants pour construire une base de faits globale pour inférer les règles de l'ontologie sur les faits. L'utilisateur pose sa requête sous la forme d'une expression logique. L'ontologie revêt l'apparence d'un schéma global par lequel accéder aux données. La construction d'une base de faits compromet la scalabilité du système.

**SEMWEB** [BISH97] est orienté vers l'interopérabilité de systèmes d'information géographiques (SIG). Un schéma fédéré est construit à partir du vocabulaire issu d'une ontologie. L'ontologie est une hiérarchie de termes conçue pour une application particulière. Le schéma fédéré est enrichi par des formules logiques du premier ordre et forme un contexte de référence appelé proxy contexte. Chaque source d'information possède son propre contexte local qui doit être adapté au proxy contexte à l'aide de fonctions de transformation. L'interrogation se fait sur le schéma du proxy contexte qui, grâce aux formules logiques et aux fonctions de transformation, peut interroger les sites participants et recomposer les données dans un résultat homogène. Cette approche ressemble davantage à une évolution de l'approche fédérée classique sans la lourdeur de l'intégration de schéma : l'utilisateur ne peut accéder aux informations que par rapport à un contexte de référence commun à tout le monde.

Dans le second cas, ontologies multiples, chaque contexte peut être décrit à partir d'un univers conceptuel spécifique, les langages de description sémantiques sont eux-même hétérogènes. Il est donc nécessaire de pouvoir interpréter le contenu d'une ontologie dans une autre ontologie. Un dictionnaire de mise en correspondance sémantique est généralement utilisé pour relier l'interprétation des univers conceptuels entre deux ontologies.

**Le projet Observer** [KASH94, MENA99] est basé sur une hiérarchie de serveurs d'ontologies décrits en logique terminologique pour définir des domaines d'utilisation. Chaque SI doit définir ses propres données par rapport aux concepts d'une ou plusieurs ontologies. Des règles de réécritures et des fonctions de transformation définissent l'interprétation locale des informations. Un utilisateur pose une requête terminologique sur les concepts d'une ontologie. Cette requête peut être envoyée sur tout le domaine et seuls les serveurs dont le contexte est en accord avec celui de la requête fournissent des réponses. La réconciliation sémantique se base sur des relations inter-ontologies prédéfinies 2-2 et sur des relations de synonymie.

**InfoSleuth** [FOWL99] est basé sur une architecture agent. Des agents ressources conservent les informations fournies par les sources de données en utilisant le vocabulaire d'agents ontologiques. Chaque ressource doit être enregistrée à un agent "broker" pour assurer la localisation des informations. Un agent utilisateur fournit une interface d'interrogation aux utilisateurs et gère une requête à l'aide des fonctionnalités des agents "broker". La découverte d'information est supportée à l'aide d'agents spécialisés qui étendent l'architecture pour gérer plusieurs ontologies simultanément grâce à des relations inter-ontologies prédéfinies.

**Kraft** [VISS99a, VISS99b] utilise aussi une architecture à base d'agents mais ne réclame pas la définition de relations inter-ontologies pour réconcilier plusieurs domaines, les informations sont comparées et regroupées par affinité sémantique. Kraft est basé sur le langage Ontolingua [FIKE97] (extension de Kif [GINS91]) pour représenter les définitions ontologiques des informations. La réconciliation contextuelle construit des ontologies intermédiaires qui permettent de classer dans des relations de généralisation / spécification les concepts issus de domaines différents. La méthode permet de rapprocher les informations similaires et fournit une interprétation des concepts d'une ontologie dans une autre. Ontolingua est un langage riche mais aussi relativement complexe qui rend difficile la comparaison de concepts.

**SCOPES** [OUKS99] propose également une réconciliation contextuelle en utilisant les informations ontologiques comme point de départ. Chaque cas possible de rapprochement est évalué et le meilleur est retenu. Le nombre de cas à tester pour répondre à une requête risque vite d'exploser en complexité dans le cas d'ontologies complexes.

L'utilisation d'ontologies pour décrire un univers de discours est le point commun de nombre de solutions. Définir un serveur d'ontologies comme référence sémantique ne va pourtant pas sans problèmes : (1) Identifier les concepts d'une ontologie réclame un consensus de la part des intervenants, (2) la modélisation de l'ontologie est une étape délicate, (3) l'univers sémantique est limité par le contenu de l'ontologie. La prise en compte de plusieurs serveurs d'ontologies fait réapparaître le problème des mises en correspondance 2-2 entre représentations hétérogènes, bien connu des outils de traduction de schéma [NICO97].

## 2.2 Approche sémantique dans DILEMMA

DILEMMA est basé sur une intégration incrémentale des informations pertinentes découvertes dynamiquement sur des SI. DILEMMA profite de la robustesse de l'approche médiation de schéma et la combine aux techniques de rapprochement sémantique de la médiation de contexte. Les informations possèdent une description contextuelle qui guide leur intégration. Chaque source d'information possède un contexte qui simplifie le travail d'intégration : (1) il permet la compréhension des données partagées, (2) il supporte la réconciliation contextuelle pour comparer des contextes et identifier les informations en relation sémantique, (3) il assure la localisation des informations, (4) il facilite la génération des informations d'intégration et adapte les données à un contexte référence.

