HAL
open science

Mediation and Data Source Selection for Large Scale Virtual Organizations

Alexandra Pomares

To cite this version:

Alexandra Pomares. Mediation and Data Source Selection for Large Scale Virtual Organizations. Databases [cs.DB]. Université de Grenoble, 2010. Español. NNT : . tel-00717263

HAL Id: tel-00717263

https://theses.hal.science/tel-00717263v1

Submitted on 12 Jul 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MEDIACIÓN Y SELECCIÓN DE FUENTES DE DATOS EN ORGANIZACIONES VIRTUALES DE GRAN ESCALA

Universidad de los Andes

Alexandra Pomares Quimbaya

Disertación enviada como requisito para optar por el título de Doctor en Ingeniería

Supervisado por
Claudia Roncancio
y
José Abásolo

Jurado

Claudia Roncancio,	Director
José Abásolo,	Director
Rubby Casallas,	Presidente
Marta Rukoz,	Revisor
Sandra de Amo,	Revisor
Marta Millán,	Examinador

Julio de 2010

UNIVERSITÉ DE GRENOBLE

N ° attribué par la bibliothèque

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UDG

Spécialité: « Informatique »

préparée au laboratoire **LOGICIELS, SYSTÈMES, RÉSEAUX (LSR) – IMAG**

dans le cadre de l'École Doctorale

« Mathématiques, Sciences et Technologies de l'Information, Informatique (MSTII) »

présentée et soutenue publiquement par

Alexandra POMARES QUIMBAYA

Le 26 juillet 2010

Titre :

Médiation et sélection de sources de données pour des organisations virtuelles distribuées à grande échelle

Directeurs de thèse :

Mme. Claudia RONCANCIO et M. José ABÁSOLO

JURY

Mme. Rubby CASALLAS,	Président
Mme. Marta RUKOZ,	Rapporteur
Mme. Sandra DE AMO,	Rapporteur
Mme. Claudia RONCANCIO,	Directeur de thèse
M. José ABÁSOLO,	Co-directeur de thèse
Mme. Marta MILLÁN,	Examinateur

a Marco y Mariana

Agradecimientos

Agradezco a la doctora Rubby Casallas, profesora de la Universidad de los Andes y vicedecana de posgrados e investigación de la Facultad de Ingeniería por haber aceptado presidir mi jurado de tesis. Así mismo, agradezco a la doctora Marta Rukoz, profesora en la Universidad Paris Ouest Nanterre La Défense y a la doctora Sandra de Amo, profesora de la Universidad de Uberlândia, por haber aceptado ser revisoras de esta tesis. Agradezco todos sus comentarios y sugerencias que me permitieron mejorar este documento. Agradezco también a la doctora Marta Millán, profesora de la Universidad del Valle, por haber aceptado ser parte de los miembros del jurado y por todos sus comentarios constructivos que me permitieron mejorar mi trabajo.

Agradezco a la Pontificia Universidad Javeriana por haberme permitido llevar a cabo este sueño. Gracias, no sólo por su apoyo económico sino por su invaluable apoyo profesional. Agradezco a Germán Chavarro, Cesar Bustacara, Enrique González y María del Mar Angulo por toda su colaboración y diligencia durante el desarrollo de mi doctorado.

Agradezco al doctor Jean-Pierre Giraudin, profesor de la Universidad Pierre Mèndes France, y a la doctora Christine Verdier, profesora de la Universidad Joseph Fourier, por haberme acogido con tanta calidez en el equipo SIGMA del laboratorio de informática de Grenoble. Así como también al doctor Harold Castro, director del equipo COMMIT en la Universidad de los Andes, por su constante colaboración.

Especialmente agradezco a la doctora Claudia Roncancio, profesora del INPG, por su invaluable apoyo y colaboración, por toda la confianza que depositó en mi durante estos años y por sus invaluables enseñanzas a nivel profesional y personal. Así mismo, agradezco al Doctor José Abásolo, profesor de la Universidad de los Andes, por toda su ayuda, comprensión y disponibilidad durante el desarrollo de esta tesis. Sin sus aportes, discusiones y su paciencia no hubiera sido posible el desarrollo de este trabajo. Gracias a los dos por sus consejos y su amistad.

Agradezco también a la doctora María del Pilar Villamil profesora de la Universidad de los Andes, al doctor Cyril Labbée, profesor de Universidad Joseph Fourier y al doctor Van-Dat Cung, profesor del INPG, por sus aportes y ayuda durante el desarrollo de esta tesis. Así mismo, a la doctora Claudia Jiménez, profesor de la Universidad de los Andes y al doctor Yves Denneulin, profesor del INPG, por sus aportes y apoyo que me permitieron enriquecer este trabajo.

Quisiera también agradecer a todos mis amigos y colegas que de alguna u otra forma aportaron al desarrollo de esta tesis. En especial quiero agradecer a Amal, Charlotte, Amin, Jorge Luis, Anseem, Raja, Diana, Gabriel, Sattisvar, Carlos, Carlos Jaime, Lulú, Luz María, Carole, Mouaidad, Giang, Marcia, Walter y Lina María por todo lo que compartimos, por apoyarme y escucharme en los momentos más difíciles. En Colombia agradezco a Enrique, Natalia, Diego, Jorge, Eduardo, Adriana, John, Mario, Oscar por su ayuda incondicional. A todas las personas que me apoyaron durante mi estancia en Francia a Carine, Linda, Manu, Yolanda, Germán, Pablo, Natalia y Mateo mil gracias por su

compañía y ayuda.

Finalmente, agradezco a toda mi familia porque sólo con su apoyo y comprensión pude lograr esta meta. Para Marco y Mariana faltan palabras para expresar la gratitud que siento hacia ustedes, los amo con todo mi corazón. Gracias hija por que fuiste el motor que siempre me dio fuerzas para seguir adelante, gracias por tu compañía en Francia, tu vida me llena de alegría. Marco gracias por todo el amor que me has dado, por tu paciencia, por la soledad que viviste para que yo lograra este sueño. Te amo. A mis hermanos Maritza, Camilo y Javier gracias por su ayuda y comprensión. A mis padres gracias, los quiero con toda mi alma, gracias por sus enseñanzas y por la fuerza que siempre me han dado para seguir adelante. Papi muchas gracias por todas esas llamadas. A mi familia política mil gracias por toda su ayuda. A mis suegros por su apoyo incondicional. A mis cuñados y cuñadas por su colaboración cada vez que lo necesité.

Índice general

1. Introducción **1**
1.1. Motivación: Integración de Datos en Organizaciones Virtuales . . 1
1.2. Contexto: Mediación de Información Heterogénea 3
1.3. Problema y Objetivos de la Investigación 5
1.3.1. Problema de Investigación: Optimización de la Selección de Fuentes de Datos en Organizaciones Virtuales 5
1.3.2. Objetivos de Investigación 5
1.4. Contribución 6
1.5. Organización del Documento 7

I Organizaciones Virtuales y Estado del Arte en el Procesamiento Distribuido de Consultas **9**

2. Organizaciones Virtuales **10**
2.1. Introducción 10
2.2. Definición 10
2.3. OV en el Sector Salud 12
2.4. Características de los Contextos de Datos en las Organizaciones Virtuales de Gran Escala 17
2.5. Arquitectura de Información para OV de Gran Escala 18
2.6. Síntesis 19

3. Procesamiento de Consultas en Contextos Distribuidos y Heterogéneos **20**
3.1. Introducción 20
3.2. Sistemas de Mediación 20
3.2.1. Arquitectura Lógica de Referencia 21
3.2.2. Funciones del Nivel de Adaptación 23
3.2.3. Funciones del Nivel de Mediación 24
3.3. Mediación en Comunidades Virtuales 30
3.3.1. Dimensiones del Estudio 30
3.3.2. Web Oculta (*Deep Web*) 32
3.3.3. Sistemas Peer to Peer(P2P) 35

3.3.4. Mallas de Datos 42
3.4. Síntesis 44

4. Selección de Fuentes de Datos 46
4.1. Introducción 46
4.2. Estrategias Orientadas a Fuentes de Datos Estructuradas 47
4.2.1. Orientadas a Capacidades 47
4.2.2. Orientadas a Calidad 50
4.2.3. Orientadas a Oferta y Demanda 53
4.2.4. Flexible (Multi-Utilidad) 54
4.3. Estrategias Orientadas a Fuentes de Datos no Estructuradas . . . 55
4.3.1. Orientados a Fuentes Cooperativas 56
4.3.2. Orientados a Fuentes No Cooperativas 57
4.4. Síntesis 58

II Selección de Fuentes de Datos en Organizaciones Virtuales de Gran Escala 61

5. Una Arquitectura de Mediación para Organizaciones Virtuales: ARIBEC 62
5.1. Introducción 62
5.2. Principios de Diseño de ARIBEC 64
5.3. Arquitectura de Datos de ARIBEC 69
5.3.1. Niveles de Abstracción de los Datos 69
5.3.2. Catálogo de VDOs 70
5.3.3. Base de conocimiento de la OV 73
5.3.4. Otros Elementos 73
5.4. Arquitectura Funcional de ARIBEC 74
5.4.1. Componentes del Nivel de Mediación 75
5.4.2. Proceso de Evaluación de Consultas 78
5.5. Base de Conocimiento de la Organización Virtual 80
5.5.1. Estructura de la Base de Conocimiento de la Organización Virtual 80
5.5.2. Representación de Metadatos en la Base de Conocimiento 83
5.5.3. Roles de Conocimiento de las Fuentes de Datos 84
5.5.4. Inserción de Metadatos en la Base de Conocimiento . . . 85
5.5.5. Consultas sobre la Base de Conocimiento de la OV 87
5.5.6. Creación y Actualización de la Base de Conocimiento . . 88
5.6. Síntesis 89

6. OptiSource: Estrategia de Selección de Fuentes de Datos para Organizaciones Virtuales 91
6.1. Selección de Fuentes de Datos como un Problema de Decisión . . 92
6.1.1. Objetivo de la Selección 92

6.1.2. Implicaciones de la Fragmentación no Disyunta en el Proceso de Selección de Fuentes 93
6.1.3. Complejidad del Problema de Selección 95
6.2. Estrategia Propuesta: OptiSource 96
6.2.1. Principio de OptiSource: Fuentes Dominantes 96
6.2.2. OptiSource al interior de ARIBEC 98
6.3. Estimación del Beneficio de Usar una Fuente de Datos 100
6.3.1. Aproximación Intuitiva 101
6.3.2. Aproximación Optimista 101
6.3.3. Aproximación usando Roles 102
6.3.4. Aproximación usando Roles y Volumen 106
6.4. Conformación de los Conjuntos de Integración 107
6.4.1. Identificación de Conjuntos de Integración 107
6.4.2. Cartografía de la Consulta 108
6.5. Optimización de la Asignación de Condiciones 110
6.5.1. Selección de Fuentes como un Problema de Asignación . . 111
6.5.2. Modelo Matemático 111
6.5.3. Identificación de Fuentes de Datos Dominantes 113
6.6. Relación de OptiSource con el componente de Coordinación de la Ejecución 114
6.6.1. Coordinación de la Ejecución 114
6.6.2. Uso de los resultados de la coordinación de ejecución . . . 115
6.7. Identificación de Roles de Fuentes en OV 115
6.7.1. Interpretación de Procesos de Negocio 116
6.7.2. Identificación de Roles Usando Minería de Datos 121
6.7.3. Interpretación del Procesamiento de Consultas 121
6.8. Síntesis 122

7. Selección de Fuentes MultiEscala para Organizaciones Virtuales 123
7.1. Problemática: Desequilibrio entre el Tiempo de Planeación y el Beneficio Obtenido 124
7.2. Contexto de Datos de la OV 125
7.2.1. Perfil de Datos: Una Abstracción del Contexto de Datos de la OV 126
7.2.2. Configuración para Obtener el Contexto 128
7.2.3. Atributos Derivados 129
7.2.4. Patrones del Perfil de Datos 130
7.2.5. Principio del Proceso de Selección de Fuentes Multiescala 130
7.3. Clasificación de Estrategias de Acuerdo a los Patrones del Perfil de Datos 132
7.4. Algoritmo de Elección de la Estrategia de Selección de Fuentes de Datos 133
7.5. Síntesis 135

III Implementación y Validación de la Propuesta 136

8. Análisis y Evaluación de OptiSource 137
8.1. Evaluación de Precisión y Exhaustividad de OptiSource 137
8.1.1. Métricas Evaluadas 138
8.1.2. Metodología de Experimentación 140
8.1.3. Resultados Obtenidos 142
8.2. Análisis de Sensibilidad de OptiSource 149
8.2.1. Metodología de Análisis 149
8.2.2. Resultados de la Experimentación 150
8.3. Comparación de OptiSource con Otras Estrategias 152
8.3.1. iDrips 152
8.3.2. Estrategias de selección aplicadas a OV de gran escala . . 156
8.4. Prototipo 158
8.4.1. Implementación de Componentes 158
8.4.2. Elección de Lenguajes y Marcos de Trabajo 159
8.5. Síntesis 160

9. Conclusiones y Perspectivas 162
9.1. Contribuciones 162
9.2. Conclusiones y Perspectivas de Investigación 165

A. Publicaciones Realizadas 179

B. Modelo de Optimización 181

C. Base de Conocimiento y Metadatos en OptiSource 182
C.1. Clases Raíz de la Base de Conocimiento 182
C.2. Recursos de la Organización Virtual 182
C.2.1. Clases Asociadas 183
C.2.2. Propiedades *DataType* 183
C.3. Unidades de la Organización Virtual 183
C.3.1. Clases Asociadas 183
C.3.2. Propiedades *DataType* 184
C.4. Conceptos de Dominio Organización Virtual: El Caso de la Salud 184
C.5. Propiedades *ObjectType* 185
C.5.1. Propiedades que Relacionan Recursos y Unidades 185
C.5.2. Propiedades que Relacionan Unidades y Unidades 186
C.5.3. Propiedades que Relacionan Recursos de datos y Recursos de datos 186
C.5.4. Propiedades que Relacionan Recursos y Conceptos de Dominio 186
C.5.5. Propiedades que Relacionan Unidades y Unidades 187

Índice de figuras

2.1. Proceso de Negocio Generador de Réplicas de Pacientes 15
2.2. Contexto de Datos en una OV en el Sector Salud 16

3.1. Arquitectura de Referencia de los Sistemas de Mediación 21
3.2. Ejecución de Consultas en la Arquitectura de Referencia de Mediación 22

4.1. Clasificación de las Técnicas de Selección de Fuentes de Datos . . 59
4.2. Ausencia de Estrategia de Selección para Contextos con Replicación y Solapamiento 60

5.1. Niveles de Abstracción de los Datos en ARIBEC 70
5.2. Ejemplo de Niveles de Abstracción de los Datos 71
5.3. Arquitectura de Datos de ARIBEC 72
5.4. Ejemplo de VDO 73
5.5. Consulta usando interfaz VDO 74
5.6. Consulta usando directamente el lenguaje de consulta 74
5.7. Arquitectura Funcional de ARIBEC 75
5.8. Diagrama de Secuencia de Evaluación de Consultas en ARIBEC 79
5.9. Proceso de Reescritura Independiente por Grupos de Fuentes . . 80
5.10. Clases Núcleo de la Base de Conocimiento de ARIBEC 81
5.11. Porción de la taxonomía creada a partir de la clase *VODomainConcept* 82
5.12. Relaciones entre individuos de la clase VODomainConcept e individuos de la clase VOUnit y VOResource 84
5.13. Clases, instancias y hechos de conocimiento en la base de conocimiento 85
5.14. Inserción de Relación EsAutoridad 86
5.15. Inserción de Relación EsEspecialista 86
5.16. Inserción de Relación EsContenedor 87
5.17. Respuesta de la Base de Conocimiento a las Consultas 88

6.1. Problemática de Integración de Fuentes en OV con Fragmentación 93
6.2. Dominancia entre Fuentes de Datos 97
6.3. Fases de OptiSource 98

6.4. Proceso de Selección Iterativo con OptiSource 99
6.5. Aproximaciones para estimar el beneficio 101
6.6. Rango del Factor del Rol 105
6.7. Cartografia en OptiSource 110
6.8. Atributos Extendidos 117
6.9. Proceso de Negocio para una OV en Salud 118
6.10. Mapa de Replicación para la OV en Salud 119

7.1. Visibilidad de los Lentes 127
7.2. Configuración del Perfil de Datos 129
7.3. Patrones de Perfil de Datos 131
7.4. Árbol de Decisión de las Estrategias de Mediación 132

8.1. Esquema de Recuperación de Información 138
8.2. Precisión caso instancias concentradas. Número de fuentes: 501-1000. Nivel de conocimiento: Medio 143
8.3. Precisión caso instancias dispersas. Número de fuentes: 501-1000. Nivel de conocimiento: Medio 143
8.4. Precisión caso propiedades concentradas. Número de fuentes: 501-1000. Nivel de conocimiento: Medio 144
8.5. Comparación *Fall Out* OptiSource y Estrategia Intencional. Número de fuentes: 501-1000. Nivel de conocimiento: Medio 145
8.6. Precisión extensional variando el nivel de conocimiento. Número de fuentes: 501-1000. Tipo de Consulta: Concentración de instancias. 146
8.7. Exhaustividad K variando el nivel de conocimiento. Número de fuentes: 501-1000. Tipo de Consulta: Concentración de instancias. 146
8.8. MedidaF variando el nivel de conocimiento 146
8.9. *Fall Out* extensional variando el nivel de conocimiento. Número de fuentes: 501-1000. Tipo de Consulta: Concentración de instancias. 146
8.10. Precisión extensional variando el número de fuentes. Nivel de conocimiento: Medio. Tipo de consulta: Concentración de instancias.147
8.11. Exhaustividad K variando el número de fuentes. Nivel de conocimiento: Medio. Tipo de consulta: Concentración de instancias. . 147
8.12. Escalabilidad en tiempo de acuerdo al número de fuentes y de condiciones 148
8.13. Impacto General de Beneficio 151
8.14. Promedio de Cambios en el Conjunto de Experimentos 151
8.15. Impacto del Beneficio en la Asignación - 15 % 151
8.16. Impacto del Beneficio en la Asignación - 50 % 152
8.17. Comparación del comportamiento de OptiSource e iDrips 154
8.18. Eficiencia de OptiSource e iDrips 155
8.19. Adaptación de la estrategia de caminos de navegación para OV . 157
8.20. Componentes Implementados de ARIBEC 158
8.21. Interfaz de los componentes 159

C.1. Clases Núcleo de la Base de Conocimiento 182
C.2. Recursos de las OV 183
C.3. Unidades de la OV 184
C.4. Subclases de VODomainConcept 185
C.5. Relaciones entre individuos de la clase VODomainConcept e individuos de la clase VOUnit y VOResource 187

Capítulo 1

Introducción

Esta tesis aborda el problema de ejecución de consultas en organizaciones virtuales con contextos de datos complejos y de gran escala. Este primer capítulo introduce la motivación y el contexto de la investigación. Así mismo, presenta los problemas que busca solucionar la investigación, los objetivos planteados y las principales contribuciones. Finalmente, se describe la estructura del documento.

1.1. Motivación: Integración de Datos en Organizaciones Virtuales

Una organización virtual es una red de organizaciones autónomas que comparten competencias y recursos, están dispersas geográficamente, interactúan bajo una alianza explícita de colaboración y su comunicación y operación están basadas en tecnologías de información. Las organizaciones virtuales(OV) facilitan a organizaciones independientes compartir competencias y recursos [FKT01], tales como datos, infraestructura de cómputo e instrumentos científicos, entre otros. Esta investigación está enfocada a OV que reúnen un alto número de participantes (entre 300 y 1000) que trabajan alrededor de una misma área o dominio y que buscan aliarse a nivel local, regional, nacional o mundial para compartir información y recursos. El contexto de datos en estas OV de gran escala involucra típicamente un alto número de fuentes de datos autónomas y heterogéneas que deben ser compartidas y consultadas de forma transparente a través de la OV.

La administración de datos en estas OV genera nuevos retos debido a que las técnicas tradicionales de mediación de bases de datos heterogéneas y distribuidas no son las más adecuadas. Éstas deben ser escalables para soportar un gran número de fuentes autónomas, garantizando al mismo tiempo respuestas de la calidad requeridas en ambientes organizacionales. Sin embargo, la magnitud del contexto de datos, sumado a su fragmentación, replicación y la incertidumbre de su localización dificultan esta tarea.

En la presente investigación se utilizó a las OV del sector salud como caso de estudio. El análisis sobre estas OV y sobre dos proyectos de integración en salud, uno a nivel regional y otro a nivel nacional, permitieron identificar las limitaciones de los enfoques tradicionales de integración y las características que hacen complejo el procesamiento de consultas en estos contextos de datos. Así mismo, el análisis permitió identificar que las características organizacionales de la OV hacen que su contexto de datos tenga ciertas características que en otros contextos no es posible suponer y que fueron identificadas y utilizadas para favorecer el procesamiento de consultas. Por ejemplo, los procesos organizacionales de las OV permiten obtener conocimiento acerca de las fuentes de datos que en otros contextos distribuidos no es posible tener.

El proyecto de integración a nivel nacional llamado SISPRO fue liderado por el gobierno colombiano en el año 2003. Su objetivo era "Construir un Sistema de Información, que bajo el concepto de la Protección Social, permita tomar decisiones que apoyen la elaboración de políticas, el monitoreo regulatorio y la gestión de servicio"[MdPS08]. Este sistema utiliza un enfoque de consolidación [KC04] para integrar la información de salud, riesgos profesionales, pensiones y asistencia social de la población colombiana. Posteriormente, la información integrada es utilizada para el cálculo de indicadores relacionados con la prestación de servicios de salud. A pesar de que el enfoque inicial del proyecto buscaba satisfacer las necesidades de información de la entidad de control y administración de salud en Colombia, durante el desarrollo del proyecto se identificó la necesidad de tener integrada no sólo la información básica de pacientes, sino también la información médica detallada de más de 33 millones de personas que permitiera mejorar la prestación de servicios y la investigación clínica. Desafortunadamente, el enfoque de consolidación usado en SISPRO no era escalable a estas nuevas necesidades de volumen de información, por lo que fueron postergadas a una fase posterior del proyecto.

El gobierno francés, por su parte, ha venido desarrollando proyectos de integración de información médica de pacientes desde hace un tiempo. El caso más notable es el sistema desarrollado en la región Rhône-Alpes [CR10] para compartir la información de pacientes de forma segura entre los profesionales de la salud. La plataforma desarrollada conecta 30 hospitales de la región y cubre 2 millones de documentos médicos de más de 200,000 pacientes. Estos documentos se encuentran almacenados remotamente y son accedidos en tiempo real cuando un profesional de la salud autorizado los requiere. Este proyecto representa importantes avances en la integración de datos de pacientes, especialmente en el aseguramiento de la confidencialidad de los datos de salud y en la arquitectura híbrida de integración (consolidada y federada). Sin embargo, la integración de información está dirigida a consultas acerca de un paciente en particular. Su arquitectura no contempla la evaluación de consultas masivas de documentos médicos de pacientes. Por ejemplo, esta arquitectura no permite obtener las historias clínicas de pacientes con diagnósticos o características específicas que faciliten la investigación y evaluación de tratamientos.

En el proyecto SISPRO se detectaron problemas de mantenibilidad y escalabilidad de los modelos consolidados cuando el contexto de datos involucra

grandes volúmenes de datos y de fuentes. La mantenibilidad se dificulta por que cada vez que es adicionada o actualizada información a las fuentes originales, la bodega de datos consolidada queda desactualizada y por lo tanto se hace necesario ejecutar nuevamente los procesos de extracción, tranformación y carga a la bodega. Así mismo, la escalabilidad del sistema a nivel de volumen de datos y de fuentes se reduce de acuerdo a las capacidades de almacenamiento y de procesamiento de un servidor central.

Por otro lado, en el caso francés las arquitecturas de integración que combinan consolidación y federación de fuentes permitieron identificar un incremento considerable en la capacidad de escalamiento y facilidad de mantenibilidad. Bajo esta arquitectura de integración, los datos, en la mayoría de los casos, permanecen en las fuentes originales evitando cuellos de botella y puntos críticos de falla. Sin embargo, dejaron al descubierto la ausencia de estrategias de planeación capaces de determinar las mejores fuentes para evaluar una consulta cuando ésta no define una condición sobre una propiedad única (por ejemplo pacientes con id=1). Esta ausencia se debe al manejo de un índice único sobre los identificadores de paciente.

Las limitaciones de los enfoques actuales de integración de datos al ser aplicados en los contextos complejos de las OV de gran escala y las nuevas características que surgen en este tipo de contextos, gracias al carácter organizacional de las OV, fueron los elementos que dieron lugar a esta investigación. La motivación fundamental está en la necesidad de crear nuevas estrategias escalables y flexibles de integración de datos que satisfagan los requerimiento de las OV. Específicamente, la investigación se enfoca en la optimización del proceso de selección de fuentes de datos como estrategia para escalar el problema tradicional de consultas en sistemas distribuidos a ambientes complejos de gran escala. Los aportes de esta investigación no sólo contribuyen al área de gestión de consultas en contextos complejos sino que también aportan a la solución de un problema real en el sector salud.

1.2. Contexto: Mediación de Información Heterogénea

Diferentes propuestas para soportar consultas en contextos de datos con fuentes heterogéneas y distribuidas han sido estudiadas desde hace más de 15 años. La mayoría están basadas en los sistemas de mediación [Wie92] cuyo objetivo es proveer una vista integrada de fuentes de datos heterogéneas y distribuidas respetando su autonomía [RS97b]. La arquitectura de los sistemas de mediación [GMPQ+97, RS97b, TRV98, Kos00a] está compuesta por cuatro niveles de abstracción: el nivel de fuentes de datos, el nivel de adaptación (*wrappers*), el nivel de mediación y el nivel de aplicación. El procesamiento de consultas en este tipo de sistemas se coordina en el nivel de mediación y se hace en dos etapas: la planeación y la ejecución. En la planeación se **seleccionan** las fuentes que pueden resolver total o parcialmente la consulta que se **reescribe** en subconsultas

usando el modelo externo de cada fuente. En la **ejecución** el mediador envía las subconsultas a cada fuente y se encarga de **coordinar la ejecución**, recibiendo las respuestas y procesando las operaciones necesarias para integrarlas.

Considerando la necesidad de modificar el proceso de selección de fuentes para poder soportar las características de gran escala y complejidad de las OV, el contexto de esta investigación lo componen los proyectos en el área de mediación que abordan el proceso de selección de fuentes de datos.

En la primera generación de sistemas de integración para fuentes de datos estructuradas como Information Manifold [LRO96], TSIMMIS [GMPQ+97], DISCO[TRV98] y Minicon [PH01], el proceso de selección de fuentes de datos está basado en filtrar las fuentes de datos de acuerdo a las capacidades de procesamiento de las fuentes (por ejemplo el número de condiciones requeridas, atributos que pueden definirse en el predicado). Aunque estas propuestas son eficientes en diferentes contextos, sus estrategias de filtrado no son útiles en las OV en donde las capacidades de las fuentes son similares.

Otro grupo de estrategias como iDrips y Streamer [DL02] seleccionan las fuentes de datos implícitamente al elegir los planes de consultas que generan mayor utilidad; las fuentes de datos que no están en estos planes son descartadas. A pesar de que estas estrategias pueden ser útiles en las OV, el cálculo de la utilidad funciona bajo supuestos de información que no está disponible en las OV o que es muy costosa de obtener y difícil de mantener. Adicionalmente, en el caso de Streamer los planes son considerados independientes, lo cual no es cierto en las OV en donde diferentes planes pueden entregar resultados similares, debido a que las fuentes que participan pueden proveer las mismas respuestas .

Otras propuestas como QPIAD [WKC+07] y la Orientada a calidad [NFL04] usan estadísticas detalladas de la calidad de los datos contenidos en las fuentes para reducir el número de planes. Infortunadamente, estas estadísticas son difíciles de obtener en las OV debido al alto número de fuentes autónomas y al volumen de datos contenidos en ellas. De la misma forma, la propuesta *Caminos de navegación*[1][BKN+06] utilizada para seleccionar caminos entre fuentes supone el conocimiento de qué instancias están contenidas en cada fuente y cómo están relacionadas las instancias de las diferentes fuentes. Una vez más esta información no está disponible en las OV. La propuesta de balanceo de proveedores [QRLV07] es adecuada para ambientes de datos en donde las fuentes son completas y un solo plan es suficiente para obtener el conjunto de respuestas requeridas. No obstante, esto no es apropiado en las OV donde las fuentes son incompletas y por lo tanto es necesaria la ejecución de diferentes planes de consulta.

Finalmente, las propuestas de mediación a gran escala usando arquitecturas *Peer-to-Peer* (PIER [HHL+03], PinS [VRL06], PIAZZA [TIM+03], PeerDB [OTZ+03] , SomeWhere [AGR07]) son capaces de mediar en contextos en donde hay gran número de fuentes de datos. Sin embargo, sus algoritmos de reescritura y selección de fuentes no contemplan el hecho de que múltiples fuentes pueden

[1] La propuesta no tiene un nombre específico para facilidad de lectura en esta investigación será denominada *Caminos de navegación*.

entregar el mismo conjunto de respuestas, produciendo respuestas redundantes cuando son aplicados en OV de gran escala.

1.3. Problema y Objetivos de la Investigación

1.3.1. Problema de Investigación: Optimización de la Selección de Fuentes de Datos en Organizaciones Virtuales

1. Ninguna de las estrategias actuales de selección de fuentes de datos es adecuada para los contextos complejos de las organizaciones virtuales de gran escala.

   Las estrategias de selección de datos existentes actualmente han sido diseñadas para contextos de datos con características diferentes a las de las OV o funcionan bajo supuestos que no son ciertos en las OV. Como consecuencia, al ser aplicados en los contextos de las OV funcionan ineficientemente o simplemente no funcionan. Su funcionamiento ineficiente se debe a que algunas de ellas suponen independencia entre las fuentes, otras están enfocadas a contextos de datos en donde las fuentes son completas. Las estrategias que no funcionan se debe a que utilizan metadatos que no están disponibles en las OV para reducir el número de fuentes o no garantizan la propiedad de autonomía de las fuentes que en el caso de las OV es fundamental. El detalle del porqué estas estrategias no pueden ser usadas directamente en las OV de gran escala será presentado en los capítulos 3 y 4.

2. Los sistemas de mediación disponibles usan una única estrategia de selección de fuentes de datos para todos los tipos de consulta incurriendo en costos de planeación innecesarios o en ineficiencia durante la ejecución de la consulta.

   La aplicación de una única estrategia de selección de fuentes puede introducir lo que se podría llamar un desequilibrio entre el tiempo de planeación y la reducción de fuentes seleccionadas cuando son aplicadas en OV. Por ejemplo, la planeación de una consulta cuyo predicado puede ser evaluado en un conjunto reducido de fuentes no merece el uso de estrategias de planeación complejas pues tan sólo es necesario el esquema de las fuentes para identificar las fuentes que se deben involucrar durante la planeación. Por el contrario, consultas que involucran un gran número de fuentes deben ser planeadas minuciosamente para lograr un balance entre la completitud de la respuesta y el número de fuentes consultadas.

1.3.2. Objetivos de Investigación

1. Diseñar una estrategia de selección de fuentes de datos para organizaciones virtuales de gran escala.

El propósito es proponer una nueva aproximación del proceso de selección de fuentes en los sistemas de mediación que responda a las necesidades de los contextos de datos de las organizaciones virtuales de gran escala.

2. Definir un modelo de mediación de datos para organizaciones virtuales escalable y adaptable a las necesidades de información.

   El propósito es rediseñar el nivel de mediación tradicional para que sea capaz de adaptar la estrategia de planeación de consultas de acuerdo al contexto de datos de la organización virtual y a la consulta que se va a ejecutar.

1.4. Contribución

Las contribuciones más importantes de este trabajo al área de integración de información son las siguientes:

1. Análisis de los sistemas de mediación desde el punto de vista de las OV

   Los sistemas de mediación y de integración de datos han evolucionado de manera importante durante los últimos años gracias a los avances en las tecnologías de información y comunicación. La diversidad de contextos distribuidos y de necesidades de integración han dado lugar a la creación de diferentes propuestas de mediación. La primera contribución de este trabajo es la clasificación de estas propuestas y la comparación de las estrategias de selección de fuentes de datos que utilizan. Además de proporcionar un panorama más claro de las estrategias disponibles actualmente, este trabajo de análisis del estado del arte permitió identificar las estrategias de selección de datos más apropiadas de acuerdo a las características de los contextos de datos.

2. Especificación del contexto de datos de las organizaciones virtuales

   Los contextos de datos de las organizaciones virtuales involucran un alto número de variables que deben ser consideradas durante la ejecución de consultas. Este trabajo de investigación hace una caracterización de las variables involucradas en las organizaciones virtuales que pueden impactar el procesamiento de las consultas. La caracterización es denominada perfil de datos pues permite analizar a la organización virtual desde diferentes puntos de vista y con niveles de detalle diferentes.

3. OptiSource: Optimizador de selección de fuentes para organizaciones virtuales

   OptiSource utiliza una nueva aproximación para resolver el problema de selección de fuentes como un problema de toma de decisión dentro de un ambiente complejo. Usando las características propias de la OV, OptiSource estima el beneficio de usar las fuentes de datos en la ejecución de

una consulta y optimiza la asignación de fuentes para evaluar las condiciones del predicado de la consulta. Esta optimización se lleva a cabo usando un modelo de optimización combinatoria que identifica el conjunto de fuentes que maximizan el beneficio, en términos de completitud de la respuesta, minimizando el número de fuentes contactadas y respetando las restricciones de procesamiento de las fuentes. OptiSource es apropiado en ambientes donde el número de fuentes es alto, donde las respuestas a las consultas están fragmentadas y distribuidas en múltiples fuentes y donde no hay exigencias de exhaustividad en las respuestas.

4. Procesador de consultas multiescala para organizaciones virtuales

   La necesidad de satisfacer diferentes tipos de usuario y diferentes tipos de análisis, hace que las OV requieran un procesador de consultas que se adapte a los requerimientos de análisis cambiantes. Parte de la contribución de esta investigación es el diseño de un procesador de consultas multiescala para OV cambiantes en el tiempo donde diferentes estrategias de procesamiento de consultas cohabitan. Este procesador es capaz de elegir el algoritmo de selección de fuentes más apropiado, considerando la noción de consulta y la evolución del contexto de datos de la OV. La evolución del contexto de datos de la OV se da porque su estructura básica cambia (e.g., más o menos participantes) o porque el conocimiento que se tiene de ella cambia.

1.5. Organización del Documento

El documento está estructurado en tres grandes partes:

1. Marco Conceptual y Estado del Arte: Organizaciones Virtuales y Procesamiento Distribuido de Consultas

   El Capítulo 2 presenta un marco conceptual alrededor de las OV y caracteriza sus contextos de datos. Posteriormente, en el Capítulo 3 se presenta la arquitectura de referencia de los sistemas de mediación, y se analizan diferentes implementaciones en sistemas que buscan compartir datos en comunidades virtuales con características similares a las de las OV. Finalmente, en el Capítulo 4 se evalúan las propuestas de selección de fuentes de datos de los sistemas de mediación actuales y se identifican los puntos que pueden ser aplicados en la arquitectura de mediación para OV.

2. Selección de Fuentes de Datos en Organizaciones Virtuales de Gran Escala

   La segunda parte presenta en tres capítulos la propuesta de esta tesis. El Capítulo 5 presenta la arquitectura de mediación propuesta para consultar y compartir información en las OV. Posteriormente, el Capítulo 6 describe de forma detallada OptiSource, la estrategia que optimiza la selección de fuentes de datos en OV de gran escala. Finalmente, el Capítulo 7 presenta el diseño del procesador de consulta multi-escala definido para elegir la

mejor estrategia de selección de fuentes de acuerdo al contexto de datos de la OV.

3. Implementación y Validación de la Propuesta

   La tercera parte presenta la validación de OptiSource en el Capítulo 8. Este capítulo presenta el análisis y la evaluación realizados sobre OptiSource desde diferentes puntos de vista. Así mismo, presenta las características del prototipo utilizado para hacer estas pruebas.

Parte I

Organizaciones Virtuales y Estado del Arte en el Procesamiento Distribuido de Consultas

Capítulo 2

Organizaciones Virtuales

2.1. Introducción

La creciente importancia de las relaciones business-to-bussiness (B2B) y el avance en las tecnologías de información y comunicación, han propiciado la creación de un nuevo modelo organizacional llamado Organización Virtual [Mow03]. En este nuevo modelo, diferentes organizaciones autónomas y distribuidas colaboran y comparten recursos y competencias bajo acuerdos explícitamente definidos que buscan lograr un objetivo común [Let01, FKT01]. Esta investigación está enmarcada en la necesidad que tienen las OV de crear una infraestructura estable que les permita compartir y consultar datos provenientes de fuentes de datos distribuidas que hacen parte de la OV. El objetivo de este capítulo es introducir el concepto de organización virtual, sus características y en especial los contextos de datos que se generan cuando múltiples organizaciones autónomas, pero relacionadas lógicamente, forman una OV.

El capítulo está estructurado de la siguiente manera. En primer lugar la Sección 2.2 define lo que para este trabajo se considera una OV y el tipo de OV al que se enfoca la investigación. Posteriormente, la Sección 2.3 presenta el ejemplo de OV de gran escala que se utilizó como caso de estudio en toda la investigación. La Sección 2.4 identifica las características de los contextos de datos de las OV de gran escala. La Sección 2.5 ilustra los retos de una arquitectura de información creada para OV. Finalmente, la Sección 2.6 concluye el capítulo.

2.2. Definición

Las diversas interpretaciones del concepto de virtualidad han dado lugar a diferentes definiciones de organización virtual. Para efectos de esta investigación se tomará la Definición 1 que integra diferentes visiones [Let01, FKT01, CXWL05, NEE08].

Definición 1 ***Organización Virtual***
Una organización virtual(OV) es una red de organizaciones autónomas que comparten competencias y recursos, están dispersas geográficamente, interactúan bajo una alianza explícita de colaboración y su comunicación y operación están basadas en tecnologías de información.

Esta definición abarca diversos tipos de OV. Desde las OV de pequeña escala, en donde un grupo reducido de organizaciones se alían para satisfacer un requerimiento puntual del mercado, hasta OV de gran escala cuyos participantes colaboran bajo alianzas estables y de largo plazo para compartir recursos que los fortalezca mutuamente. Las necesidades de información y comunicación en cada tipo de OV pueden ser diferentes y por lo tanto la infraestructura tecnológica requerida también lo es. Este trabajo está enfocado en fortalecer la arquitectura de información de las OV denominadas en esta investigación como de gran escala.

Las **OV de gran escala** están caracterizadas por tener un alto número de organizaciones participantes que trabajan alrededor del mismo dominio de conocimiento (genética, física, psicología, etc.) o sector (educación, salud, gobierno, etc.), y que se asocian a nivel regional, nacional o mundial para satisfacer los requerimientos de información que una organización participante por si sola no puede satisfacer. Este tipo de OV, llamadas de gran escala por el volumen de sus participantes, combina la flexibilidad de las comunidades de intercambio de datos en Internet con la formalidad de las organizaciones, asegurando un nivel de prestación de servicios con mayor calidad, confiabilidad y seguridad. Por lo tanto su arquitectura de información debe ser lo suficientemente escalable, flexible y segura para soportar un gran número de fuentes de datos distribuidas, autónomas y heterogéneas sobre las cuales se realizan diferentes tipos de análisis y consultas.