DILEMMA adopte une technique de réconciliation contextuelle indépendante d'un langage ontologique, c'est-à-dire basée sur la spécification de contextes locaux. Des informations identifiées sur des contextes hétérogènes mais en recouvrement peuvent être rapprochées sans la définition de relations inter-ontologies. La réconciliation contextuelle repose sur une technique de comparaison des concepts composant deux contextes. La similarité des concepts est évaluée en fonction de leur ressemblance taxonomique, de leurs propriétés intrinsèques et de leur voisinage sémantique (c'est-à-dire des relations entre concepts). La définition d'un concept dans DILEMMA, unité sémantique élémentaire, peut être représentée à l'aide d'une adaptation du triangle sémantique connu des applications de cognition artificielle [GRUM94]. La *Figure 1* définit un concept comme un lien entre trois notions : le référent, le signifiant et le signifié. **Le référent** est ce que représente un concept, il s'agit ici d'un objet informatif qui constitue la classe objet d'une information partagée. L'objet informatif, aussi appelé I-Objet, est l'élément de base pour échanger des données à travers notre modèle de médiation. **Le signifiant** regroupe toutes les connaissances permettant de donner une signification au concept, c'est-à-dire les informations taxonomiques (synonymes, antonymes), les propriétés (sémantique de valeurs, contraintes) et les relations avec le voisinage (subsumption, rôle). **Le signifié** est la classe conceptuelle qui représente l'abstraction d'une entité.

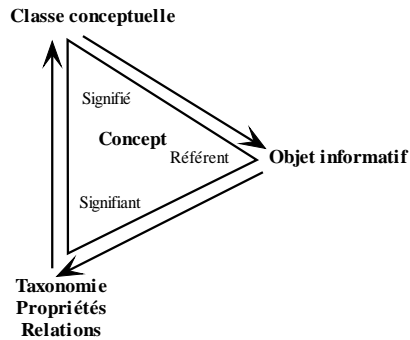


Figure 1 : Définition cognitive d'un concept dans DILEMMA

Les SI exportent des classes d'objets dans le modèle de médiation et fournissent un contexte local. Ces classes sont définies par rapport à leur contexte local, le modèle de médiation décrit donc à la fois la structure et la sémantique des données. La réconciliation sémantique entre les contextes de deux SI permet d'importer les classes pertinentes et de les intégrer.

### 3 Un scénario de coopération

Cette section décrit un exemple d'interopération entre systèmes hétérogènes. Soit une association qui désire fournir un service d'information sur les concerts d'artistes de variétés à travers l'Europe.

<b>S1(nbC, artistN, dateC, libreP, venduP, prixP)</b>	
<i>nbC (integer)</i>	: Numéro identifiant d'un concert (local au système)
<i>artistN (string)</i>	: Nom de l'artiste
<i>dateC (date)</i>	: Date du concert
<i>libreP (integer)</i>	: Nombre de places libres
<i>venduP (integer)</i>	: Nombre de places vendues
<i>prixP (float)</i>	: Prix d'une place (en Euro)
<b>S2(id, nom, seance1..n, nbP, totP, ticket)</b>	
<i>id (integer)</i>	: Numéro identifiant d'un concert (local au système)
<i>nom (string)</i>	: Nom de l'artiste
<i>seance1..n (date)</i>	: Date de la séance
<i>nbP (integer)</i>	: Nombre de places libres
<i>totP (integer)</i>	: Nombre total de places
<i>ticket (float)</i>	: Prix d'une place (en Francs)
<i>S2 possède une fonction de transformation des Francs en Euro sous la forme d'une macro <b>ftoe</b>.</i>	

Tableau 1: Schémas des SI s1 et s2.

Les informations proposées par cette association regroupent pour chaque concert le nom de l'artiste, la date du concert, le nombre de places encore libres, le prix des places et la salle où il a lieu. Un utilisateur peut interroger ce système pour obtenir une vue intégrée des informations distribuées sur diverses sources d'information. On considère, pour simplifier l'exemple, l'existence de deux systèmes *s1* et *s2* relatifs à deux salles de concert et susceptibles de donner de l'information sur des concerts. Le système *s1* est administré par un SGBD relationnel (une table S1) alors que *s2* utilise un tableau pour stocker les données (feuille de calcul S2). L'organisation des informations est décrite dans le Tableau 1.

Les sources *s1* et *s2* posent divers problèmes d'hétérogénéités. Les modèles de données sont différents, des conflits de nom apparaissent entre les entités similaires, des conflits de valeur apparaissent au niveau du prix des places, des conflits cognitifs apparaissent dans la manière de considérer les dates de concert. Chaque système commence par exporter le schéma de ses

informations dans un format spécifique à DILEMMA. La *Figure 2* décrit la démarche à suivre pour qu'une source d'information puisse coopérer dans un environnement DILEMMA. La représentation locale des données est traduite dans le modèle de médiation sous la forme de classes d'objets. Un spécialiste du domaine construit le contexte de la source d'information et lie les classes à ce contexte pour définir la sémantique de chaque classe. Le contexte peut alors être enregistré au niveau d'un annuaire qui répertorie les sites DILEMMA.