A pesar de la autonomía de las organizaciones participantes en una OV de gran escala, la relación lógica que existe entre ellas, por el hecho de pertenecer a un mismo dominio o sector, y los vínculos que las enlazan debido a las alianzas establecidas al decidir pertenecer a la OV, generan proximidad y relaciones entre las fuentes de datos que proveen a la OV. Por ejemplo, en una OV en el sector salud todas los esquemas de datos de las fuentes son cercanos, ya que todas manejan conceptos comunes (por ejemplo Paciente, Historia Clínica, etc.). Adicionalmente, el contenido de las fuentes (instancias de los conceptos) puede estar también relacionado como consecuencia de la ejecución de un proceso de negocio o por la naturaleza del sector. Por ejemplo, en el sector salud muchas fuentes pueden tener información de un mismo paciente debido a su movilidad, o debido a procesos regulatorios que exigen reportar información de quienes prestan los servicios de salud hacia quienes regulan la prestación de servicios de salud. Estas relaciones entre organizaciones y por ende entre fuentes, diferencian los contextos de datos de las OV de otros contextos que involucran fuentes distribuidas y heterogéneas. Es necesario entonces diseñar una arquitectura de información adecuada que responda a las necesidades y características de las OV. Las siguientes secciones de este capítulo analizan cuáles son estas necesidades

y características con el fin de identificar los requerimientos de una arquitectura de información apropiada para OV de gran escala.

2.3. OV en el Sector Salud

Esta sección ilustra las características y requerimientos de una OV de gran escala en el sector salud. Ésta fue tomada como ejemplo representativo debido a la importancia social y política del sector salud, que ha promovido la cooperación entre instituciones "para ofrecer más atención, más calidad, más integración y más velocidad"[BRBT07]. OV como BIRN [Org07], eDiaMoND, la grilla de información de salud en Canadá [Bea03], demuestran la evolución del sector hacia este nuevo modelo organizacional como una manera de fortalecer la prestación de servicios de salud y la investigación en las diferentes especialidades.

La OV en la que se entrará en detalle es considerada la evolución natural de los proyectos de integración en el sector salud que se han venido desarrollado en diferentes países [MdPS08], [CR10]. Su objetivo de creación es construir una infraestructura estable que permita compartir e intercambiar información médica proveniente de fuentes de datos heterogéneas que ayuden a fortalecer la administración y prestación de servicios médicos, así como también la investigación clínica.

La puesta en marcha de esta OV de salud es la motivación de este proyecto no sólo por su impacto a nivel social sino también por los grandes retos que representa definir una arquitectura de datos estable, escalable y robusta que soporte los requerimientos de información de todos los actores del sector. A continuación se describen las características de esta OV, con énfasis en sus contextos de datos.

Participantes

Los participantes en esta OV pueden clasificarse en cinco grupos. Cada uno de ellos será descrito teniendo en cuenta el objetivo de su vinculación en la OV y el aporte en recursos de datos.

- Agencias Gubernamentales: Este grupo incluye las organizaciones de administración, monitoreo y control en el sector salud. Los recursos de datos que estas organizaciones proveen a la OV son el grupo de fuentes con información consolidada de pacientes y resúmenes de prestación de servicios de salud. Su participación en la OV tiene como propósito tener una vista global de la prestación de servicios que les permita tomar decisiones con mayor información de soporte y definir programas estratégicos basados en la vista completa de los datos. Así mismo, estas organizaciones buscan mejorar el proceso de entrega de la información de pacientes que por ley debe ser proveída por las organizaciones prestadoras de servicios de salud.

- Organizaciones Prestadoras del Servicio de Salud: En este grupo están incluidas todas las organizaciones, instituciones y profesionales independientes que prestan servicios de salud. Este grupo provee el conjunto más

grande de fuentes de datos que contienen información de procedimientos y consultas médicas. Su objetivo dentro de la OV es tener disponible una fuente de información más amplia de pacientes y tratamientos que les permita tener historias clínicas más completas y tener disponible mayor cantidad de casos para hacer investigaciones clínicas de mayor envergadura.

- Organizaciones Promotoras del Servicio de Salud: En este grupo están incluidas todas las empresas que organizan y garantizan, directa o indirectamente, la prestación de los servicios de salud. Este grupo provee fuentes con información demográfica de paciente y con información resumida de procedimientos realizados a estos pacientes. Su objetivo en la OV es poder analizar la información de tratamientos de un conjunto amplio de pacientes para definir mejor sus políticas, en especial en los tratamientos de enfermedades costosas. Algunas instituciones pertenecientes a este grupo, también pertenecen al grupo de organizaciones prestadoras del servicio de salud.

- Instituciones de Investigación: El propósito de este grupo de organizaciones es realizar acciones de investigación que enriquezcan la labor médica. Su objetivo dentro de la OV es, por una lado, proveer resultados de su investigación a toda la comunidad, y por otro, obtener información más completa que permita mejorar su actividad investigativa y tener más recursos de cómputo para realizar sus experimentos.

- Pacientes: Es el grupo de participantes más amplio. Incluye a todos los individuos que reciben servicios de salud. Su principal intención es obtener su registro médico completo.

A pesar de que los nombres de estos participantes pueden variar de un país a otro, las funciones que desarrollan son similares en todos los países.

Requerimientos de Información

Los requerimientos de información de la OV están asociados a los objetivos de cada grupo de participantes. Los requerimientos más importantes son:

- Investigación Clínica y Medicina Basada en la Evidencia: La OV debe garantizar la obtención de la evidencia médica que responda de la manera más acertada a las preguntas de un profesional de la salud durante el desarrollo de su actividad. Ejemplos de evidencia médica requerida son:

  - Imágenes Médicas y metadatos de tejidos y órganos humanos
  - Historia Genética
  - Análisis Celular
  - Tratamientos completos de enfermedades y en especial de las poco frecuentes

- Registro médico electrónico y distribuido: La OV debe permitir obtener a partir de un grupo amplio de fuentes de datos los registros médicos de los pacientes que cumplen con ciertas características demográficas, diagnósticas o de tratamientos realizados.
- Administración de Salud: La OV debe permitir obtener vistas completas y responder preguntas relacionadas con:
  - Análisis de cobertura de políticas
  - Validación de afiliación a los servicios de salud
  - Análisis del costo de tratamientos
  - Análisis epidemiológico

Contexto de Datos

El contexto de datos de la OV en salud está conformado por las fuentes de datos que proveen las organizaciones participantes. Estas fuentes están distribuidas en nodos regionales y están disponibles en la OV vía redes de área amplia. Aunque las alianzas establecidas en la OV definen niveles de servicio y permanencia de los participantes, la autonomía de las organizaciones les permite estar fuera de línea durante ciertos periodos de tiempo.

Cada organización que hace parte de la OV desempeña un rol que determina su función dentro de la OV en términos de procesos, recursos y privilegios. Los recursos de datos que cada organización provee son registrados a la OV indicando su ubicación y el tipo de instancias que almacena (por ejemplo instancias de pacientes). Debido a la naturaleza de la prestación del servicio de salud las instancias de pacientes están fragmentadas y replicadas en diferentes fuentes de datos. Adicionalmente, los procesos de negocio existentes en el sector y las políticas gubernamentales que exigen el envío de información por parte de los prestadores y promotores de servicios de salud hacen que los datos de pacientes se encuentren en diferentes fuentes. En ciertos casos no hay claridad sobre cuáles fuentes contienen las instancias originales y cuáles las replicadas.

La Figura 2.1 ilustra un ejemplo de proceso de negocio que genera replicación de las instancias de pacientes. En este proceso de negocio un hospital psiquiátrico A (HA) especializado en terapias de corta duración para pacientes de bajo riesgo tiene un acuerdo con otro hospital psiquiátrico B (HB) especializado en cuidado temporal o permanente de pacientes. Típicamente, los pacientes con enfermedades psiquiátricas van primero a HA, y de acuerdo a su diagnósticos son o no remitidos a HB. Como consecuencia todos los pacientes con desórdenes psiquiátricos importantes van de HA a HB. Adicionalmente, cada vez que HA o HB atienden a un paciente con esquizofrenia deben enviar su información para ser analizada en un instituto de investigación C (RC), como parte de un programa de investigación patrocinado por el gobierno. Mensualmente HA y HB deben enviar también un reporte a la organización reguladora de los servicios de salud del gobierno (GD) con la identificación de los pacientes y un resumen de los procedimientos realizados durante el mes.

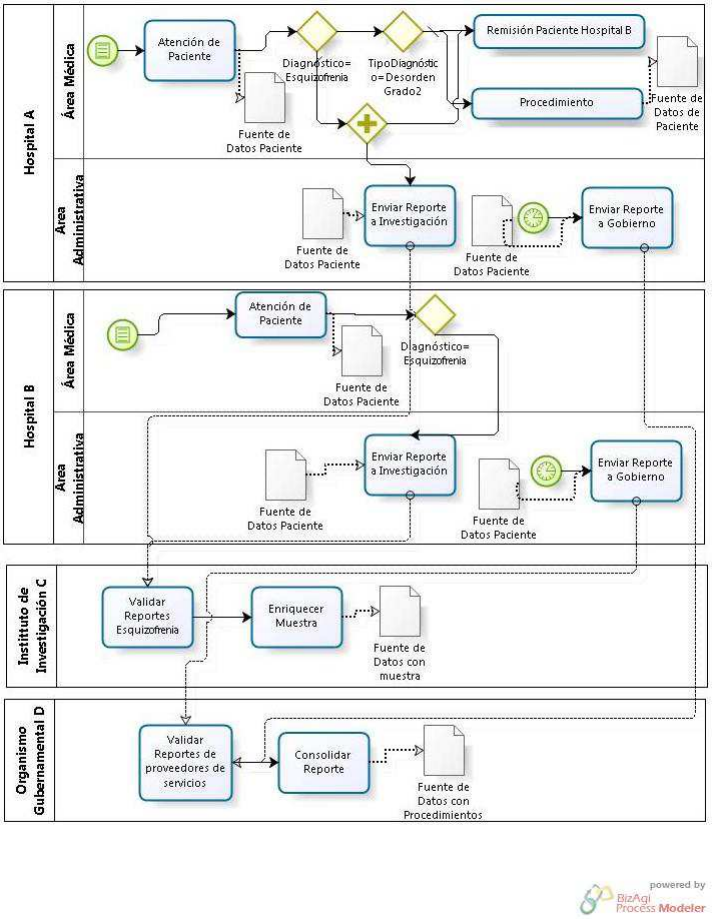

Figura 2.1: Proceso de Negocio Generador de Réplicas de Pacientes

Como resultado de la ejecución de los múltiples procesos de negocio existentes, el contexto de datos de la OV presenta un alto nivel de distribución y replicación de instancias que dificulta saber exactamente dónde se encuentra la información requerida. La Figura 2.2 ilustra un ejemplo de distribución en la OV. En este ejemplo seis fuentes de datos (representadas por óvalos) proveen instancias de pacientes (representadas usando rectángulos). Ninguna de las fuentes incluye todas las propiedades de pacientes (fragmentación vertical) y ninguna fuente contiene todas las instancias de pacientes (fragmentación horizontal). Adicionalmente, los fragmentos verticales no son disyuntos pues múltiples fuentes (DS1 a DS4) proveen las propiedades de paciente *DatoDemográfico* (e.g. género,edad) y de diagnóstico, y múltiples fuentes (DS5 y DS6) proveen propiedades de *Afiliación*. Los fragmentos horizontales tampoco son disyuntos.

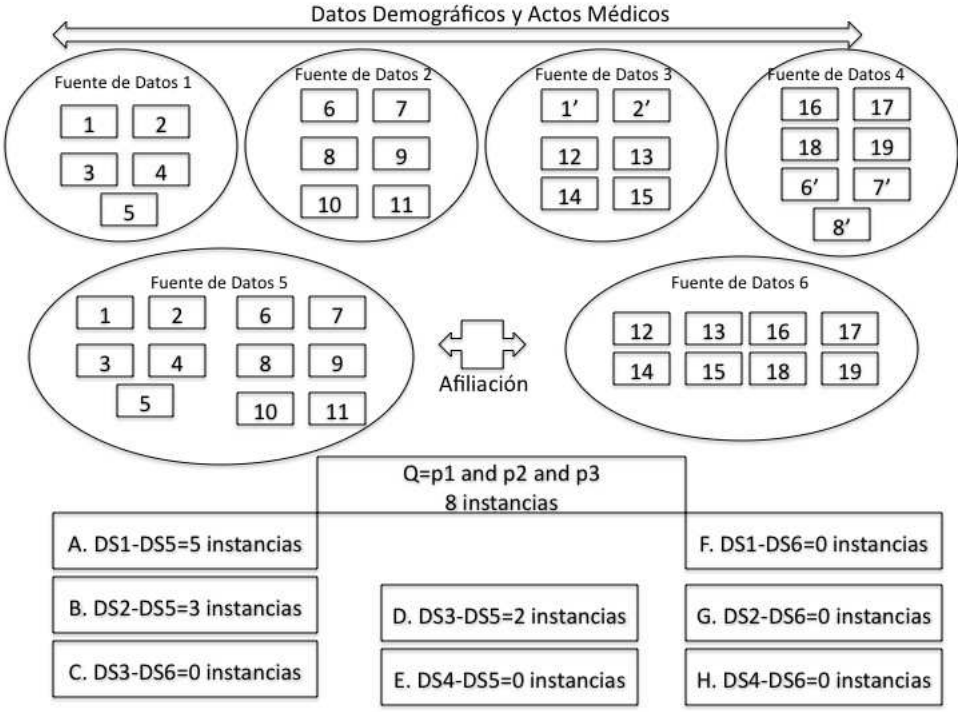

Figura 2.2: Contexto de Datos en una OV en el Sector Salud

Las instancias 1 y 2 y las instancias 6 a 8 están contenidas en dos fuentes de datos que proveen las mismas propiedades. Aunque las instancias 1 y 2 compartidas por DS1 y DS3 tienen el mismo identificador de paciente, no es posible establecer de antemano si son o no réplicas, por esta razón las instancias 1 y 2 en DS3 se diferencian usando el símbolo " ' ".. Esto también ocurre para las instancias 6,7 y 8 en DS2 y en DS4.

Contextos de datos similares se presentan en múltiples OV de gran escala. Consideremos por ejemplo una OV en el sector educativo cuyo objetivo sea compartir datos de estudiantes. La movilidad estudiantil hace que las instancias de estudiante estén distribuidas en múltiples fuentes de datos. Así mismo, de acuerdo al rol de cada participante dentro del proceso educativo (colegio, universidad, institutos de fomentos a la educación, ministerio, etc.) las propiedades que se tienen de un estudiante varían. Igualmente, debido a los procesos en los que participan múltiples organizaciones se genera replicación de instancias en múltiples fuentes. Casos semejantes se encuentran en OV en el sector judicial, en el sector financiero y en todo aquel sector y dominio en donde organizaciones independientes pero relacionadas lógicamente intercambian información.

2.4. Características de los Contextos de Datos en las Organizaciones Virtuales de Gran Escala

El análisis de la OV en salud y de diferentes OV de gran escala [LWZ06, CER], [Org07], [Pro10], [Gri09], permitió identificar un conjunto de características comunes en sus contextos de datos, que serán descritas a continuación.

1. Conceptos de dominio compartidos. El hecho de pertenecer a una misma OV hace que los datos almacenados en las fuentes de datos provistas por las organizaciones participantes estén relacionados a los mismos conceptos dentro del dominio de la OV.

2. Alto volumen de fuentes y de información. La información almacenada en las fuentes puede estar en el orden de los *GigaBytes*, *TeraBytes* e incluso *PetaBytes*. Esto hace difícil la clasificación de lo que cada fuente contiene en los catálogos de metadatos.

3. Heterogeneidad de las fuentes de datos. La autonomía de las organizaciones participantes es sinónimo de heterogeneidad de las fuentes que proveen a la OV. Las OV deben solucionar problemas de heterogeneidad estructural, sintáctica y semántica de las fuentes.

4. Alta distribución geográfica. La distancia geográfica entre las organizaciones participantes en una OV hace necesario que su enlace deba hacerse usando conexiones de redes de área amplia o infraestructuras que provean conexión de alta velocidad como las mallas de datos.

5. Copias Difusas. Un fenómeno de copias inexactas puede aparecer entre las fuentes de datos. Este tipo de copias no tienen un protocolo explícito de consistencia pues es el resultado de acciones independientes de diferentes proveedores de datos.

6. Fragmentación vertical no disyunta. Las propiedades de los conceptos de dominio están distribuidas en diferentes fuentes de datos. Sin embargo, debido a la proximidad semántica de las fuentes de datos los fragmentos no son disyuntos, lo que significa que la misma propiedad de un concepto de dominio puede estar en varias fuentes de datos.

7. Fragmentación horizontal no disyunta. Producto de la replicación, varias fuentes de datos pueden proveer el mismo fragmento o instancia de un concepto de dominio (e.g. dos fuentes proveen información acerca del mismo experimento), pero también varias fuentes pueden proveer propiedades diferentes (e.g. dos fuentes proveen información acerca del mismo experimento pero propiedades diferentes) y valores diferentes para una misma instancia (e.g. dos fuentes proveen las mismas propiedades acerca del mismo experimento pero los valores de las propiedades son diferentes).

8. Incertidumbre en la localización de los datos. A pesar de la existencia de catálogos de metadatos, los usuarios de la OV tienen incertidumbre acerca de cuál es la mejor fuente de datos para obtener la información requerida.

9. Requerimientos variables de información. Como consecuencia del gran número de usuarios, el tipo, el alcance y los requerimientos de las consultas son variables. A pesar de que es posible establecer parámetros comunes requeridos en las consultas de los usuarios, la evolución en las necesidades de información hace que los metadatos estáticos de las fuentes no sean suficientes para satisfacer efectivamente los requerimientos de información de los usuarios.

10. Diferentes restricciones de seguridad. La autonomía de las organizaciones proveedoras de información y los tipos de acuerdos definidos en la OV hace que los datos compartidos por cada organización puedan tener diferentes restricciones de seguridad. Las OV pueden involucrar diferentes políticas por cada fuente de datos.

2.5. Arquitectura de Información para OV de Gran Escala

Teniendo en cuenta las características de los contextos de datos de las OV de gran escala, el principal reto de una arquitectura de información diseñada para apoyar su funcionamiento es proveer a los usuarios de la OV de una vista integrada de la información disponible en la OV.

Esta vista debe permitirles realizar consultas de forma eficiente y transparente aún cuando físicamente involucren diferentes fuentes de datos autónomas, distribuidas y heterogéneas. A diferencia de otros contextos distribuidos, proveer una vista integrada se dificulta debido a las características del contexto de datos. Por un lado, la magnitud y la confidencialidad de las fuentes impiden mantener en una sola fuente los datos integrados. Por otra parte, la evaluación de consultas distribuidas es compleja producto de los fragmentos disyuntos que generan solapamiento entre fuentes, y de la incertidumbre acerca de la localización de los datos. Con estas características el esquema por si solo no puede ser usado para elegir las fuentes que deberían participar en la ejecución de la consulta, y los catálogos de metadatos no son suficientes para hacer consultas eficientemente. El solapamiento hace que sólo sea posible tener certeza de entregar una respuesta completa seleccionando todas las fuentes que son capaces de resolver las condiciones de la consulta, lo cual es muy costoso en tiempo y recursos, y en muchos casos no es necesario. En el caso en que se usen metadatos, si las fuentes tienen grandes volúmenes de datos sería necesario almacenar un catálogo muy detallado y de gran tamaño que no sólo es muy costoso sino también muy difícil de mantener a largo plazo. En resumen, además de resolver los problemas derivados por la heterogeneidad y la distribución, en las OV es necesario reconocer cuáles son las fuentes de datos más útiles para resolver una

consulta, de acuerdo a su contenido y a su relación con la consulta. Para lo cual, es preciso identificar qué tanto puede aportar una fuente de datos durante la ejecución de una consulta y así, dirigir las consultas sólo a aquellas fuentes con mayor probabilidad de entregar respuestas. En las OV es necesario llegar a un balance entre las fuentes seleccionadas para ejecutar una consulta y el número de respuestas que se desean obtener.

2.6. Síntesis

El nuevo modelo organizacional, llamado Organización Virtual (OV), es una red de organizaciones autónomas que comparten recursos, interactúan bajo una alianza explícita de colaboración y su comunicación y operación están basadas en tecnologías de información. Este modelo está caracterizado por la dispersión geográfica de las organizaciones participantes, por estar construida bajo los lineamientos de una alianza explícita, por tener una estructura dinámica y por no estar legalmente establecida. La integración y colaboración entre los participantes de una OV está mediada por sistemas de información flexibles que permiten a los participantes compartir competencias y recursos. Este trabajo está enfocado en fortalecer la infraestructura de las OV de gran escala, en donde un alto número de organizaciones participantes que trabajan alrededor del mismo dominio o sector, se asocian a nivel regional, nacional o mundial para satisfacer los requerimientos de información que una organización participante por si sola no puede satisfacer.

Los contextos de datos de las OV de gran escala están compuestos por múltiples fuentes de datos heterogéneas y distribuidas que almacenan instancias de conceptos del dominio de la OV. Las fuentes de datos en estos contextos están caracterizadas por manejar un alto volumen de información, por tener fragmentos verticales y horizontales no disyuntos con respecto a las instancias de los conceptos, y por la existencia de réplicas o de copias difusas. En general, los usuarios de estos contextos tienen requerimientos variables de información y en muchos casos no conocen con precisión la localización de las fuentes de datos más apropiadas para obtener la información requerida.

Evaluar consultas en estos contextos no es una labor sencilla. Además de resolver los problemas derivados por la heterogeneidad y la distribución, es necesario reconocer durante la planeación cuáles son las fuentes de datos más útiles para resolver una consulta, de acuerdo a su estructura y su contenido. En las OV es necesario llegar a un balance entre las fuentes seleccionadas para ejecutar una consulta y el número de respuestas que se desean obtener.

Capítulo 3

Procesamiento de Consultas en Contextos Distribuidos y Heterogéneos

3.1. Introducción

El capítulo 2 presentó la necesidad de consultar información dispersa y heterogénea en las OV como un problema de procesamiento de consultas en ambientes distribuidos y heterogéneos, que a diferencia de otros ambientes de datos cuenta con mayor información acerca de las fuentes de datos. Esta nueva información es derivada de los procesos, las relaciones y la estructura de la OV.

Este capítulo introduce la arquitectura de mediación y los sistemas de integración que la implementan con el fin de contextualizar el trabajo desarrollado en esta tesis.

El capítulo está estructurado de la siguiente manera. La Sección 3.2 presenta la arquitectura de integración de datos usando el enfoque de mediación. Posteriormente, la Sección 3.3 analiza las arquitecturas de mediación utilizadas en diferentes comunidades virtuales. Finalmente, la Sección 3.4 concluye el capítulo.

3.2. Sistemas de Mediación

El principio de los sistemas de mediación es crear un módulo de software que permita el acceso homogéneo a un conjunto de fuentes de datos pre-existentes, autónomas, distribuidas y posiblemente heterogéneas. El acceso homogéneo significa que el usuario sólo debe utilizar un lenguaje de consultas para obtener la información, independientemente del lenguaje utilizado por cada una de las fuentes [Wie92]. A continuación se presentan las características de estos sistemas.

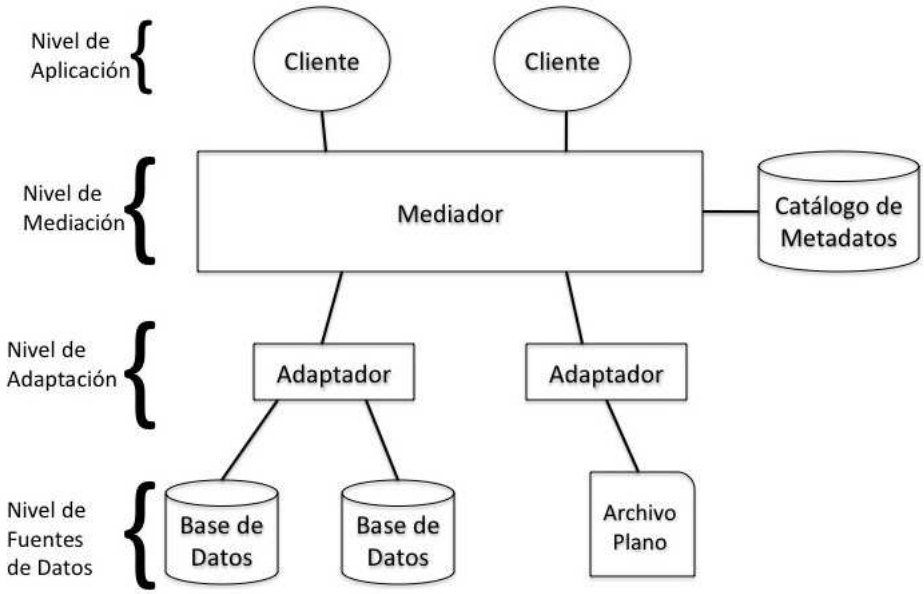

Figura 3.1: Arquitectura de Referencia de los Sistemas de Mediación

3.2.1. Arquitectura Lógica de Referencia

La arquitectura lógica de referencia de los sistemas de mediación ([GMPQ+97, RS97b, TRV98, Kos00a]), ilustrada en la Figura 3.1, está compuesta por cuatro niveles de abstracción: el nivel de fuentes de datos, el nivel de adaptación (*wrappers*), el nivel de mediación y el nivel de aplicación.

La Figura 3.2 ilustra el funcionamiento general de un sistema de mediación. Las fuentes de datos que se van a integrar son encapsuladas usando componentes adaptadores (*wrappers*) que las presentan de forma homogénea al nivel de mediación. Cada adaptador presenta la fuente (o fuentes) que encapsula a través de un **modelo externo** que describe la vista que la fuente tiene sobre el **esquema global** del sistema. El **esquema global** es definido y expuesto en el nivel de mediación. Representa la vista integrada de las fuentes, es usado por las aplicaciones en la definición de consultas y es definido por quien crea el sistema de mediación.

Cuando una consulta (en términos del esquema global) es realizada en el sistema, el mediador realiza la planeación y coordina la ejecución de la consulta. El proceso de planeación utiliza un catálogo de metadatos que incluye el modelo externo de las fuentes y otra información generalmente de tipo estadístico. El plan determina qué fuentes de datos son seleccionadas para participar en el procesamiento de la consulta, qué subconsulta evaluará cada una, y cómo las respuestas que cada fuente provea serán integradas. Una vez definido el plan, el mediador inicia la coordinación de la ejecución de la consulta enviándole al

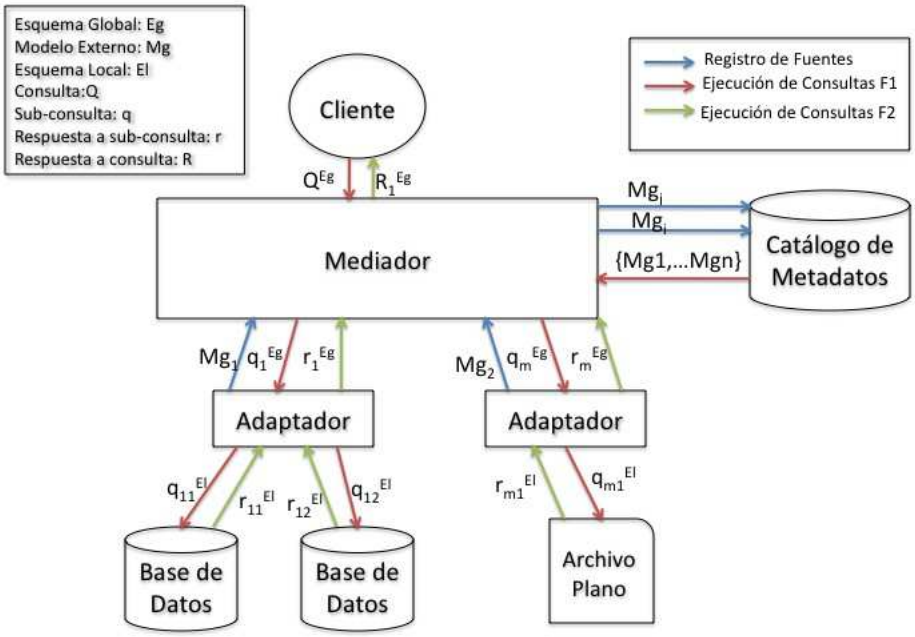

Figura 3.2: Ejecución de Consultas en la Arquitectura de Referencia de Mediación

adaptador de cada fuente de datos seleccionada la subconsulta que es responsable de ejecutar. El adaptador traduce la subconsulta, que se encuentra en términos del **esquema global**, a términos locales, de tal forma que pueda ser entendida por la fuente de datos. Cuando la subconsulta es resuelta por la fuente de datos, el adaptador traduce la respuesta de tal forma que corresponda a los conceptos del **esquema global**, y la entrega al mediador. El mediador integra los resultados recibidos de los adaptadores y se los entrega a la aplicación.

Esta arquitectura lógica ha sido implementada en muchos proyectos de integración pues logra independizar la solución a los problemas de heterogeneidad y distribución utilizando los niveles de adaptación y de mediación, respectivamente. Sin embargo, para adaptar esta arquitectura a las necesidades de los proyectos de integración, y para lograr que sea robusta y escalable, las implementaciones han hecho modificaciones a la arquitectura original. La Sección 3.3 presentará un conjunto de estas implementaciones. A continuación se describen en detalle las funciones de los niveles de adaptación y de mediación que son el núcleo de esta arquitectura.

3.2.2. Funciones del Nivel de Adaptación

La principal responsabilidad del nivel de adaptación es encapsular las fuentes de datos resolviendo los problemas de heterogeneidad con respecto al esquema global. En otras palabras, el adaptador actúa como traductor entre el nivel de mediación y el nivel de fuentes de datos. Para hacerlo, utiliza un conjunto de reglas que le permiten traducir las solicitudes del mediador al lenguaje de consultas que comprende la fuente que encapsula y viceversa. Para garantizar la encapsulación de fuentes, el adaptador realiza varías funciones, éstas serán descritas a continuación:

1. Registro de fuentes. Para que una fuente de datos pueda hacer parte de un sistema de mediación, el adaptador que la encapsula debe registrarla en el nivel de mediación. Esta fase de registro tiene por objetivo darle a conocer al mediador el contenido y/o las capacidades de las fuentes usando el lenguaje establecido por el mediador. Esta descripción es llamada el modelo externo, pues refleja lo que cada fuente quiere compartir al exterior. El nivel de detalle del modelo externo de las fuentes varía de acuerdo al lenguaje definido. Por ejemplo en el lenguaje *Garlic Data Language(GDL)* [RS97b] es posible definir la vista del esquema global que cada fuente conoce, los atributos que pueden ser modificados y las funciones que pueden ser ejecutadas en ellas. Por su parte el lenguaje basado en plantillas, propuesto en [GMPQ+97], permite definir la vista del esquema global que cada fuente conoce, y por cada vista es posible definir si un atributo puede, debe o no debe ser especificado en una consulta. Otro lenguaje llamado D2RQ [Ber07] permite describir las vistas que tienen las fuentes de datos relacionales sobre un esquema global definido a través de una ontología en OWL [Hor05].

2. Traducción de consultas globales en consultas locales. Esta función se encarga de recibir las consultas que el mediador envía y traducirlas al lenguaje que usa la(s) fuente(s) que encapsula. Para hacerlo, el adaptador almacena el procedimiento necesario para obtener las instancias que corresponden al modelo externo y para resolver los problemas de heterogeneidad [SL90]. Idealmente un adaptador debe resolver las diferencias en la estructura, en el lenguaje de consultas y en la semántica de los datos entre el esquema global del sistema y el esquema local de la fuente. A nivel estructural, el esquema global puede referirse a un atributo usando un nombre diferente al que realmente está almacenado en la fuente. Por ejemplo, el esquema global puede referirse al atributo *nombre* y localmente el atributo *nombre* está dividido en dos atributos llamados *primer nombre* y *segundo nombre*. A nivel de lenguaje, el mediador y la fuente pueden usar lenguajes diferentes. Por ejemplo, el mediador puede usar Datalog mientras que las fuentes pueden usar el lenguaje de consultas SQL. Las diferencias a nivel semántico ocurren cuando hay un desacuerdo entre el significado de un concepto en el esquema global y en el esquema local. Por ejemplo, a nivel global el concepto *utilidad* puede significar la *ganancia*

después de costos, mientras que a nivel local el concepto *utilidad* significa la *ganancia* después de *costos* y de *impuestos*.

3. Traducción de respuestas locales en respuestas globales. De forma inversa las respuestas a nivel local deben ser traducidas en términos del esquema global resolviendo nuevamente la heterogeneidad existente entre la fuente y el esquema manejado a nivel del mediador.

4. Participación en la planeación de la consulta. Algunos adaptadores además de cumplir las funciones anteriores, incluyen un conjunto de métodos que le permiten al nivel de mediación tener más información en el momento de planear una consulta [CHS+95, RS97a],[Kos00a]. Ejemplos típicos de esta información es cuánto tiempo durará la ejecución de la consulta, cuántas respuestas a una consulta podrá entregar, cuánto costará la ejecución de una consulta, etc. Durante la planeación de una consulta, el nivel de mediación puede solicitar a los adaptadores la ejecución de los métodos que proveen esta información, de tal manera que la optimización sea más precisa.

La definición de las reglas que permiten traducir de un esquema global a uno local y viceversa no es una labor sencilla y consume mucho tiempo. Sin embargo, ya existen herramientas que facilitan esta definición apoyándose en el aprendizaje de máquina. El trabajo presentado en [Nea07] hace una propuesta semi-automática de definición de mapeos en donde sólo es necesario hacer el mapeos manual para un grupo pequeño de fuentes de datos y el algoritmo analiza las otras fuentes para calcular automáticamente su mapeo con respecto al esquema global. Así mismo, desde los inicios de la arquitectura de mediación se han definido lineamientos generales para construir más rápidamente adaptadores como se puede observar en el proyecto GARLIC[RS97a].

3.2.3. Funciones del Nivel de Mediación

La responsabilidad del nivel de mediación es proveer a los usuarios una interfaz de consulta uniforme sobre un conjunto de fuentes de datos autónomas. Para proveer esta interfaz, el nivel de mediación realiza varías funciones que serán descritas a continuación:

1. Almacenamiento del esquema global. El esquema global representa la vista integrada de todas las fuentes a través de un conjunto de relaciones virtuales, en el sentido en que sus instancias están distribuidas en múltiples fuentes. Este esquema es diseñado manualmente de acuerdo a las necesidades de la aplicaciones que usan el sistema y es usado para definir las consultas en el nivel de aplicación.

2. Almacenamiento del catálogo de metadatos con la siguiente información de las fuentes:

- Modelo externo de las fuentes de datos. Determina qué partes del esquema global son almacenadas por cada fuente de datos. Este modelo externo le permite al mediador saber cuál es la estructura del contenido de las fuentes, los atributos que se encuentran en cada fuente y las restricciones en los contenidos de las fuentes. El lenguaje utilizado para hacer esta descripción de las fuentes es establecido por el nivel de mediación. El modelo externo es también conocido como vista, pues representa la vista que las fuentes tienen del esquema global.
- Estadísticas de los adaptadores. Las estadísticas almacenadas varían de acuerdo al sistema. Algunas estadísticas son el tiempo promedio en resolver una consulta, el número de respuestas entregadas dado un tipo de consulta, el costo de usar el adaptador, etc.

3. Planeación de la consulta: Una de las funciones más importantes del nivel de mediación es planear la ejecución de las consultas. Esta planeación es realizada utilizando las vistas que cada una de las fuentes tiene sobre el esquema global (modelo externo), y el conjunto de estadísticas sobre los adaptadores. La planeación incluye las siguientes fases.

   - Reescritura. El mediador reescribe la consulta en términos de las vistas (modelo externo) de las fuentes.
   - Descomposición. Evalúa los posible planes físicos de las reescrituras y elimina aquellas que no tienen ningún plan físico factible. Un plan físico es factible si respeta las limitaciones de las fuentes.
   - Optimización: La optimización típicamente busca elegir los planes menos costosos o los planes que involucran a fuentes con mayor calidad. En el primer caso, el costo es evaluado usando estadísticas de ejecuciones pasadas, o métodos de calibración en donde se envían consultas de prueba a los adaptadores para saber su comportamiento en la consulta real [Kos00b]. En el segundo caso, la calidad es evaluada analizando consultas pasadas o ejecutando algoritmos de análisis sobre las fuentes, que enriquecen las estadísticas del mediador [NLF99, WKC+07].

   La optimización de los planes de consulta representa importantes retos en los sistemas de mediación, pues dada la autonomía de las fuentes, la información y las estadísticas que se tienen sobre ellas no es ni muy completa ni muy fiable, dificultando la identificación de los planes óptimos.

4. Coordinación de la Ejecución. El mediador envía las subconsultas a cada una de las fuentes seleccionadas y realiza las operaciones necesarias para retornar a la aplicación una respuesta integrada. Los operadores más frecuentemente ejecutados en el mediador son el *join*, el *order by* y el *union*. De acuerdo a la transparencia de heterogeneidad provista por los adaptadores, la labor de coordinación de la ejecución de la consulta

puede ser más o menos compleja. Por ejemplo, un aspecto crítico en los mediadores es cómo garantizar que las instancias de un concepto provistas por dos fuentes diferentes tengan el mismo identificador. Si el nivel de adaptación no garantiza esta coherencia entre identificadores, el nivel de mediación debe determinar cuándo dos instancias corresponden a la misma. Teniendo en cuenta que esta investigación busca integrar datos en OV, una de las hipótesis es que una instancia tiene el mismo identificador independientemente de cuál sea el adaptador que la provea.

La planeación de la consulta es la función más crítica dentro de un sistema de mediación. Por esta razón, se explicarán más en detalle sus fases.

Reescritura

La reescritura de una consulta Q usando un conjunto de vistas V es el proceso de encontrar una expresión Q' que usa sólo las vistas V, y es equivalente o contiene de manera máxima a la consulta Q [BLR97]. Las vistas se refieren a los elementos del modelo externo. La reescritura de consultas aplicado al contexto de los sistemas de integración debe considerar que las vistas contienen predicados complejos que expresan distinciones entre el contenido de las fuentes. Adicionalmente, aunque las vistas de dos fuentes de datos diferentes pueden ser iguales a nivel de definición, no son reemplazables debido a que cada fuente puede proveer diferentes instancias.

El método de reescritura depende de la manera como se mapeó el esquema global (E_g) del sistema a las vistas definidas en los modelos externos de las fuentes (M_g) . Los enfoques de mapeo más conocidos son llamados *Global-as-View* y *Local-as-View*. Para analizarlos se usará un ejemplo en el área de la salud donde el esquema global está compuesto por tres relaciones:
-paciente(p,n,d), almacena todos los identificadores (p) de pacientes, sus nombres (n) y su año de nacimiento (d).
-tipoProcedimiento(t,n), almacena todos los identificadores (t) de tipos de procedimientos y sus nombres (n).
-procedimiento(p,t), almacena la relación entre pacientes y los tipos de procedimientos efectuados.
Supóngase que la consulta que el nivel de mediación debe reescribir es:

Q(p):-paciente(p,n), tipoProcedimiento(t,"Vasectomía"),procedimiento(p,t)

, La manera de hacer la reescritura usando cada uno de los enfoques se presenta a continuación.