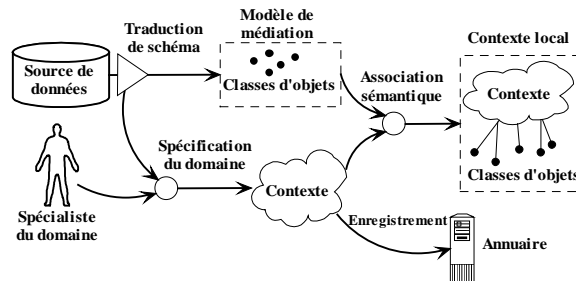


Figure 2 : Aspect fournisseur

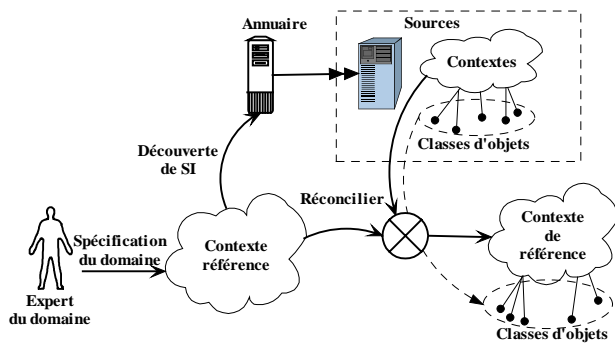


Figure 3 : Aspect consommateur

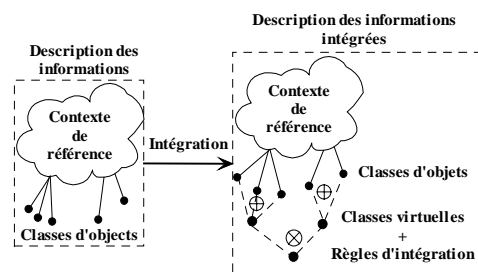


Figure 4 : Intégration des informations

L'association doit découvrir les sites potentiellement intéressants et en extraire les informations en adéquation avec ses besoins. La *Figure 3* décrit la démarche qui consiste à comparer et réconcilier des contextes pour importer des informations. Un expert du domaine est chargé de définir le contexte de l'application qui devient le contexte de référence lorsqu'un système joue le rôle de consommateur d'information. La spécification du contexte est comparée avec la totalité ou un sous-ensemble des contextes des sites enregistrés dans l'annuaire. Les contextes sources sont réconciliés avec le contexte de référence, les classes d'objets pertinentes sont importées et adaptées à la sémantique du contexte de référence. Les classes importées peuvent alors être interprétées dans le domaine de l'application.

La *Figure 4* décrit la démarche permettant l'intégration des classes d'objets dans des classes d'objets virtuelles. Les classes virtuelles assurent une transparence d'accès totale, elles fournissent une interface homogène et intégrée des informations partagées. La construction des classes d'objets virtuelles consiste à repérer les classes d'objets similaires ou en relation en utilisant le contenu du contexte de référence et à générer des règles d'appariement entre ces classes. Les classes d'objets et les règles d'intégration forment une base de connaissance statique pour la résolution de requêtes. L'aspect dynamique repose sur la possibilité d'augmenter de manière incrémentale le contenu de cette connaissance en utilisant la réconciliation contextuelle pour importer et intégrer de nouvelles classes d'objets.



## 4 Le langage de médiation

La description des informations partagées utilise un langage de médiation qui décrit d'une part les classes d'objets dans un modèle orienté objet de type F-logic [KIFE95] et d'autre part la sémantique de ces classes à travers un contexte issu de la logique terminologique [BRAC85, MAC88].

### 4.1 Représentation des classes d'objets

Une classe d'objets représente un ensemble d'objets qui répondent à une structure commune. Une classe est définie par son nom et par la liste de ses méthodes qui caractérisent les propriétés (attributs et opérations) des objets de la classe. Un objet est représenté par son état (les valeurs des propriétés) et par un identifiant interne.

Soient  $C$  une classe,  $m_{i=1..n}$  ses méthodes et  $T_{i=1..n}$  les types retournés par les méthodes, une classe s'écrit `class C[m1 ⇒ T1; ...; mi ⇒ Ti; ... mn ⇒ Tn]`. Un type  $T_i$  peut être soit un type de base (integer, float, string, date), soit une référence à une classe. Une méthode peut être multivaluée, dans ce cas elle est notée  $m_i \Rightarrow T_i$ , et paramétrée avec une ou plusieurs valeurs de type primitif ou objet  $p_{i=1..m}$ , dans ce cas elle est notée  $m_i(p_1, \dots, p_i, \dots, p_m) \Rightarrow T_i$ . Soient  $O$  un objet de la classe  $C$  et  $V_{i=1..n}$  les valeurs respectives des méthodes  $m_{i=1..n}$  pour cet objet,  $O$  est noté formellement  $O : C[m_1 \rightarrow V_1; \dots; m_i \rightarrow V_i; \dots m_n \rightarrow V_n]$ . La référence à une méthode  $m$  d'une classe  $C$  ou d'un objet  $O$  se note également sous la forme d'un chemin  $C.m$  ou  $O.m$ .

Les classes sont organisées dans une hiérarchie d'héritage. Soient deux classes  $C1$  et  $C2$ , si  $C1$  hérite de  $C2$  ( $C1$  est plus spécifique que  $C2$ ) alors  $C1$  possède les méthodes de  $C2$  et la relation d'héritage est notée  $C1::C2$ . Les méthodes héritées peuvent naturellement être redéfinies. Toutes les classes descendent d'une classe commune appelée `vfo` (Very First Object) qui possède deux méthodes, l'une `repository` pour localiser la source d'information propriétaire de la classe, l'autre `description` qui contient une définition littérale de ce que représente les objets. Les classes redéfinissent ces deux méthodes en instanciant directement leurs valeurs. La classe `vfo` est formellement définie par `class vfo[repository ⇒ string; description ⇒ string]`. Des restrictions d'accès peuvent être appliquées sur les objets : une classe `public` peut être exportée sans restriction, une classe `protected` peut être exportée une seule fois, une classe `private` est restreinte à une utilisation locale.

Les sources d'informations  $s1$  et  $s2$  fournissent les classes  $cs1$  et  $cs2$  à la coopération. Les informations purement locales comme les identifiants disparaissent, les attributs de la relation du site  $s1$  et les titres de colonnes de la feuille de calcul du site  $s2$  deviennent les méthodes des classes  $cs1$  et  $cs2$ . Les formats des nombres et des dates sont uniformisés dans les types primitifs du modèle. La représentation du nombre de séances numérotées de  $s2$  exploite les concepts du modèle en utilisant une méthode `seance` paramétrée avec le numéro de séance. La fonction de transformation de Francs en Euro de  $s2$  devient aussi une méthode paramétrée.

```

class cs1[repository => 's2.paris.fr';
    description => 'Concerts de la salle Omnisport de Paris Bercy';
    artistN => string; dateC => date; libreP => integer; prixP => float].

class cs2[repository => 's1.bruxelles.be';
    description => 'Système d'information du Forum National de Bruxelles';
    nom => string; seance(integer) => date; nbP => integer;
    ticket => float; ftoe(float) => float].

```

## 4.2 Représentation des contextes

Un contexte dans DILEMMA est un réseau sémantique, c'est-à-dire un ensemble de concepts terminologiques [BRAC85, MAC88] en relation les uns avec les autres. Comprendre un concept, c'est permettre l'interprétation des classes d'objets qui lui sont associées. Un contexte est composé de plusieurs éléments sémantiques : le **concept** qui forme l'unité sémantique de base, le **rôle** qui modélise une relation binaire entre concepts, la **subsumption** qui spécifie des relations de généralisation / spécialisation entre les concepts d'une part et les rôles d'autre part, la **taxonomie** qui définit des relations de synonymie et d'antinomie entre les termes de la terminologie du contexte, **l'interprétation** qui permet à une classe d'objet d'instancier un concept.

Formellement un contexte se décrit comme un schéma de classes conceptuelles. Une classe conceptuelle est la représentation d'un concept dans le modèle de médiation DILEMMA. Ce modèle utilise une notation similaire à la notion de classe d'objets. Si  $C$  est une classe conceptuelle et  $r_{i=1..n}$  ses rôles qui mettent en relation les classes conceptuelles  $C_{i=1..n}$  avec  $C$ , alors on peut écrire `concept C[r1 => C1; ...; ri => Ci; ... rn => Cn]`. La relation de subsumption entre deux classes  $C_1$  et  $C_2$  se note  $C_1::C_2$ . Des synonymes et des antonymes peuvent être associés aux concepts et aux rôles, la liste de synonymes  $S$  du concept  $C$  se note `synonym C {S}`. Des propriétés sont également associées aux concepts : typage des informations liées à un concept et contraintes.

Le contexte ci-dessous décrit le domaine du système *s1*. Le concept CONCERT est relié aux concepts ARTISTE, NOMBRE\_PLACE et DATE\_FRANCAISE par les rôles respectifs Affiche, Contient et DateConcert. Le concept ARTISTE est relié à CHANSON par le rôle Interprete. Le concept PLACE est relié à NOMBRE\_PLACE et PRIX par les rôles Quantite et Coute. Différentes relations de subsumption sont également définies, par exemple la classe conceptuelle MONNAIE subsume EURO et PRIX.

```

concept CONCERT[ Affiche => ARTISTE; Contient => NOMBRE_PLACE;
    DateConcert => DATE_FRANCAISE].
concept ARTISTE::PERSONNE[Interprete => CHANSON].
concept PLACE[Quantite => NOMBRE_PLACE; Coute => PRIX].
concept DATE_FRANCAISE::DATE.
concept CHANSON::MUSIQUE.
concept PLACE_LIBRE::NOMBRE_PLACE.
concept PLACE_VENDUE::NOMBRE_PLACE.
concept PRIX::EURO::MONNAIE.

```

Les classes d'objets sont des interprétations des classes conceptuelles : les classes d'objets et leurs méthodes deviennent des instances des classes conceptuelles. Si  $C$  est une classe d'objet dont la sémantique est proche du concept décrit par la classe conceptuelle  $C_1$ , si  $m$  est une méthode de  $C$  et si  $C.m$  est sémantiquement proche de la classe conceptuelle  $C_2$ , alors  $C$  est instance de  $C_1$  et  $C.m$  est instance de  $C_2$ , cette interprétation du contexte est notée



Cette section présente les objectifs de la réconciliation et la technique utilisée pour interpréter des données distantes dans un contexte utilisateur de référence. Les classes d'objets pertinentes doivent être associées au contexte de référence en tant qu'instances d'un ou plusieurs de ces concepts. La *Figure 7* présente le contexte défini par l'association du scénario qui joue le rôle de contexte de référence.

Le rapprochement sémantique est composé de trois phases : (1) la réconciliation contextuelle entre le contexte de référence et un ou plusieurs contextes sources, (2) l'importation des classes d'objets pertinentes à la vue du résultat de la réconciliation et le placement de ces classes dans le contexte de référence, (3) l'intégration des classes importées dans des classes virtuelles.

## 5.1 Technique de rapprochement

La réconciliation compare la description de deux contextes pour identifier les concepts similaires ou sémantiquement très proches. Les informations à disposition pour cette comparaison sont de type terminologique : informations taxonomiques (nom, synonymes, antonymes), relations de subsumption, ensemble de rôles. Lorsque deux concepts ont été définis comme similaires, les classes d'objets instances du concept issu du contexte source sont importées dans le système consommateur et liées au concept similaire du contexte référence. La définition d'une distance sémantique est nécessaire pour évaluer les similarités et les différences entre concepts.

### 5.1.1 Notion de distance sémantique

La distance sémantique entre deux classes conceptuelles  $C_1$  et  $C_2$ , de deux contextes différents, est le taux de ressemblance entre ces concepts. Elle se note  $d(C_1, C_2)$  avec  $d(C_1, C_2) \in \mathfrak{R}^+ / 0 \leq d(C_1, C_2) \leq 1$ . Le calcul de cette distance repose sur une analyse taxonomique des termes qui caractérisent les classes conceptuelles et par une comparaison du voisinage de celles-ci. L'analyse taxonomique utilise la liste des synonymes et des antonymes des deux classes virtuelles pour calculer un taux ressemblance taxonomique noté  $\Delta_{C_1C_2}$ . La comparaison du voisinage prend en compte à la fois les rôles et les concepts reliés à  $C_1$  et  $C_2$ . La comparaison fournit une combinaison de possibilités, chacun des  $n$  rôles de la classe conceptuelle  $C_1$  peut être comparé avec les  $p$  rôles de  $C_2$ . Chaque possibilité fournit un taux de ressemblance du voisinage  $\Delta_{\text{voisinage}}$  dont le meilleur est retenu pour calculer  $d(C_1, C_2)$ . La distance sémantique entre  $C_1$  et  $C_2$  est définie par  $d(C_1, C_2) = \Delta_{C_1C_2} \times \text{Max}(\Delta_{\text{voisinage}_i})$  avec  $i$  le nombre de combinaisons possibles. Formellement si  $S(x)$  caractérise l'ensemble des synonymes de l'entité  $x$ ,  $A(x)$  l'ensemble des antonymes de  $x$  et  $\text{card}()$  la cardinalité d'un ensemble, on définit  $\Delta_{C_1C_2}$  et  $\Delta_{\text{voisinage}_i}$  comme suit :