1. **Global-as-View (GAV)**: en este enfoque el contenido de cada elemento g del esquema global E_g, se describe en términos de las vistas del modelo externo M_g de las fuentes. Si un elemento no puede ser mapeado al menos a una vista de una fuente de datos, el elemento es eliminado del esquema global. Siguiendo con el ejemplo, el Cuadro 3.1 presenta el mapeo:

Esquema Global	Vistas	Mapeo
tipoProcedimiento(t,n)	V1(t,n)	tipoProcedimiento(t,n):- V1(t,n)
procedimiento(p,t)	V2(t,n)	tipoProcedimiento(t,n):- V2(t,n)
paciente(p,n,d)	V3(p,t)	procedimiento(p,t):-V3(p,t)
	V4(p,n,d)	paciente(p,n,d):-V4(p,n,d)
	V5(p,n,d)	paciente(p,n,d):-V5(p,n,d)

Cuadro 3.1: Caracterización GAV

Debido a que las fuentes de datos son autónomas no es posible suponer que las vistas V1 y V2 ni V4 y V5 retornarán el mismo conjunto de tuplas. Por ésto la relación *tipoProcedimiento* y *paciente* está definida dos veces.

La reescritura de la consulta Q(p) siguiendo el método GAV se ejecuta en dos fases:

a) Reemplazar los elementos de la consulta por las vistas (modelos externos) que las definen. Esto quiere decir que cada uno de los términos definidos en la consulta deben ser reemplazados por la definición que está contenida en el nivel de mediación. Retomando el ejemplo, la reescritura de Q sería la siguiente consulta conjuntiva:

Q'(p):-V4(p,n,d), V1(t,"Vasectomía"),V3(p,t)
Q'(p):-V4(p,n,d), V2(t,"Vasectomía"),V3(p,t)
Q'(p):-V5(p,n,d), V1(t,"Vasectomía"),V3(p,t)
Q'(p):-V5(p,n,d), V2(t,"Vasectomía"),V3(p,t)

b) Limpieza de la consulta reescrita para evitar redundancias.

El mapeo GAV facilita el proceso de reescritura. Sin embargo, la adición o la eliminación de una fuente implica adaptar la definición del esquema global.

2. **Local-as-View (LAV)**: En este enfoque el modelo externo M_g de cada fuente de datos debe ser caracterizado en términos de una vista q_G sobre el esquema global E_g. Continuando con el ejemplo, el mapeo entre las vistas de los modelos externos y el esquema global es presentado en el Cuadro 3.2:

   La reescritura usando el método LAV varía de acuerdo al algoritmo utilizado. Para ilustrar cómo se realizaría la reescritura de la consulta Q(p), se utilizará el algoritmo Minicon propuesto en [PH01] como una evolución de algoritmos anteriores. Este algoritmo hace la reescritura en tres pasos.

   a) Paso 1: Considera todas las vistas e intenta crear uno o más mapeos de las relaciones (sub-objetivos) de la consulta a las relaciones (sub-objetivos) de las vistas. Estos mapeos son llamados MCD. Para el ejemplo los mapeos creados son:

Esquema Global	Vistas	Mapeo
tipoProcedimiento(t,n) procedimiento(p,t) paciente(p,n,d)	V1(t,n) V2(t,n) V3(p,t) V4(p,n,d) V5(p,n,d)	V1(t,n) :- tipoProcedimiento(t,n) V2(t,n) :- tipoProcedimiento(t,n) V3(p,t) :- procedimiento(p,t) V4(p,n,d) :- paciente(p,n,d),d <1960 V5(p,n,d) :- paciente(p,n,d)

Cuadro 3.2: Caracterización LAV

-MCD V4
mapeo a variables: p ->p; n ->n; d ->d;
sub-objetivos cubiertos: paciente(p,n,d)
-MCD V5
mapeo a variables: p ->p; n ->n; d ->d;
sub-objetivos cubiertos: paciente(p,n,d)
-MCD V1
mapeo a variables: t ->t; "Vasectomía" ->n;
sub-objetivos cubiertos: tipoProcedimiento(t,"Vasectomía")
-MCD V2
mapeo a variables: t ->t; "Vasectomía" ->n;
sub-objetivos cubiertos: tipoProcedimiento(t,"Vasectomía")
-MCD V3
mapeo a variables: p ->p; t ->t;
sub-objetivos cubiertos: procedimiento(p,t)

b) Paso 2: Crea las reescrituras de la consulta combinando los mapeos obtenidos en el primer paso y restringiendo el número de reescrituras a máximo el número de relaciones(sub-objetivos) de la consulta.

Q'(p) :- V4(p,n,d),V1(t,"Vasectomía"),V3(p,t)
Q'(p) :- V5(p,n,d),V1(t,"Vasectomía"),V3(p,t)
Q'(p) :- V5(p,n,d),V2(t,"Vasectomía"),V3(p,t)

En este caso la reescritura usando V1,V2 y V3 es eliminada por que el número de reescrituras no puede superar el número de sub-objetivos de la consulta, que para el ejemplo era de tres.

c) Paso 3: Las redundancias de las reescrituras son eliminadas.

El mapeo LAV favorece la extensibilidad del sistema, pues para agregar una nueva fuente basta con agregar el conjunto de aserciones. Sin embargo, el proceso de reescritura es mucho más extenso, en especial cuando el número de fuentes es alto, ya que el número de mapeos puede crecer exponencialmente con respecto al número de fuentes.

Fuente de Datos	Vista	Limitación
1	V1(t,n)	b,f
3	V3(p,t)	b,f - f,b
4	V4(p,n,d)	b,f,f - f,b,f

Cuadro 3.3: Restricciones de Acceso

Descomposición

La descomposición es la fase que determina si el plan físico de una reescritura es o no factible, es decir puede ser ejecutado por las fuentes de datos de acuerdo a sus limitaciones. Los proyectos de integración actuales determinan esta factibilidad validando las limitaciones de acceso y/o de tratamiento de las fuentes que participan en la reescritura. Las limitaciones de acceso definen qué atributos deben o no estar especificados para que la consulta pueda ejecutarse. Las limitaciones de tratamiento, definen qué operadores lógicos en las condiciones pueden o no ser ejecutados en una fuente. Para ilustrar esta fase se considerará la propuesta presentada en [YLGMU99]. En esta propuesta las limitaciones de acceso de las fuentes son definidas en plantillas que definen las limitaciones sobre cada atributo. Las limitaciones, llamadas "adornos" son las siguientes:
-f (free): El atributo puede o no estar especificado en la consulta
-u (unspecifiable): el atributo no puede estar especificado en la consulta
-b (bound): el atributo debe estar especificado en la consulta
-c[S](constant): el atributo debe estar especificado y adicionalmente debe ser elegido del conjunto de constantes S
-o[S](optional): el atributo puede o no estar especificado. En el caso en que sea especificado debe ser elegido del conjunto de constantes S

Supóngase que los adaptadores establecieron las limitaciones de acceso presentadas en el Cuadro 3.3.

Dada la reescritura: Q'(p) :- V4(p,n,d),V1(t,"Vasectomía"),V3(p,t)
El algoritmo evalúa la factibilidad de la reescritura evaluando la factibilidad de los planes físicos derivados de ella. Si el nodo raíz de por lo menos uno de los planes físicos respeta las limitaciones de acceso, el algoritmo valida las limitaciones de los nodos internos. Si en al menos un plan todos los nodos respetan las limitaciones, la reescritura es factible. En este caso los posibles planes físicos son:

- $P1 = V4 \bowtie V3 \bowtie V1$
- $P2 = V1 \bowtie V3 \bowtie V4$

Como puede observarse, el plan P1 no es factible pues la vista V4 exige que el atributo p o n estén especificados para resolver la consulta. Sin embargo, el plan P2 es factible por lo que la reescritura es factible y no es eliminada.

Optimización de los Planes

La optimización de los planes en el nivel de mediación puede ser llevado a cabo usando heurísticas, usando aproximaciones basada en costos o aproximaciones basadas en utilidad. Un ejemplo de heurística [OV99] es descomponer la consulta en subconsultas lo más pequeñas posibles, cada una de las cuáles es ejecutada en una fuente de datos diferente. En este caso la descomposición es más sencilla, pero el mediador debe ejecutar gran parte del trabajo de integración de respuestas. Adicionalmente, el número de mensajes puede crecer.

Las aproximaciones basadas en costos, pueden tener grandes dificultades en los sistemas de mediación, debido a la dificultad de tener información de las fuentes autónomas. Para reducir el impacto de la falta de estadísticas, una aproximación de optimización consiste en definir un modelo de costo genérico para todos los adaptadores y ajustar ciertos parámetros del modelo para cada adaptador ejecutando un conjunto de consultas de prueba. Esta aproximación es llamada calibración y ha sido propuesta en diferentes proyectos como en [ROH99, GGT96].

Las consultas de prueba son muy útiles para permitir al nivel de mediación usar estrategias de optimización conocidas en las bases de datos distribuidas como los *semi-joins*, que buscan reducir los costos de comunicación [AHY86, HF01] o técnicas de paralelización para mejorar el tiempo de respuestas [LC96, Gea06].

Otras estrategias de optimización se enfocan en descartar los planes de acuerdo a la utilidad que representan en términos de respuestas. En este caso el objetivo de optimización es seleccionar los planes que involucran a las fuentes que mayor utilidad representan a la consulta con respecto al número de respuestas o de calidad de respuestas. Teniendo en cuenta que este tipo de optimización es muy relevante para la ejecución de consultas en OV, el Capítulo 4 se enfocará especialmente en estas estrategias.

3.3. Mediación en Comunidades Virtuales

Como se mencionó, la arquitectura de mediación ha sido la inspiración de muchos proyectos de integración, que la han adaptado a las necesidades de cada contexto. Esta sección presenta un grupo de proyectos de mediación cuyo objetivo es compartir información en comunidades constituídas formalmente o informalmente. Todos los sistemas analizados están constituídos por diferentes nodos autónomos que comparten información y están distribuídos físicamente. El análisis de estos sistemas fue realizado en torno a un conjunto de dimensiones que serán presentadas en la Sección 3.3.1 que permitieron identificar las decisiones tomadas en cada uno de los sistemas.

3.3.1. Dimensiones del Estudio

El objetivo del análisis de estos trabajos es identificar la manera como implementan los niveles de la arquitectura de referencia de mediación, y en especial

la manera como planean las consultas. Las dimensiones evaluadas sobre cada uno de los niveles fueron las siguientes:

Nivel de Aplicación

- Expresividad de las Consultas: Los tipos de consulta aceptados en los sistemas pueden ser usando palabras clave, consultas de tipo Atributo-Operador-Valor, o consultas más ricas como las soportadas en las bases de datos relacionales o como las consultas definidas sobre ontologías en lenguajes como SPARQL [EP07].

Nivel de Adaptación

- Tipo de fuentes de datos soportadas: Los sistemas pueden mediar entre fuentes de datos estructuradas, no estructuradas o semi-estructuradas.
- Almacenamiento de los datos: Esta dimensión evalúa la capacidad del sistema de decidir la ubicación de las fuentes de datos. En algunos casos el sistema tiene control sobre la ubicación de las fuentes. En estos casos las fuentes pueden ser replicadas, distribuidas, trasladadas de un nodo a otro para efectos de eficiencia y disponibilidad. En otros casos la ubicación de las fuentes es definida por el proveedor de las fuentes y el sistema no tiene la posibilidad de decidir dónde estarán ubicadas.

Nivel de Mediación

- Esquema de referencia: Esta dimensión busca identificar si el sistema está soportado en un esquema global de datos sobre el cual se deben hacer las consultas o si existen múltiples esquemas entrelazados, o si hay una combinación de los dos.
- Estructuración de metadatos: Se refiere a la información que los mediadores almacenan de las fuentes. Esta información puede incluir el modelo externo de las fuentes con respecto al esquema global, estadísticas, palabras claves, etc.
- Almacenamiento del catalogo de metadatos: La manera de almacenar el catálogo de metadatos puede ser centralizado o distribuido.
- Método de reescritura: Esta dimensión analiza el enfoque de mapeo que utiliza y la estrategia de reescritura.
- Método de descomposición: Además de dividir las consultas globales en subconsultas, esta dimensión evalúa si existe algún mecanismo para establecer si las fuentes de datos, además de resolver las consultas a nivel de esquema, son capaces de evaluar la consulta, de acuerdo a sus capacidades o contenido.

- Características de las respuestas: Esta dimensión evalúa si el conjunto de respuestas entregadas por el nivel de mediación es completo o abarca sólo una porción de las respuestas disponibles.

Los sistemas de mediación fueron agrupados en tres grupos de acuerdo a la motivación por la cuál fueron creados: Web Oculta, *Peer-to-Peer* y mallas de datos. Los proyectos en el grupo Web Oculta (*Deep Web*) están motivados por la obtención de información en la web que usualmente está oculta para los buscadores tradicionales. Más allá de todas las propuestas creadas por los motores de búsqueda en la web, los proyectos de la web oculta son de interés en este proyecto pues permitieron identificar la dificultad de mediar consultas sobre las bases de datos estructuradas y semi-estructuradas que están detrás de los portales en internet. La motivación de los proyectos P2P es compartir archivos y datos en comunidades informales. Las mallas de datos por su parte están motivadas en crear infraestructuras físicas de red y datos que apoyen a comunidades científicas que manejan grandes volúmenes de información.

3.3.2. Web Oculta (*Deep Web*)

Los sistemas de mediación en este grupo tienen como principal objetivo permitir el acceso efectivo a las bases de datos disponibles en la web. Existen diversos proyectos trabajando por crear nuevos métodos y estrategias para descubrir y mediar el acceso a la web oculta [HMYW04, CLB04, CHZ05, ZHC05, GM05, SMM+08]. En esta sección nos enfocaremos en dos de ellos que por sus características permiten ver diferentes formas para resolver el problema.

MetaQuerier [CHZ05, ZHC05]

MetaQuerier es un sistema que permite buscar e integrar bases de datos en la web. Después de recibir una consulta de usuario, busca las mejores bases de datos disponibles en la web para evaluar la consulta, la traduce usando la plantilla de consulta de cada una de estas bases de datos, une los resultados y se los entrega al usuario final.

1. Nivel de Aplicación Para realizar las consultas en el *MetaQuerier* los usarios definen el dominio en el que desean buscar información (e.g autos, viajes, medicina). En cada dominio el usuario puede expresar consultas de tipo atributo-operador-valor haciendo uso de una interfaz unificada, donde los atributos disponibles varían de acuerdo al dominio.

2. Nivel de Adaptación: MetaQuerier está enfocado en integrar la información de bases de datos estructuradas que sólo pueden ser consultadas usando las interfaces de páginas web. Para homogenizar el acceso a las bases de datos MetaQuerier tiene un componente que busca las fuentes y extrae de ellas las plantillas requeridas para hacer consultas. Cuando el mediador debe evaluar un predicado usando un tipo de dato que no es soportado por la plantilla de la fuente de datos elegida, un componente adaptador transforma los tipos de datos originales de la consulta a los requeridos.

3. Nivel de Mediación: Debido a la naturaleza de las bases de datos disponibles en la web, obtener la información de cada una de ellas es una labor que requiere mucho tiempo. La estrategia de MetaQuerier es utilizar un componente que lanza consultas de prueba sobre cada base de datos para capturar cuál es su semántica. Estas consultas son enfocadas a ciertos tópicos, lo que permite identificar cuál es el dominio de la base de datos. Al finalizar, el componente obtiene las bases de datos disponibles, extrae de ellas las interfaces que utilizan para hacer consultas y las agrupa de acuerdo al dominio al que pertenecen. Adicionalmente, en cada dominio descubre cuáles son las equivalencias semánticas. En resumen, el modelo externo de las fuentes de datos es extraído automáticamente por un componente del sistema y corresponde a las plantillas que cada base de datos soporta como consulta y a las capacidades en términos de qué atributos son obligatorios o no en cada plantilla.

   El modelo de mapeo utiliza el enfoque LAV en donde cada plantilla está descrita en términos de la interfaz unificada. Sin embargo, en la última versión de MetaQuerier [ZHC05] no existe una interfaz unificada, y por lo tanto cada usuario puede usar la interfaz que más le convenga. Para hacer la reescritura en estos casos, *MetaQuerier* compara el predicado de la consulta y las plantillas disponibles usando un enfoque basado en capacidades. Al finalizar reescribe las consultas usando las plantillas de las fuentes que subsumen el predicado. La descomposición de la consulta envía a cada fuente la mayor cantidad posible de predicados, de tal forma que se maximize el uso de las bases de datos. El enfoque de mediación busca entregar la mayor cantidad de respuestas posibles.

   El proyecto *MetaQuerier* no especifica la manera como se almacenan los metadatos utilizados por el nivel de mediación.

Integrador WISE [HMYW04]

WISE es una herramienta que integra las interfaces web de bases de datos. El objetivo es proveer al usuario una interfaz única para hacer las consultas en un dominio específico.

1. Nivel de Aplicación: Se provee un única interfaz con atributos que pueden ser definidos o elegidos por el usuario. Estos atributos varían de acuerdo a las interfaces que se están integrando.

2. Nivel de Adaptación: Las fuentes de datos son bases de datos estructuradas y semi-estructuradas disponibles a través de interfaces web. Estas interfaces no definen explícitamente cuál es su esquema o sus metadatos, por esta razón el adaptador debe obtener autónomamente esta información analizando las páginas web.

   La información que provee el nivel de adaptación es la siguiente:

$$F = (S, \{A_1, A_2, ..., A_n\})$$

donde S es información específica del sitio web como el dominio y la URL. Cada A_i representa un atributo que a su vez debe presentarse como una tupla del tipo:

$$A_i = (L, P, DT, DF, VT, U, Re, \{Ej, Ej+1, \ldots, Ek\}, Ca)$$

donde L es la etiqueta del atributo, P es la posición dentro de la interfaz de A_i, DT es el dominio de A_i, DF es el valor por defecto, VT es el tipo de valor que puede tomar. U son las unidades del valor (si aplica). $\{E_j, E_{j+1}, \ldots, E_k\}$ es la lista de los posibles valores, Re es el tipo de relación de los elementos del dominio, Ca identifica la lista de elementos de restricción.

3. Nivel de Mediación: Para integrar las interfaces de múltiples bases de datos disponibles en la web, se llevan a cabo varias fases:

   - Normalización: Todos los valores de los atributos se modifican para que queden en las mismas condiciones. Por ejemplo, todos se dejan en minúsculas, se eliminan los números, se remueven caracteres especiales, etc.
   - Análisis de relaciones semánticas: Utilizando un diccionario de palabras se identifican sinónimos, hiperónimos, e hipónimos. Se exploran seis posibles relaciones entre atributos: coincidencia exacta, coincidencia aproximada de cadenas, similitud de coseno de nombres, coincidencia sinónima, coincidencia de hiperonomía, coincidencia de hiponomía.
   - Identificación de atributos coincidentes: Se usan los nombres, la especificación del campo (e.g. el tipo de dato), los valores y los patrones, para determinar si dos atributos son coincidencias positivas o coincidencias predictivas.
   - Creación de clusters de atributos: Los atributos coincidentes positivos se agrupan en clusters para identificar el nombre que va a tener el atributo en la interfaz global. Así mismo, los atributos que aún no se han identificado se agrupan en un solo cluster de coincidencias predictivas. Sobre estos últimos se llevan a cabo análisis adicionales para identificar cuáles son coincidentes.
   - Generación de la nueva interfaz integrada.

Web Oculta y las Organizaciones Virtuales

La web oculta puede considerarse una OV no explícita en donde un conjunto de fuentes pertenecientes a organizaciones de uno o más dominios son compartidas y puestas a disposición de los usuarios finales. Los proyectos alrededor de la web oculta son de utilidad en las OV pues proporcionan estrategias para resolver la heterogeneidad y para obtener de forma

automática las interfaces utilizadas por las bases de datos para recibir consultas. Adicionalmente, el mecanismo utilizado para capturar cuál es el tópico de una fuente de datos usando consultas de prueba es de gran utilidad en las OV para detectar qué tipo de información está contenida en cada fuente. No obstante, las estrategias de selección de fuentes buscan entregar la mayor cantidad de respuestas posibles pero no utilizan ningún método para identificar cuáles de estas fuentes entregarán más respuestas, por esta razón los métodos que utilizan pueden seleccionar un gran número de fuentes no relevantes antes de encontrar las respuestas requeridas por el usuario.

3.3.3. Sistemas Peer to Peer(P2P)

Los sistemas peer to peer (p2p) son sistemas distribuidos con características particulares de "auto-organización, comunicación simétrica y control distribuido" [RM06], diseñados para soportar un alto volumen de nodos autónomos, dinámicos y altamente distribuidos. En los últimos años, han tenido un gran auge en ambientes abiertos como el de internet, especialmente para implementar sistemas de intercambio y búsqueda de información.

Los sistemas p2p pueden clasificarse de acuerdo al nivel de centralización o de acuerdo a la estructura de la red [ATS04]. En el primer caso, pueden existir arquitecturas puramente descentralizadas donde todos los nodos actúan como clientes o como servidores de forma dinámica; arquitecturas parcialmente centralizadas en donde algunos nodos asumen roles más importantes y son asignados dinámicamente; y arquitecturas descentralizadas híbridas en donde hay servidores centrales sobre los cuales se realizan las operaciones de localización, pero posteriormente la comunicación es realizada directamente entre los nodos. La segunda clasificación de acuerdo a la estructura de la red, define dos tipos de sistemas, los p2p no estructurados y los p2p estructurados. En los p2p no estructurados la ubicación de recursos es independiente de la topología. En los p2p estructurados la topología es controlada y los recursos se ubican en nodos específicos, generalmente a partir de una función de hash que posteriormente facilita su localización. La primera estructura no garantiza respuestas completas , pues por lo general su esquema de localización se hace a partir de inundación en donde se definen tiempos máximos de espera, que no necesariamente corresponden al tiempo necesario para ubicar todos los recursos asociados. La segunda, garantiza comprehensividad gracias a su manera de indexar usando funciones hash que permite conocer con exactitud el nodo donde fue ubicado urecurso al aplicar la misma función de hash durante la localización.

La primera generación de sistemas de mediación estaban enfocados a compartir e intercambiar archivos como Gnutella [Gnu06] , los inicios de eDonkey [EDo06] y Napster [Nap06]. Éstos fueron creados sobre sistemas P2P centralizados y no estructurados Su funcionalidad está limitada a localizar archivos basados en su correspondencia con una palabra clave o usando un conjunto de propiedades reducidas. La planeación de consultas en estos sistemas es mínima pues están basados en modelos de inundación como mecanismo para enviar

a procesar consultas, lo que no escala bien en consultas complejas o cuando existen muchas consultas.

La evolución por motivos legales, tecnológicos y de requerimientos de usuario ha hecho que los sistemas de mediación de segunda generación se crearan sobre sistemas P2P parcialmente centralizados o descentralizados y estructurados, y que adicionalmente tuvieran más riqueza en el tipo de consultas que soportan y en el tipo de fuentes que integran. Este es el caso de PIER [HHL+03], PIAZZA[TIM+03], APPA [AM07], PeerDB [OTZ+03], y el proyecto de Anillo de Datos [AP09], que son catalogados como sistemas de administración de datos en grid (PDMS por sus siglas en inglés). A continuación se presentará el análisis realizado sobre los PDMS.

PIER [HHL+03, HCH+05]

PIER ofrece un motor de consultas masivamente distribuido que combina la escalabilidad de internet con aproximaciones de bases de datos. La arquitectura de PIER está compuesta por tres niveles. El nivel de aplicaciones, el nivel PIER, que es el que realiza la mediación, y el nivel DHT que es donde se encuentran las fuentes de datos.

1. Nivel de Aplicación: PIER permite al usuario definir consultas usando el lenguaje SQL. PIER provee 14 operadores lógicos y 26 operadores físicos.

2. Nivel de Adaptación: PIER está enfocado a fuentes de datos con modelos de datos relacionales. Los adaptadores de las fuentes de datos son responsables de insertar, actualizar y borrar los registros de las bases de datos (o referencias a estos registros) en el sistema PIER.

   Cada registro entregado por los adaptadores a PIER es distribuido en una red DHT. Para hacerlo, PIER le asigna a cada uno de ellos un identificador compuesto por: $(nombredelarelacin, valordelallaveprimaria, enteroaleatorio)$, donde el entero aleatorio sirve para identificar registros de tablas que no tengan llave primaria. La distribución de los registros (o de las referencias a los registros) permite conocer con certeza qué nodos son los responsable de almacenar los registros de una relación.

   Los adaptadores de las fuentes de datos son responsables de invocar periódicamente en PIER un método que renueva su vinculación al sistema y que asegura que los datos que él provee sigan disponibles en el sistema.

3. Nivel de Mediación: PIER no maneja un esquema global pues cada registro es auto-descriptivo. La estrategia utilizada en PIER para proveer un motor de consultas escalable a millones de nodos es distribuir los identificadores de cada registro en índices basados en funciones de hash que permiten obtener eficientemente los registros (o las referencias a los registros) que corresponden a una consulta.

   Para planear una consulta, incluso las que tienen operadores de desigualdad u operaciones de mínimos y máximos, PIER utiliza un índice que le

permite llegar rápidamente a los registros que cumplen con el predicado. Este índice es creado sobre un árbol PHT (Prefix Hash Tree [RRHS04]) que tiene la capacidad de distribuir las referencias de tal manera que los registros con valores cercanos quedan en un mismo nodo hoja o en una hoja cercana, permitiendo hacer una recuperación eficiente.

En el caso en que la consulta tenga múltiples operadores, PIER deja en una cola los datos que ya han sido evaluados y estos son usados por el siguiente operador, para seguir evaluando las condiciones restantes. Una vez se conocen los identificadores de los registros que cumplen el predicado, PIER contacta los nodos que almacenan estos registros. La latencia de los índices determina qué tan correctas y completas son las respuestas de PIER.

PIAZZA [TIM+03]

PIAZZA es un sistema de manejo de datos soportado en una infraestructura P2P que permite mediar fuentes con diferentes esquemas sin necesidad de crear un esquema global.

1. Nivel de Aplicación: Cada usuario define las consultas en el esquema que prefiera.

2. Nivel de Adaptación: PIAZZA soporta cualquier tipo de fuentes de datos siempre y cuando sea posible expresar un mapeo de su esquema a otros esquemas. Los datos no son almacenados, sino una descripción de ellos.

   Teniendo en cuenta que cada usuario puede usar el esquema preferido para hacer consultas, los nodos que quieren compartir su información en el sistema deben describir su mapeo al esquema de otro nodo que ya se encuentre en el sistema. El lenguaje para definir los mapeos en PIAZZA es llamado PPL (*PIAZZA Peer Language*), éste permite hacer la descripción de las relaciones de igualdad e inclusión usando el enfoque GAV o el enfoque LAV. La creación de mapeos es asistida por algoritmos que comparan esquemas. Adicionalmente, cada nodo proveedor puede entregar un resumen de los datos que contiene, expresados de la forma atributo-valor.

3. Nivel de Mediación: Cuando el nivel de mediación recibe la consulta lleva a cabo un proceso transitivo de reescritura que valida todos los mapeos que hay desde el nodo en el que se lanzó la consulta hacia los otros nodos. De esta forma propaga la consulta a todos los nodos relevantes. Adicionalmente, el catálogo de metadatos centralizado, que contiene los resúmenes de los datos de cada nodo, permite descartar la reescritura al esquema de los nodos que no entregarán resultados o que entregarán resultados redundantes.

   Como se puede observar, debido a la manera como están descritos los mapeos, la reescritura utilizada en PIAZZA sigue un enfoque transitivo donde se reescriben las consultas desde el nodo inicial hasta el último nodo

que es posible mapear. Esta forma de reescritura hace que la completitud de las respuestas entregadas esté directamente relacionada con la calidad de los mapeos definidos.

APPA [AM07]

APPA es un sistema de manejo de datos en P2P creado para ser usado en aplicaciones avanzadas que requieren gran cantidad de datos.

1. Nivel de Aplicación: A pesar de que en APPA existe un esquema común, las consultas pueden hacerse en el esquema de cualquiera de los nodos registrados.

2. Nivel de Adaptación: APPA define un esquema común al que deben mapearse todos los nodos que quieran pertenecer al sistema. Estos mapeos deben mantenerse mientras se desee compartir la información.

   A pesar de que los datos son compartidos por un nodo, para mejorar la persistencia de los datos, APPA replica los datos en varios nodos. Si un nodo no está disponible los datos que éste proveía aún pueden ser obtenidos. La replicación de los datos en la red P2P utiliza un servicio que asigna las réplicas (o porciones de las réplicas) a un nodo de acuerdo a una función de hash, que posteriormente facilita su obtención.

3. Nivel de Mediación: A nivel de mediación los metadatos que se mantienen son los mapeos entre el esquema local de los nodos y el esquema común. Estos metadatos son almacenados en la red P2P usando un función de hash.

   Cuando el nivel de mediación recibe una consulta, ésta es reescrita inicialmente al esquema común. Posteriormente, se reescribe a los esquemas de los nodos usando el enfoque LAV. Por cada relación incluída en la consulta, APPA selecciona el mejor nodo candidato usando una función de costo. Posteriormente, la descomposición de la consulta genera las subconsultas que cada nodo debe evaluar. Las respuestas obtenidas son integradas y entregadas a la aplicación. Como en otros sistemas basados en P2P pueden ser incompletas por la volatilidad de los nodos.

PeerDB [OTZ+03]

PeerDB es un sistema P2P descentralizado creado para compartir datos distribuidos estructurados sin necesidad de usar un esquema global. Este sistema está construído sobre BestPeer, una plataforma para aplicaciones P2P que combina la tecnologías de las redes P2P con los agentes móviles.

1. Nivel de Aplicación: A nivel de aplicación se provee una interfaz de usuario que permite hacer consultas tipo SQL usando el lenguaje local.

2. Nivel de Adaptación: Las fuentes de datos soportadas por PeerDB son bases de datos relacionales. Éstas son administradas por un componente llamado sistema manejador de objetos que se comunica con el manejador de bases de datos del nodo. Por cada relación creada en un nodo, se almacena un grupo de metadatos en un diccionario local. Las relaciones que quieren ser compartidas se almacenan en un diccionario exportado.

3. Nivel de Mediación: El mediador almacena los diccionarios locales y los diccionarios exportados de las fuentes de datos. En ambos diccionarios se mantienen el nombre de la relación, el nombre de los atributos y por cada atributo se almacenan un conjunto de palabras claves que permitirán posteriormente hacer el mapeo con las relaciones de otros nodos. Estas palabras claves pueden ser sinónimos, descriptores o cualquier otra palabra que haga más claro el significado del atributo. Por ejemplo si un atributo se llama *lon*, la palabra clave puede ser *longitud*.

   El mediador tiene un componente llamado *DBAgent* cuya función es crear el ambiente para que los agentes que ayudan a ejecutar la consulta operen. Cuando una consulta llega a un nodo se ejecutan dos fases. En la primera fase el *DBAgent* crea un *MasterAgent* que es el encargado de manejar toda la consulta. El *MasterAgent* aplica una función de concordancia para ver si el diccionario local puede resolver la consulta, si es así le envía al usuario las relaciones que pueden ser provistas por el nodo local. Al mismo tiempo crea un *RelationMatchingAgent* que va a los nodos vecinos a evaluar si sus diccionarios exportados contienen relaciones que concuerden con las relaciones de la consulta, usando la función de concordancia. Además de la consulta estos agentes tienen la dirección IP del nodo en el que fueron creados y un TTL que decrece cada vez que va consultando nodos. A medida que los *RelationMatchingAgent* encuentran relaciones se las van enviando al nodo que los lanzó. Al finalizar todas las relaciones obtenidas son presentadas al usuario quien define cuáles de ellas quiere consultar. En PeerDB, dada una consulta Q (R,A,C) y una relación D con atributos T, el grado en el que D concuerda con Q se calcula usando la función de concordancia 3.1:

$$Match(Q, D) = \frac{(wt_r * r) + (wt_a * N_{match}(A \cup C, T))}{wt_r + (wt_a * N(A \cup C))}. \qquad (3.1)$$

   En la función de concordancia presentada en la fórmula 3.1 wt_r y wt_a son pesos asignados para reflejar la importancia de la concordancia entre la relación y el atributo respectivamente. r toma valores 1 o 0. r es 1 si y sólo si D y R comparten palabras claves. $N_{match}(A \cup C, T)$ se refiere al número total de palabras claves que concuerdan de los atributos que están en Q y los que están en D. $N(A \cup C)$ es el número total de palabras claves diferentes.

   En la segunda fase el *MasterAgent* recibe las relaciones que el usuario quiere consultar y crea los *DataRetrievalAgent* que son los encargados de

ir a cada uno de los nodos responsables de cada relación elegida y de reformular la consulta usando el diccionario exportado de cada nodo. Una vez que se tienen las consultas, se crea un *WorkAgent* cuya función es ejecutar la consulta en el componente administrador de objetos. Finalmente, las respuestas son enviadas al nodo que inició la consulta. El número de respuestas incompletas depende del TTL establecido en los *RelationMatchingAgent*.

Adicionalmente, en el nivel de mediación se maneja un caché de datos, que almacena las respuestas más recientes obtenidas de otros nodos. Éstas son usadas para entregar más rápidamente las respuestas.

Anillo de Datos [AP09]

El anillo de datos (Data Ring) es una infraestructura creada para permitir a los usuarios administrar y compartir sus datos de forma sencilla. En esta infraestructura, el usuario determina qué datos quiere compartir y el anillo de datos se encarga de indexarlos para dejarlos disponibles, replicarlos para garantizar su disponibilidad y reorganizar su almacenamiento para tener mejor desempeño en las consultas.

1. Nivel de Aplicación: El nivel de aplicación ofrece dos maneras de hacer consultas. La primera de ellas definiendo consultas sencillas sobre un esquema global en forma de patrones de árboles (*tree-pattern queries*). La segunda, usando un lenguaje declarativo que permite definir consultas tradicionales sobre los recursos almacenados en un nodo específico, consultas que permiten monitorear cambios en un recurso o consultas abiertas que permiten consultar recursos que generan nueva información de manera constante, como es el caso de los sensores.

2. Nivel de Adaptación: El anillo de datos está compuesto por nodos que desean compartir datos o servicios. Los datos son presentados al anillo a través de una descripción en XML Activo (*ActiveXML*). Los servicios son presentados usando el lenguage *Web Services Description Language*(WSDL). Cada nodo contiene un adaptador que se encarga de traducir la consulta a su lenguaje local y ejecutarla.

3. Nivel de Mediación: El catálogo de metadatos incluye la descripción en *ActiveXml* de los datos que cada nodo conoce, las capacidades de las fuentes, la localización de los nodos y estadísticas de distribución de carga de trabajo. Los archivos en activeXML se distribuyen utilizando una red basada en tablas de hash distribuido (*DHT* por sus siglas en inglés). La distribución funciona de la siguiente manera. Cada elemento del archivo xml se identifica usando un identificador estructural *sid* que está compuesto por tres parámetros (comienzo,fin, nivel). El comienzo es el número del *tag* que abre el elemento dentro del documento xml. El fin es el número del tag que cierra el elemento dentro del documento xml. El nivel corresponde al nivel en el que se encuentra el elemento dentro del documento xml. Los

sid permiten definir si un elemento es ancestro o no de otro elemento. El catálogo indexa las etiquetas y las palabras asociadas a cada elemento que son llamados *términos*. Las etiquetas son indexadas usando el identificador compuesto por (p,d,sid,l) donde p es el nodo que la almacena, sid es el identificador estructural del documento y l es la etiqueta. Para el caso de las palabras la estructura es similar (p,d,sid,w), donde w es la palabra.

Cuando el anillo de datos recibe una consulta en forma de patrón el mediador evalúa los términos involucrados y busca su información en el catálogo. De esta manera identifica qué nodos tienen información relevante y les envía a ellos la consulta.

Para que la ejecución de consultas sea robusto, el anillo de datos unifica el proceso de optimización y ejecución de consultas con un proceso de recuperación ante fallas que permite continuar con la ejecución de una consulta incluso cuando el nodo que se iba a consultar sale del sistema.

PDMS y las Organizaciones Virtuales

Los PDMS han desarrollado novedosas estrategias para mantener distribuidos los metadatos que utiliza el nivel de mediación y en algunos casos, como en el proyecto PIER, para distribuir los datos que son compartidos. En las OV estas técnicas de distribución de metadatos son de gran utilidad para evitar que sólo un sitio mantenga registrados los metadatos de todas las fuentes disponibles y que por lo tanto se convierta en un cuello de botella, más aún cuando son OV de gran escala. Sin embargo, es necesario adaptar estas estrategias de distribución si son aplicadas en las OV. A diferencia de las redes P2P, en las OV existe un nivel de compromiso de disponibilidad mayor por parte de los proveedores de las fuentes haciendo innecesario el uso de estrategias costosas usadas en los PDMS para soportar la volatilidad de nodos.

Por otra parte, los manejadores de datos soportados en la infraestructura P2P han desarrollado grandes avances para evitar la centralización del proceso de mapeo de los esquemas locales de las fuentes de datos al esquema global del mediador. La distribución y flexibilidad que proveen las propuestas de PDMS como PIAZZA y PeerDB para realizar los mapeos benefician la escalabilidad del sistema y facilitan la incorporación de nuevos recursos. Este concepto de mapeo distribuido es útil en las OV de gran escala porque permite incorporar más fácilmente nuevos recursos a compartir.

Finalmente, a pesar de que la evolución de los PDMS ha estado enmarcada principalmente a garantizar la escalabilidad en número de fuentes y en número de consultas, los métodos de optimización utilizados al resolver una consulta no son suficientes cuando son aplicados en contextos de OV de gran escala. Éstos no consideran aspectos tales como cuál es la mejor fuente para resolver una consulta a nivel de contenido. Si bien en ciertos

casos como en el proyecto APPA se selecciona el mejor nodo candidato para resolver una consulta, la selección está más enfocada en reducir costo que en aumentar la utilidad. Por esta razón los métodos de optimización usados en los PDMS no son suficientes para las OV.

3.3.4. Mallas de Datos

Una malla de datos es una infraestructura de "servicios que ayudan a descubrir, transferir y manipular grandes volúmenes de datos almacenados en repositorios distribuidos y también, crear y administrar copias de estos repositorios" [ESIT07]. Estos servicios pueden clasificarse en [CFK+99]: servicios base, que son aquellos indispensables, independientemente de las necesidades específicas del problema que se quiere resolver; y los servicios de alto nivel, que modelan necesidades específicas para solucionar problemas particulares. La infraestructura de malla de datos es considerada la nueva generación de sistemas de administración de datos inter-organizacionales pues proporciona los servicios para administrar datos distribuidos y heterogéneos provenientes de organizaciones autónomas. A continuación se describen los principios de evaluación de consultas de un grupo representativo de este tipo de sistemas.

SLA-based Data Integration on Database Grids [Nea07]

Este proyecto propone un modelo de integración de datos basado en acuerdos de calidad de servicio para ejecutar consultas que implican la integración de la información de múltiples bases de datos autónomas en la grid.

1. Nivel de Aplicación: SLA permite la definición de consulta tipo SQL. Adicionalmente, el usuario puede definir el nivel de servicio requerido por parte de las fuentes que van a resolver la consulta.

2. Nivel de Adaptación: El sistema está dirigido a integrar fuentes de datos estructuradas y semi-estructuradas representadas a través de un servicio grid de datos que permite homogenizar su acceso. Todas las fuentes que van a compartir datos deben proveer metadatos que describan las características generales de la fuente como métodos de acceso y descripción de capacidades. Adicionalmente, debe describir los niveles de servicio de la fuente incluyendo información de los esquemas de datos que provee el servicio, costo del procesamiento de una consulta, el precio de usar el servicio, la cantidad de datos proveída; y atributos opcionales como privilegios y restricciones de seguridad y reglas de compensación que se ejecutan en el caso en que se incumpla un acuerdo de servicio. El esquema de las fuentes debe estar definido en términos de una vista sobre el esquema global (o de referencia).

3. Nivel de Mediación: El principio de mediación es la definición de un plan de consulta no sólo a partir del esquema de las bases de datos autónomas sino también de los niveles de servicio que prestan. Los niveles de servicio

son descritos a través de un modelo de descripción de recursos. El mediador provee un esquema global de referencia que es utilizado para hacer consultas y al cual todas las fuentes deben mapearse.

Para ejecutar una consulta el modelo propuesto involucra los siguientes elementos:

- Administrador de Consultas: Analiza la consulta y genera el conjunto de subconsultas que hacen parte del plan de ejecución. Para hacerlo utiliza un algoritmo tradicional de reescritura bajo el enfoque GAV.
- Descubridor de Recursos: Invoca un agente que compara el esquema de las subconsultas con el esquema de las fuentes para identificar qué fuentes coinciden y así generar las soluciones candidatas de la consulta.
- Evaluador de Recursos: Apoya al descubridor de recursos para que filtre las fuentes de acuerdo al nivel de servicio ofrecido.
- Extractor de Datos: Accede a las bases de datos definidas en las soluciones de consulta para obtener los datos solicitados de cada una de ellas.
- Agente de Integración: Responsable de la integración de los datos de las diferentes bases de datos extraídas.
- Administrador Global de Consultas: Mapea la respuesta integrada al formato requerido por la consulta.

A partir de este modelo es posible resolver consultas de agregación en donde el resultado es la unión de los resultados obtenidos en fuentes de datos diferentes, consultas que incluyan *joins* entre bases de datos diferentes y consultas complejas que además de lo anterior requieran algún tipo de ordenamiento o procesamiento como min y max.

Arquitectura de Mediación para Sistemas de Información Grid [WBT05]

Este proyecto propone un servicio llamado Grid Data Mediation Service (GDMS) que permite reducir la complejidad de las aplicaciones que requieren tener acceso a diferentes fuentes de datos heterogéneas. El servicio actúa como un mediador entre las aplicaciones de usuario y los servicios proporcionados por OGSA-DAI. GDMS provée a las aplicaciones transparencia de localización, nombre, esquema y lenguaje.

1. Nivel de Aplicación: GDMS permite definir consultas tipo SQL en términos de un esquema global.

2. Nivel de Mediación: Los metadatos almacenados en el nivel de mediación son los mapeos entre los esquemas de las fuentes y el esquema global. Adicionalmente, se guarda información acerca de las réplicas existentes en el sistema.

   Cuando una consulta llega, el servicio ejecuta tres etapas:

- Reescritura de consultas, en donde se traduce del esquema global a los esquemas de las fuentes de datos lógicas usando un modelo de reescritura GAV.
- Optimización de consulta de acuerdo al estado físico de la grid para poder decidir cuál es la réplica más adecuada para usar durante la ejecución de la consulta.
- Ejecución de consulta, que se realiza comunicándose con los adaptadores OGSA-DAI [KAA+05].

Las respuestas de este servicio son completas con respecto a los nodos que están disponibles en el sistema en el momento de recibir la consulta.

3. Nivel de Adaptación: La arquitectura propuesta del servicio incluye un mapeo de todas las fuentes a un esquema global que denominan *Virtual Data Source*.

Mallas de Datos y las Organizaciones Virtuales

Las mallas de datos son la infraestructura que más se ha relacionado con las OV. Sus técnicas para almacenar y administrar grandes volúmenes de datos son su principal característica gracias a estar soportadas en recursos con alto poder de almacenamiento. Las estrategias que utilizan los servicios de consulta en las mallas de datos para planear las consultas se han dirigido a seleccionar las fuentes de acuerdo a los niveles de servicio ofrecido por las fuentes calculado de acuerdo al esquemas de datos, al costo del procesamiento de una consulta, el precio de usar la fuente y a la cantidad de datos proveída. A pesar de que esto es útil para descartar fuentes, no es suficiente para detectar cuáles son las fuentes que responderán a la consulta y de estas cuáles son las más relevantes.

3.4. Síntesis

La mediación de datos distribuidos y heterogéneos es un problema que se ha explorado hace más de 20 años. La arquitectura de referencia de mediación está compuesta por cuatro niveles: el nivel de aplicación, el nivel de mediación, el nivel de adaptación y el nivel de fuentes de datos. La inteligencia para planear y ejecutar las consultas que involucran múltiples fuentes está en el nivel de mediación. Este nivel se encarga de recibir la consulta en el lenguaje global del sistema y reescribirla en términos de las vistas que cada fuente de datos tiene sobre el esquema. Finalmente, la consulta reescrita genera un conjunto de subconsultas que cada fuente de datos seleccionada debe responder.

Esta arquitectura de referencia ha sido la base para crear gran cantidad de sistemas de integración. Sin embargo, su implementación en diferentes contextos requiere adaptaciones para asegurar escalabilidad, robustez y desempeño.

Los sistemas de mediación de la web oculta tienen como objetivo permitir el acceso efectivo a las bases de datos disponibles en la web. Para hacer esto de forma adecuada el nivel de adaptación es enriquecido con componentes que le permiten adquirir información de fuentes de datos, pues estas no son cooperativas. En los proyectos MetaQuerier [CHZ05, ZHC05] y Integrador WISE [HMYW04] se pueden observar diferentes aproximaciones para hacer esta extracción de metadatos de las fuentes de datos.

Por su parte los sistemas de mediación creados sobre sistemas P2P deben ser capaces de mediar un gran número de fuentes volátiles. Para poder proveer sistemas escalables y con alto desempeño, estos sistemas han modificado principalmente el nivel de mediación distribuyendo la capacidad de mediación a múltiples nodos y evitando el uso de un esquema global como es el caso de PIER [HHL+03], PIAZZA[TIM+03], PeerDB [OTZ+03]. Adicionalmente, para garantizar la disponibilidad de los datos se manejan réplicas dentro del sistema de mediación como es el caso de APPA [AM07] o mecanismos de tolerancia a fallas como es el caso del proyecto de Anillo de Datos [AP09].

Finalmente, los sistemas de mediación creados para administrar datos interorganizacionales sobre mallas de datos han dirigido sus esfuerzos a garantizar calidad en la prestación de servicios y equilibrio entre los proveedores de datos. Por esta razón, durante la planeación de las consultas, además de tener en cuenta el esquema de las fuentes de datos, también son analizados los niveles de calidad provistos por las fuentes [Nea07] y el estado de la infraestructura [WBT05] de tal forma que la elección de las fuentes satisfaga más al usuario final, al mismo tiempo que se mantienen equilibradas las cargas de trabajo en el sistema.

Todos estos proyectos de mediación de datos en comunidades de gran escala fueron la motivación de este proyecto. La arquitectura propuesta para compartir datos en las OVs toma elementos de ellos, y enriquece el nivel de mediación para que sea capaz de elegir las fuentes de datos no sólo por sus atributos externos (por ejemplo disponibilidad, distancia), o por su esquema, sino también por su contenido y relevancia para resolver una consulta. Si bien los sistemas de mediación existentes ya han resuelto problemas como la heterogeneidad, la distribución, la gran escala, ninguno de ellos se ha concentrado en dos de los problemas más importantes en las OVs de gran escala. El primero de ellos es cómo reducir el número de fuentes utilizadas cuando las fuentes tienen esquemas similares y el segundo de ellos es como integrar las respuestas a nivel de instancia.

Capítulo 4

Selección de Fuentes de Datos

4.1. Introducción

La selección de fuentes de datos es el proceso de identificar el conjunto de fuentes de datos que deben participar en el procesamiento de una consulta. Este proceso es también conocido como selección de servidores y selección de base de datos, de acuerdo al contexto de aplicación. El objetivo de este proceso es descartar durante planeación las fuentes de datos que no se requieren estrictamente para la consulta, bien sea porque no tienen nada que aportar ó porque lo que tienen lo pueden aportar otras fuentes debido al solapamiento o replicación entre fuentes. El proceso de selección busca evitar la ejecución de consultas en fuentes que entregarán resultados vacíos o duplicados [GWJD03]. La selección de fuentes de datos tiene un alto impacto en los sistemas de gran escala debido al sobrecosto que genera tener en cuenta un alto número de fuentes durante el proceso de reescritura y descomposición de las consultas.

La investigación en esta área ha sido promovida principalmente por los requerimientos de la web, por los sistemas de intercambio de archivos en comunidades virtuales y por las bases de datos distribuidas. En todos estos contextos es necesario reducir el número de fuentes a consultar para evitar la planeación exhaustiva de consultas que ralentiza su solución. Este capítulo presenta los principales enfoques de selección de fuentes encontrados en la literatura con el fin de identificar las propuestas de selección más indicadas para cubrir los requerimientos de las OV de gran escala.

Para realizar el análisis de los principales enfoques encontrados, éstos fueron divididos en dos grandes grupos: las estrategias que están dirigidas a contextos con fuentes de datos estructuradas, principalmente provenientes del mundo de las multi-bases de datos , y las que están dirigidas a fuentes no estructuradas, provenientes de las necesidades de algunas comunidades en la web. La Sección 4.2 presenta el primer grupo y la Sección 4.3 el segundo. Finalmente, la Sección 4.4 concluye el capítulo analizando la utilidad de estas estrategias al ser aplicadas en las OV.

4.2. Estrategias Orientadas a Fuentes de Datos Estructuradas

Estas estrategias están enmarcadas en la necesidad de resolver consultas sobre un conjunto de fuentes de datos autónomas que están distribuidas y que almacenan información de forma estructurada, generalmente en bases de datos relacionales.

4.2.1. Orientadas a Capacidades

Las propuestas orientadas a capacidades fueron las precursoras en el tema de mediación de fuentes de datos heterogéneas. Sus estrategias de planeación propusieron los enfoques de reescritura que actualmente conocemos como Local-As-View y como Global-As-View los cuales fueron presentados en el Capítulo 3. Además de la selección de fuentes que se hace implícitamente durante el proceso de reescritura, y que la mayoría de estrategias utilizan, la estrategias orientadas a capacidades seleccionan las fuentes usando metadatos que especifican sus limitaciones al ejecutar un tipo de consulta.

1. Information Manifold [LRO96]

   Contexto. El Information Manifold es un sistema creado para proveer acceso uniforme a una colección de fuentes de datos estructuradas, heterogéneas y distribuidas en la web. En este contexto las fuentes son incompletas por lo que el resultado es la unión de los resultados obtenidos de múltiples fuentes. El principio para mediar el acceso a estas fuentes está en la descripción del contenido y de las capacidades de evaluación de cada una de las fuentes.

   Metadatos. Cada fuente de datos es descrita usando tres elementos: la vista que tiene sobre el esquema global, el contenido de cada vista y las capacidades. Las vistas y el contenido son definidas usando consultas conjuntivas sobre el esquema global, es decir usando el enfoque LAV (fue este proyecto el que definió este enfoque de reescritura). Las capacidades son definidas usando lo que ellos denominaron registros de capacidades. Un registro de capacidad es un tupla con cinco elementos $(S_in, S_out, S_sel, min, max)$, donde S_in son los atributos que pueden entrar, S_out son los atributos que pueden ser retornados, S_sel son los atributos sobre los cuales la fuente de datos puede aplicar selecciones, min y max representan el número mínimo y máximo de atributos que pueden entrar a la fuente, respectivamente.

   Estrategia de Planeación. La estrategia de planeación utiliza el algoritmo de reescritura llamado *Bucket* que supone que la consulta está escrita usando *Datalog* [AHV95]. En este algoritmo se crea un bucket por cada sub-objetivo de la consulta (cada relación) y se le asocia a cada uno las fuentes de datos que pueden retornar tuplas para él. En este paso se eliminan las fuentes que por su descripción no pueden satisfacer

la consulta. Posteriormente, se consideran todas las posibles combinaciones entre las fuentes de datos tomando una de cada *bucket*, y verificando si la combinación (incluyendo el orden de las fuentes) produce un plan semánticamente correcto, o si se puede hacer semánticamente correcto adicionando operaciones al plan. Un plan es semánticamente correcto si puede ser ejecutado dadas las capacidades de las fuentes. Finalmente, se eliminan las redundancias de cada plan.

Aplicación en las OV. El algoritmo de selección del *Information Manifold* supone la descripción del contenido de las fuentes. Esta suposición permite descartar las fuentes que no satisfacen la consulta desde la etapa de reescritura. Esta estrategia aplicada directamente en las OV tiene dos inconvenientes. El primero de ellos es que en una OV no es escalable ni viable definir el contenido de las fuentes para cada atributo que puede ser parte de la condición de la consulta. Sería necesario encontrar una manera de describir el contenido de forma más general y que al mismo tiempo diferenciara a las fuentes. El segundo inconveniente es que hacer un producto cartesiano de todas las posibles fuentes que pueden resolver un sub-objetivo de la consulta no sería escalable cuando cada sub-objetivo puede ser resuelto por un número muy alto de fuentes, y no sería necesario si las respuestas que provee una fuente pueden obtenerse de otra fuente como sucede en las OV.

2. TSIMMIS [YLGMU99]

   Contexto. TSIMMIS es un mediador creado para contextos de datos en donde múltiples fuentes de datos estructuradas, distribuidas y heterogéneas tienen diferentes capacidades de procesamiento. Las capacidades de una fuente determinan qué características debe tener una consulta para que pueda ser evaluada por la fuente. El mediador debe dirigir las consultas a aquellas fuentes que pueden resolverlas teniendo en cuenta sus limitaciones de capacidades.

   Metadatos. Para realizar la planeación de consultas TSIMMIS utiliza las vistas de las fuentes de datos con respecto al esquema global del sistema y las capacidad de evaluación de consultas descritas usando el lenguaje de "adornos". En este lenguaje cada atributo de una vista es asociado a un grupo de adornos. Los adornos disponibles son:

   - f (free): El atributo puede o no estar especificado en la consulta.
   - u (unspecifiable): el atributo no puede estar especificado en la consulta.
   - b (bound): el atributo debe estar especificado en la consulta.
   - c[S](constant): el atributo debe estar especificado y adicionalmente debe ser elegido del conjunto de constantes S.
   - o[S](optional): el atributo puede o no estar especificado. En el caso en que sea especificado debe ser elegido del conjunto de constantes S.

Como puede observarse los adornos permiten expresar de manera formal los atributos que pueden estar especificados en las condiciones de las consultas, las restricciones de estas condiciones y la estructura que puede tener la consulta. Adicionalmente, para elegir los mejores planes TSIMMIS utiliza una función de costo utilizando el tamaño esperado de las fuentes de datos y la selectividad de las condiciones en cada fuente.

Estrategia de Planeación. Cuando una consulta llega al mediador ésta se reescribe en términos de las vistas de las fuentes usando el enfoque GAV. Las reescrituras generadas son entregadas a un componente que las descompone, validando las limitaciones de capacidades de las fuentes para generar planes factibles. Este proceso de descomposición se lleva a cabo en tres pasos: evaluación de concordancia, generación de secuencias y optimización. La evaluación de concordancia genera una tabla en donde por cada condición de la consulta se identifica qué fuentes la pueden evaluar y qué restricciones tiene cada fuente. Posteriormente la generación de secuencias define el orden en que cada subconsulta debe ejecutarse para respetar las limitaciones de las fuentes y determina qué operaciones deben ejecutarse en el mediador. El algoritmo usado en este segundo paso depende del tipo de consulta. En el caso en que la consulta requiera fusionar el resultado proveniente de muchas fuentes (como es el caso en las OV), se generan todos los órdenes posibles para evaluar las condiciones y los planes factibles asociados a cada orden. En estos planes se genera una consulta por cada condición y por cada fuente que es capaz de resolverla. Finalmente, el paso de optimización elige de los planes factibles el más eficiente evaluando en cada uno el costo de las consultas de selección en cada fuente y de las consultas de *semi-join*.

Aplicación en las OV. La selección de fuentes en esta propuesta es un proceso implícito dentro de la planeación en donde son descartadas las fuentes que no tienen las capacidades de evaluar la condición. A pesar de que el algoritmo usado para planear las consultas de fusión es útil para entregar respuestas rápidamente al usuario, al incluir en los planes a todas las fuentes que pueden resolver una consulta se hace la suposición de independencia entre fuentes lo cual no es cierto en las OV.

3. Mediación a Gran Escala

   Las estrategias de PIER [HHL+03], PIAZZA[TIM+03], PeerDB [OTZ+03], APPA [AM07] y el Anillo de Datos [AP09] presentadas en la Sección 3.3 también utilizan el enfoque basado en capacidades. Sin embargo, su estrategia de mediación no sólo involucra la planeación sino también la búsqueda eficiente de los metadatos que permiten conocer qué fuentes son seleccionadas. En ese sentido, la manera como estas estrategias distribuyen los metadatos puede hacer más escalable la mediación de fuentes en las OV.

4.2.2. Orientadas a Calidad

El principio de reducción de estas estrategias es la calidad de los datos contenidos en las fuentes. Esta calidad por lo general es medida en términos de que tan completos son los registros contenidos en las bases de datos.

1. QPIAD [WKC+07, KFCK07, WKK+09]

   Contexto. QPIAD es una estrategia de reescritura para sistemas de mediación de fuentes estructuradas y heterogéneas que permite obtener las respuestas relevantes a una consulta, incluso si éstas son respuestas incompletas. Una respuesta es incompleta si tiene valor nulo en uno de los atributos restringidos en la consulta, pero puede ser demostrado que es parte de la respuesta. Por ejemplo en una sistema de mediación de fuentes de datos con información de automóviles una tupla con los atributos modelo=Accord y fabricante=nulo es una respuesta incompleta para una consulta solicitando la información de todos los automóviles fabricados por Honda. En resumen, QPIAD está diseñado para contextos en donde las fuentes de datos tienen baja calidad y por lo tanto es necesario hacer el máximo esfuerzo para obtener la mayor cantidad de respuestas. La respuesta entregada al usuario es la unión de las respuestas entregadas por cada fuente.

   Estrategia de Planeación. La planeación y la ejecución están entrelazadas en dos fases. La primera fase reescribe la consulta original usando la descripción de las fuentes, y las envía a todas las fuentes que son capaces de evaluarla. Las respuestas obtenidas de esta primera fase son respuestas certeras pues evalúan todas las condiciones de la consulta. La segunda fase aprende dependencias funcionales entre los atributos, usando las respuestas certeras y una muestra de registros previamente obtenida durante la fase de alimentación del sistema. Las dependencias encontradas son usadas para generar nuevas reescrituras de la consulta original. Estas nuevas reescrituras se diferencian de las primeras en las condiciones que solicitan. Por ejemplo, para la consulta solicitando información de automóviles cuyo fabricante=honda, la nueva reescritura puede ser modelo=accord, debido a la correlación encontrada entre fabricante y modelo y entre honda y accord. Aunque las reescrituras generadas pueden obtener respuestas posibles, no todas tienen la misma relevancia. En consecuencia, antes de que QPIAD envíe las consultas, el algoritmo ordena las consultas usando la precisión y la exhaustividad estimadas. Sólo las primeras k consultas son enviadas a las fuentes. Donde k es el número máximo de consultas soportadas por una fuente de datos.

   Metadatos. El algoritmo de QPIAD se basa en un conjunto de metadatos que se generan en una fase previa de alimentación al sistema de mediación. Estos metadatos son una muestra de las fuentes de datos que permiten calcular la correlación entre atributos (por ejemplo modelo y fabricante), la distribución de los valores (por ejemplo Honda y Accord) y la selectividad de las consultas. Adicionalmente, QPIAD debe tener las vistas que cada

fuente tienen sobre el esquema global y el número máximo de consultas permitidos en una fuente.

Aplicación en las OV. QPIAD es una estrategia muy interesante para obtener respuestas completas en contextos de datos cuyas fuentes de datos tienen problemas de calidad. Esta estrategia incluye algoritmos innovadores para aprender la dependencia funcional entre los valores de los atributos que pueden ser útiles en muchos ámbitos. Sin embargo, el uso de QPIAD en contextos de datos que involucren muchas fuentes puede producir problemas de escalabilidad debido al incremento en el número de reescrituras. Aunque QPIAD mejora la calidad del resultado, es muy costoso pues el número de consultas enviadas a las fuentes puede llegar a ser muy alto. Aplicar este algoritmo a las OV pues permitiría obtener respuestas más completas incluso si la calidad de las fuentes no es tan alta. Sin embargo, para garantizar su escalabilidad en las OV, sería necesario aplicar primero una estrategia de selección que permita filtrar las fuentes y posteriormente aplicar la técnica de reescritura de QPIAD sobre las fuentes que tengan registros incompletos.

2. Integración Basada en la Calidad en Sistemas de Información Heterogéneos [NLF99, NFL04]

   Contexto. Este trabajo propone un modelo de integración de fuentes de datos estructuradas y heterogéneas en contextos que no requieren una respuesta completa pero si de alta calidad. En esta propuesta los planes son priorizados teniendo en cuenta la completitud de las fuentes que participan en el plan y el solapamiento entre las fuentes.

   Metadatos. Debido a que esta propuesta usa las fuentes con mayor calidad para evaluar una consulta, su planeación utiliza gran cantidad de metadatos para estimar esta calidad. Los metadatos requeridos son:

   - Proporción de instancias contenidas en la fuente de datos frente al total de instancias en el sistema.
   - Densidad de las instancias con respecto a una consulta que corresponde al promedio de la densidad de los atributos que están involucrados en la consulta.
   - Tipo de solapamiento entre todas las fuentes disponibles. Las fuentes pueden ser disyuntas, independientes (pueden tener instancias en común pero son independientes), tener un nivel de solapamiento cuantificado, pueden estar contenidas o pueden ser iguales.
   - Precio de evaluar una consulta, donde el precio está definido por tipo de consulta y no por consulta específica.
   - Tiempo de respuesta estimado, donde el tiempo está definido por tipo de consulta y no por consulta específica.
   - Frecuencia con que los datos de una fuente de datos son actualizados

Estrategia de Planeación. Dada una consulta el sistema la reescribe utilizando el enfoque LAV, cada reescritura es considerada un plan. Los planes pueden producir diferentes resultados pues involucran diferentes fuentes y por lo tanto más de una de ellas debe ser ejecutada para entregar la respuesta lo más completa posible. Para elegir los mejores planes se utiliza un modelo de calidad que evalúa cada plan en términos de completitud, disponibilidad y costo de las fuentes que participan en él. La completitud del plan se evalúa teniendo en cuenta la completitud de las fuentes que hacen parte de él. Esta última se calcula utilizando la estimación de cobertura y la densidad de la fuente. La cobertura es la proporción de instancias contenidas en la fuente del total de instancias en todo el sistema. La densidad mide si los atributos tienen realmente valores o si son nulos. Una vez se tienen los valores de cobertura y densidad de todas las fuentes del plan, estos valores son combinados teniendo en cuenta el solapamiento existente entre las fuentes. En el caso en que las fuentes sean disyuntas, la cobertura será vacía pues no tienen instancias en común. Por el contrario si una fuente está contenida en otra, la cobertura será la cobertura de la fuente de datos más pequeña.

Aplicación en las OV. El modelo de calidad presentado en este sistema de mediación tiene en cuenta las situaciones de solapamiento y de incompletitud de las fuentes que se presentan en una OV. Su estrategia para medir la completitud de una fuente podría ser empleada en una OV si se tuvieran disponibles todos los metadatos requeridos. Sin embargo, estos metadatos no están disponibles en las OV o son difíciles de obtener. Antes de usar esta estrategia de planeación sería necesario obtener los metadatos de solapamiento, densidad y cobertura de las fuentes.

3. Caminos de Navegación [BKN+06]

   Contexto. Esta propuesta está diseñada para contextos en donde las fuentes de datos pueden ser estructuradas o no estructuradas y contienen información relacionada. Por ejemplo una fuente contiene la información genética de una enfermedad y otra contiene las publicaciones realizadas alrededor de esa enfermedad. Las instancias contenidas en una fuente puede tener uno o más enlaces con las instancias contenidas en otras fuentes. Estas relaciones son conocidas y plasmadas en un grafo. Las consultas son consultas de navegación que definen cuál es la clase origen deseada y posibles fuentes origen y las clases destino. Por ejemplo la clase seleccionada puede ser Enfermedad con la palabra clave Cáncer y la clase final puede ser Publicaciones. En estos contextos el problema de resolver una consulta se traduce en un problema de seleccionar el camino adecuado para ir de una fuente inicial a una fuente final obteniendo el mayor número de respuestas posibles y minimizando el costo.

   Metadatos. Esta estrategia parte del supuesto de que se conocen las relaciones entre todas las fuentes de datos, y que adicionalmente, estas relaciones están cuantificadas a través de un beneficio y de un costo. Esto

quiere decir que si dos fuentes están relacionadas se conoce el número de instancias que se obtienen al consultar la clase destino dado el valor de las instancias de la clase origen. Adicionalmente, esta estrategia utiliza el nivel de solapamiento entre dos caminos que indica el número de instancias que tienen en común dos caminos diferentes.

Estrategia de Planeación. Dada una consulta y el grafo que relaciona las fuentes de datos, la planeación se lleva a cabo de la siguiente manera. Se buscan todos los posibles caminos que hay desde las fuentes que tienen la clase inicial hasta las fuentes que tienen la clase destino. Cada camino tiene asociado un costo y un beneficio calculado a partir de la cardinalidad de objetos obtenidos en la fuente destino. Para elegir los mejores planes se proponen diferentes algoritmos. El que les permitió obtener mejores resultados fue un algoritmo llamado *MSCT-greedy*. Este algoritmo ordena los caminos de acuerdo a la proporción costo-beneficio. Posteriormente elige el camino con menor proporción, lo elimina de la lista y ajusta el costo-beneficio de los planes restantes teniendo en cuenta el solapamiento entre fuentes. La siguiente fase elige el siguiente mejor camino hasta que se exceda el presupuesto disponible para la ejecución de la consulta.

Aplicación en las OV. El algoritmo de planeación utilizado es útil en las OV en donde es posible conocer con exactitud los niveles de solapamiento entre fuentes. Así mismo, para usar este algoritmo, sería necesario conocer el costo y el beneficio de ejecutar una consulta siguiendo un camino de una fuente inicial a una fuente final. Si bien la propuesta supone la existencia de esta información, para poder ser aplicada en OV sería necesario primero tener mecanismos para obtenerla. Adicionalmente, para poder usar la estrategia de selección en OV de gran escala es muy importante también optimizar la enumeración de todos los posibles caminos de tal forma que el mejor camino pueda ser detectado rápidamente.

4.2.3. Orientadas a Oferta y Demanda

1. SQLB: Un Marco de Trabajo de Ubicación de Consultas para Consumidores y Proveedores Independientes [QRLV07]

   Contexto. Proponen un modelo de mediación para ambientes donde los usuarios son considerados consumidores que pueden salir del sistema si no están satisfechos y las fuentes de datos son consideradas proveedores que están interesados en satisfacer cierto tipo de requerimientos. En este contexto es importante tener en cuenta las intenciones de los usuarios y las intenciones de los proveedores de fuentes de datos para hacer el proceso de asignación de consulta. Así mismo, para asegurar el equilibrio de carga entre los proveedores, es necesario tener en cuenta la equidad con todos los proveedores en el momento de la asignación de la consulta. Una consulta en este sistema es definida como q = (c, d, n), donde c es el identificador del consumidor, d es la descripción de la tarea por hacer, n es el número

de proveedores a los que el consumidor quiere asignar la consulta. La variable d incluye además de la consulta el nivel de calidad requerido.

Metadatos. Cada fuente de datos debe describir el tipo de consultas que es capaz de evaluar y el tipo de consultas que prefiere evaluar. Las preferencias pueden estar asociadas a diferentes aspectos como al valor de una condición o al número de registros que genera una consulta.

Estrategia de Planeación. SQLB utiliza un procedimiento de intermediación para encontrar los proveedores que pueden resolver la tarea (la consulta). De acuerdo a los requerimientos del consumidor y a las preferencias de los proveedores, el algoritmo calcula una calificación para cada fuente. Posteriormente ordena a los proveedores con respecto a esta calificación en ese tipo de consulta y elige los n primeros proveedores, donde n es el número de proveedores definido en la consulta, o elige todos los proveedores en el caso en que n sea muy grande.

Aplicación en las OV. Esta estrategia es aplicable en las OV si las organizaciones participantes tienen preferencias en el tipo de consultas que evalúan, si las fuentes son capaces de evaluar todas las condiciones de una consulta y si son independientes. En ese caso cada proveedor definiría las vistas que es capaz de evaluar y las vistas que prefiere consultar. Por cada vista se debería definir cuál es la calidad del servicio provisto, por ejemplo en número de tuplas. El usuario por su parte definiría la calidad de servicio esperado en número de tuplas y el número máximo de fuentes que desea consultar. A pesar de su posible aplicación, el tipo de OV en el que está enfocado este trabajo no cumple con las características mencionadas.

4.2.4. Flexible (Multi-Utilidad)

En este grupo se encuentran las estrategias que seleccionan las fuentes de datos a partir de funciones de utilidad genéricos que pueden ser adaptados a diferentes necesidades.

1. iDrips y Streamer [DL02]

   Contexto. Esta propuesta presenta formalmente el problema de encontrar de manera eficiente los mejores planes para la ejecución de una consulta en un sistema de integración de fuentes estructuradas. El problema de ordenar los planes de consulta en los sistemas de integración se produce cuando las fuentes de datos son incompletas, y ningún plan de consulta garantiza producir todas las respuestas. Esta situación aumenta el número de planes de consulta, que tienden a tener una diferencia significativa en su utilidad (por ejemplo la cobertura). Por lo tanto, el problema es encontrar primero los planes con mejor utilidad a fin de iniciar la ejecución tan pronto como sea posible. Para obtener el mejor plan se identifican tres propiedades estructurales del problema que permite el diseño de soluciones eficientes.

a) Monotonía de utilidad, lo que significa que es posible encontrar un orden total de las utilidades de las fuentes de datos.

b) Similitud entre fuentes, lo que significa que las fuentes similares puede ser reemplazadas entre ellas en cualquier plan sin afectar la utilidad del plan.

c) Independencia de los planes y la reducción de la utilidad, dos planes son independientes si la utilidad de un plan no depende de la ejecución de otro plan. Por otra parte, la reducción de la utilidad significa que la utilidad de un plan nunca aumenta si se desplaza hacia abajo en el ordenamiento de planes.

Metadatos. Los metadatos requeridos varían de acuerdo a la función de utilidad utilizada. En el caso en que la función de utilidad sea la cobertura sería necesario conocer el número de instancias que entregaría una fuente de datos dada una consulta.

Estrategia de Planeación. iDrips funciona bajo el supuesto de similitud de fuentes. Actúa en dos fases:

a) La fase inicial, agrupa las fuentes similares y crea planes abstractos que son capaces de resolver todos los sub-objetivos de la consulta. Cada plan abstracto representa un conjunto de planes concretos que tienen asociada una utilidad. Esta medida es un intervalo que contiene la utilidad de los planes concretos del conjunto.

b) La segunda fase, obtiene planes de consulta en orden decreciente de la utilidad seleccionando sólo el plan más dominante en cada iteración. Un plan abstracto dominante es un plan que tiene por lo menos un plan concreto cuya utilidad no sea inferior a la utilidad de todos los planes concretos en otros planes abstractos. Cada iteración de iDrips selecciona el mejor plan, lo elimina de los planes abstractos, y reconstruye las relaciones de dominación que sean necesarios.

El algoritmo Streamer es una modificación de iDrips que no vuelva a calcular las relaciones de dominio en cada iteración, ya que asume que los planes son independientes.

Aplicación en las OV. Aunque estos algoritmos son muy útiles en los sistemas de integración de datos, al aplicarlos en una OV su eficiencia depende de la función de utilidad elegida. Sólo podrían ser aplicados si la función de utilidad es capaz de representar los niveles de intersección entre las fuentes y la cobertura de cada plan.

4.3. Estrategias Orientadas a Fuentes de Datos no Estructuradas

Hasta el momento se han explorado las estrategias de selección de fuentes que se utilizan en ambientes compuestos principalmente por fuentes de datos estruc-

turadas o semi-estructuradas. Esta sección analiza las estrategias propuestas cuando las fuentes de datos no son estructuradas como archivos de texto, imágenes, videos, etc. Todas la estrategias utilizan estructuras de información que caracterizan a las fuentes de datos. La caracterización se puede hacer en términos del contenido de cada fuente de datos o en términos de atributos de la fuente como calidad de los datos, el costo de acceso, la frescura, etc. La planeación limita el grupo de fuentes de datos que participan en el plan de consulta de acuerdo al conocimiento disponible de las fuentes de datos. La diferencia de las propuestas en estos contextos no está en cómo realizan la planeación sino en cómo obtienen la información para hacer la planeación. Esta sección clasifica las estrategias en dos grupos: las que presumen que las fuentes son cooperativas y las que no.

4.3.1. Orientados a Fuentes Cooperativas

Estas estrategias presumen que las fuentes de datos proveen resúmenes de su contenido en un escenario cooperativo. Esta aproximación es apropiada cuando todas las fuentes están reguladas o son controladas por una misma entidad. En estos casos la estandarización de cómo entregar los resúmenes es el aspecto complicado.

1. Summary-Based Network for Single-Hop Semantic Routing[EMH+06] Este proyecto se centra en la búsqueda de imágenes similares en un sistema P2P. Los datos de cada nodo son resumidos a través de un histograma representado en un vector. Cada componente del vector contiene una agrupación. El valor de cada componente en un nodo p_i define la cantidad de documentos de p_i que pertenecen a la agrupación. Los resúmenes son dados a conocer usando un algoritmo de propagación de rumores aleatorio. El documento de agrupación global necesario para soportar esta aproximación es realizado de forma distribuida usando una implementación de algoritmo k-medias [KMN+02]. Para hacer sus resúmenes, cada nodo recibe los centroides de las agrupaciones globales.El proceso de selección de fuentes consiste en ordenar los nodos de acuerdo al número de documentos que cada nodo contienen en el cluster más cercano a los requerimientos de la consulta. Cada nodo contiene representaciones compactas de los otros nodos. El desempeño de la estrategia se mide en términos de cuántos nodos tuvieron que ser contactados en promedio para encontrar las top n imágenes que coincidían con la consulta.

2. Selección de Servidores Usando la Frecuencia de Términos

   Esta técnica esta dirigida a la búsqueda de bases de datos de textos. Está basada en la colección de la frecuencia de términos de los documentos. Aunque este tipo de descripción de las fuentes no requiere esfuerzo manual para crearlos o actualizarlos, la exactitud de la descripción de los datos puede ser deficiente de acuerdo al método usado para obtener la frecuencia de términos.

El protocolo START[GCGMP97] (Stanford Protocol for Internet Retrieval and Search) es propuesto como el protocolo de recuperación y búsqueda de documentos que facilita la definición de las fuentes de documentos. Los resúmenes usando este protocolo contienen la lista de las palabras que aparecen en la fuente con algunos atributos como dónde se encuentran en el documento (por ejemplo, en la sección de título) y las estadísticas para cada palabra incluyendo la frecuencia de las palabras y la frecuencia en el documento.

Los algoritmos más representativos que implementan esta técnica son CORI [CLC95, XC98], gGIOSS y bGIOOS [GGM95, GGMT99] y CVV [YL96]. bGIOSS por ejemplo, representa los documentos como palabras con información de su posición. Las consultas son expresiones compuestas por palabras, y conectores como *y, o, no* y operaciones de proximidad. La respuesta a una consulta es el conjunto de todos los documentos que satisfacen la expresión booleana. El proceso para seleccionar un fuente de datos tiene en cuenta la proporción de documentos que contienen las palabras dadas con respecto al número de documentos en la base de de documentos.

Aplicación en las OV. Las estrategias de selección orientadas a fuentes no estructuradas cooperativas utilizan una noción que puede ser muy útil en las OV: la priorización entre fuentes de acuerdo a su contribución. Si bien, las fuentes de datos en las OV son principalmente estructuradas, su contenido puede ser descrito de forma resumida por tipos de registros indicando la proporción de registros relativa a lo contenido en otras fuentes.

4.3.2. Orientados a Fuentes No Cooperativas

En este enfoque el conocimiento de cada fuente de datos se obtiene directamente por el sistema de procesamiento de consultas. El conocimiento se puede obtener solicitando la evaluación de una consulta específica o durante una fase de entrenamiento donde el procesador de consultas crea un catálogo que puede ser útil para cualquier consulta.

1. QProber (Querying Probes) [IGS01, IG02]

   Este proyecto de bibliotecas digitales distribuidas utiliza consultas de prueba para clasificar de forma automática una base de datos en una taxonomía y luego aplica un algoritmo eficaz para la extracción del resumen del contenido.

   QProber utiliza una combinación de aprendizaje automático y técnicas de consultas en bases de datos. Las técnicas de aprendizaje de máquina crean inicialmente clasificadores de documentos. Posteriormente, extraen reglas de clasificación de los clasificadores de documentos, y transforman estas reglas en un conjunto de consultas que se pueden enviar a la interfaz de búsqueda de las bases de datos de texto disponibles. El algoritmo utiliza

el número de respuestas obtenidas para hacer la clasificación de las bases de datos de documentos.

La selección de fuentes de datos utiliza el árbol de clasificación para saber qué fuentes pueden entregar documentos relacionados con la consulta.

2. Query Based Sampling of Text Databases[CC01]

   En esta técnica, la descripción de las fuentes de datos se crea ejecutando consultas y examinando cuáles son los documentos que fueron entregados en la respuesta. La hipótesis es que una muestra lo suficientemente imparcial de los documentos puede ser construido a partir de la unión de muestras sesgadas obtenidas mediante un muestreo basado en consultas. El proceso de muestreo de consultas consiste en enviar consultas de un término y hacer un análisis de los documentos obtenidos a partir de ella, para extraer las palabras y las frecuencias de los top n documentos. La selección de las fuentes de datos se realizan utilizando las descripciones aprendidas.

3. Lightweight Probe [HT99]

   Este es un método de clasificación de servidores propuesto para las consultas que involucran un número considerable de términos. La estrategia consiste en enviar un número pequeño de términos de la consulta a todos los servidores disponibles, cada una de los cuales responde con un paquete pequeño que contiene la información de la frecuencia de los términos. Los datos de frecuencia se utilizan para ordenar los servidores de acuerdo a su utilidad en la consulta.