$$\begin{cases} \Delta_{C_1C_2} = 0 \text{ si } \text{card}(S(C_1) \cap A(C_2)) \neq 0 \text{ ou } \text{card}(S(C_2) \cap A(C_1)) \neq 0, \\ \text{sinon} \\ \Delta_{C_1C_2} = \text{card}(S(C_1) \cap S(C_2)) / \text{card}(S(C_1)) \\ \text{et } \Delta_{\text{voisinage}_i} = (\sum \Delta_{r_1r_2} \cdot \Delta_{C_{1r_1}C_{2r_2}}) / (\sum \Delta_{r_1r_2}) \end{cases}$$

avec  $\Delta_{r_1r_2}$  le taux de ressemblance entre le rôle  $r_1$  de  $C_1$  et le rôle  $r_2$  de  $C_2$ ,  $\Delta_{C_{1r_1}C_{2r_2}}$  le taux de ressemblance des classes  $C_{1r_1}$  et  $C_{2r_2}$ .  $C_{1r_1}$  (resp.  $C_{2r_2}$ ) est la classe conceptuelle en relation avec  $C_1$  (resp.  $C_2$ ) par le rôle  $r_1$  (resp.  $r_2$ ). Une combinaison de voisinage  $\Delta_{\text{voisinage}_i}$  est la somme des taux de ressemblance entre les classes conceptuelles reliées aux concepts  $C_1$  et  $C_2$  pondérées

par le taux de ressemblance des rôles mis en jeu. Les valeurs  $\Delta_{r_1r_2}$  et  $\Delta_{C_{1r_1}C_{2r_2}}$  sont calculées sur la base d'une analyse taxonomique :

$$\left\{ \begin{array}{l} \Delta_{r_1r_2} = \text{card}(S(r_1) \cap S(r_2)) / \text{card}(S(r_1)) \\ \text{et} \\ \Delta_{C_{1r_1}C_{2r_2}} = d(C_{1r_1}, C_{2r_2}) \quad (\text{définition récursive}) \end{array} \right.$$

### 5.1.2 Algorithme de rapprochement contextuel

La distance sémantique entre classes conceptuelles permet la découverte des classes similaires, c'est-à-dire sémantiquement proches. L'application de l'algorithme de rapprochement contextuel généralise ce calcul de distance à l'ensemble des concepts constituant les contextes à rapprocher. Chaque concept d'un contexte peut être comparé à tous les concepts d'un autre contexte, un grand nombre de combinaison de rapprochement est donc envisageable. Le taux de correction du rapprochement  $\Delta_{\text{contexte}}$  est calculé en fonction de la somme des distances sémantiques entre les concepts comparés :

$$\Delta_{\text{contexte}} = (\sum d(C_1, C_2)) / n_1$$

avec  $n_1$  le nombre de concepts dans le contexte de référence. Le rapprochement fournissant le meilleur  $\Delta_{\text{contexte}}$  est retenu et les classes d'objets du contexte fournisseur qui correspondent aux concepts ayant fusionnés avec le contexte de référence sont reliées à ce contexte de référence.

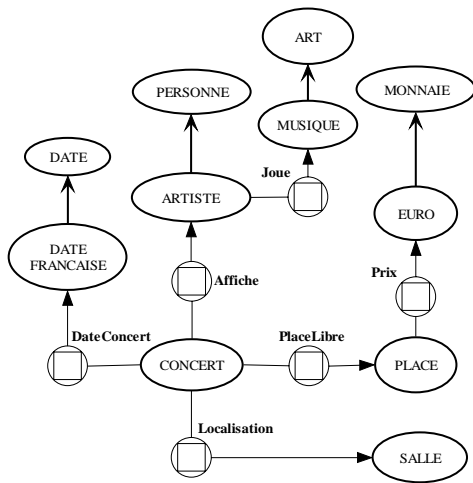


Figure 7 : Contexte de référence

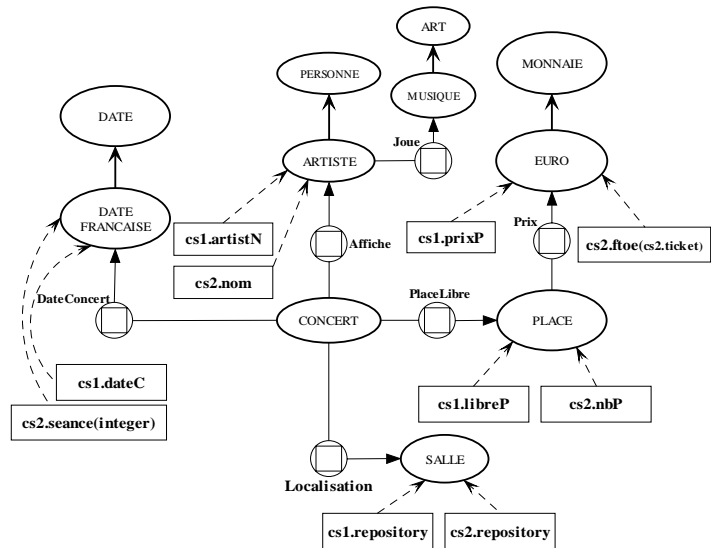


Figure 8 : Contexte référence après rapprochement

La Figure 8 présente le résultat du rapprochement contextuel du contexte de l'association (contexte de référence) avec chacun des contextes des SI  $s_1$  et  $s_2$ . Les informations importées possèdent maintenant une signification sur le contexte de référence de l'application qui désire les utiliser.