Aplicación en las OV. Las estrategias de selección orientadas a fuentes no estructuradas y no cooperativas utilizan mecanismos para extraer metadatos de las fuentes de datos que posteriormente serán usados durante el proceso de selección. Dada la autonomía de las fuentes en las OV, utilizar este tipo de mecanismos puede ser beneficioso para conocer más en detalle la información que está contenida en cada fuente de datos y de esta manera poder clasificar el tipo de contribución de cada fuente. En relación a esta clasificación, la propuesta presentada en [IGS01, IG02] permite ver la utilidad de organizar las fuentes de una OV en una taxonomía de tal forma que sea posible mejorar la búsqueda de fuentes relevantes.

4.4. Síntesis

Esta sección presentó las principales técnicas de selección existente en los sistemas de mediación actuales. La Figura 4.1 resume las estrategias analizadas.

Estas técnicas son de gran utilidad en los contextos para los cuales fueron creados. Adicionalmente, representan un gran avance en las diferentes fases de la planeación de consultas, como por ejemplo en la definición de los enfoques de reescritura LAV y GAV. Sin embargo, no pueden ser aplicadas directamente

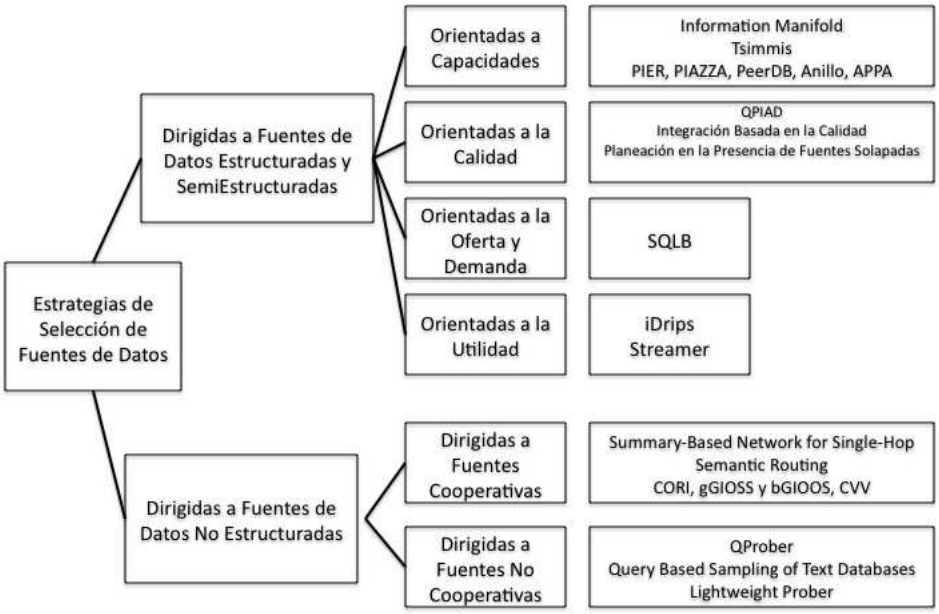

Figura 4.1: Clasificación de las Técnicas de Selección de Fuentes de Datos

en las OV por varias razones. En el caso de las estrategias orientadas a capacidades no se contempla la posible redundancia entre las fuentes, esto genera respuestas redundantes y planes costosos cuando son aplicadas a las OV. Así mismo, las estrategias orientadas a calidad y a utilidad hacen suposiciones de tener disponible un conjunto de metadatos que no están disponibles en las OV o que son muy difíciles de mantener actualizados. Por su parte las estrategias orientadas a oferta y demanda son útiles sólo en las OV en donde los proveedores de fuentes tienen preferencias bien definidas en el tipo de consultas que evalúan, y en donde las fuentes son independientes y son capaces de evaluar todas las condiciones de una consulta.

Finalmente, las estrategias orientadas a fuentes de datos no estructuradas no pueden ser aplicadas directamente en las OV pues el tipo de consultas a las que están dirigidas sólo definen un término o palabra clave. Sin embargo, son de gran relevancia para el procesamiento de consultas en OV, pues las propuestas de clasificar las fuentes de datos en taxonomías permiten describir de manera condensada el contenido de una fuente de datos asegurando mayor escalabilidad cuando el número de fuentes es alto, como es el caso en las OV. La aplicación de este concepto de jerarquización de fuentes será explorado en la propuesta de esta investigación.

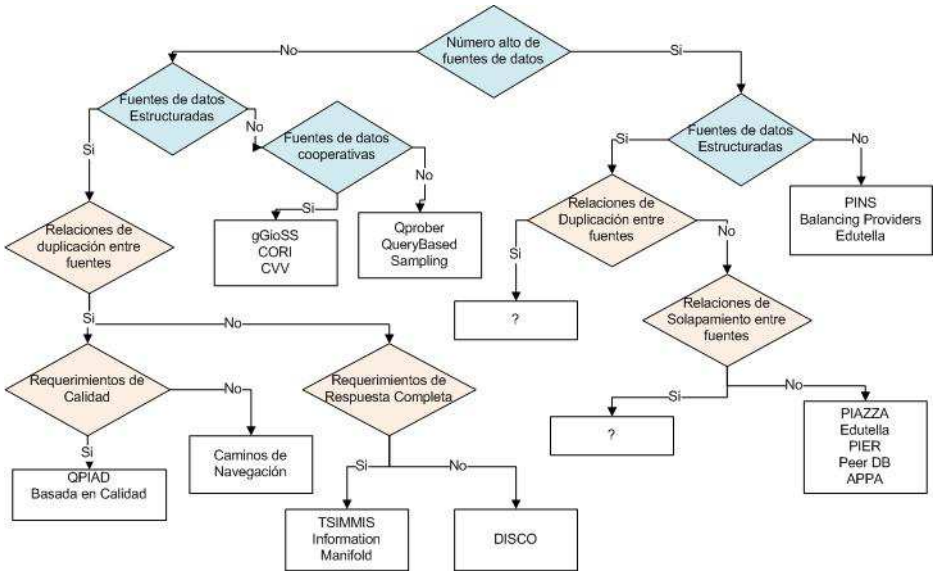

Figura 4.2: Ausencia de Estrategia de Selección para Contextos con Replicación y Solapamiento

En resumen, el análisis del estado del arte alrededor de las estrategias de selección de fuentes de datos permitió identificar que actualmente no existe ninguna estrategia de selección creada para contextos con un alto número de fuentes de datos estructuradas, en donde existen situaciones de solapamiento y duplicación entre fuentes, como los contextos de las OV. La Figura 4.2 ilustra este vacío que motivó la creación de una nueva estrategia de selección en este proyecto de investigación.

Parte II

Selección de Fuentes de Datos en Organizaciones Virtuales de Gran Escala

Capítulo 5

Una Arquitectura de Mediación para Organizaciones Virtuales: ARIBEC

5.1. Introducción

El análisis de las características y necesidades de las OV (Capítulo 2) y la evaluación de las principales propuestas de mediación y de selección de fuentes (Capítulo 3 y 4) motivaron la definición de una arquitectura diseñada para satisfacer los requerimientos de información de las OV. Esta arquitectura soluciona los problemas clásicos de integración causados por la heterogeneidad y distribución de las fuentes, al mismo tiempo que asegura la evaluación eficiente de diferentes tipos de consultas. Los principales retos de esta arquitectura de mediación, diseñada para OV, son los siguientes:

1. **Incertidumbre en la localización de los datos:** La ejecución de consultas distribuidas es guiada principalmente por el conocimiento que se tiene acerca de cada una de las fuentes de datos disponibles. Sin embargo, en las OV sólo se tiene un conocimiento parcial de estas fuentes, generalmente asociado al esquema y no al contenido. Tener información detallada del contenido es complejo en las OV, debido a que las fuentes son principalmente estructuradas y describir en detalle lo que contiene cada una de ellas puede ser muy costoso en tiempo y en espacio de almacenamiento. Adicionalmente, aún si fuese posible mantener catálogos de lo que contiene cada fuente puede que un gran porcentaje de esta información no sea utilizada durante la evaluación de consultas al no ser solicitada o incluida como restricción en la consulta. El reto de una arquitectura de mediación para OV es cómo llegar a un equilibrio entre el costo de almacenar meta-

datos de las fuentes de datos y la utilidad que estos metadatos representan durante la evaluación de consultas.

2. **Integración de respuestas a nivel de instancia:** A diferencia de otros contextos, la fragmentación de fuentes se da a nivel vertical y horizontal. Conformar una respuesta en estos contextos puede requerir reunir (*join*) las porciones de distintas fuentes para armar una instancia, y posteriormente unir todas las instancias para entregar la respuesta al usuario. Adicionalmente, considerando que la fragmentación de diferentes instancias puede variar, el conjunto de fuentes que deben participar para crear una instancia puede ser distinto. Por ejemplo, en la OV del sector salud la distribución de la información de una instancia de paciente A, puede ser completamente diferente a la distribución de un paciente B, debido a su ubicación, movilidad o enfermedad, por lo cual el proceso de integración no puede ser predefinido. El reto de una arquitectura de mediación para OV es encontrar para una consulta el mínimo conjunto de fuentes de datos necesarias para conformar las respuestas requeridas por el usuario sin necesidad de visitar todas las fuentes disponibles.

3. **Requerimientos de usuario variables:** Como resultado de los diferentes tipos de usuario los requerimientos de cada grupo de usuarios pueden ser diferentes. Un grupo puede requerir hacer consultas que abarcan un amplio grupo de respuestas (por ejemplo solicitar todos los pacientes del país con cáncer de seno), mientras que otros pueden hacer consultas con restricciones más selectivas frente al contexto general de datos (por ejemplo solicitar pacientes mujeres, de la tercera edad con cirrosis hepática que habiten en Bogotá). La definición de una estrategia estándar para optimizar la evaluación de consultas no es muy adecuada teniendo en cuenta que cierto tipo de consultas, por su naturaleza selectiva, deben ser dirigidas únicamente a las fuentes que realmente son relevantes para ellas. El reto es cómo el sistema de mediación puede utilizar diferentes estrategias dependiendo del tipo de consulta.

4. **Restricciones de confidencialidad:** En las OV pueden existir diferentes políticas de confidencialidad de los datos según la fuente de la cual provengan. Estas diferencias aumentan la complejidad de la evaluación de las consultas al tener que incluir una fase que verifique la validez de los planes de ejecución no sólo a nivel lógico y físico sino también a nivel de aseguramiento de la confidencialidad de los datos. A pesar de que éste es un gran reto para la mediación, su solución no está incluida en el alcance de esta investigación. Sin embargo, la arquitectura fue diseñada para que pueda ser extendida utilizando la propuesta de seguridad presentada en [CG08].

La arquitectura de mediación propuesta, llamada ARIBEC (ARquitectura de Integración Basada En Cartografía), tiene por objetivo proporcionar a los usuarios de una OV una vista comprensible de los datos que hacen parte de ella,

ocultándoles la complejidad de la mediación entre fuentes. El diseño de ARIBEC está basado en la arquitectura de referencia de mediación que fue enriquecida para responder a los retos impuestos por los contextos de datos de las OV. Este capítulo presenta la arquitectura de ARIBEC e ilustra detalladamente uno de sus componentes núcleo del nivel de mediación: la base de conocimiento de la OV.

La organización del capítulo es la siguiente: La Sección 5.2 presenta los principios que dirigieron el diseño de ARIBEC. Posteriormente, las secciones 5.3 y 5.4, ilustran la arquitectura de datos y la arquitectura funcional de ARIBEC, respectivamente. La Sección 5.5 presenta uno de los componentes núcleo de ARIBEC llamado *Base de Conocimiento*. Finalmente, la Sección 5.6 concluye el capítulo.

5.2. Principios de Diseño de ARIBEC

ARIBEC fue construido para mediar la consulta entre fuentes heterogéneas, distribuidas, en su mayoría estructuradas y que proveen una porción de los datos acerca de un dominio del conocimiento, asegurando escalabilidad y correctitud en la respuesta. A pesar de que los datos en las OV pueden ser estructurados, semiestructurados o no estructurados, ARIBEC está enfocada en solucionar los problemas de integración de fuentes con datos estructurados. Esta decisión fue tomada debido a que los contextos en los que se manejan los otros tipos de datos la respuesta por lo general es la unión de respuestas de cada fuente y no la integración de datos a nivel instancia. Por ejemplo, si se comparten imágenes, la respuesta a una consulta es el conjunto de imágenes obtenidas por cada fuente, en estos casos no se intenta integrar una imagen con otra. Por el contrario, si se comparten registros de pacientes, la respuesta a una consulta involucra la integración de los datos de un mismo paciente obtenidos en fuentes diferentes. En adelante, cuando nos refiramos a fuentes de datos, estaremos haciendo referencia a fuentes con datos estructurados.

Como se ilustró en el Capítulo 3, las propuestas actuales de mediación emplean estrategias, especialmente en planeación, que no escalan correctamente cuando hay fragmentación no disyunta entre fuentes y cuando el número de fuentes es alto, o utilizan metadatos que no están disponibles en las OV. La propuesta para el nivel de mediación es el principal aporte de esta investigación pues se propone una nueva manera de planear consultas y de coordinar su ejecución que no sólo es de utilidad en las OV sino en otros contextos distribuidos con características similares. A continuación se ilustran de forma general las decisiones de diseño tomadas en ARIBEC para enriquecer el nivel de mediación:

1. **Planeación basada en el conocimiento:** En ambientes de gran escala y con fuentes estructuradas es complejo tener información detallada de las fuentes de datos que permita diferenciarlas a partir de su contenido, aún cuando tengan el mismo esquema. Esta dificultad se debe al número alto de registros que puede llegar a tener cada fuente y a la diversidad de

valores en los atributos de cada registro que dificulta la descripción condensada del contenido en cada fuente. Para solucionar esta dificultad, la decisión tomada en ARIBEC fue describir el aporte que cada fuente ofrece para cierto tipo de consultas a través de un "resumen" que evita definir en detalle el contenido de cada fuente pero incrementa la capacidad de diferenciarlas durante la planeación de consultas. ARIBEC determina si este resumen será más o menos detallado de acuerdo al grado de similitud de las fuentes en su contenido y al tipo de consultas que se lancen en el sistema. Por ejemplo, si dadas dos fuentes que contienen pacientes, una de ellas es especialista en pacientes con cáncer y la otra en pacientes con insuficiencia renal, diferenciar su contenido en relación al diagnóstico de pacientes sólo requiere asociar la fuente al diagnóstico general y no a especializaciones, como Cáncer de próstata. En estos casos una descripción general del contenido será suficiente para determinar la utilidad de una fuente en una consulta que restrinja la propiedad diagnóstico. En otros casos, será necesario detallar más el contenido de cada fuente para poder detectar su utilidad. Por ejemplo, en una OV en el sector salud en donde las consultas utilizan como filtro los diagnósticos generales de pacientes (por ejemplo diagnóstico = Glaucoma), no es necesario tener metadatos asociados a tratamientos (por ejemplo provisión de medicamento = Timolol) , ni diagnósticos detallados (diagnostico = GlaucomaCongenital). Los tratamientos no serán utilizados en las consultas por lo que mantener este tipo de metadatos será inoficioso, y los diagnósticos detallados no serán necesarios si las consultas siempre se hacen en un nivel superior.

Para soportar esta nueva forma de describir los metadatos de las fuentes, ARIBEC apoya su estrategia de mediación en una ontología que permite describir de forma resumida el conocimiento extensional (contenido) de las fuentes. El usar una ontología y no una estructura plana de metadatos permite a ARIBEC describir las fuentes usando diferentes niveles de detalle y permite inferir metadatos así no se hayan declarado explícitamente. Por ejemplo, en una OV en el sector salud es posible definir que una fuente tiene información de pacientes con Cáncer y que otra tiene información de pacientes con Cáncer de Seno. El detalle en el que se describe la información depende de la utilidad que genere tener el conocimiento durante la planeación de consultas. Adicionalmente, si una consulta solicita pacientes con *Cáncer* y una fuente declaró conocer pacientes con *Cáncer de Seno*, a través de inferencia esta información puede obtenerse automáticamente.

2. **Agrupación de Conceptos de Interés:** Tradicionalmente, en los sistemas de mediación de fuentes estructuradas, el usuario final describe qué parte de la ontología de referencia desea consultar usando un lenguaje declarativo como *Datalog* (ver [PH01, KLSS95]). La definición de estas consultas requiere por parte del usuario final de un nivel de conocimiento técnico que dificulta el proceso de formulación de consultas. Para facilitar el proceso de consultas ARIBEC incorpora el concepto de VDO (*Virtual Data Object*). Un VDO representa un concepto de interés para un grupo

de usuarios dentro de la OV que puede involucrar uno o más subconceptos y sobre el cual los usuarios pueden realizar consultas sin tener conocimientos en lenguajes de definición de consultas. La definición 2 describe formalmente un VDO como un subconjunto de clases de una ontología de referencia que tiene sentido lógico para el usuario.

Definición 2 ***Objeto Virtual de Datos (VDO)***
Dada una ontología de dominio $O(C, P, A)$ *compuesta por un conjunto de clases* $C = c_1, ..., c_n$*, propiedades* $P = p_1, ..., p_m$ *y axiomas* $A = a_1, ...a_p$*, un objeto virtual de datos VDO se define como*
$VDO(C', P', A') \equiv O'(C', P', A')$ *tal que* $C' \subseteq C \wedge P' \subseteq P \wedge A' \subseteq A$.

La identificación de subconjuntos de clases que son de interés para grupos de usuario al interior de la OV no sólo facilita la formulación de consultas, sino que, como se verá más adelante, facilitan su optimización. Un VDO puede tener asociadas una o más instancias que le dan valor a cada uno de las clases y propiedades que hacen parte de él. Los VDOs son considerados virtuales porque sus instancias están particionadas y distribuidas en diferentes fuentes de datos de la OV.

3. **Integración de fuentes compatibles**: Teniendo en cuenta que las respuestas requeridas en una OV deben ser creadas a partir de la integración (*join*) de las respuestas de diferentes fuentes, una arquitectura de mediación para OV debe ser capaz de reconocer cuándo las instancias provistas por una fuente deben ser integradas con las instancias provistas por otra fuente. Sin la existencia de una lógica de integración, la conformación de respuestas podría ser muy costosa si por cada fuente que participa en la consulta se intentara integrar sus respuestas con las respuestas de otras fuentes que participan en la consulta. Es necesario incorporar al proceso de mediación una lógica de integración que señale cuáles fuentes deberían ser integradas, evitando integrar fuentes que no tienen instancias en común que satisfagan la consulta. La lógica de ARIBEC para guiar el proceso de integración es utilizar los metadatos disponibles durante la planeación de la consulta para identificar cuáles fuentes proveen instancias que satisfacen los filtros de la consulta y cuáles de estas fuentes son potencialmente compatibles (ver Definición 3).

   Dos fuentes son compatibles si su intersección no es vacía. Para identificar cuáles son estas fuentes de datos "compatibles" con respecto a las instancias, ARIBEC relaciona la información organizacional de la OV como alianzas, procesos, relaciones, etc. para detectar qué fuentes tienden a tener información del mismo grupo de instancias. La lógica de integración fue diseñada a partir de la hipótesis de que todas las fuentes manejan el mismo identificador de instancia.

   Definición 3 ***Fuentes Compatibles***
   Dos fuentes de datos DS_i *y* DS_j *son compatibles si dada la extensión de*

DS_i, $ext(DS_i)$ y la extensión de DS_j, $ext(DS_j)$, $ext(DS_i) \sqcap ext(DS_j) \neq \phi$, donde $ext(DS_i)$ y $ext(DS_j)$ son el conjunto de instancias contenidas en la fuente de datos DS_i y DS_j respectivamente.

4. **Evaluación intencional y extensional de la consulta:** La fragmentación de datos en las OV genera incompletitud en las fuentes de datos pues ninguna de ellas contiene todas las instancias de los conceptos del dominio y en muchos casos ninguna contiene todas las propiedades. Entregar una respuesta completa implica consultar todas las fuentes que pueden resolver la consulta e integrar las respuestas obtenidas a nivel de instancia para posteriormente entregarle al usuario el resultado consolidado. Sin embargo, consultar todas las fuentes puede ser muy costoso y poco escalable cuando el número de fuentes es muy grande. La tarea más crítica en estos contextos es identificar correctamente las fuentes de datos que deberían participar en la evaluación de la consulta y la porción de la consulta que debería evaluar cada una de ellas. Para hacer esta identificación el principio de ARIBEC es analizar las fuentes con respecto a la consulta desde el punto de vista intencional y cuando sea necesario desde el punto de vista extensional. La capacidad intencional (ver Definición 4) de una fuente se refiere a la capacidad que tiene de resolver una consulta desde el punto de vista de su esquema. Si el esquema de una fuente no contiene al menos uno de los atributos solicitados en una consulta su capacidad intencional es nula. Sean:
$VDO(C, P, A)$, un objeto virtual de datos .
$Q(VDO, propiedades(p_1, p_2, ..., p_n), condiciones(c_i, ..., c_m))$, una consulta sobre el VDO donde la tupla *propiedades* determina cuáles propiedades del VDO son solicitadas en la consulta ($p_i \in P$) y la tupla *condiciones* determina las condiciones sobre las instancias del VDO ($c_k = (p_i = val)$) requeridas.
DS_l una fuente de datos de la OV con un esquema relacional
$S = \{a_1, a_2, ..., a_m\}$.

Definición 4 ***Capacidad Intencional de una Fuente***
DS_l es capaz de resolver intencionalmente Q ssi $\exists(a_j, p_i) \mid a_j mapp_i$ donde map es una función de mapeo entre las relaciones a_j y las propiedades p_i.

La capacidad extensional (ver Definición 5) se refiere a la capacidad que tiene una fuente de resolver una consulta desde el punto de vista de su contenido. Si el predicado de una consulta exige que las respuestas cumplan una condición c_1 y una condición c_2, si una fuente de datos no contiene ninguna instancia cuyos valores coincidan con la condición c_1 y c_2 o con al menos una de ellas, su capacidad extensional es nula.

Sean:
$VDO(C, P, A)$, un objeto virtual de datos .
$Q(VDO, propiedades(p_1, p_2, ..., p_n), condiciones(c_i, ..., c_m))$, una consulta

sobre el VDO donde la tupla *propiedades* determina cuáles propiedades del VDO son solicitadas en la consulta ($p_i \in P$) y la tupla *condiciones* determina las condiciones sobre las instancias del VDO ($c_k = (p_i = val)$) requeridas.
DS_l una fuente de datos de la OV con un esquema relacional
$S = \{a_1, a_2, ..., a_m\}$.

Definición 5 ***Capacidad Extensional de una Fuente***
DS_l es capaz extensionalmente de resolver Q ssi $\exists c_k \mid \sigma_{c_k}(DS_l) \neq \phi$, donde $\sigma_{c_k}(DS_l)$ representa los registros (tuplas,instancias, etc.) que satisfacen la condición c_k y $c_k \in$ condiciones.

Para efectos prácticos la capacidad intencional y extensional serán evaluadas de acuerdo a los atributos utilizados en el predicado de la consulta (condiciones) y no a todos los atributos de salida. La evaluación intencional y extensional de ARIBEC genera una estructura denominada la *Cartografía de la Consulta*. La cartografía representa la distribución de las instancias que resuelven la consulta. Entre más precisa sea esta cartografía más eficiente será el procesamiento de la consulta. Dada una consulta
$Q(VDO(C, P, A), propiedades(p_1, p_2, ..., p_n), condiciones(c_i, ..., c_m))$, sobre un VDO que se encuentra distribuido en la OV, donde *propiedades* son las propiedades solicitadas del VDO y *condiciones* son las condiciones del predicado de Q, la cartografía de Q en la OV cuyo conjunto de fuentes de datos es $DS_{VO} = \{DS_1, ..., DS_l\}$ es presentada en la Definición 6.

Definición 6 ***Cartografía***
$Cartography(Q) = (DS_p, priority_{DS_p}, SubQ_i), ..., (DS_k, priority_{DS_k}, SubQ_j)$, donde DS_l es una fuente de datos que puede resolver la consulta Q ya sea a nivel intencional o extensional, $priority_{DS_l}$ es un valor entero que determina el orden en que debe consultarse DS_l para obtener un conjunto completo de respuestas a Q y $SubQ_i$ es la porción de la consulta Q que puede evaluar la fuente DS_l. .

5. **Selección de fuentes adaptable al tipo de consulta:** El uso de una única estrategia de planeación de consultas independientemente de los requerimientos de usuario y del contexto de la OV puede generar ineficiencias en el procesamiento de las consultas en las OV. El contexto de datos puede cambiar y con ellos la estrategia más adecuada. ARIBEC tiene en cuenta esta naturaleza cambiante y ofrece un mecanismo de adaptación que selecciona la estrategia según la visión actual de este contexto. Este mecanismo evita usar estrategias de selección de análisis exhaustivo, es decir estrategias que analizan en detalle los metadatos de las fuentes, para consultas que involucran un número reducido de fuentes. ARIBEC selecciona en tiempo de ejecución la estrategia más apropiada para planear la

consulta, teniendo en cuenta el contexto de la OV y la consulta que se va a evaluar. Una estrategia es considerada apropiada si el aumento en la precisión de la selección supera el esfuerzo dedicado durante la selección, con respecto a una estrategia que usa sólo la estructura de las fuentes para llevar a cabo la selección. La estrategia más apropiada es la que obtiene mayor precisión con menor esfuerzo.

Las siguientes secciones presentan la arquitectura propuesta bajo estos principios. Inicialmente, se ilustran los niveles de abstracción de datos que se manejan en ARIBEC. Posteriormente, se describen las funcionalidades haciendo énfasis en la manera como se lleva a cabo el proceso de evaluación de consultas. A pesar de que ARIBEC proporciona diferentes funcionalidades tales como el registro de usuarios, el registro de fuentes, el registro de organizaciones participantes, etc., el énfasis en la presentación de la arquitectura estará en describir los componentes relevantes en la funcionalidad de evaluación de consultas pues las propuestas originales de esta tesis se sitúan en este nivel.

5.3. Arquitectura de Datos de ARIBEC

Como se mencionó anteriormente el principal objetivo de ARIBEC es permitirle a los usuarios de la OV realizar consultas de forma eficiente y con transparencia de distribución, fragmentación, replicación y heterogeneidad de los datos. Esta sección describe la arquitectura de datos que se propone para proporcionar esta transparencia.

5.3.1. Niveles de Abstracción de los Datos

ARIBEC proporciona tres niveles de abstracción (tipo ANSI/SPARC [OV99]) ilustrados en la Figura 5.1.

En el nivel interno se encuentran las fuentes de datos registradas para compartir sus datos. Estas fuentes exponen su modelo de datos. El nivel externo es una capa semántica que representa la visión que los usuarios tienen del mundo. Este nivel está compuesto por un conjunto de objectos virtuales de datos (VDO ver Definición 2). El nivel conceptual representa al mundo a través de una definición semántica y general del dominio de la OV, utilizando una ontología de referencia. Este nivel no considera necesidades específicas de los usuarios ni restricciones de los modelos locales. La ontología de referencia debe ser definida por los miembros de la OV. Los enlaces entre los niveles representan los mapeos necesarios para llegar de una consulta en el nivel externo a los datos almacenados en el nivel interno y a la inversa. La Figura 5.2 ilustra un caso puntual de estos niveles de abstracción para una OV en el sector salud. En el nivel externo se presenta un VDO llamado *Paciente* que está asociado a un conjunto de conceptos definidos en la ontología de referencia como *Persona* y *Acto Médico*. A su vez, cada concepto de la ontología de referencia está ligado a la información contenida en una o más fuentes de datos.

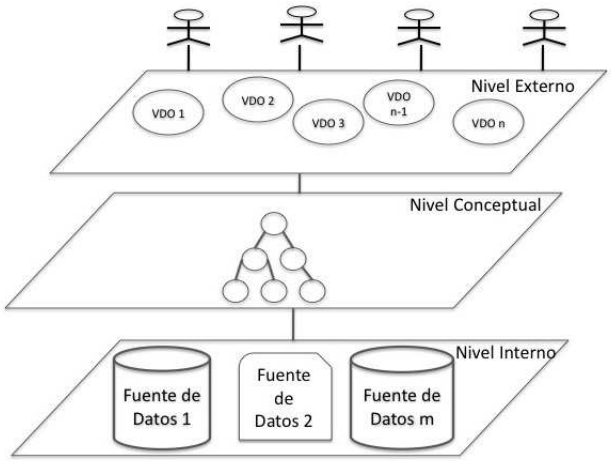

Figura 5.1: Niveles de Abstracción de los Datos en ARIBEC

La Figura 5.3 ilustra las estructuras de datos que apoyan las funciones de mediación para llegar desde el nivel externo de abstracción de datos hasta el nivel interno. Cada estructura de datos es administrada por uno de los niveles de la arquitectura de referencia de mediación (ver Sección 3.2).

5.3.2. Catálogo de VDOs

El objetivo de este catálogo es mantener el conjunto de VDOs que se han definido en la OV. Los usuarios de la OV pueden elegir y restringir un VDO de este catálogo para formular una consulta. Por ejemplo, en la OV en el sector salud un VDO puede ser *Paciente* (ver Figura 5.4). La intención de crear los VDOs es evitar que los usuarios no técnicos deban definir consultas usando lenguajes complejos y poco amigables. Cada VDO es visto por el usuario final como una plantilla sobre la cuál puede definir sus consultas.

Un **VDO** está compuesto por un nombre, una estructura, una definición y opcionalmente unas políticas de confidencialidad. A continuación se describe cada uno de ellos.

- El **nombre** permite diferenciar diversos VDOs. Cada nombre está orientado a los usuarios de la OV. Ejemplos de nombres de VDO son Paciente, Gen, Experimento, Publicación.

- Las **propiedades** son los elementos asociados a cada VDO. El usuario puede determinar cuáles de las propiedades desea obtener y usar para restringir el conjunto de instancias de VDOs obtenidas.

- La **definición** relaciona las propiedades del VDO con la ontología de referencia de la OV. Técnicamente esta definición es una consulta sobre la ontología de la OV. El lenguaje usado para hacer esta consulta varía de

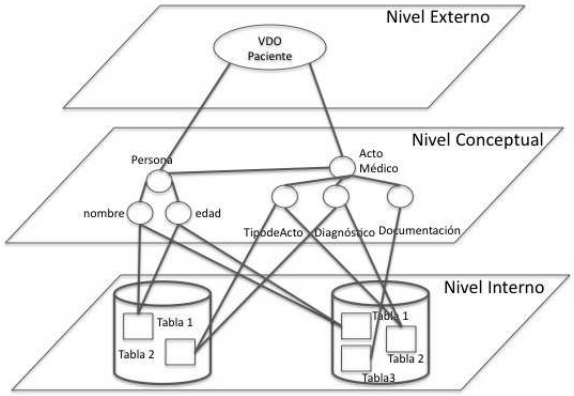

Figura 5.2: Ejemplo de Niveles de Abstracción de los Datos

acuerdo al lenguaje en el que está definida la ontología de la OV. Para tener mayor poder de expresividad se propone el uso del lenguaje OWL [Hor05] para definirla y del lenguaje SPARQL [EP07] para consultarla. Se eligieron estos lenguajes, pues a diferencia de otros lenguajes como XML, OWL tiene una semántica formal basada en la lógica descriptiva que permite inferir nuevo conocimiento a partir de la declaración de conocimiento explícito.

- La **definición de confidencialidad** establece los permisos para cada propiedad y el tipo de condiciones que puede definir cada tipo de usuario. Por ejemplo, es posible definir que cierto tipo de usuarios no puedan solicitar las propiedades que permitan identificar a un paciente, como el nombre, la identificación, etc.

La Figura 5.4 ilustra un ejemplo del VDO Paciente y la Figura 5.5 una consulta. Realizar una consulta en esta interfaz es muy sencillo pues sólo le exige al usuario seleccionar el VDO en el cual está interesado y limitar una o más de sus propiedades. En la Figura 5.5 el usuario estableció condiciones sobre la propiedad *género* y sobre la propiedad *diagnóstico*. Adicionalmente, el usuario eliminó las propiedades de *Afiliación*.

La respuesta a una consulta sobre un VDO es el grupo de instancias del VDO que satisfacen las condiciones definidas en la consulta. La creación de una instancia de VDO puede requerir consultar múltiples fuentes pues cada propiedad puede ser proveída por diferentes fuentes. Adicionalmente, dada la fragmentación vertical y horizontal de las instancias la materialización de una instancia puede requerir fuentes de datos completamente diferentes a la materialización de otra instancia. Por ejemplo, al consultar en una OV en el sector salud los procedimientos llevados a cabo sobre pacientes con *diabetes* a los que se les practica *hemodiálisis* es necesario primero identificar cuáles son estos pacientes y luego obtener todos sus procedimientos. Para un paciente A estos

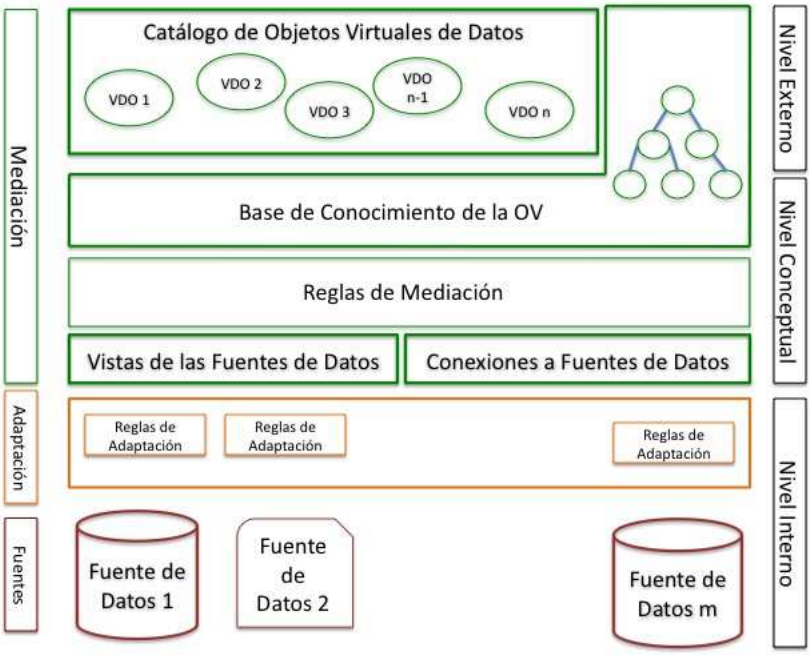

Figura 5.3: Arquitectura de Datos de ARIBEC

procedimientos pueden provenir de la fuentes de datos 1 y para un Paciente B pueden provenir de la fuente de datos 2.

Internamente una consulta sobre un VDO corresponde a una consulta típica en otros sistemas de mediación. La Figura 5.6 presenta un ejemplo de la representación interna de una consulta de un VDO en el lenguaje SPARQL[EP07], utilizado en ARIBEC. A pesar de que la mayoría de usuarios sólo pueden ver el catálogo de VDOs para realizar sus consultas, usuarios más avanzados pueden definir directamente consultas sobre la ontología de referencia usando el lenguaje de consultas SPARQL. Sin embargo, todos los esfuerzos de optimización de consultas están enfocados a la solución de consultas sobre los VDOs pues son los objetos de mayor relevancia para la OV. Si el usuario requiere hacer consultas que involucren más de un VDO será necesario que defina la consulta directamente usando la representación interna, es decir el lenguaje usado para consultar la ontología. En estos casos los usuarios deben tener conocimientos técnicos en el área de ontologías.

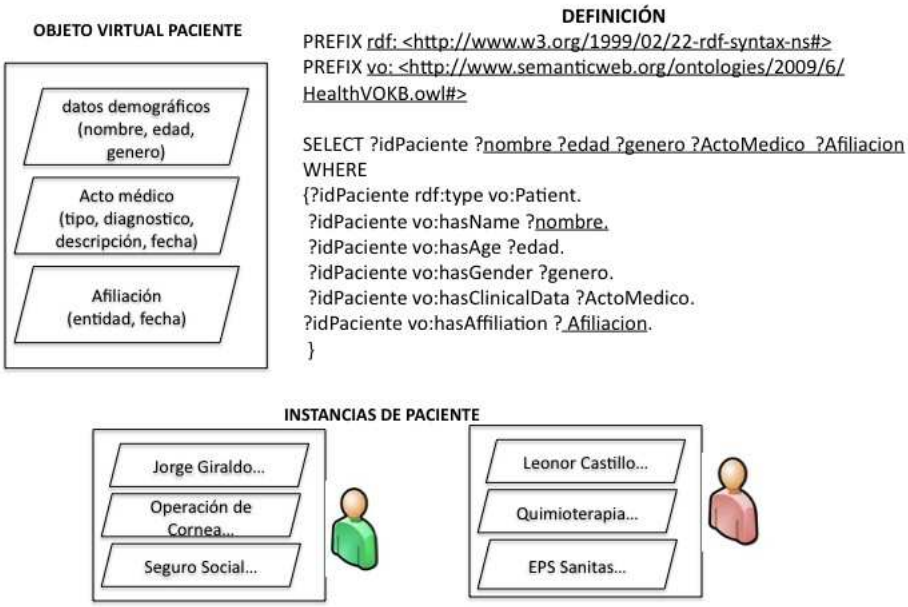

Figura 5.4: Ejemplo de VDO

5.3.3. Base de conocimiento de la OV

La base de conocimiento de la OV tiene dos intenciones. La primera de ellas es representar la ontología de referencia del dominio de la OV (por ejemplo persona, acto médico, diagnóstico, etc. en una OV en el sector salud). La segunda, es relacionar esta descripción del dominio con el contexto de la OV. Debido a la importancia de esta base de conocimiento en el procesamiento de consultas, ésta será detallada en la Sección 5.5.

5.3.4. Otros Elementos

En el Nivel de Mediación

Reglas de Mediación: Las reglas de mediación almacenan toda la lógica, heurísticas y procedimientos llevados a cabos durante el proceso de mediación. La lógica de mediación determina la manera como se planean las consulta y cómo se coordina la ejecución de las consultas haciendo uso de múltiples adaptadores. La Sección 5.4 detallará esta lógica.
Vistas de las Fuentes: Las **vistas de las fuentes** representan la visión que tiene cada una de las fuentes con respecto a la ontología de referencia del do-

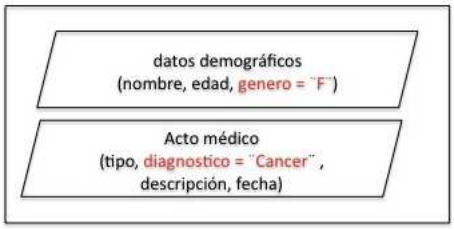

Figura 5.5: Consulta usando interfaz VDO

CONSULTA

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX vo: <http://www.semanticweb.org/ontologies/2009/6/
HealthVOKB.owl#>

SELECT ?idPaciente ?nombre ?edad ?genero ?ActoMedico ?Afiliacion
WHERE
{?idPaciente rdf:type vo:Patient.
 ?idPaciente vo:hasName ?nombre.
 ?idPaciente vo:hasAge ?edad.
 ?idPaciente vo:hasGender ?genero.
 ?idPaciente vo:hasClinicalData ?ActoMedico.
 ?ActoMedico vo:hasDiagnosis ?Diagnosis.
?idPaciente vo:hasAffiliation ?Afiliacion.
?Diagnosis rdf:type vo:Cancer.
FILTER regex(?genero, "F").
}
```

Figura 5.6: Consulta usando directamente el lenguaje de consulta

minio definida en la base de conocimiento. Estas vistas corresponden al modelo externo que las fuentes exponen y que es usado para realizar la reescritura de las consultas (ver Sección 3.2). La definición de estas vistas se basa en la propuesta presentada en [PH01] que utiliza el método de reescritura LAV.

Conexiones a las Fuentes de Datos: Las **conexiones a las fuentes de datos** almacenan la ruta de acceso a los adaptadores de las fuentes de datos que hacen parte de la OV. Esta información incluye los datos de autenticación como usuario y contraseña.

En el Nivel de Adaptación y Fuentes

Reglas de Adaptación: Esta estructura almacena los procedimientos necesarios para traducir conceptos locales de las fuentes a conceptos globales de la OV y viceversa. Esto incluye la resolución de los problemas de heterogeneidad sintáctica, semántica y esquemática. Las reglas de adaptación varían de acuerdo a cada fuente de datos.

Fuentes de Datos: Las fuentes de datos contienen físicamente la información de la OV. Aunque las fuentes pueden contener gran cantidad de información, las vistas descritas en el catálogo de vistas determinan lo que será visible de las fuentes en la OV.

5.4. Arquitectura Funcional de ARIBEC

Esta sección ilustra los principales componentes de ARIBEC que permitieron materializar las decisiones de diseño descritas en la Sección 5.2.

La Figura 5.7 ilustra la arquitectura global de ARIBEC. Cada nivel es a su vez un componente que puede ser reemplazado por otro que cumpla con el

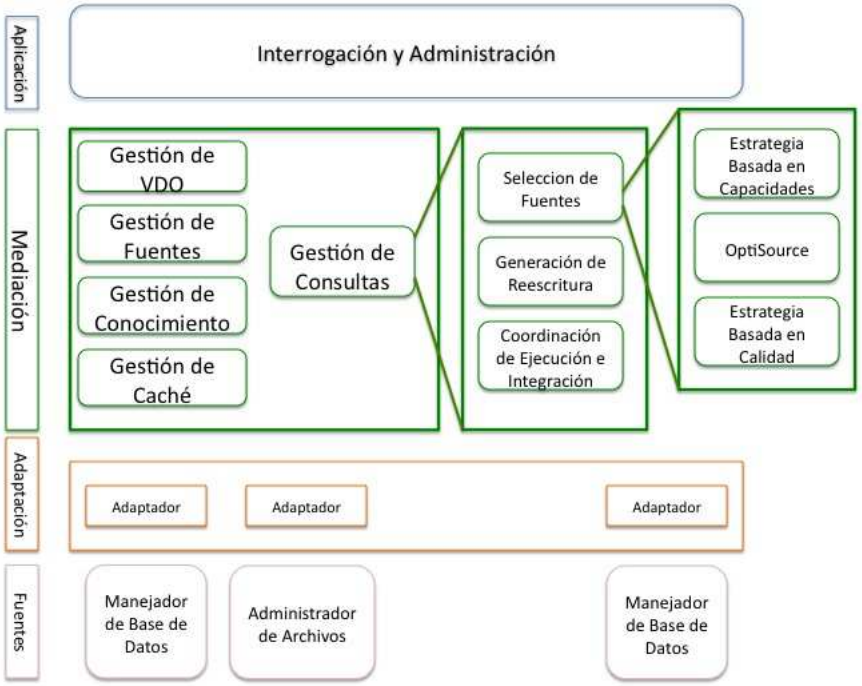

Figura 5.7: Arquitectura Funcional de ARIBEC

contrato de servicios establecidos para cada uno.

Los niveles fuentes y adaptación implantan los componentes clásicos para manejar fuentes heterogéneas. El nivel de mediación planea y coordina la evaluación de consultas. El diseño interno de este nivel es parte de la propuesta de esta investigación y será descrito a continuación.

5.4.1. Componentes del Nivel de Mediación

Las características de incertidumbre en la localización de información, variabilidad en los requerimientos de usuario y la necesidad de integrar respuestas a nivel de instancia hicieron necesario diseñar un conjunto de componentes específicos para la mediación en OV que no se encuentran en otros sistemas de mediación. Estos nuevos componentes fueron estructurados en cinco grandes componentes cuya funcionalidad se describe a continuación.

- Componente de Gestión de Conocimiento: A diferencia de otros sistemas de mediación que manejan catálogos de metadatos o taxonomías para describir la información de las fuentes, en ARIBEC se tomó la decisión de crear un componente de gestión de conocimiento implementado a través de una ontología que diera mayor riqueza y flexibilidad para expresar el

conocimiento. Por un lado la base de conocimiento permite describir el conocimiento en diferentes niveles de detalle, pero adicionalmente permite inferir nuevo conocimiento. Además, la base de conocimiento no sólo fue pensada para describir información de las fuentes sino para actuar como base de conocimiento para la OV. En ella se mantiene información de los recursos de la OV y de los proveedores de estos recursos, permitiendo enriquecer aún más los tipos de inferencias que se pueden hacer. Por ejemplo, si una fuente es proporcionada por una Clínica especialista en enfermedades del ojo, incluso si no se tiene información detallada de las fuentes es posible conocer de antemano el tipo de instancia que proporcionan estas fuentes. El componente de gestión de conocimiento es el responsable de administrar la base de conocimiento de la OV. Esta administración incluye su almacenamiento, la gestión de consultas a la base, la alimentación con nuevos hechos de conocimiento y la consistencia de la base. La Sección 5.5 ilustra las decisiones tomadas en la definición de este componente.

- Componente de Gestión de VDO: Como se mencionó en la Sección 5.3, para facilitar la formulación de consultas, ARIBEC ofrece a los usuarios un conjunto de VDOs que pueden ser usados como base para hacer consultas. El objetivo de este componente es administrar la creación, modificación o eliminación de objetos virtuales de datos, por parte de los usuarios habilitados para hacerlo. Este componente almacena por cada VDO establecido en la OV su identificación y su definición.

- Componente de Gestión de Caché: Para aumentar la eficiencia en la ejecución de consultas, en especial en OV de gran escala, ARIBEC propone el uso de una memoria caché de tipo semántico que permite agilizar la evaluación de consultas. La caché semántica almacena temporalmente información sobre consultas ejecutadas en el pasado que permite agilizar la ejecución de las nuevas consultas sobre VDOs. La arquitectura de esta caché extiende la caché dual presentada en [dJD+07, dJLR07] y está compuesta por tres estructuras de almacenamiento:

  - Caché de cartografía: Almacena cartografías de consultas.
  - Caché de consultas: Almacena las condiciones definidas en una consulta de VDO y los identificadores de las instancias que se obtuvieron en consultas pasadas asociadas a estas condiciones.
  - Caché de instancias: Almacena los valores de las propiedades de las instancias de VDOs obtenidos en consultas pasadas.

  La arquitectura propuesta para esta caché así como sus decisiones de actualización y reemplazo pueden consultarse en [PAR08].

- Componente de Gestión de Fuentes: Este componente se encarga de administrar la información de conexión de las fuentes (adaptadores) registradas y las vistas que éstas proveen con respecto a la ontología de referencia del dominio de la OV. Esta información será útil durante la evaluación de la consulta.

- Componente de Gestión de Consultas: Es el componente principal del nivel de mediación pues es responsable de procesar las consultas de usuario. La complejidad del componente hizo necesario dividirlo en tres componentes responsables de las diferentes fases de la evaluación de una consulta: el componente de selección de fuentes, el componente de reescritura de consultas y el componente de coordinación de ejecución y de integración.

  - El **componente de selección de fuentes** es responsable de seleccionar las fuentes de datos que participarán durante la consulta, produciendo su cartografía. Como se mencionó anteriormente, la cartografía identifica las fuentes de datos que pueden participar en la evaluación de una consulta. Este componente recibe como entrada una consulta y retorna su cartografía. El nivel de precisión y el costo de obtener esta cartografía varían de acuerdo a la estrategia de selección utilizada. Como se ilustró en el Capítulo 4, algunas estrategias de selección disponibles utilizan sólo el conocimiento intencional de las fuentes (esquema) para seleccionarlas, otros usan la calidad, etc. En las OV este proceso es crítico pues los fenómenos de fragmentación, el alto número de fuentes, la incertidumbre en la localización de datos y la variedad de requerimientos de usuario hacen que seleccionar las fuentes que realmente deben participar en la evaluación sea complicado. Para las consultas que involucran propiedades que no están fragmentadas en muchas fuentes, seleccionar las fuentes indicadas puede ser logrado evaluando únicamente el esquema de las fuentes, en otros casos se debe realizar un verdadero esfuerzo para reducir el número de fuentes a consultar. Para garantizar un equilibrio entre el costo (en tiempo y esfuerzo) y la precisión del proceso de selección, el componente de selección de ARIBEC utiliza una aproximación, llamada multiescala, que mantiene un conjunto de estrategias de selección que elige dinámicamente de acuerdo al tipo de consulta y al contexto de datos. El objetivo de esta aproximación es mantener estrategias "livianas" que sólo utilizan la parte intencional de las fuentes para reducir las fuentes a consultar y estrategias exhaustivas que reducen las fuentes a partir de su contenido, y elegir una u otra de acuerdo a la necesidad. Las estrategias orientadas a la parte intencional y a calidad ya existen; sin embargo, para soportar la selección en contextos complejos, de gran tamaño y con fenómenos de fragmentación horizontal y vertical no disyunta se requieren de estrategias mucho más exhaustivas a nivel extensional que las ya existentes. La estrategia propuesta para este tipo de contextos es llamada OptiSource y es parte fundamental de la propuesta de esta investigación. El Capítulo 6 presenta OptiSource y el Capítulo 7 presenta cómo se elige dinámicamente la estrategia de selección de fuentes.

  - El **componente de reescritura** reescribe la consulta usando las vistas definidas de las fuentes que pertenecen a la cartografía, este componente utiliza el enfoque *Local-As-View*.

  - El **componente de coordinación de ejecución** optimiza la ejecución del proceso de integración teniendo en cuenta la distribución de las fuentes

y la selectividad de las condiciones de la consulta. El modelo de coordinación propuesto utiliza los principios de espacios de tuplas de LINDA [CG89].

5.4.2. Proceso de Evaluación de Consultas

La Figura 5.8 ilustra, a través de un diagrama de secuencia, la interacción entre los componentes descritos anteriormente para llevar a cabo la evaluación de consultas en ARIBEC.

(1-3) El componente de gestión de consultas lanza el procesamiento de las consultas correctas sintácticamente.
(4-5) Se solicita al componente de caché la información que pueda ser relevante para la evaluación de la consulta.
(6-8) Para las porciones de la consulta de las cuales no se obtuvo información en la caché se procede a la planeación solicitando al componente de selección de fuentes crear la cartografía de la consulta. Para crearla, este componente determina en tiempo de ejecución cuál es la estrategia de selección apropiada para obtener la cartografía. El Capítulo 7 ilustra la manera como se toma esta decisión. En resumen, se contrasta la consulta con el contexto de datos para evaluar si es necesario una estrategia exhaustiva o una que sólo tenga en cuenta el conocimiento intencional. La precisión de la cartografía y el costo de planeación varían de acuerdo a la estrategia de selección de fuentes utilizado.

Adicionalmente, dependiendo del tipo de estrategia de selección utilizada durante la creación de la cartografía, se determinan los grupos de fuentes cuya integración (join) dará resultados no vacíos. Esta tarea evita enviar mensajes innecesarios durante la ejecución de la consulta.
(9-11) Posteriormente, las vistas de las fuentes de datos obtenidas en la cartografía son utilizadas para llevar a cabo la reescritura de la consulta. Teniendo en cuenta que el problema general de encontrar reescrituras a una consulta usando un número finito de vistas es NP-Completo en el tamaño de la consulta y en la presencia de un conjunto grande de vistas [LMSS95], cuando el algoritmo de selección utilizado identifica dentro de la cartografía grupos de fuentes "compatibles", ARIBEC realiza un proceso de reescritura independiente por cada grupo de fuentes. A partir de la selección de fuentes se generan múltiples procesos de reescritura, uno por cada grupo de fuentes, como se ilustra en la Figura 5.9. De esta manera se evita que el algoritmo de reescritura genere reescrituras, que aunque lógicamente correctas, no aportarán ningún resultado, pues las vistas que involucran no tienen instancias en común.
(12-15) Una vez obtenidas las reescrituras de la consulta, el plan es enviado para ser ejecutado. La coordinación de la ejecución de la consulta implica llevar a cabo un proceso de optimización física de las consultas que determine el mejor orden de ejecución de la consulta, y la mejor manera de integrar las instancias obtenidas de cada fuente. Por esta razón, ARIBEC lleva a cabo un proceso de optimización adaptable en donde se determina cuál es el siguiente grupo de

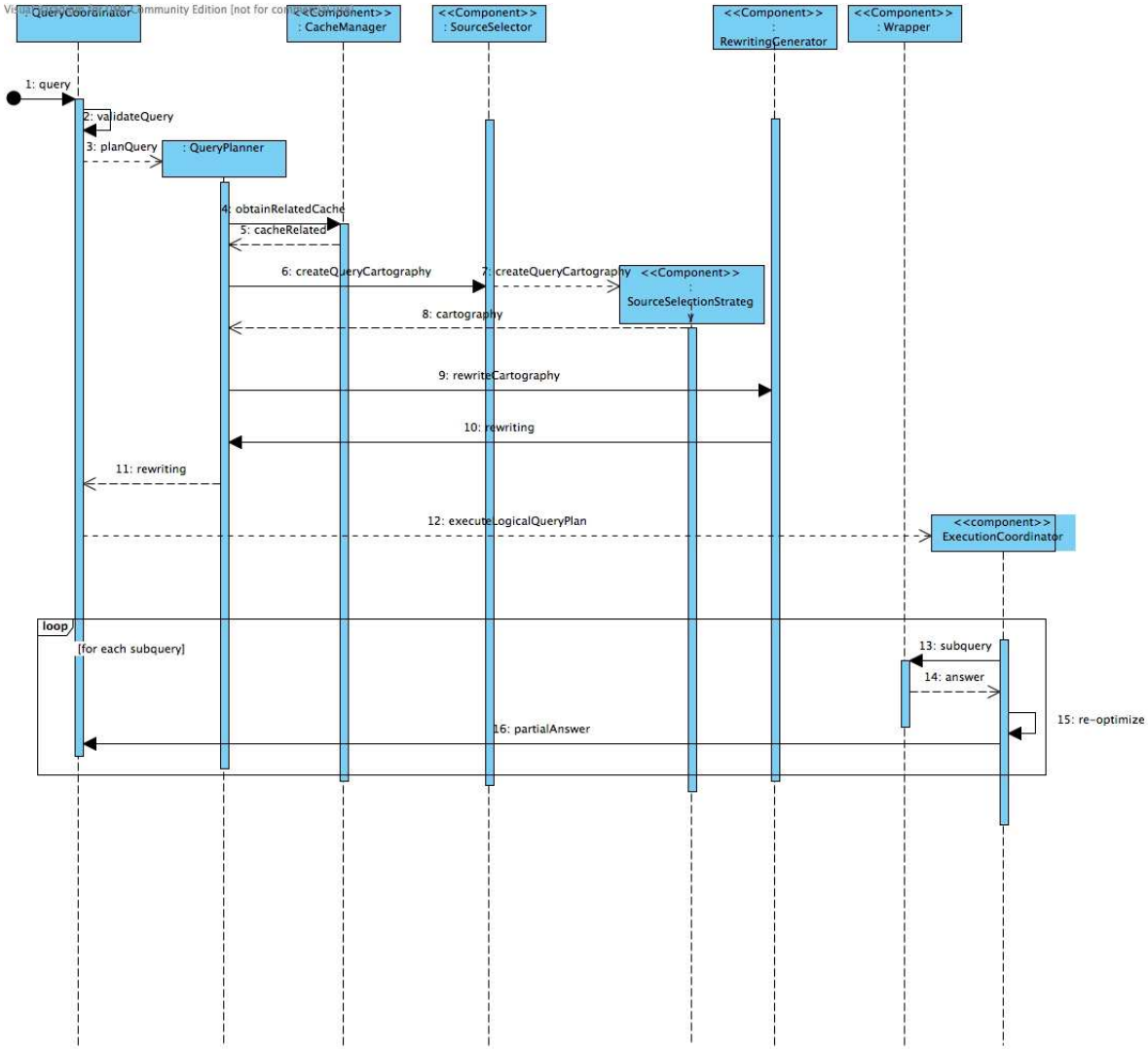

Figura 5.8: Diagrama de Secuencia de Evaluación de Consultas en ARIBEC

mejores fuentes a ser consultadas a partir de las respuestas obtenidas por parte de las fuentes previas. Este proceso adaptable será presentado en el Capítulo 6.

Figura 5.9: Proceso de Reescritura Independiente por Grupos de Fuentes

5.5. Base de Conocimiento de la Organización Virtual

La falta de conocimiento acerca del contexto actual de los datos de la OV afecta la eficiencia del proceso de evaluación de consultas. Los catálogos de metadatos que se usan en los sistemas de mediación actuales (ver Capítulo 4) no son suficientes para planear eficientemente consultas en las OV o suponen la disponibilidad de información que es difícilmente obtenible en las OV, como el número de instancias que satisfacen un predicado contenidas en una fuente.

La propuesta de ARIBEC es usar un catálogo de metadatos que permita refinar el conocimiento que se tiene del contenido de las fuentes a medida que evoluciona la OV, asegurando un equilibrio entre el volumen de metadatos almacenados y su utilidad al planear consultas. ARIBEC promueve el almacenamiento de metadatos que ayudan a diferenciar las fuentes que tienen esquemas similares, sin necesidad de almacenar información detallada del contenido (por ejemplo a nivel de instancia) que no sólo no está disponible en las OV sino que no es viable almacenar, dado el volumen de fuentes y el volumen de datos al interior de cada fuente. Este catálogo de metadatos está implementado a través de una base de conocimiento que conceptualiza el dominio de la OV en una ontología, y también mantiene metadatos del contexto de la OV en diferentes niveles de detalle. Esta sección describe cómo está estructurada esta base de conocimiento, qué metadatos almacena y cómo es utilizada en el nivel de mediación.

5.5.1. Estructura de la Base de Conocimiento de la Organización Virtual

La base de conocimiento permite mantener metadatos de las fuentes de datos y de toda la OV. Conocer quiénes hacen parte de la OV, qué proveen y có-

mo interactua es fundamental para tener control sobre la OV. Las fuentes son provistas por organizaciones participantes que desempeñan un rol dentro de la OV y el contenido que está en su interior está asociado a diferentes conceptos del dominio de la OV. Por ejemplo, una fuente de datos *BD1* provista por la organización Instituto Cancerológico, contiene información acerca del concepto Paciente y del concepto Acto Médico pues almacena los actos médicos practicados sobre pacientes con diagnóstico cáncer. Reflejar las relaciones entre las fuentes de datos y las organizaciones que las proveen, y entre las fuentes y los conceptos de dominio que están asociados es parte de lo que almacena la base de conocimiento.

La base de conocimiento describe la ontología de referencia de la OV y la relación entre los conceptos que hacen parte de esta ontología y los participantes de la OV, y entre estos conceptos y los recursos de la OV. La ontología de referencia almacenada es consultada para definir VDOs o consultas. Las relaciones son consultadas para mejorar el proceso de planeación de consultas.

La construcción de la base del conocimiento se hizo usando el lenguaje de descripción de ontologías OWL [Hor05] pues facilita la descripción detallada de los conceptos, sus jerarquías y sus relaciones. OWL permite conceptualizar un dominio a través de *Clases* que describen las características de un grupo de *Individuos* del mundo, *Propiedades* que describen las relaciones entre estos individuos (*ObjectType Properties*), o entre los individuos y los tipos de datos (*DataType Properties*) y *Axiomas* sobre las clases y sobre las propiedades que permiten refinar sus definiciones.

La estructura básica de la base de conocimiento está conformada por tres clases núcleo: *VOUnit, VODomainConcept,VOResource*, como lo ilustra la Figura 5.10.

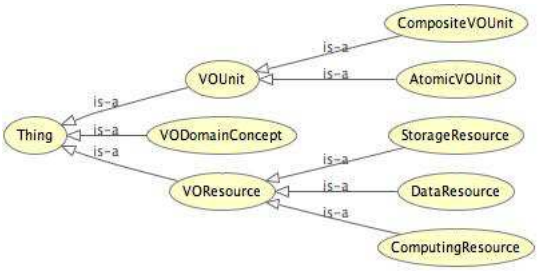

Figura 5.10: Clases Núcleo de la Base de Conocimiento de ARIBEC

La clase *VODomainConcept* es la superclase de las clases que describen los conceptos del dominio de la OV. Sus subclases definen el esquema global de la OV pues incluyen todos los conceptos que son de interés para la OV, y sus propiedades. Las subclases de *VODomainConcept* son diferentes para cada OV. La Figura 5.11 por ejemplo ilustra parte de las subclases de *VODomainConcept* para una OV en el sector de la salud.

La clase *VOResource* es la superclase de todos los recursos de la OV. De

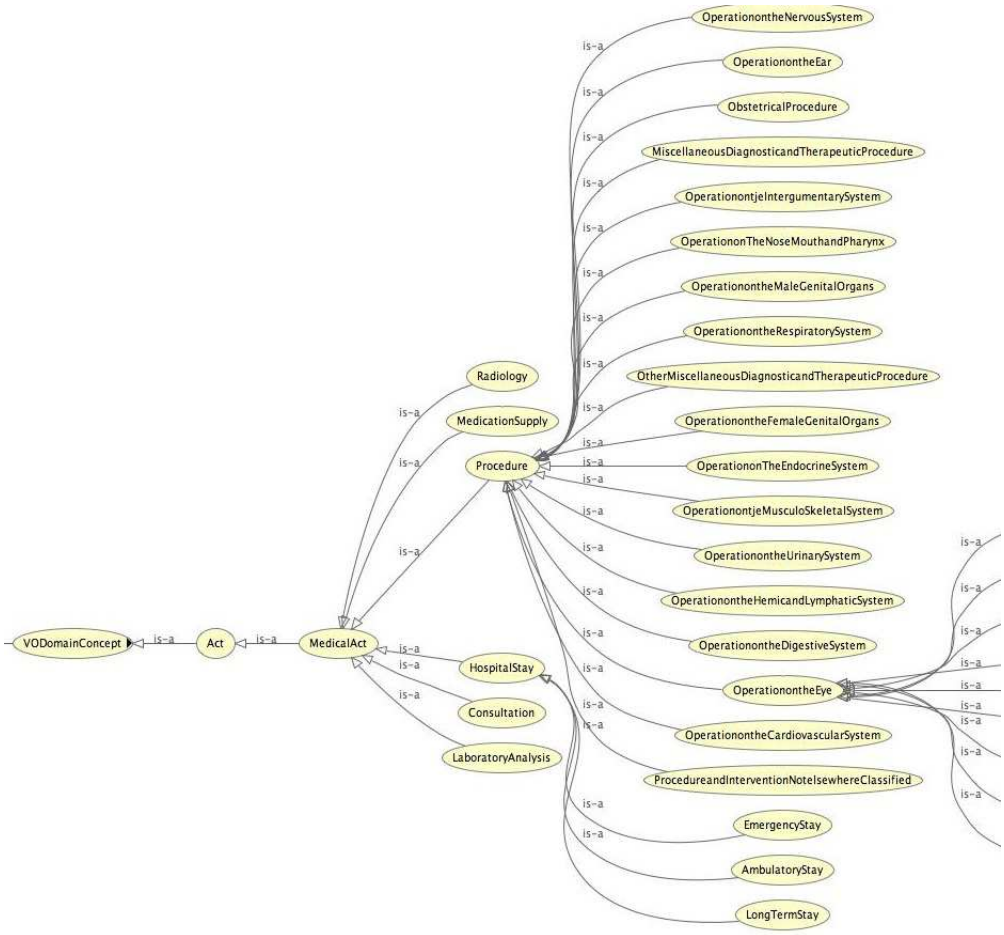

Figura 5.11: Porción de la taxonomía creada a partir de la clase *VODomainConcept*

acuerdo al tipo de OV el tipo de recursos puede variar. Algunas OV comparten recursos físicos, como recursos de cómputo o instrumentos, otras recursos lógicos, como bases de datos. Debido a que las OV de interés en este trabajo son las que comparten datos, los recursos más importantes en este tipo de OV son los recursos de datos, los recursos de almacenamiento y de procesamiento.

El objetivo de la clase *VOUnit* es representar a las organizaciones que hacen parte de la OV. Estas organizaciones, llamadas de ahora en adelante unidades de la OV, pueden ser atómicas o pueden involucrar asociaciones de organizaciones. La clase *AtomicVOUnit* describe las propiedades de las organizaciones atómicas y la clase *CompositeVOUnit* describe las propiedades de las asociaciones. El detalle de las propiedades de estas clases puede consultarse en el Apéndice C.

5.5.2. Representación de Metadatos en la Base de Conocimiento

La intención de la base de conocimiento es darle mayor flexibilidad y riqueza a la descripción de los metadatos que se utilizan en los sistemas de mediación. Esto se logra describiendo de forma resumida qué contiene cada fuente de datos conscientes de que el conocimiento que se tiene del contenido aumenta con el tiempo, gracias a las consultas previas o a los análisis que se realicen posteriores al registro de la fuente. Por ejemplo, inicialmente se puede conocer únicamente que una fuente contiene información de actos médicos sobre pacientes, pero luego puede conocerse que en realidad la información que se tiene es de procedimientos en los ojos. Para asegurar esta evolución del conocimiento en el tiempo, es necesario describir los metadatos en una estructura que permita aumentar de forma transparente el nivel de detalle que se tiene de las fuentes. La propuesta de ARIBEC es expresar estos metadatos como relaciones entre individuos de la clase *VOResource* con individuos de las clases *VDODomainConcept* y entre los individuos de la clase *VOUnit* con los individuos de las clases *VDODomainConcept*. De tal manera que a medida que se tiene más conocimiento del contenido de las fuentes éste puede ser expresado a través de la actualización de la relación con clases del dominio más especializadas.

Los individuos de las clases hijas de *VODomainConcept* se encuentran en las fuentes de datos, por ejemplo los individuos de la clase Paciente, Acto Médico, Enfermedad, etc. están contenidos en las bases de datos. La intención de ARIBEC no es almacenar estos individuos en la base de conocimiento[1], sino reflejar qué tipo de individuos contiene cada fuente de datos. Para hacerlo, se relaciona cada fuente de datos con los conceptos de dominio cuyas instancias están contenidas en ella. Por ejemplo, asociar una fuente de datos DB1 con la clase Paciente. Estas relaciones pueden hacerse en un nivel alto dentro de la taxonomía de las clases *VODomainConcept*, por ejemplo relacionar una fuente de datos con la clase *MedicalAct*, o pueden hacerse en niveles más detallados, por ejemplo relacionar la fuente de datos con la clase *OperationontheEye*.

En la práctica, para definir estas relaciones entre fuentes y conceptos del dominio, lo que se hace es relacionar los individuos de las clases *VOResource* y *VOUnit* con los individuos de las subclases de *VODomainConcept*. Como se mencionó anteriormente las subclases de *VODomainConcept* no tienen individuos asociados (pues sus individuos están en las fuentes de datos), por lo cual se propone la creación de un individuo artificial por cada subclase de *VODomainConcept* que permita hacer estas relaciones. Por el contrario, los individuos de las clases *VOResource* y *VOUnit* si son almacenados en la OV pues corresponden a los recursos de la OV y a los participantes de la OV, respectivamente.

La propiedad de la ontología utilizada para relacionar estos individuos es llamada *juegaRoldeConocimiento* y representa la competencia de las unidades de la OV con respecto a los conceptos del dominio o el almacenamiento de instancias de un VDO en los recursos de la OV respectivamente. Adicionalmente, para

[1] El objetivo es construir un sistema de integración y no una base de datos consolidada con todas las fuentes de la OV.

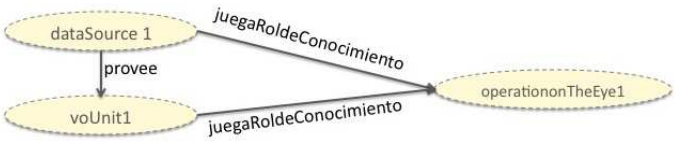

Figura 5.12: Relaciones entre individuos de la clase VODomainConcept e individuos de la clase VOUnit y VOResource

relacionar a los individuos de las clases *VOUnit* y *VOResource* se utiliza la propiedad *provee* que permite definir quién provee qué recurso en la OV. La Figura 5.12 ilustra una relación entre los individuos de estas clases.

5.5.3. Roles de Conocimiento de las Fuentes de Datos

A pesar de que la propiedad *juegaRoldeConocimiento* permite declarar el conocimiento de los individuos de *VOUnit* y de *VOResource* con respecto a los individuos de la clase *VODomainConcept*, esta propiedad no es efectiva para diferenciar dos unidades o dos recursos de datos cuando están asociadas a un mismo concepto. Considere por ejemplo una fuente de datos A y una fuente de datos B que están relacionadas usando la propiedad *juegaRoldeConocimiento* con el individuo artificial de la clase *Paciente* al que se le han practicado procedimientos del tipo *OperationontheEye*. Si la fuente de datos A tiene 5 registros de pacientes con estas características y la fuente de datos B tiene 100, pues ésta última proviene de una clínica especialista en este tipo de operaciones, la propiedad *juegaRoldeConocimiento* no permite describir este conocimiento. Una opción sería asociar cada relación con el número de instancias que contiene cada fuente. Sin embargo, este tipo de metadatos es difícil de mantener en el tiempo, pues cada día las fuentes registran nueva información. Más que vincular a la relación un valor numérico es necesario vincular a la relación los diferentes niveles de conocimiento que pueden existir entre las fuentes de datos y entre los participantes con respecto a un concepto de dominio. Para esto se propone especializar la propiedad *juegaRoldeConocimiento* para describir de forma más precisa la contribución de una fuente de datos con respecto a las instancias del VDO. Estas especializaciones serán usadas posteriormente para reflejar la habilidad de las fuentes para resolver predicados de consultas sobre VDOs. De acuerdo al análisis de los posibles roles que pueden jugar las unidades de las OV con respecto a los conceptos del dominio, ARIBEC propone tres especializaciones de la propiedad *juegaRoldeConocimiento* que serán llamadas de ahora en adelante roles: *autoridad, especialista* y *contenedora*.

La definición formal de cada rol se ilustra en las definiciones 7, 8 and 9. En estas definiciones todas las instancias de *VDOj* almacenadas en la fuente *DSi* se denota como $ext(DSi, VDOj)$[2]. *U* designa al conjunto de fuentes que hacen parte de la OV. Todas las instancias de *VDOj* disponibles en la OV se

[2]La extensión contiene los identificadores de instancia

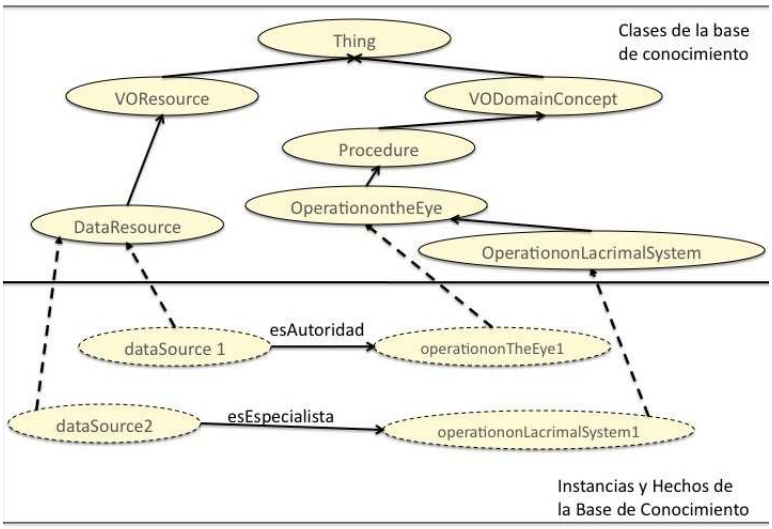

Figura 5.13: Clases, instancias y hechos de conocimiento en la base de conocimiento

denotan como $ext(U, VDOj)$. El subconjunto de $ext(DSk, VDOj)$ que satisface la condición p se denota $ext(DSk, VDOj)^p$ y $card()$ es la cardinalidad de una función.

Definición 7 ***Rol Autoridad.*** *Una fuente de datos DSi juega el rol de autoridad con respecto a una condición en una consulta VDOj ssi almacena todas las instancias de VDOj disponibles en U que satisfacen p.*

$EsAutoridad(DSi, VDOj, p) \Longrightarrow ext(U, VDOj)^p \subset ext(DSi, VDOj)$

Definición 8 ***Rol Especialista.*** *Una fuente de datos DSi juega el rol especialista con respecto a una condición p en una consulta sobre VDOj ssi la mayoría de instancias de VDOj almacenadas en DSi satisfacen p.*

$EsEspecialista(DSi, VDOj, p) \Longrightarrow card(ext(DSi, VDOj)^p) \geq card(ext(DSi, VDOj)^{\neg p})$

Definición 9 ***Rol Contenedor.*** *Una fuente de datos DSi juega el rol contenedor con respecto a la condición p de la consulta sobre VDOj ssi DSi contiene al menos una instancia de VDOj que satisface p.*

$EsContenedor(DSi, VDOj, p) \Longrightarrow ext(U, VDOj)^p \cap ext(DSi, VDOj)^p \neq \varnothing$

La Figura 5.13 ilustra la estructura de la base de conocimiento incluyendo las clases, individuos y hechos de conocimiento que describen las relaciones entre un conjunto de fuentes de datos y un conjunto de conceptos del dominio.

5.5.4. Inserción de Metadatos en la Base de Conocimiento

Cuando una fuente de datos es registrada en ARIBEC, la fuente de datos es creada como un nuevo individuo de la clase *DataResource*. Para reflejar los

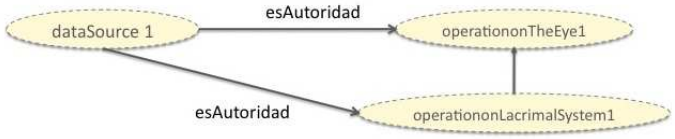

Figura 5.14: Inserción de Relación EsAutoridad

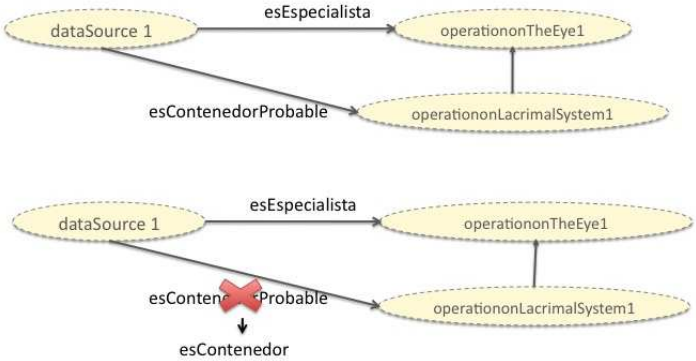

Figura 5.15: Inserción de Relación EsEspecialista

conceptos involucrados en las vistas declaradas por la fuente, se agrega una relación entre el individuo artificial del concepto definido y la fuente de datos usando la propiedad *juegaRoldeConocimiento*.

Si se detecta que la fuente de datos es autoridad con respecto a un concepto, para hacer más eficiente el proceso de búsqueda posterior, se adiciona una nueva relación usando la propiedad *esAutoridad* con la clase en cuestión y con todas las clases hijas de la clase en cuestión. La Figura 5.14 ilustra un ejemplo de esta inserción. En este ejemplo la relación de autoridad se detectó con respecto a la clase *OperationonTheEye*, por lo que sus subclases también fueron relacionadas con el individuo *dataSource1*.

Si se detecta que la fuente de datos es especialista con respecto a un concepto, se adiciona una nueva relación usando la propiedad *esEspecialista* con la clase en cuestión y se relacionan todas las clases hijas con la propiedad *esContenedorProbable*, que corresponde a una especialización de la clase esContenedor. Cuando se verifique que existe una relación de autoridad, especialista o de contenencia de la fuente con respecto a la clase con la que tiene una relación *esContenedorProbable*, se elimina esta relación y se reemplaza para la nueva relación. Así mismo, si se verifica que no es contenedora, se elimina la relación. La Figura 5.15 ilustra un ejemplo de esta relación.

Si se detecta que la fuente de datos es contenedora con respecto a un concepto, se adiciona una nueva relación usando la propiedad *esContenedor* con la clase en cuestión únicamente. La Figura 5.16 ilustra un ejemplo de esta relación.

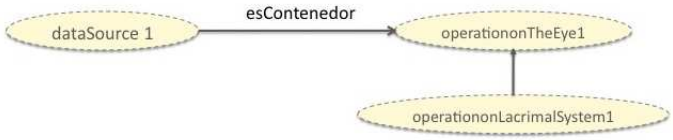

Figura 5.16: Inserción de Relación EsContenedor

5.5.5. Consultas sobre la Base de Conocimiento de la OV

Como se mencionó anteriormente la base de conocimiento es consultada para obtener información del contexto de datos que facilita la planeación de las consultas. Estas consultas buscan identificar qué fuentes pueden resolver las condiciones del predicado de una consulta. Por ejemplo, para planear la Consulta 1 expresada en SPARQL[EP07] el componente de gestión de consultas realiza la Consulta 2 y la Consulta 3 a la base de conocimiento. La Consulta 2 solicita todas las fuentes que tienen un rol asociado a la clase *Patient* y la Consulta 3 solicita las fuentes que tienen un rol asociado a la clase *OperationonTheEye*. Las $?X$ obtenidas en la Consulta 2 y en la Consulta 3 son las fuentes de datos que cumplen al menos un rol de conocimiento asociado a los conceptos involucrados en la consulta. Para obtener específicamente el rol que desempeñan basta con hacer una consulta en la base de conocimiento.

Consulta 1 *SELECT ?X*
WHERE ?X rdf:type vo:Patient.
?X vo:hasClinicalData ?procedure.
?procedure rdf:type vo:OperationonTheEye

Consulta 2 *SELECT DISTINCT ?X*
WHERE
?X rdf:type vo:DataResource.
?X vo:juegaRoldeConocimiento ?y.
?y rdf:type vo:Patient

Consulta 3 *SELECT DISTINCT ?X*
WHERE
?X rdf:type vo:DataResource.
?X vo:juegaRoldeConocimiento ?y.
?y rdf:type vo:OperationonTheEye

Una de las ventajas de usar la base de conocimiento implementada a través de una ontología es que el motor de inferencia de la ontología puede obtener metadatos así no se hayan declarado explícitamente. Por ejemplo, si no existe

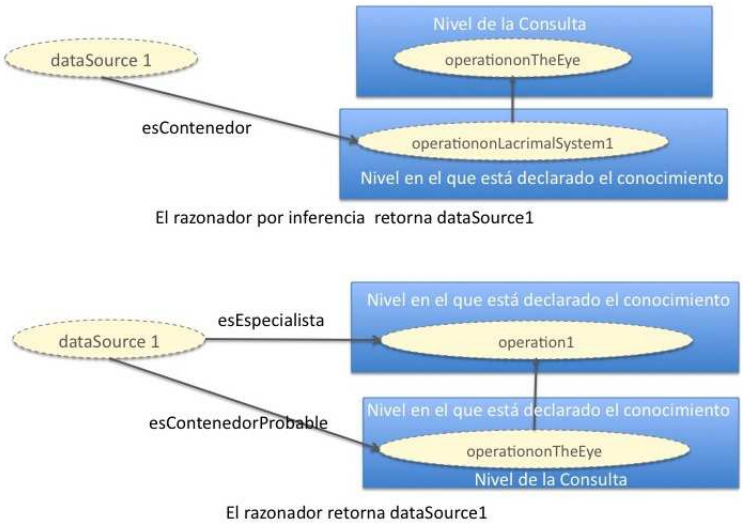

Figura 5.17: Respuesta de la Base de Conocimiento a las Consultas

ninguna fuente de datos asociada a la clase *OperationonTheEye*, pero existe una fuente asociada a la clase *OperationonCornea* (subclase de OperationonTheEye) el motor de inferencia entregará el identificador de esta fuente de datos. Adicionalmente, gracias a la manera como se insertan los hechos de conocimiento, si alguna superclase de *OperationonTheEye* está asociada a una fuente de datos usando las relaciones *esAutoridad* o *esEspecialista*, la clase *OperationonTheEye* también estará asociada a la fuente. La Figura 5.17 ilustra un ejemplo de cómo actuaría el motor de inferencia de acuerdo al nivel de la taxonomía en el que se realice la consulta y de acuerdo al nivel de la taxonomía en el que está declarado el conocimiento.

5.5.6. Creación y Actualización de la Base de Conocimiento

El punto más crítico en la creación de la base de conocimiento de la OV es la definición de todos los conceptos del dominio de la OV y sus relaciones, es decir las subclases de la *VODomainConcept*. Para definir esta ontología se propone la aplicación de metodologías para creación de ontologías [GPFLCG03, DNMN09, CBS$^+$09] que pueden ser aplicadas para crear esta conceptualización. Así mismo, se propone el uso de métodos de creación de ontologías semiautomáticos [JKK07, AKM$^+$03] que permiten crear ontologías a partir de elementos ya existentes como páginas web, bases de datos, documentos de texto, etc. que luego pueden ser refinadas por un experto del dominio. En el caso de estudio en el sector salud se reutilizó una ontología creada como modelo de referencia para estandarizar la transferencia de información entre en-

tidades del sector llamada "Modelo de Referencia de Información HL7" [Int08] y se integró con conceptos de historia clínica definidos por expertos.

Por otra parte, la definición de las subclases de *VOResource* y de *VOUnit* es mucho más simple ya que las clases son similares para todas las OV por lo cual su creación no representa un esfuerzo alto en la OV.

La actualización de la base de conocimiento está asociada a la creación, eliminación o modificación de clases, propiedades, individuos y relaciones entre individuos. La tecnología de ontologías fue seleccionada para describir los metadatos en las OV por que su naturaleza facilita la evolución de conceptos gracias a los motores de inferencia que reclasifican el conocimiento contenido cada vez que hay cambios.

La operación de creación no afecta el conocimiento anterior por lo cual se propone crearlo usando un editor o una interfaz de programación que permita agregar el nuevo elemento que se quiere incorporar a la ontología.

Las consecuencias de la operación de eliminación varían de acuerdo al elemento que se elimine. En el caso en que se elimine una clase será necesario eliminar toda las instancias creadas explícitamente como pertenecientes a estas subclases. La eliminación de individuos a su vez requiere la eliminación de todas las relaciones asociadas a ellas. La eliminación de propiedades requiere la eliminación de las relaciones entre individuos usando la propiedad eliminada.

Por último, la modificación de clases existentes hace referencia a la modificación de superclases, restricciones o de clases disyuntas. En los tres casos los cambios realizados son soportados automáticamente por el motor de inferencia de la ontología que los usará durante la siguiente clasificación. De igual manera, las modificaciones en el rango o el dominio de una propiedad serán usadas por el motor de inferencia en el momento de clasificar la ontología.

5.6. Síntesis

Este capítulo presenta ARIBEC, una arquitectura de mediación creada específicamente para OV. ARIBEC enriquece el nivel de mediación de la arquitectura de referencia para proveer un servicio de evaluación de consultas escalable, adaptable y comprensible para los usuarios de la OV cuando las fuentes de datos son en su mayoría estructuradas (por ejemplo bases de datos relacionales). ARIBEC está basada en la planeación usando la cartografía de la consulta que identifica la capacidad que tienen las fuentes de datos de resolver una consulta a nivel intencional(esquema) y extensional(contenido). Así mismo, la planeación se basa en un conjunto de hechos de conocimiento que permiten clasificar las fuentes de datos de acuerdo a su importancia para resolver una consulta.

La arquitectura de datos de ARIBEC presenta el mundo a través de un conjunto de objetos virtuales de datos (VDO) cuyas instancias son creadas a partir de las porciones de conocimiento que tiene cada una de las fuentes. La definición de una consulta es simple gracias a la predefinición de VDOs a los usuarios; sin embargo, entregar las instancias correspondientes a una consulta es una labor compleja debido a la fragmentación y replicación de las fuentes en

las OV. La propuesta de ARIBEC para resolver esta complejidad se ve reflejada en su arquitectura funcional y física. En la arquitectura funcional el proceso de planeación se adapta de acuerdo al tipo de consulta y al tipo de contexto. Adicionalmente, se añade un proceso de selección de fuentes previo a la labor de reescritura que reduce el número de fuentes de acuerdo a su esquema y contenido y agrupa las fuentes cuya intersección de instancias es no vacía. Esta decisión de diseño evita la generación de reescrituras innecesarias que no generan resultados y disminuye el contacto de fuentes innecesarias. De la misma forma, para asegurar la escalabilidad de la evaluación de consultas, la arquitectura física propone un sistema parcialmente descentralizado que distribuye el procesamiento y el almacenamiento de la información que soporta la planeación de consultas.

Finalmente, este capítulo presenta la base de conocimiento sobre la cual se apoyan las funcionalidades del nivel de mediación de ARIBEC para obtener información del contexto de datos. Esta base almacena la ontología de referencia de la OV y mantiene información de las fuentes de datos y de los participantes. A través de la base de conocimiento es posible registrar el conocimiento que se tiene del contenido de las fuentes en diferentes niveles de detalle lo que da mayor flexibilidad y escalabilidad al proceso de mediación, pues los metadatos no son tan costosos de almacenar, como los metadatos que se guardan a nivel de instancia, pero permiten diferenciar las fuentes que contienen el mismo esquema.

Capítulo 6

OptiSource: Estrategia de Selección de Fuentes de Datos para Organizaciones Virtuales

El Capítulo 5 ilustra las principales decisiones de diseño de ARIBEC, una arquitectura de mediación creada para OV. Como se mencionó, el nivel de mediación de ARIBEC incluye un proceso de selección que busca reducir las fuentes a sólo aquellas que pueden resolver la consulta a nivel intencional o extensional. Debido a que los procesos de selección existentes no están dirigidos a contextos de datos con fragmentación no disyunta y replicación entre fuentes, como sucede en las OV de gran escala, este capítulo presenta una propuesta dirigida a estos contextos. La propuesta, llamada OptiSource, es una estrategia de selección de fuentes creada para ambientes de datos complejos existentes en las OV de gran escala. OptiSource fue diseñado para resolver consultas conjuntivas que no necesariamente requieren respuestas completas.

La organización del capítulo es la siguiente. La Sección 6.1 presenta la selección de fuentes en OV de gran escala como un problema de decisión. La Sección 6.2 presenta OptiSource, la propuesta para seleccionar fuentes de datos en OV de gran escala. Posteriormente, la Sección 6.3 ilustra el modelo de estimación de beneficio utilizado en OptiSource para calcular la contribución de una fuente dentro de una consulta. La Sección 6.4 presenta la estrategia usada por OptiSource para evitar integrar las respuestas de fuentes de datos que no comparten instancias. Posteriormente, la Sección 6.5 ilustra el componente de optimización de OptiSource que permite encontrar la mejor combinación de fuentes para ejecutar una consulta. La Sección 6.7 ilustra los métodos propuestos para obtener los metadatos de las fuentes utilizados por OptiSource. Finalmente, la Sección 6.8 concluye el capítulo.

6.1. Selección de Fuentes de Datos como un Problema de Decisión

6.1.1. Objetivo de la Selección

El proceso de selección en los sistemas de mediación es el proceso de identificar el conjunto de fuentes de datos que deben participar en el procesamiento de una consulta. Dada una consulta sobre un objeto virtual de datos (VDO) cuyas instancias se encuentran fragmentadas, distribuidas y replicadas en múltiples fuentes, el proceso de selección debe decidir cuáles fuentes son necesarias para obtener las instancias que cumplen con las condiciones de la consulta. Por ejemplo, en la Consulta 4 (expresada en SPARQL) que solicita el identificador de todos los pacientes mujeres que hayan sido diagnosticadas con cáncer, la selección de fuentes debe identificar qué fuentes pueden resolver la consulta a nivel intencional y qué fuentes lo pueden hacer a nivel extensional. A nivel intencional se debe identificar qué fuentes contienen la clase *Patient* (el VDO) y de éstas cuáles contienen las propiedades *hasGender*, *hasClinicalData* y *hasDiagnosis* (ó al menos una de estas). A nivel extensional, es necesario identificar cuáles de estas fuentes contienen instancias de pacientes con las condiciones *hasGender* igual a "F" y cuya propiedad *hasClinicalData* tenga como rango un elemento (individuo) cuya propiedad *hasDiagnosis* esté asociada a un elemento de tipo *Cáncer*. Considerando el solapamiento intencional de las fuentes en las OV, y siguiendo con el ejemplo de una OV en el sector salud, una gran proporción de fuentes contienen la clase *Patient* y a su vez una gran proporción contienen las propiedades *hasGender*, *hasClinicalData* y *hasDiagnosis*. Por esta razón el proceso de selección debe enfocarse en la evaluación extensional para poder reducir el número de fuentes involucradas. Teniendo en cuenta que no es requisito que las respuestas sean completas, el proceso de selección debe elegir las fuentes que más instancias provean a la consulta y evitar aquellas cuyo aporte es nulo o es mínimo ya sea porque no contienen un gran número de instancias en relación a lo que proveen otras fuentes o porque su contenido es redundante con otra fuente que aporta más a la consulta.

Consulta 4 `SELECT ?idPaciente`