Le nombre de rapprochements possibles peut croître très vite, une phase de pré-analyse est alors préférable pour éliminer très tôt les cas improbables. Tous les cas sont d'abord évalués en limitant le calcul de la distance sémantique à l'analyse taxonomique sans étude de voisinage. La définition d'un seuil  $\delta$  permet de ne retenir que les rapprochements dont  $\Delta_{\text{contexte}} \geq \delta$  pour approfondir la comparaison sur le voisinage.

## 5.2 Fusion d'objets

### 5.2.1 Règles d'intégration

Le modèle de médiation permet de définir trois types de règles d'intégration : les règles de contraintes qui réduisent les objets à considérer en fonction de prédicats, les règles de fusion qui agrègent les instances de classes d'objets similaires dans des instances d'une classe virtuelle et les règles de jointure qui combinent les informations de plusieurs classes d'objets en fonction d'une ou plusieurs propriétés communes. Seules les règles de fusion sont abordées dans ce papier.

Soient  $C_1$  et  $C_2$  des classes d'objets importées respectivement de  $s1$  et  $s2$ , soient  $M(C_1)$  et  $M(C_2)$  l'ensemble des méthodes de  $C_1$  et  $C_2$ , si  $C_1$  et  $C_2$  sont des instances du concept  $C$ , alors  $\exists C_v$  une classe virtuelle telle que  $C_v = C_1 \oplus C_2$  avec  $M(C_v) = M(C_1) \cap_s M(C_2)$ . L'opérateur d'intersection  $\cap_s$  définit l'intersection sémantique des méthodes de deux classes.

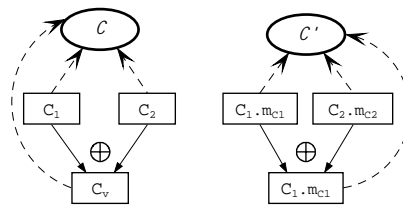


Figure 9 : Fusion des classes et des méthodes similaires

Formellement le résultat de  $M(C_1) \cap_s M(C_2)$  est un ensemble de triplets  $\langle m_{C_v}, m_{C_1}, m_{C_2} \rangle$  avec  $m_{C_v} \in M(C_v)$ ,  $m_{C_1} \in M(C_1)$  et  $m_{C_2} \in M(C_2)$  tels que si  $m_{C_1}:C'$  et  $m_{C_2}:C'$  alors  $m_{C_v}:C'$  et  $m_{C_v}$  fusionne les méthodes  $m_{C_1}$  et  $m_{C_2}$ . La Figure 9 résume la fusion des classes dans une classe virtuelle. Si le typage associé au niveau des propriétés du concept  $C'$  est  $T'$  alors la classe virtuelle  $C_v$  et son interprétation s'écrivent en tenant compte de  $n$  triplets  $\langle m_{C_v}, m_{C_1}, m_{C_2} \rangle$  :

```
interpretation C_v:C[m_{1C_v}:C'_1; ...; m_{nC_v}:C'_n].
class C_v[m_{1C_v} => T'_1; ...; m_{nC_v} => T'_n].
```

La Figure 8 de l'exemple laisse apparaître que les classes  $cs1$  et  $cs2$  sont les instances d'un même concept CONCERT. Pratiquement, le nom d'une classe  $C_v$  reprend celui de la classe conceptuelle dont  $C_v$  est instance et le nom d'une méthode  $m_{C_v}$  est obtenu à partir des rôles qui relient  $C$  à  $C'$ . Une classe virtuelle Concert est donc créée, les noms de méthodes correspondent aux rôles de CONCERT.

```
interpretation Concert:CONCERT[ Affiche:ARTISTE; DateConcert:DATE_FRANCAISE;
                               PlaceLibre:PLACE; Prix:EURO;
                               Localisation:SALLE => string].

class Concert[Repository => 'assoc.com';
              Description => cs1.Description + cs2.Description;
              Affiche => string; DateConcert => date; PlaceLibre => integer;
              Prix => integer; Localisation => string].
```

Si  $C_1$  et  $C_2$  sont fusionnées dans une classe virtuelle  $C_v$ , alors deux règles de fusion sont définies pour que les objets de chacune des deux classes puissent être accédés par la classe virtuelle. Pour chaque triplet  $\langle m_{C_v}, m_{C_1}, m_{C_2} \rangle$  de l'intersection sémantique les méthodes  $m_{C_1}$  et  $m_{C_2}$  sont

mappées sur la méthode `mCV`. Une règle de fusion est une règle d'unification dans laquelle la tête définit la classe virtuelle et le corps représente une classe dans laquelle récupérer les instances. Les règles suivantes créent un objet `X` instance de `CV` si `Y` instance de `C1` ou `C2` et les valeurs des méthodes de `CV` sont calculées à partir des valeurs des méthodes de `C1` ou `C2`.

```
rule {X, Y, Vi=1..n}
X:CV[ ...; Mi → Vi; ...] :- Y:C1[ ...; m1,j → Vi; ...].
X:CV[ ...; Mi → Vi; ...] :- Y:C2[ ...; m2,k → Vi; ...].
```

Les objets des classes `cs1` et `cs2` sont fusionnés dans la classe virtuelle `Concert`. Les variables permettent d'unifier les valeurs des objets. Le paramètre `_` de la méthode `seance(_)` permet de récupérer tous les concerts pour tous les numéros de séance. A noter l'utilisation de la méthode `ftoe` de la classe `cs2` pour adapter le prix au format attendu par `Concert`.

```
rule {X, A, D, PL, P, L F}
X:Concert[Affiche → A; DateConcert → D; PlaceLibre → PL; Prix → P;
  Localisation → L]:-
  Y:cs2[repository → L; nom → A; seance(_)-> D; nbP → PL;
    ticket → F; ftoe(F) → P].

rule {X, A, D, PL, P, L F}
X:Concert[Affiche → A; DateConcert → D; PlaceLibre → PL; Prix → P;
  Localisation → L]:-
  Y:cs1[repository → L; artistN → A; dateC → D; libreP → PL;
    prixP → P].
```

La classe virtuelle peut être interrogée comme toute classe d'objet, assurant une transparence complète quant à la localisation et à la résolution des conflits de données. L'ajout ou la suppression d'un SI se traduit simplement par l'ajout ou la suppression d'une règle.