```
WHERE
{?idPaciente rdf:type vo:Patient.
 ?idPaciente vo:hasGender ?genero.
 ?idPaciente vo:hasClinicalData ?ActoMedico.
 ?ActoMedico vo:hasDiagnosis ?Diagnosis.
 ?Diagnosis rdf:type vo:Cancer.
 FILTER regex(?genero, "F").
}
```

Teóricamente, llevar a cabo esta selección significa identificar cuáles fuentes contienen las propiedades de VDO requeridas en la consulta, cuáles contienen instancias que concuerdan con todo o parte del predicado de la consulta, cuáles

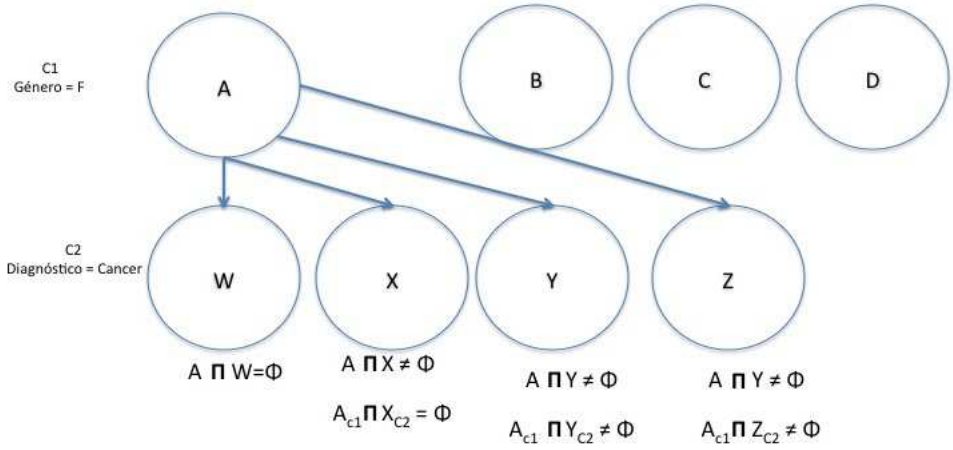

Figura 6.1: Problemática de Integración de Fuentes en OV con Fragmentación

tendrán una mayor contribución en la respuestas, y cuáles pueden ser eliminadas por ser redundantes en términos de instancias con otras fuentes. Sin embargo, identificar cuáles son estas fuentes en la práctica no es una labor trivial debido a las características de los contextos de datos de las OV, como se expondrá a continuación.

6.1.2. Implicaciones de la Fragmentación no Disyunta en el Proceso de Selección de Fuentes

A diferencia de las bases de datos distribuidas tradicionales en donde se sabe con certeza qué fuentes deben reunirse para evaluar una consulta, pues se conoce el diseño de la fragmentación vertical y horizontal, en los contextos de datos de las OV el solapamiento de fragmentos verticales y la incertidumbre de la fragmentación horizontal dificultan esta tarea. A continuación se presentan las razones de esta dificultad.

Fragmentación Vertical. Considerando la fragmentación vertical de los VDOs en las fuentes es posible que las propiedades contenidas en la consulta no estén en todas las fuentes de datos. Esta fragmentación es crítica al evaluar el predicado de la consulta, pues no es posible asegurar que una instancia corresponde al predicado hasta que todas las condiciones sean evaluadas. Para dar una respuesta correcta al usuario es necesario evaluar parcialmente el predicado en diferentes fuentes y luego reunir (join) las instancias para obtener el conjunto final de respuestas.

Fragmentación Horizontal. Debido a la fragmentación horizontal de los VDOs en las fuentes, la elección de qué fuentes se deben reunir para evaluar completamente un predicado no es una tarea simple. Considere la situación

presentada en la Figura 6.1, en donde existen cuatro fuentes de datos que contienen instancias que cumplen la condición de género femenino y otras cuatro que contienen instancias cuyo diagnóstico es cáncer. El primer grupo de fuentes no contiene la propiedad que permite evaluar el diagnóstico y el segundo grupo no contiene información de género. Para validar cuáles de las instancias contenidas en el primer grupo deben estar en la respuesta es necesario reunir las respuestas del grupo 1 con las del grupo 2. La solución intuitiva sería intentar reunir cada fuente del grupo 1 con cada fuente del grupo 2. Esta solución de reunión obtendrá todos los resultados existentes en el sistema. Sin embargo, su costo en número de mensajes y en tiempo es muy alto cuando el número de fuentes en alguno de los grupos es alto. Otra opción es elegir al azar una fuente del grupo 1 y reunirla con una fuente del grupo 2. En este caso, la respuesta obtenida puede ser vacía si las fuentes integradas no tenían instancias en común. Es necesario entonces reunir las fuentes que comparten instancias, es decir que son compatibles extensionalmente, y así evitar esfuerzos de integración cuyo resultado será vacío.

En la Figura 6.1 se ilustra si las fuentes de datos tienen o no instancias en común con la fuente de datos A usando el símbolo $\sqcap$. En los casos en donde hay instancias en común se presenta a través de un subíndice c_i si éstas coinciden con las condiciones de la consulta c_i que cada fuente evalúa. En esta figura se puede apreciar cómo la fuente de datos A sólo es compatible con las fuentes de datos X, Y y Z, pero sólo comparte instancias que concuerdan con la consulta con Y y Z, por esta razón debe ser evitado integrarla con W y X. El problema principal está en cómo detectar cuáles fuentes son realmente compatibles si no se tiene certeza de la localización de los datos, ni del contenido preciso de cada fuente. La estrategia de selección OptiSource debe resolver este problema.

Copias de Instancias. Considere que la integración entre A e Y arroja 5 instancias que concuerdan con el predicado, y la integración de A con Z arroja 20 instancias que concuerdan con el predicado. Si las 5 instancias validadas con Y están contenidas en las 20 instancias validadas con Z, Y puede ser descartada. Incluso si las 5 instancias de Y no están contenidas en Z, Y puede ser descartada si el usuario no está interesado en una respuesta completa y las 20 instancias de W son consideradas suficientes.

Requerimientos del Proceso de Selección

En conclusión, el proceso de selección de fuentes en OV de gran escala debe:

1. Elegir las fuentes que están habilitadas para resolver la consulta a nivel intencional y extensional

2. Identificar las agrupaciones de fuentes de datos compatibles extensionalmente, es decir que tienen instancias en común.

3. Identificar las fuentes redundantes, es decir las que no aportan nada nuevo ó no es necesario consultar pues ya hay suficientes respuestas.

6.1.3. Complejidad del Problema de Selección

Como se ilustró en la sección anterior, las implicaciones del contexto de las OV de gran escala vuelven complejo el proceso de selección. Más allá de identificar las fuentes relacionadas, la selección se convierte en un problema de decisión en donde se debe determinar si una fuente participará o no en la evaluación de una condición de la consulta.

Las entradas de este problema de decisión son:

- La consulta $Q(VDO, propiedades(p_1, p_2, ..., p_q), condiciones(c_i, ..., c_m))$
- Las n fuentes de datos disponibles $DS_i, 1 \leqslant i \leqslant n$ en la OV
- Los metadatos M_i declarados en la base de conocimiento de las OV que relaciona a las fuentes de datos $DS_i, 1 \leqslant i \leqslant n$ con el VDO_A

La salida es el grupo de conjuntos de fuentes cuyas respuestas deberían ser integradas $(CI_1, CI_2, ..., CIp)$ donde $CI_i = (DS_k, c_1), ..., (DS_l, c_m), 1 \leqslant k, l \leqslant n$. Una pareja (DS_k, c_l) significa que la fuente de datos DS_k puede evaluar la condición c_l. Una fuente puede evaluar varias condiciones c_j y una condición debe ser evaluada mínimo por una fuente de datos. Si una o más condiciones de la consulta no pueden ser evaluadas en mínimo una fuente de datos, la consulta es considerada insatisfacible.

El problema de selección de fuentes en los contextos de datos de las OV de gran escala se hace complejo debido a las siguientes características:

1. Gran número de posibles soluciones: cuando el número de fuentes que pueden resolver parcialmente las condiciones de la consulta es superior al 20 % de las fuentes disponibles, el conjunto de posible soluciones crece exponencialmente. Por esta razón hallar entre ellas las mejores soluciones puede tener una complejidad exponencial con respecto al número de fuentes. Considere por ejemplo un ambiente con 500 fuentes de datos y una consulta con un predicado compuesto por cinco condiciones de la forma $p_1 = v_1$ y $p_2 = v_2$ y $p_3 = v_3$ y $p_4 = v_4$ y $p_5 = v_5$, si cada propiedad p_i puede ser evaluada en 100 fuentes diferentes y ninguna fuente puede evaluar más de una condición, conformar el conjunto completo de instancias que coinciden con el predicado puede requerir integrar los resultados de cada fuente que puede resolver una condición, con todas las demás fuentes que pueden resolver las otras conclusiones. En resumen el número de posibles integraciones sería del orden de $100X100X100X100X100$ en el peor caso. Identificar cuál es la mejor manera de evaluar la consulta en este tipo de contextos no es una labor trivial y requiere de un gran esfuerzo para reducir los posibles planes de consulta que no arrojarán resultados o que arrojarán resultados redundantes con los de otro plan.

2. Fuertemente restringido: el proceso que selecciona de forma adecuada las fuentes que se van a integrar debe considerar restricciones impuestas por las fuentes de datos como sus capacidades de ejecución, la completitud de

sus datos y la calidad del contenido. En ciertos contextos las fuentes más relevantes por su calidad y número de registros no pueden ser consultadas directamente debido a exigencias en cuanto al número de resultados que entregan o en cuanto al número de condiciones en el predicado que exigen para resolver una consulta.

3. Objetivos contradictorios: La selección debe considerar por un lado, obtener el conjunto de respuestas completas disponibles en el sistema, y por otro, reducir el número de fuentes contactadas a únicamente las necesarias para evitar sobrecostos en tiempo y en procesamiento. Estos objetivos son contradictorios, por lo que la decisión debe hacerse buscando un balance entre los dos.

6.2. Estrategia Propuesta: OptiSource

Dada la complejidad del problema de selección de fuentes en las OV de gran escala, esta sección presenta OptiSource, una estrategia propuesta para obtener los conjuntos de integración apropiados durante la evaluación de una consulta en contextos complejos. OptiSource está dirigida a resolver consultas que involucran gran cantidad de fuentes de datos a nivel intencional, pero pocas a nivel extensional en contextos donde hay fragmentación no disyunta entre fuentes. Esta estrategia utiliza técnicas de estimación y optimización para decidir cuáles fuentes deberían ser consultadas si se quiere llegar a un equilibrio entre la completitud de la respuesta y el número de fuentes consultadas.

6.2.1. Principio de OptiSource: Fuentes Dominantes

Con el fin de evitar el contacto de fuentes de datos innecesarias durante la ejecución de consultas, el principio de OptiSource consiste en seleccionar las fuentes de datos dominantes para cada una de las condiciones de la consulta. Una fuente de datos es dominante si su contribución en términos de instancias es mayor que la de las otras fuentes de datos que pueden evaluar la condición (Ver Definición 6.2.1). La contribución de una fuente es estimada de acuerdo a su contribución individual con respecto a una condición de la consulta, es decir qué tantas instancias que satisfagan la condición aporta para resolver una consulta, y a su contribución en conjunto cuando es integrada con otras fuentes, de las instancias aportadas que satisfacen la condición qué tantas realmente satisfacen las otras condiciones evaluadas en otras fuentes. Las dimensiones utilizadas para estimar la contribución varían de acuerdo al método de estimación utilizado. La Figura 6.2 ilustra un gráfico de dominancia entre fuentes para una condición. OptiSource se encarga de identificar cuál es la fuente dominante para cada condición de la consulta. Una fuente de datos puede ser la dominante en varias condiciones.

Definición 10 ***Fuente Dominante***
Dada una consulta $Q(VDO, propiedades(p_1, p_2, ..., p_n), condiciones(c_i, ..., c_m))$,

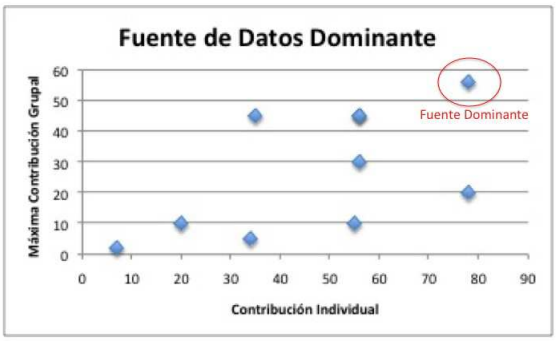

Figura 6.2: Dominancia entre Fuentes de Datos

sobre un objeto VDO que se encuentra distribuido en la OV. Dadas dos fuentes de datos DS1 y DS2 donde DS1 tiene una contribución individual CI_1 con respecto a c_i y una contribución grupal CG_1 con respecto a Q, y donde DS2 tiene una contribución individual CI_2 con respecto a c_i y una contribución grupal CG_2 con respecto a Q. DS1 domina a DS2 en la condición c_i si $(DS1.CI_1 > DS2.CI_2$ y $DS1.CG_2 \geq DS2.CG_2)$ o $(DS1.CI_1 > DS2.CI_2$ y $DS1.CG_2 \geq DS2.CG_2)$.

El proceso llevado a cabo por OptiSource para seleccionar las fuentes dominantes se resume en la Figura 6.3. El objetivo es identificar cuáles son las fuentes dominantes y cómo éstas deben ser integradas. Para lograr su objetivo OptiSource está dividido en tres fases: la estimación del beneficio, la optimización de la asignación y la renovación de conocimiento. Estas fases fueron inspiradas por los principios de la inteligencia de negocios adaptable presentada en [MSMC06] que, al igual que OptiSource, busca resolver un problema de decisión complejo. En el caso de OptiSource el problema de decisión es determinar cuáles son las fuentes que, por su dominancia, deben resolver la consulta.

En resumen, OptiSource funciona de la siguiente manera. En la fase de estimación de beneficio se utilizan los hechos contenidos en la base de conocimiento para determinar si una fuente de datos es útil para resolver una consulta ya sea a nivel intencional o extensional. Si es así, se estima el beneficio de utilizarla durante la evaluación de las condiciones de la consulta; esto corresponde a su *contribución individual*. Como se verá en la Sección 6.3 la precisión de la estimación varía de acuerdo al tipo de conocimiento disponible en la base de conocimiento de la OV y al método utilizado para calcularlo. Adicionalmente, para poder evaluar la contribución de las fuentes cuando son integradas con otras fuentes, esta fase identifica los conjuntos de integración viables. Es decir los grupos de fuentes que tienen una alta probabilidad de tener instancias en común y que por lo tanto tienen una *contribución en conjunto* positiva.

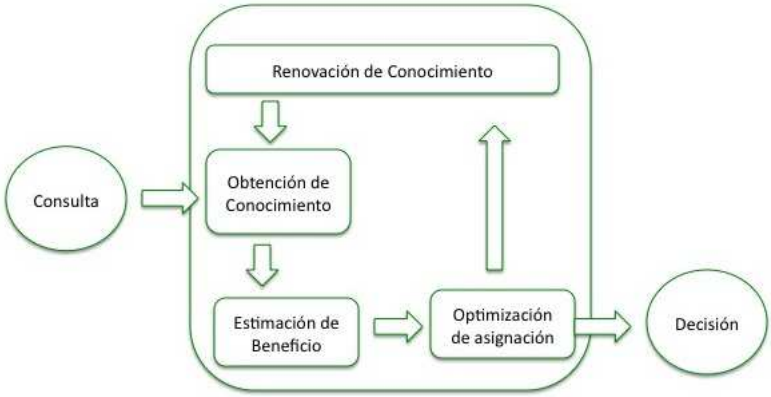

Figura 6.3: Fases de OptiSource

La fase de optimización utiliza un modelo matemático para identificar dentro de cada conjunto de integración las fuentes que, al ser integradas, representan la mejor *contribución en conjunto* a la consulta. El objetivo de este modelo es maximizar el beneficio obtenido al usar un conjunto de fuentes en la evaluación de una consulta minimizando el número de fuentes consultadas. La Sección 6.5 ilustrará en detalle el componente que realiza esta fase.

Finalmente, en la fase de renovación de conocimiento se evalúa la decisión de selección tomada por las fases anteriores con respecto a los resultados obtenidos y se renueva la base de conocimiento de acuerdo a los hallazgos obtenidos una vez la consulta es ejecutada. Esta fase del proceso de selección monitorea los resultados obtenidos durante la coordinación de ejecución de la consulta. La relación entre OptiSource y el componente de coordinación de ejecución de la consulta será presentado en la Sección 6.6.

6.2.2. OptiSource al interior de ARIBEC

OptiSource fue diseñado para ser usado como estrategia de selección de fuentes de datos en ARIBEC. Como se mencionó en el Capítulo 5, ARIBEC puede elegir diferentes estrategias de selección de acuerdo a las características del contexto y al tipo de consulta. El diseño de OptiSource está dirigido a contextos con alto número de fuentes estructuradas que se encuentren solapadas a nivel intencional y extensional y que además puedan presentar fenómenos de duplicación entre fuentes. En esta sección se presenta en primer lugar cómo OptiSource interactua con ARIBEC durante el procesamiento de consultas y cómo OptiSource utiliza los hechos de conocimiento almacenados en ARIBEC.

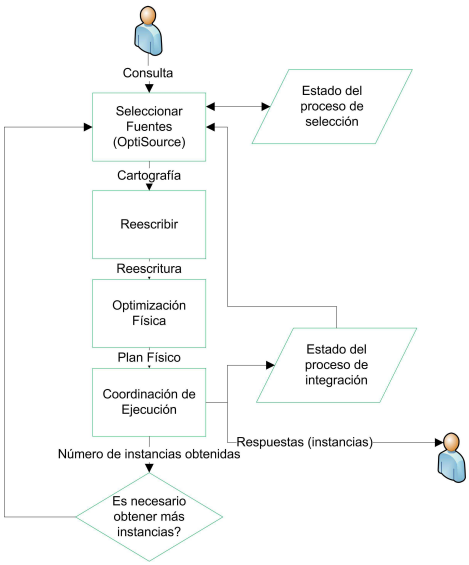

Figura 6.4: Proceso de Selección Iterativo con OptiSource

Selección iterativa de fuentes dominantes

Debido a que OptiSource elige las fuentes de datos dominantes para cada condición de la consulta, la respuesta entregada por estas fuentes puede que no incluya todas las instancias de VDOs existentes en el contexto de datos que concuerdan con el predicado de la consulta[1]. Esta respuesta incompleta en ciertos casos puede ser suficiente para el usuario, pero en otros casos el usuario puede requerir una respuesta que incluya más instancias o incluso una respuesta completa. Para soportar esto, OptiSource permite realizar diferentes iteraciones del proceso de selección, en donde cada iteración obtiene las fuentes dominantes eliminando las fuentes dominantes de la anterior iteración. La Figura 6.4 ilustra este proceso iterativo. En esta figura se puede observar cómo OptiSource registra en un repositorio temporal el estado del proceso de selección de tal manera que sea posible realizar múltiples iteraciones. Así mismo, el proceso de selección puede consultar el estado de integración de la consulta en donde se registra qué fuentes han respondido a una consulta y qué resultados han entregado. Esto no sólo permite renovar el conocimiento sino también identificar las nuevas fuentes dominantes de acuerdo al estado actual de integración. Por ejemplo, si ya se han obtenido todas las instancias existentes en la OV que cumplen con una condición, las fuentes de datos que aún no han sido consultadas y que también resuelven esta condición no serán dominantes pues su contribución será innecesaria si sólo son capaces de evaluar esta condición.

[1] Como se mencionó en el Capítulo 2 las fuentes de datos en las OV son incompletas debido a la fragmentación.

Roles de Fuentes de Datos

Como se mencionó en el Capítulo 5 el conocimiento que las fuentes tienen acerca de los VDO es descrito en la base de conocimiento a través de los roles autoridad, especialista y contenedor. Estos roles reflejan el grado de importancia que tiene en la extensión de una fuente cierto tipo de instancias de VDO. Un tipo de instancia de VDO es descrito a través de la restricción de una o más propiedades del VDO a un conjunto específico de valores. Por ejemplo, el tipo de instancias *Paciente Niño* del VDO *Paciente* corresponde a todas la instancias de *Paciente* donde la propiedad edad es menor a 13 años. OptiSource utiliza los roles definidos en la base de conocimiento (ver definiciones 7, 8 y 9) como insumo de su algoritmo de selección. Para OptiSource el hecho de que una fuente tenga un rol con respecto a un tipo de instancia de VDO es sinónimo de su habilidad para resolver consultas que tengan asociadas las condiciones restringidas que definen el tipo de instancia.

Los roles pueden ser obtenidos usando tres aproximaciones diferentes: (1) Manualmente (por ejemplo el DBA definió el conocimiento), (2) Interpretando los procesos de la OV, (3) Automáticamente extrayendo este conocimiento de las fuentes. La Sección 6.7 presentará las estrategias propuestas en este proyecto para las dos últimas aproximaciones.

Ahora que ya se conoce el principio y el funcionamiento general de OptiSource, las siguientes secciones ilustrarán cómo se llevan a cabo sus fases.

6.3. Estimación del Beneficio de Usar una Fuente de Datos

Calcular la contribución de una fuente de datos en una consulta no es una labor trivial en los sistemas distribuidos y heterogéneos, principalmente por la falta de estadísticas y de metadatos. Sin embargo, la connotación organizacional de las OV permite inferir conocimiento de las fuentes que no se encuentra en otros sistemas distribuidos. Los roles de las organizaciones proveedoras de conocimiento y los procesos organizacionales suministran conocimiento que ayuda a describir con mayor precisión las fuentes de datos. Esta sección presenta cómo puede ser calculada la contribución individual de una fuente con respecto a una condición de la consulta usando una medida de beneficio que utiliza aspectos organizacionales para estimar la eficacia de una fuente de datos resolviendo una consulta.

El beneficio es un valor numérico que permite organizar en orden decreciente la contribución de las fuentes de datos con respecto a una condición. Si dos fuentes tienen el mismo beneficio podrán estar en la misma posición. La fuente con mayor beneficio ocupará la primera posición. Para calcular este beneficio se pueden usar diferentes estrategias como puede observarse en la Figura 6.5 en donde cada punto representa una estrategia para estimar el beneficio de usar una fuente en una consulta. De acuerdo al nivel de conocimiento, la estimación del beneficio será más o menos fiable. Elegir la aproximación para calcular

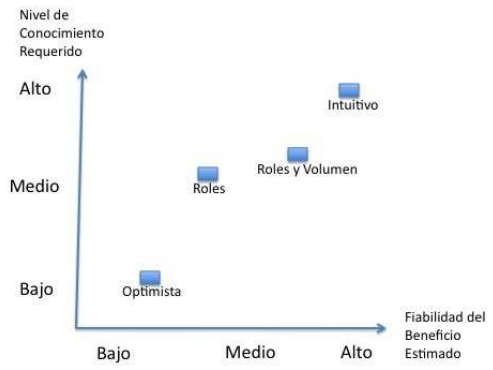

Figura 6.5: Aproximaciones para estimar el beneficio

el beneficio depende en gran medida de la posibilidad de tener información detallada de las fuentes y de la necesidad de reducir el contacto de fuentes innecesarias. Como se verá más adelante OptiSource utiliza la aproximación basada en roles y en volumen que permite tener una fiabilidad media-alta en la estimación con un nivel de conocimiento medio.

6.3.1. Aproximación Intuitiva

La aproximación intuitiva para calcular el beneficio de usar una fuente en una consulta es priorizarla de acuerdo al número de instancias que satisfacen el predicado de la consulta. En este caso la fuente con más instancias que concuerdan con el predicado estaría en el primer lugar. Si bien esto sería el caso ideal, no es posible su aplicación en las OV por dos razones: la falta de conocimiento de cuántas instancias que cumplen el predicado están contenidas en cada fuente, y la fragmentación vertical que impide conocer con exactitud cuántas instancias contenidas en una fuente cumplen con una condición que no puede ser evaluada en la fuente.

6.3.2. Aproximación Optimista

Esta aproximación tiene como principio que el tamaño de la fuente determina el número de instancias que podrá aportar en una consulta, por esta razón si una fuente tiene más instancias la posibilidad de que entregue instancias que resuelvan una consulta será mayor. Esta aproximación optimista puede ser aplicable en contextos en donde unas pocas fuentes concentran la información de un gran número de fuentes de menor tamaño. Sin embargo, la hipótesis de contenido en las fuentes es en la mayoría de contextos de datos muy fuerte como para ser el único insumo del cálculo de beneficio.

6.3.3. Aproximación usando Roles

Los enfoques tradicionales usados en las fuentes de datos distribuidas no hacen ninguna diferenciación entre las fuentes de datos según su contribución a una consulta. Desafortunadamente, tratar a todas las fuentes por igual hace difícil la identificación de cuáles son las más útiles para resolver una consulta, desde el punto de vista de su contenido. Gracias a la dimensión organizacional de las OV, es posible distinguir ciertas diferencias entre las fuentes de acuerdo a las organizaciones participantes que las proporcionaron. En la OV en el sector salud por ejemplo, dos fuentes de datos pueden aportar información sobre los procedimientos realizados a pacientes. Si uno de ellos es proporcionado por una organización que se especializa en el diagnóstico del cáncer y el otro en una organización que se especializa en procedimientos pediátricos, las fuentes tendrán características diferentes en contenido, especialmente para las propiedades diagnóstico y edad del VDO paciente. Estos metadatos pueden ser utilizados para diferenciar el rol que puede jugar una fuente durante la ejecución de la consulta. Una fuente especializada en Cáncer será más relevante que una fuente especializada en Glaucoma para resolver una consulta en donde se solicita la información de paciente con diagnóstico Cáncer. Así mismo, de acuerdo al grado de conocimiento que se tenga del contenido de las fuentes, el rol que puede jugar una fuente puede ser declarado en un nivel más o menos detallado. Por ejemplo, la fuente especialista en Cáncer, puede que sólo sea especialista en diagnósticos de Cáncer de Seno. En el caso en el que se tenga este conocimiento detallado el rol de especialista puede declararse directamente con el diagnóstico Cáncer de Seno. Si por el contrario, no se dispone de este conocimiento detallado el rol puede asociarse a un nivel más alto dentro de la taxonomía de enfermedades, vinculando el rol de especialista a Cáncer. La asociación de los roles a los elementos de la ontología de la OV proporciona una alta flexibilidad en el momento de describir el conocimiento acerca de las fuentes, pues permite tener diferentes niveles de detalle de acuerdo al conocimiento o a la importancia de una fuente dentro de la OV.

Estimar el beneficio usando esta aproximación relaciona el rol de una fuente de datos con respecto a un concepto con el beneficio que se obtendrá si es utilizada. Los roles son obtenidos de la base de conocimiento de la OV que, como se ilustró en el Capítulo 5, almacena las relaciones de conocimiento entre las fuentes y los conceptos del dominio a través de tres especializaciones de la propiedad *juegaRoldeConocimiento*: *autoridad, especialista* y *contenedora*.

Los roles permiten priorizar las fuentes con respecto a un tipo de VDO cuando las fuentes están solapadas extensionalmente. Por ejemplo, dada una consulta sobre el VDO Paciente que solicita todos los identificadores de pacientes con la condición *diagnóstico=Cancer* y un conjunto de fuentes $S\{DS_1, DS_2, DS_3\}$ cuyos roles con respecto a *diagnóstico=Cancer* son respectivamente Contenedor, Autoridad y Especialista, la priorización de las fuentes sería DS_2, DS_3, DS_1, pues el beneficio de usar DS_2 se espera que sea mayor en número de instancias y en fiabilidad de información por su condición de autoridad; de la misma forma se espera que la DS_3 genere más instancias y mayor detalle de información por

Rol Nominal	Rol Numerizado
Autoridad	1
Especialista	0.66
Contenedor	0.33

Cuadro 6.1: Numerización del Rol

su condición de especialista que DS_1, que es contenedor. Sin embargo, en el caso en que la declaración de conocimiento de DS_2 no esté hecha en el nivel de Cáncer sino en una subclase de Cáncer llamada Cáncer de Seno la priorización no sería muy precisa usando únicamente el rol. En este caso puede que la fuente DS_3 especialista en *Cáncer* sea más relevante que DS_2 que es autoridad en *Cáncer de Seno*. Es necesario entonces diferenciar las fuentes usando el rol y el nivel en el que se encuentra declarado el rol. Para hacerlo la propuesta es numerizar el rol y el nivel, y combinarlos usando una función que refleje estas diferencias entre las fuentes. A continuación se presenta cómo fue hecha esta numerización y cómo son combinados los dos valores.

Numerización del Rol

Para numerizar el rol se propone el uso de la técnica de numerización 1 a 1 que permite identificar un cierto orden o magnitud en los valores del atributo originalmente nominal. El Cuadro 6.1 presenta la numerización realizada del rol. Este método puede ser adaptado o modificado de acuerdo a las necesidades de la OV.

Numerización del Nivel del Conocimiento

Para numerizar el nivel en el que se encuentra el conocimiento se procedió de la siguiente manera. Si el rol de la fuente de datos está asociado a la clase exacta que se utilizó en la consulta su valor es 0. Si el rol está en un nivel inferior o superior su valor variará de acuerdo al tipo de rol. El Algoritmo 1 ilustra la lógica de numerización. Fue necesario tener en cuenta el rol y el nivel pues la implicación de que una fuente sea autoridad en un nivel superior es diferente a la implicación de que sea una especialista o contenedor en un nivel superior. Por ejemplo si la consulta se hace en el nivel *Cancer de Seno* y el conocimiento está declarado en el nivel *Cancer*, si la fuente de datos es autoridad quiere decir que contendrá todas las instancias incluso las de *Cancer de Seno*, mientras que si es especialista o contenedor no necesariamente implica que tenga instancias de paciente con *Cancer de Seno*. Por esta razón, el Algoritmo 1 modifica el valor de la variable *factordeNivel* para reflejar si el nivel de conocimiento está o no al mismo nivel de la consulta, y si no es así valora qué tan alejado está el conocimiento del nivel en el que se definió la consulta.

Algoritmo 1 Numerización del Nivel