### 5.2.2 Inférence d'une requête

Une requête fournit les objets de la classe interrogée, si cette classe est virtuelle des règles d'intégration sont inférées pour construire dynamiquement des objets de la classe virtuelle à partir des objets qui composent la classe virtuelle. L'exemple suivant demande toutes les informations sur les concerts de Céline Dion dont il reste encore des places. Cette requête est inférée sur les faits et les règles définies au niveau du médiateur sur lequel elle est posée.

```
query {D, PL, P, L}
X:Concert[Affiche → 'Céline Dion'; DateConcert → D; PlaceLibre → PL;
  Prix → P; Localisation → L], PL > 0.
```

Le médiateur décompose la requête initiale en deux sous requêtes en prenant en compte les deux règles de fusion. Chacune des règles instancie la requête et se comporte comme une demande d'objets pour les sites participants qui ont été pris en compte dans le processus d'intégration. Des demandes d'objets de la classe `cs1` et `cs2` sont envoyées au médiateur qui gère les SI `s1` et `s2`. Chacun des médiateurs distants prend en charge la résolution de sa sous-requête. Une sous-requête peut elle-même être une demande d'objets sur une classe virtuelle, chaque médiateur résout la partie connue localement de la résolution globale.

```
query {D, PL, P}
```

```
X:cs2[nom → 'Céline Dion'; seance(_)→ D; nbP → PL; ticket → F;  
      ftoe(F) → P], PL > 0.
```

```
query {D, PL, P}
```

```
X:cs1[artistN → 'Céline Dion'; dateC → D; libreP → PL; prixP → P], PL > 0.
```

Chacune de ces sous-requêtes est traitée directement par les médiateurs des systèmes *s1* et *s2*. Les objets suivants sont renvoyés par chacun des SI. Les valeurs des informations sont homogènes mais les objets restent hétérogènes.

```
{[nom → 'Céline Dion'; seance(1)→ 02/02/00; nbP → 2000;      ticket → 290;  
  ftoe(290) → 44],
```

```
[nom → 'Céline Dion'; seance(2)→ 03/02/00; nbP → 500;      ticket → 290;  
  ftoe(290) → 44]}
```

```
{[artistN → 'Céline Dion'; dateC → 22/01/00; libreP → 4000; prixP → 53]}
```

Le médiateur du système de l'association recompose les résultats dans des objets de la classe virtuelle pour obtenir des réponses syntaxiquement, schématiquement et sémantiquement homogènes.

```
{[Affiche → 'Céline Dion'; DateConcert → 02/02/00; PlaceLibre → 2000;  
  Prix → 44; Localisation → 's2.paris.fr'],
```

```
[Affiche → 'Céline Dion'; DateConcert → 03/02/00; PlaceLibre → 500;  
  Prix → 44; Localisation → 's2.paris.fr'],
```

```
[Affiche → 'Céline Dion'; DateConcert → 22/01/00; PlaceLibre → 4000;  
  Prix → 53; Localisation → 's1.bruxelles.be']}]
```

## 6 Conclusion et perspectives

Nous avons présenté dans cet article une solution de médiation de schéma étendue, appelée DILEMMA, pour l'interopérabilité de systèmes d'information. Cette approche utilise un modèle qui décrit les informations partagées en prenant en compte leur sémantique sous la forme de contextes d'utilisation. Une méthodologie basée sur le rapprochement contextuel et une architecture flexible permettent une intégration incrémentale des informations pertinentes dans le cadre d'applications web. La définition de contextes locaux, du rapprochement contextuel et d'une distance sémantique réalisent l'interprétation, la localisation et l'utilisation d'informations partagées et hétérogènes sans nécessiter la définition d'une ontologie. Nos travaux actuels vont dans le sens d'une formalisation plus poussée de la notion de distance sémantique et s'orientent vers l'implémentation de l'architecture.

## Références

- [BRAC85]: RONALD J. BRACHMAN, JAMES G. SCHMOLZE, "An Overview of the KL-ONE Knowledge Representation System". *Cognitive Science*, Volume 9, n°2, 1985, p. 171-216.
- [BRES97]: S. BRESSAN, C. HIAN GOH, K. FYNN, M. JAKOBISIAK, K. HUSSEIN, HENRY B. KON, T. LEE, STUART E. MADNICK, T. PENNA, J. QU, ANNIE W. SHUM, M. SIEGEL, "The COntext INterchange Mediator Prototype". Proceedings of ACM SIGMOD International Conference on Management of Data, 1997, p525-527.



- [BISH97]: Y. BISHR, " Semantic Aspects of Interoperable GIS". Phd Thesis, ITC Publication Series, Enschede Netherlands, ISBN 90-6164-1411, 1997.
- [CODY95]: WILLIAM F. CODY, Laura M. Haas, W. Niblack, M. Arya, Michael J. Carey, R. Fagin, M. Flickner, D. Lee, D. Petkovic, Peter M. Schwarz, J. Thomas II, Mary T. Roth, John H. Williams, Edward L. Wimmers, "Querying Multimedia Data from Multiple Repositories by Content: the Garlic Project". Proceedings of the third IFIP 2.6 working conference on Visual Database Systems, 1995, p17-35.
- [DECK99]: S. DECKER, M. ERDMANN, D. FENSEL, R. STUDER, "Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information". Eighth Working Conference on Database Semantics, DS-8, Kluwer, 1999, p. 351-369.
- [FIKE97]: R. FIKES, A. FARQUHAR, J. RICE, "Tools for Assembling Modular Ontologies in Ontolingua". Technical Report, Computer Science Department, Stanford University, Knowledge Systems Laboratory, KSL-97-03, 1997.
- [FLOR95]: D. FLORESCU, L. RACHID, P. VALDURIEZ, "Using Heterogeneous Equivalences for Query Rewriting in Multidatabase Systems". Proceedings of the Third International Conference on Cooperative Information Systems, 1995, p158-169.
- [FOWL99]: J. FOWLER, B. PERRY, M. H. NODINE, B. BARGMEYER, "Agent-Based Semantic Interoperability in InfoSleuth". ACM SIGMOD Record, Volume 28, n°1, March 1999, p. 60-67.
- [GINS91]: MATTHEW L. GINSBERG, "Knowledge Interchange Format, the KIF of Death". AI Magazine, Volume 12, n°3, 1991, p. 57-63.
- [GOH94]: C. GOH AND S. MADNICK AND M. SIEGEL, "Context Interchange: Overcoming the Challenges of Large-scale Interoperable Database Systems in a Dynamic Environment". Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94), Gaithersburg, Maryland, USA, ACM Press, ISBN 0-89791-674-3, December 1994, p. 337-346.
- [GRUB93]: T. GRUBER, "A Translation Approach to Portable Ontology Specifications". International Journal of Knowledge Acquisition for Knowledge-based Systems, Volume 5, n°2, June 1993.
- [GRUM94]: A. GRUMBACH, "Cognition artificielle. Du réflexe à la réflexion". Ed. Addison-Wesley France, ISBN 2-87908-059-2, 1994.
- [GUAR97]: N. GUARINO, "Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration", Summer School on Information Extraction, Frascati, Italy, July 1997 1997, Springer Verlag, p. 139-170.
- [KASH94]: V. KASHYAP, A. SHETH, " Semantics-Based Information Brokering". Proceedings of the 3rd International Conference on Information and Knowledge Management, Gaithersburg, Maryland, USA, ACM Press, ISBN 0-89791-674-3, December 1994, p. 363-370.
- [KIFE95]: M. KIFER, G. LAUSEN, J. WU, "Logical Foundations of Object-Oriented and Frame-Based Languages". Journal of ACM, Volume 42, n°4, 1995, p. 741-843.
- [LIU95a]: L. LIU, C. PU, "The DIOM Approach to Large-scale Interoperable Database Systems ". Technical report TR95-16, Department of Computing Science, University of Alberta, 1995.

- [LIU97]: L. LIU, C. PU, "Dynamic Query Processing in DIOM". IEEE Data Engineering Bulletin, Volume 20, n°3, 1997, p449-457.
- [MAC88]: R. MACGREGOR, "A Deductive Pattern Matcher", Proceedings of the 7th National Conference on Artificial Intelligence (AAAI'88), St. Paul, Minnesota, USA, AAAI Press / The MIT Press, ISBN 0-262-51055-3, August 1988, p. 403-408.
- [MENA99]: E. MENA, "OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies". Phd Thesis, University of Zaragoza, Spain, February 1999.
- [MOLI95]: H. GARCIA-MOLINA, D. QUASS, Y. PAPAKONSTANTINOY, A. RAJARAMAN, Y. SAGIV, JEFFREY D. ULLMAN, J. WIDOM, "The TSIMMIS Approach to Mediation: Data Models and Languages". Proceedings of the Second International Workshop on Next Generation Information Technologies and Systems, 1995.
- [NICO97]: C. NICOLLE, N. CULLOT, K. YETONGNON, "A Translation Process Between Cooperating Heterogeneous Database Systems". Proceedings of the 10th International Conference on Parallel and Distributed Computing Systems (PDCS'97), ISCA, New Orleans, Louisiana, USA, October 1997, p. 619-624.
- [OUKS99]: ARIS M. OUKSEL, I. AHMED, "Ontologies are not the Panacea in Data Integration, A Flexible Coordinator to Mediate Context Construction". Journal of Distributed and Parallel Databases, Volume 7, n°1, 1999, p. 7-35.
- [PAPA96]: Y. PAPAKONSTANTINOY, S. ABITEBOUL, H. GARCIA-MOLINA, "Object Fusion in Mediator Systems". Proceedings of 24rd International Conference on Very Large Data Bases, 1996, p413-424.
- [TOMA96]: A. TOMASIC, L. RASCHID, P. VALDURIEZ, "Scaling Heterogeneous Databases and the Design of Disco". Proceedings of the 16th International Conference on Distributed IEEE Computing Systems, IEEE Computer Society Press, May 1996, p449-457.
- [SHET90]: AMIT P. SHETH, JAMES A. LARSON, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases". ACM Computing Surveys, Volume 22, n°3, September 1990, p183-236.
- [VISS99a]: PEPIJN R. S. VISSER, MARTIN D. BEER, TREVOR J. M. BENCH-CAPON, B. M. DIAZ, MICHAEL J. R. SHAVE, "Resolving Ontological Heterogeneity in the KRAFT Project". 10th International Conference and Workshop on Database and Expert Systems Applications (DEXA'99), Florence, Italy, August 1999, p. 668-677.
- [VISS99b]: P.R.S. VISSER AND V.A.M. TAMMA, "An Experience with Ontology-Based Agent Clustering". Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends (IJCAI'99), Stockholm, Sweden, August 1999.
- [WIED92]: G. WIEDERHOLD, "Mediators in the Architecture of Future Information Systems". IEEE Computer Magazine, Volume 25, n°3, March 92, p38-49.
- [WIED95]: G. WIEDERHOLD, "The Basis for Mediation". Proceedings of the Third International Conference on Cooperative Information Systems, 1995, p140-157.
- [XML]: W3C, Extensible Markup Language (XML), <http://www.w3.org/XML>.