```
Entrada:  condicion, rol, DS
Salida:   factordeNivel
Inicio
    nivelCondicion = consultaBaseKnivelClase(condicion)
    nivelConocimiento = consultaBaseKInd (rol,condicion,DS)
    numNivelesSuperiores = consultaBaseKnivelesSup(condicion)
    numNivelesInferiores = consultaBaseKnivelesInf(condicion)
    Si nivelConocimiento = nivelCondicion entonces
        factordeNivel = 0
    Sino Si (rol = "Autoridad") entonces
        Si nivelConocimiento > nivelCondicion entonces
            factordeNivel = 0
            //Si el rol es autoridad y el conocimiento estA en un
            nivel superior la fuente continUa siendo autoridad en
            el nivel en el que estA la consulta
        SiNo
            descuento = 1/numNivelesInferiores
            factordeNivel = (nivelCondición - nivelConocimiento)
                                                    * descuento
        FinSi
    SiNo
        Si nivelConocimiento > nivelCondicion entonces
            descuento = 1/numNivelesSuperiores
            factordeNivel = (nivelConocimiento - nivelCondicion)
                                                    * descuento
        SiNo
            descuento = 1/numNivelesInferiores
            factordeNivel = (nivelCondición - nivelConocimiento)
                                                    * descuento
        FinSi
    FinSi
    Retorne (factordeNivel)
Fin
```

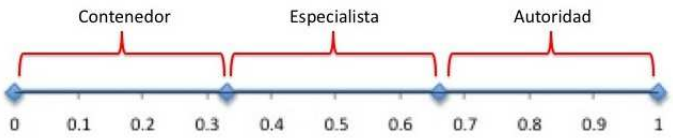

Figura 6.6: Rango del Factor del Rol

Combinación Rol y Nivel

Dado que una fuente puede tener asociado más de un rol con respecto a un tipo de VDO, el cálculo del factor del rol tiene en cuenta únicamente el rol más importante. El factor del rol representa la relevancia de una fuente de datos con respecto a un tipo de VDO. Entre mayor sea el factor del rol de una fuente, más relevante será la fuente al evaluar una consulta para el tipo de VDO en cuestión. La fórmula general para el cálculo es llamada *FactordeRol* y se presenta en la Fórmula 6.1. DSi es la fuente de datos, VDOj es el objeto virtual sobre el cual se hizo la consulta, c es la condición de la consulta que se está evaluando. Las funciones esContenedor, esEspecialista, esAutoridad retornan 1 si la fuente DSi tiene asociado el rol con respecto al VDOj cuando la condición c es verdadera, 0 en caso contrario. factorContenedor, factorEspecialista, factorAutoridad numerizan el rol y factorNivelConocimiento numeriza el nivel del conocimiento del máximo rol encontrado. La variable *diferenciaRol* es la diferencia que hay entre un rol y otro cuando son numerizados.

$$\begin{aligned} FactordeRol(DSi, VDOj, c) = \text{máx}((&esContenedor(DSi, VDOj, c) * factorContenedor()), \\ &(esEspecialista(DSi, VDOj, c) * factorEspecialista()), \\ &(esAutoridad(DSi, VDOj, c) * factorAutoridad()))- \\ &(diferenciaRol * factorNivelConocimiento(c, maxRol, DSi) \end{aligned} \tag{6.1}$$

El resultado de la Fórmula 6.1 reduce el factor del rol de acuerdo al nivel en el que se encuentra. La Figura 6.6 ilustra los rangos de movilidad de cada rol. La cota máxima sólo es posible si el rol está asociado al mismo nivel en el que se encuentra la condición, en caso contrario el valor del rol decrecerá. El valor de decrecimiento estará siempre entre el rango inmediatamente inferior y el rango inmediatamente superior evitando que fuentes con diferentes roles tengan el mismo valor.

Bajo esta aproximación el beneficio es calculado usando el Factor del Rol que permite priorizar las fuentes de acuerdo al rol que desempeñan y al nivel en el que se encuentra declarado este rol. Las fuentes de datos con roles de autoridad en el mismo nivel en el que se encuentra la condición serán las más importantes. De la misma forma y a simple vista las fuentes especialistas serían las siguientes en importancia, y por último las fuentes contenedoras. Sin embargo, en algunos casos puede que una fuente contenedora sea más importante que una fuente especialista. Considere por ejemplo dos fuentes de datos DS1 y DS2, en donde

DS1 es especialista y DS2 es contenedora. Ahora considere que DS1 almacena 100 instancias del VDO y que DS2 almacena 4000 instancias del VDO. En este caso incluso si el 100 % de las instancias de DS1 concuerdan con la clase por la cual DS1 es especialista, DS2 podría generar el mismo beneficio si sólo el 2.5 % de las instancias correspondieran con la clase por la cual DS2 es contenedora. Esto es muy común en el sector salud en donde una institución puede ser especialista en un tipo de diagnóstico pero, por su localización, precio, reconocimiento, convenios, etc. el número de pacientes no es muy alto. Por el contrario, otra institución puede no ser especialista en ningún tipo de diagnóstico pero por su tamaño, reconocimiento, etc, puede tener una mayor cobertura de pacientes. La siguiente aproximación presentada considera este aspecto.

6.3.4. Aproximación usando Roles y Volumen

Como se observó en la aproximación por roles, es posible identificar cierto grado de importancia de las fuentes al resolver una consulta. Sin embargo, puede que esto no sea suficiente para obtener un nivel de beneficio esperado cercano a la realidad. La aproximación que se presenta a continuación funciona bajo la hipótesis de que si una fuente es declarada especialista o contenedora con respecto a una condición sobre un VDO, entre mayor sea el número de instancias que contiene más probable es encontrar instancias que correspondan con esta condición. El cálculo de beneficio de las fuentes que no son autoridades en esta aproximación combina el *Factor del Rol* propuesto en la aproximación anterior con el volumen de las fuentes de datos. La Fórmula 6.2 ilustra la manera como el beneficio de una fuente es calculado a partir de estas dos medidas. Para las fuentes autoridades en el mismo nivel de la condición o en un nivel superior el beneficio es calculado con el $FactordeRol(DSi, VDOj, c)$, pues cuando una fuente es autoridad se tiene certeza de que todas las instancias que cumplen la condición están contenidas en ella. Por el contrario, si es autoridad en un nivel inferior no se tiene esta certeza y por esta razón se usa la Fórmula 6.2.

$$Beneficio(DSi, VDOj, c) = FactordeRol(DSi, VDOj, c) * \frac{card(ext(DSi, VDOj))}{\text{máx}\{card(ext(DSk, VDOj)), \forall\, DSk\, in\, U\, where\, FactordeRol(DSk, VDOj, p) > 0\}} \tag{6.2}$$

donde $DS_i : j$ es una fuente de datos i con un rol con respecto a la condición c, $ext(DSi, VDOj)$ es la extensión de DSi en cuanto a instancias de VDOj y $card(ext(DSi, VDOj))$ es la cardinalidad de esta extensión y $FactordeRol(DSi, VDOj, c)$ es el factor del rol de la fuente a la cual se le está calculando el beneficio. $card(ext(DSk, VDOj))$ es la cardinalidad de la fuente que tiene mayor número de instancias con respecto a VDOj y que tiene un rol con respecto a la condición c, esta cardinalidad es utilizada para normalizar el valor de la cardinalidad de las fuentes entre 0 y 1.

Aunque la fórmula de Beneficio podría ser más precisa si se usara la cardinalidad de la fuente teniendo en cuenta la condición -$card(ext(DSi, VDOj)^c)$-, el número exacto de instancias que satisfacen la condición no está siempre disponible en las OVs. En el caso en que esta información esté disponible será

utilizada, en caso contrario se usará la cardinalidad general de las fuentes.

6.4. Conformación de los Conjuntos de Integración

La estimación del beneficio permite identificar la contribución individual para cada una de las condiciones de la consulta. Sin embargo, aún hace falta obtener la contribución en conjunto. Para esto el primer paso es identificar qué fuentes de datos pueden crear conjuntos de integración (ver Definición 11) que dirijan los esfuerzos de integración a fuentes que contienen grupos de instancias similares. Esta sección presenta la propuesta para hacerlo utilizando la semántica organizacional de las OVs.

Definición 11 ***Conjunto de Integración***
Un conjunto de integración para una consulta Q es un grupo de fuentes de datos capaces de evaluar todas las condiciones de la consulta Q cuyo join puede producir un resultado no vacío con respecto a las condiciones de Q.

6.4.1. Identificación de Conjuntos de Integración

Como se mencionó en el Capítulo 5 las unidades que hacen parte de las OVs pueden ser atómicas, en el caso de los participantes individuales, o compuestas, cuando múltiples atómicas se asocian. Para las unidades atómicas la base de conocimiento tiene registrada la información básica como el nombre, su rol dentro de la OV, su fecha de vinculación y/o desvinculación a la OV, los procesos organizacionales en los que participa, el número de usuarios, su localización física y los recursos que provee a la OV. Para las unidades compuestas se tiene registrado su identificación, la fecha de creación y/o disolución, las unidades que hacen parte de ella, su objetivo de creación, los roles que cada unidad juega para cumplir el objetivo y los procesos organizacionales que ejecuta. La creación de conjuntos de integración toma ventaja de esta información almacenada en la base de conocimiento para determinar cuáles fuentes tienen mayor probabilidad de tener las mismas instancias.

La estrategia para conformar los conjuntos de integración es utilizar una regla heurística que se apoye en la información organizacional de la OV. Esta regla permite crear los conjuntos de integración iniciales, que posteriormente son refinados para asegurar que sean completos y no redundantes. La completitud hace referencia a que cada conjunto de integración debe ser capaz de evaluar todas las condiciones de la consulta. La redundancia evita que existan conjuntos de integración con las mismas fuentes. En el sector salud por ejemplo, la regla heurística de usar la localización de las unidades proveedoras de datos y agruparlas por ciudad es efectiva para identificar compatibilidad entre fuentes, pues la movilidad de los pacientes generalmente se da en una misma ciudad. Una lista no exhaustiva de reglas heurísticas es presentada a continuación:

1. Las unidades atómicas que pertenecen a la misma unidad compuesta tienen mayor probabilidad de tener el mismo grupo de instancias.
2. Las unidades atómicas que están localizadas físicamente en el mismo lugar (ciudad, región, país) tienen mayor probabilidad de tener el mismo grupo de instancias.
3. Las unidades que participan en el mismo proceso de negocios tienen mayor probabilidad de tener el mismo grupo de instancias.
4. Las unidades que llevan a cabo el mismo rol dentro de la OV tienen mayor probabilidad de tener el mismo grupo de instancias.

El Algoritmo 2 usa la regla heurística 1 (unidades compuestas) para crear los conjuntos de integración. Inicialmente se (1) obtiene la información de la regla heurística elegida y se crean los conjuntos de integración iniciales. Si hay conjuntos de integración redundantes, el algoritmo remueve el conjunto con menor número de fuentes. Luego, en (2) se valida la completitud de un conjunto. Un conjunto es completo si las fuentes de datos que contiene son capaces de evaluar todas las condiciones de la consulta. En (3) se identifica cuáles de las fuentes de datos contenidas en un conjunto incompleto ya están en un conjunto completo para evitar redundancias y (4) remueve los conjuntos incompletos cuyas fuentes de datos ya están en un conjunto completo. Si luego de esta validación aún hay conjuntos incompletos el algoritmo identifica las condiciones que aún no han sido evaluadas en cada uno de los conjuntos incompletos restantes (5) y los completa usando (6) las fuentes de datos con mayor contribución individual en esa condición, de los conjuntos completos. En este último paso es posible utilizar otra regla heurística, como por ejemplo usar la fuente más cercana. Al finalizar, el algoritmo entrega las fuentes de datos agrupadas en conjuntos de integración.

6.4.2. Cartografía de la Consulta

Hasta este punto OptiSource ha determinado cuál es la contribución individual de cada fuente y cuáles son los conjuntos de integración más probables. El cálculo de la contribución individual permite resolver el problema de solapamiento extensional e intencional, pues las fuentes de datos son diferenciadas y priorizadas incluso si proveen instancias del mismo tipo de VDOs. Por su parte, la detección de grupos de integración ayuda a solucionar los problemas causados por el desconocimiento de qué fuentes pueden ser compatibles. Con esta información OptiSource genera la cartografía de la consulta. La Figura 6.7 ilustra un ejemplo de la cartografía obtenida hasta este punto usando OptiSource.

Sin embargo, hasta este momento la cartografía no especifica cuáles son las fuentes de datos dominantes con respecto a la contribución grupal. La contribución grupal es importante si al interior de un conjunto de integración más de una fuente puede evaluar la misma condición con valores de beneficio individual similares. Decidir cuál fuente debe evaluar cuál condición es el objetivo de la fase de optimización, que será descrita en la siguiente sección.

Algoritmo 2 Creación de Conjuntos de Integración

```
Entrada:Q, UnidadesCompuestas{ID1(DSi,..,DSj),...,IDm(DSk,...DSm)},
QRoles(DSl:role,...,DSn:role)
Salida: Conjuntos de Integración
                          {JS1(DSp,...,DSq),...,JSn(DSt,...,DSv)}
Inicio
(1)setsIniciales{} = CrearSetsIniciales(UnidadesCompuestas,QRoles)
   setsIncompletos{} = {}
   setsCompletos{} = {}
   ParaTodo (set en setsIniciales) hacer
(2)   Si (esCompleto(set,Q))
         adicionar(setsCompletos,set)
      SiNo
        adicionar(setsIncompletos,set)
      Fin_si
   Fin ParaTodo
   Si (tamaNo(setsIncompletos) > 0)
(3)  com{} = obtenerFuentes(setsCompletos)//Las fuentes de los sets
                                                            completos
     ParaTodo (set en setsIncompletos)
       inc{} = obtenerFuentes(set) //Las fuentes de un set
                                                          incompleto
       Si (contiene(com, inc)) //com contiene inc
(4)      remueve(set, setsIncompletos)
       SiNo
(5)      condiciones{} = obtenerCondicionesFaltantes(set, Q)
         ParaTodo (cond en condiciones)
(6)         fuente = obtenerRolMasAlto(com,cond)
            adicionarFuente(set,fuente)
         Fin ParaTodo
         adicionar(setsCompletos,set)
       FinSi
     Fin ParaTodo
   FinSi
   Retornar(setsCompletos)
Fin
```

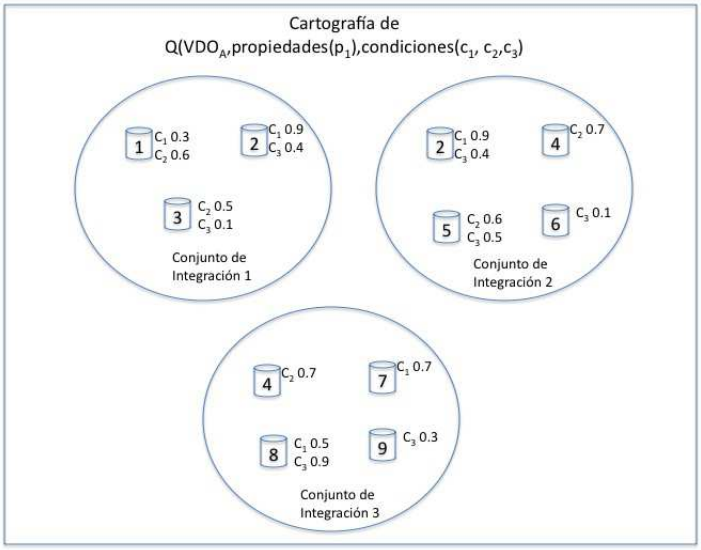

Figura 6.7: Cartografia en OptiSource

6.5. Optimización de la Asignación de Condiciones

Un conjunto de integración puede tener múltiples fuentes de datos habilitadas para evaluar una condición de la consulta, cada una con valores de beneficio diferentes con respecto a la condición. Consultar todas estas fuentes, no es necesario, ni conveniente, considerando que las fuentes pueden ser redundantes y que no se requiere el conjunto de respuestas completo.

La propuesta es ver el problema de decidir cuáles condiciones deberían ser evaluadas en qué fuentes de datos de un conjunto de integración como un problema de asignación [HL05] sujeto a restricciones de recursos que asigna una condición a una fuente, sólo si esta ayuda a maximizar el beneficio general (la contribución grupal).

Para resolverlo se creó un modelo matemático que recibe como entrada los beneficios individuales de las fuentes para cada uno de las condiciones y las capacidades de procesamiento de las fuentes, si son conocidas, y obtiene la asignación óptima de condiciones a fuentes de datos. Aunque este modelo fue creado para la selección de fuentes en OVs de gran escala, puede ser utilizado en cualquier sistema de procesamiento de consultas distribuidas durante la fase de planeación. Esta sección presenta en la Sección 6.5.1 el problema de selección de fuentes como un problema de asignación y en la Sección 6.5.2 el modelo matemático propuesto para solucionarlo. Finalmente, la Sección 6.5.3 presenta cómo la asignación realizada por el modelo de optimización genera el conjunto

de fuentes dominantes para cada conjunto de integración.

6.5.1. Selección de Fuentes como un Problema de Asignación

Para resolver el problema de elegir las fuentes más apropiadas, el problema fue modelado como una adaptación al problema de asignación. La forma general del problema de asignación es: "Existen un número de agentes y un número de actividades. Cualquier agente puede ser asignado para ejecutar una actividad, incurriendo en un costo que puede variar de acuerdo a la asignación agente-actividad. Es obligatorio ejecutar todas las actividades asignando exactamente un agente a cada actividad de tal forma que el costo total de la asignación sea minimizado." [HL05]

En el problema de selección de fuentes, existe un número de condiciones de la consulta que deben ser asignados a un número de fuentes de datos. Cada asignación genera un nivel de beneficio y consume recursos de las fuentes. El objetivo es maximizar los beneficios usando el mínimo número de fuentes al mismo tiempo que se respetan las restricciones de recursos de las fuentes.

Una condición es asignada a una fuente de datos principal como *asignación principal*, pero la condición puede ser evaluada como *asignación secundaria* en paralelo en otras fuentes que fueron seleccionadas por otras condiciones. La razón por la cual se hizo esta modificación al problema original es que una vez que una fuente de datos es consultada, es mejor evaluar en ella el máximo número posible de condiciones que permitan reducir el número de fuentes consultadas y el costo de transmitir instancias que no han sido completamente evaluadas.

En efecto, el problema de asignación de condiciones a fuentes de datos es un problema combinatorio de optimización bi-objetivo sujeto a semi-asignación y a restricciones de recursos. En práctica, se decidió controlar el objetivo de minimizar el número de fuentes de datos seleccionadas convirtiendo el objetivo en una restricción. Por defecto, el número de fuentes seleccionadas es limitado por el número de condiciones de la consulta.

6.5.2. Modelo Matemático

Dado un conjunto de fuentes de datos $I = \{1, ..., m\}$ y un conjunto de condiciones $C = \{1, ..., n\}$, los datos de entrada al modelo matemático propuesto son :

- $Ben_{i,c}$: beneficio de asignar una condición c a la fuente de datos i, $\forall i \in I$,$\forall c \in C$, como fue explicado en la fórmula 6.2 en la Sección 6.3, $Ben_{i,c} = Benefit(DSi, VDOj, c)$ para un $VDOj$;
- $MaxRes_i$: capacidad de procesamiento de la fuente de datos i, $\forall i \in I$;
- $Res_{i,c}$: recursos consumidos en asignar una condición c a la fuente i, $\forall i \in I$,$\forall c \in C$;

- $MaxAssig_i$: máximo número posible de condiciones asignadas a la fuente de datos i, $\forall i \in I$.

Es importante aclarar que en el caso en que no sea conocido el valor de $MaxRes_i$, $Res_{i,c}$, $MaxAssig_i$ los valores que serán asignados serán valores altos para $MaxRes_i$ y $MaxAssig_i$ y negativos para $Res_{i,c}$. Estas entradas fueron incluidas para permitir modelar las restricciones que ciertas organizaciones participantes imponen sobre los recursos que comparten.

Las variables de decisión son:

- $x_{i,c}$ es una variable binaria que determina si una fuente de datos i debe (=1) o no (=0) ser asignada como fuente principal para evaluar la condición c, $\forall i \in I$,$\forall c \in C$.

- y_i es una variable binaria que toma el valor 1 cuando la fuente la fuente i es asignada, $\forall i \in I$.

- $assig_{i,c}$ es una variable binaria que determina si una condición c es asignada a una fuente de datos i (=1) o no (=0), $\forall i \in I$. Esta variable representa la asignación final de condiciones a fuentes de datos. La variable $x_{i,c}$ indica las asignaciones principales mientras que la variable $assig_{i,c}$ indica las asignaciones principales y secundarias.

El programa matemático formulado es el siguiente:

$$max \sum_{c=1}^{n} \sum_{i=1}^{m} Ben_{i,c} * (x_{i,c} + assig_{i,c}), \tag{6.3}$$

sujeto a:

$$\sum_{i=1}^{m} x_{i,c} = 1, \forall c \in C\,; \tag{6.4}$$

$$\sum_{i=1}^{m} y_i \leq k\,; \tag{6.5}$$

$$\sum_{c=1}^{n} x_{i,c} \geq y_i, \forall i \in I\,; \tag{6.6}$$

$$\sum_{c=1}^{n} Res_{i,c} * assig_{i,c} \leq MaxRes_i, \forall i \in I; \tag{6.7}$$

$$\sum_{c=1}^{n} assig_{i,c} \leq MaxAssig_i, \forall i \in I\,; \tag{6.8}$$

$$x_{i,c} \leq assig_{i,c}, \forall i \in I, \forall c \in C; \tag{6.9}$$

$$assig_{i,c} \leq y_i, \forall i \in I, \forall c \in C. \tag{6.10}$$

La restricción (6.4) asegura que todas las condiciones c sean asignadas a una fuente principal. La restricción (6.5) limita el número total de fuentes consultadas a k. Por defecto k=n para iniciar el proceso de optimización. Esta restricción es en cierta forma redundante con la restricción (6.6) que previene seleccionar una fuente de datos si ninguna condición es asignada a ella; sin embargo, fue incluida para poder reducir k en la restricción (6.5) y así permitir la minimización del número de fuentes seleccionadas. Cada iteración puede reducir k a $k-1$ para validar si con la reducción de una fuente es posible mantener el mismo valor de la función objetivo.

Las restricciones (6.7) y (6.8) evitan que las asignaciones principales y secundarias excedan la capacidad de procesamiento, o el número máximo de asignaciones permitidas por fuente. Finalmente, las restricciones de acoplamiento (6.9) y (6.10) indican respectivamente que aquellas fuentes con asignaciones principales deben ser asignadas como fuentes secundarias a las demás condiciones que están en capacidad de evaluar, y que la fuente i es seleccionada cuando al menos una condición c es asignada a ella. El modelo completo puede ser consultado en el Apéndice B.

6.5.3. Identificación de Fuentes de Datos Dominantes

La solución de este modelo provee las fuentes de datos asignadas en la variable y_i y todas las asignaciones primarias y secundarias en la variable $assig_{i,j}$. La reunión de los resultados provistos por cada condición en cada fuente asignada son las instancias requeridas por el usuario. Si el número de instancias obtenidas de todos los conjuntos de integración no son suficientes para satisfacer los requerimientos de usuario, el modelo de optimización generará una nueva asignación con las fuentes de datos restantes, es decir aquellas que no estaban asignadas en y_i.

El proceso de optimización presentado en esta sección se encarga de evaluar cuál es la combinación de fuentes en cada conjunto de integración que maximizan la contribución. En este sentido la variable y_i $\forall i \in I$ indica si una fuente i es dominante o no . Si el valor de y_i es 1 la fuente es dominante, de lo contrario es no dominante. Debido a que la optimización se hace para cada uno de los conjuntos de integración, al finalizar el proceso de optimización OptiSource define para cada conjunto de integración las fuentes de datos dominantes.

La restricción 6.7 del modelo de optimización es claramente una restricción clásica del problema de la mochila (Knapsack Problem)[KPP04], por lo cual puede inferirse que nuestro modelo de optimización es NP-completo. Sin embargo, el número de condiciones de la consulta esperada es en promedio de 3, por lo que en la práctica la búsqueda de la asignación óptima es viable.

6.6. Relación de OptiSource con el componente de Coordinación de la Ejecución

El componente de coordinación de ejecución es el que realmente evalúa la consulta entregando a las fuentes relacionadas la subconsulta que deben resolver. Teniendo en cuenta que OptiSource no elige todas las fuentes sino únicamente las dominantes en cada iteración, conocer cuáles de las fuentes dominantes respondieron a la consulta y cuál fue el aporte real de cada una de ellas puede ser de gran utilidad para renovar el conocimiento y para guiar la selección de las siguientes fuentes dominantes. Estas sección presenta en primer lugar la propuesta para llevar a cabo la coordinación de la ejecución en ARIBEC y luego cómo OptiSource utiliza un resumen de los resultados de ejecución para hacer más precisa su selección.

6.6.1. Coordinación de la Ejecución

Cuando cada una de las fuentes de datos identificadas en la cartografía es capaz de evaluar todas las condiciones de la consulta, el proceso de coordinación de ejecución es trivial, pues cada fuente entrega el conjunto de instancias que corresponden con la consulta. Sin embargo, cuando una condición debe ser evaluada en una fuente y otra condición en una fuente diferente, decidir cómo se realizará la ejecución es fundamental para asegurar un buen desempeño en el sistema. La propuesta para hacer esta coordinación sigue un modelo de colaboración basado en el espacio de tuplas LINDA [CG89] y se describe a continuación.

Cada conjunto de integración con su conjunto de fuentes dominantes tiene asociado un espacio de tuplas. Cada fuente de datos perteneciente a un conjunto es representada a través de un proceso *Ejecutor* cuya prioridad depende de la prioridad que fue identificada en la cartografía. Por ejemplo, en la cartografía presentada en la Figura 6.7 la fuente de datos 1 tendría un proceso con prioridad 0,6 para la condición c2 y 0,3 para la condición c1. Las fuentes de datos que posean la condición más selectiva de la consulta tendrán asociadas un proceso de tipo *Selector*. Adicionalmente, existe un proceso *Supervisor* cuya función se describirá más adelante.

La comunicación entre procesos se hace indirectamente usando el conjunto de primitivas para almacenar(*store*) y obtener (*retrieve*) tuplas del espacio de tuplas de cada conjunto de integración. La interacción con el espacio de tuplas está determinado por el tipo de proceso y por su prioridad. La lógica de la ejecución es la siguiente:

El(los) proceso(s) *Selector*(es) inicia la ejecución poniendo en el espacio de tuplas las instancias de VDO que ya ha validado. Los procesos *Ejecutor* toman tuplas disponibles que aún no han sido evaluadas, ni por ella ni por otra fuente, en la condición que la fuente puede validar, respetando las prioridades en el caso en que más de una fuente quiera obtener la misma tupla. Si la instancia de VDO cumple con la condición que evalúo la fuente, el proceso cambia la tupla para indicar que ya fue evaluada como positiva dicha condición y la deja de nuevo en

el espacio de tuplas. Si la condición de la instancia no pudo ser evaluada porque la fuente no contenía dicha instancia, y el proceso tiene prioridad 1 (autoridad), la tupla es borrada del espacio de tuplas por el proceso; en caso contrario, el proceso cambia la tupla para indicar que ya intentó evaluarla y la deja de nuevo en el espacio de tuplas. Si la condición de la instancia pudo ser evaluada pero el resultado fue negativo, la tupla es borrada del espacio de tuplas. El proceso supervisor toma periódicamente las tuplas que ya han sido evaluadas en todas las condiciones y las entrega al usuario final al mismo tiempo que las copia en un espacio de tuplas de respuestas. Si existen tuplas que no pudieron ser evaluadas completamente dentro del conjunto de integración, el proceso supervisor las toma y las añade al espacio de tuplas de otro conjunto de integración.

Una consulta termina cuando: (1) no hay tuplas en el espacio de tuplas de ningún conjunto o (2) el usuario determina que el número de tuplas recibidas es suficiente o (3) No hay procesos activos en el espacio de tuplas o (4) cuando el tiempo de espera del sistema expira.

Este sencillo modelo de ejecución asegura una capacidad de adaptación natural a estados inestables de los recursos físicos. Una baja tasa de respuesta de los procesos es soportada a través de la comunicación asincrónica usando los espacios de tuplas. Además, puesto que cada grupo puede evaluar una condición usando procesos diferentes, existe una capacidad de adaptación a las fallas de las fuentes durante la ejecución de la consulta. Del mismo modo, el intercambio de tuplas entre diferentes conjuntos de integración evita que se pierdan instancias que podían ser resueltas en otros conjuntos.

6.6.2. Uso de los resultados de la coordinación de ejecución

El proceso de coordinación deja todas las tuplas que se entregaron al usuario en un espacio de tuplas de respuesta, cada tupla contiene qué fuente la evalúo y para qué condición, y qué fuentes no pudieron evaluarla. La propuesta para mejorar la selección de fuentes dominantes es que en cada iteración OptiSource utilice el estado del proceso de integración para detectar cuáles de las fuentes autoridades identificadas en la cartografía efectivamente están participando en la ejecución. Si una fuente autoridad es participante, todas las fuentes que evalúan la misma condición deben ser descartadas del proceso de selección.

Adicionalmente, la información obtenida permite enriquecer la base de conocimiento con nuevas relaciones de tipo contenedor entre las fuentes participantes y los conceptos de dominio relacionados con las condiciones que evaluaron. Esta adición de hechos sólo puede darse si previamente no existía una relación entre la fuente y el concepto usando el rol contenedor u otro superior.

6.7. Identificación de Roles de Fuentes en OV

Analizar cada fuente de datos para identificar su contenido y de esta forma establecer cuáles son los roles que puede jugar con respecto a los VDOs, no es

una solución viable en OV con gran número de fuentes o gran volumen de datos. Si bien la definición de roles puede ser hecha manualmente por expertos de cada organización o por los administradores de las bases de datos, esta opción no es suficiente cuando el número de fuentes es alto o cuando el nivel de cooperación está limitado. Esta sección presenta tres propuestas para obtener los roles que desempeñan las fuentes de datos. La primera propuesta está basada en el análisis de procesos de negocio de la OV para identificar el mapa de replicación de la OV y de esta manera obtener fuentes que pueden tener roles de autoridad o especialista. La segunda propuesta, presenta un enfoque inductivo de análisis de fuentes, en donde a partir de la aplicación de técnicas de minería de datos sobre un subconjunto de fuentes es posible extrapolar reglas a otras fuentes de datos. Por ejemplo, los procesos de preparación de datos pueden ser generalizados en diferentes fuentes de la OV a partir de la experiencia en una fuente de datos modelo. La aplicación de técnicas de minería permite identificar fuentes con el rol de especialista. Finalmente, la tercera propuesta se basa en el análisis de las consultas previamente ejecutadas con el ánimo de identificar fuentes de datos con el rol contenedor o una tendencia alta a ser especialistas. A continuación se presentará cada una de ellas.

6.7.1. Interpretación de Procesos de Negocio

El algoritmo de interpretación de procesos de negocio tiene como hipótesis que las OV definen cuáles son los procesos de colaboración que deben ejecutarse para lograr cumplir los objetivos estratégicos que dieron inicio a la OV. Estos procesos son definidos usando una notación, como BPMN (Business Process Management Notation) [OMG09], y describen el conjunto ordenado de actividades que son ejecutadas por diferentes roles organizacionales y en los que están involucrados diferentes tipos de recursos. A partir del análisis de la ejecución de los procesos de colaboración se obtienen los roles de las organizaciones participantes. A continuación se describe la propuesta para documentar procesos en OV usando la notación BPMN y posteriormente se presenta el algoritmo para calcular el mapa de replicación del contexto de datos de la OV que permite obtener los roles.

Definición de Procesos de Negocio en OV

Los procesos de negocio en una OV son procesos de colaboración entre diferentes organizaciones que interactuan, ejecutan acciones y aportan recursos. Cada proceso establece los roles organizacionales requeridos, los recursos que debe aportar cada rol organizacional, las acciones que deben llevar a cabo y la interacción entre diferentes roles organizacionales.

Para describir estos procesos los elementos provistos por el estándar BPMN fueron tomados y extendidos para representar con mayor precisión las tareas de cada rol organizacional y la interacción entre organizaciones.

En resumen, la definición de procesos se realiza de la siguiente manera. Los roles organizacionales que participan en un proceso son definidos usando el ele-

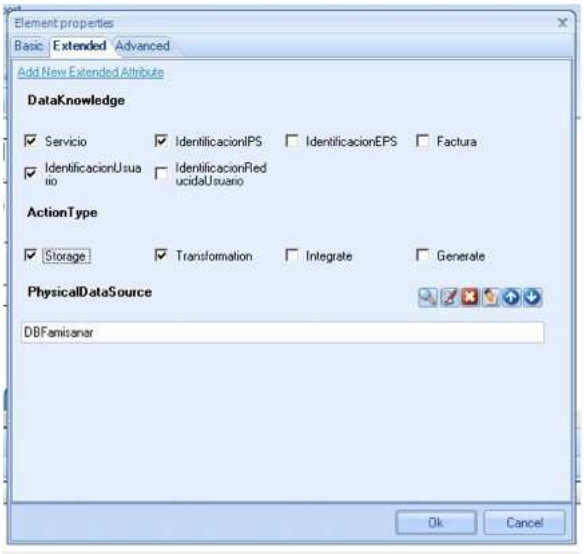

Figura 6.8: Atributos Extendidos

mento *Pool*, que son representados usando rectángulos. Las actividades que involucran entrega o recepción de datos son definidos usando el elemento *Task* de tipo *SendTask* o *ReceiveTask*. Ambos son representados gráficamente con rectángulos con esquinas redondeadas. Estos dos tipos de tareas siempre son asociados a un elemento *DataObject*, representado a través de una hoja. Como una extensión a BPMN fueron agregados tres atributos extendidos presentados en la Figura 6.8. El primero de ellos llamado *DataKnowledge* aplica a todos los elementos *Task* y *Data Object*. Los valores que puede tomar este atributo son los conceptos de dominio definidos en la OV. Para describir las acciones que las tareas realizan sobre los elementos de datos se propone utilizar un atributo llamado *ActionType* cuyo valor puede ser múltiple y representa si una tarea transforma, almacena, genera o integra datos. Finalmente, para poder asociar durante ejecución fuentes de datos reales a los elementos tipo *Data Object* que representan bases de datos, se adicionó un atributo extendido llamado *Physical Data Source*. La Figura 6.8 ilustra cómo estos nuevos elementos fueron adicionados a un software de descripción de procesos llamado *BizAgi Process Modeler* [Biz] para representar de forma más detallada las tareas y los objetos de datos.

La Figura 6.9 presenta un ejemplo de proceso de colaboración para una OV en el sector salud cuyo objetivo de colaboración es compartir información de pacientes. En este proceso se representan tres tipos de organizaciones, que realizan tareas sobre dos repositorios temporales y cuatro fuentes de datos compartidas por la OV.

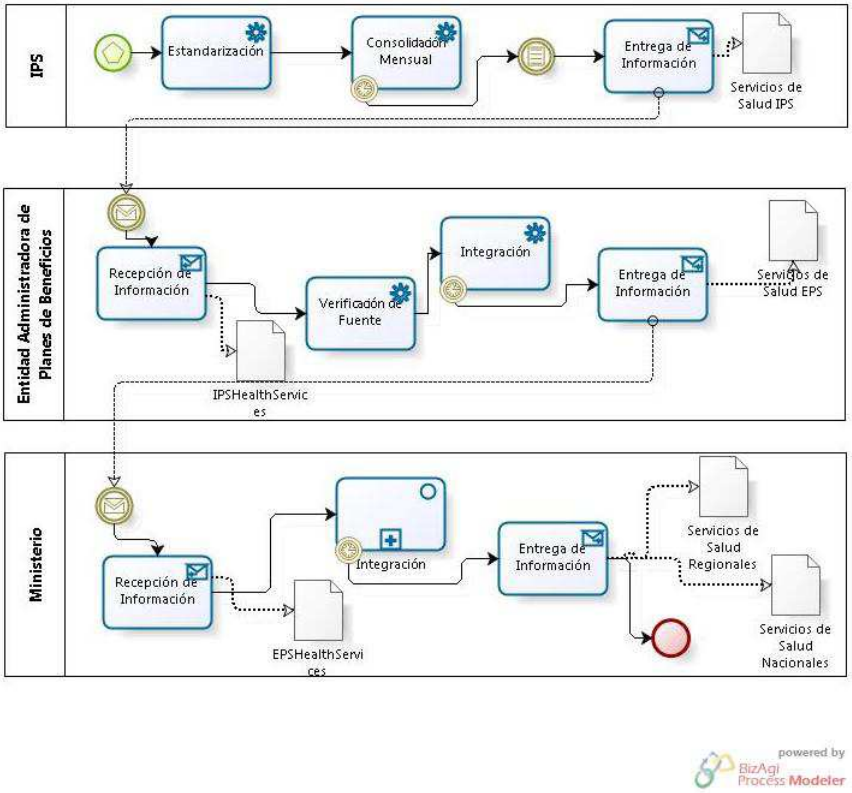

Figura 6.9: Proceso de Negocio para una OV en Salud

Análisis de Procesos de Negocio

El análisis sobre los procesos de negocio busca identificar el mapa de replicación de datos de la OV. El mapa presenta a la OV como un conjunto de fuentes de datos conectadas a través de vínculos que describen la relación que las fuentes tienen entre sí. La Figura 6.10 presenta un ejemplo de mapa de replicación en donde cuatro fuentes de datos interactuan. La existencia de un enlace entre dos puntos refleja una relación entre las instancias de conceptos del dominio contenidos en las fuentes de datos de los extremos. El punto donde inicia el conector es considerado el origen. El punto donde finaliza el conector es considerado el destino. La descripción del conector está compuesta por el listado de conceptos de dominio que fluyen de una fuente a otra y las condiciones que deben cumplirse para que este flujo se haga efectivo. Por ejemplo, entre la fuente 4 y 2 fluyen sólo los registros cuando el diagnóstico es una enfermedad infecto-contagiosa. Entre dos puntos puede existir más de un conector, reflejando así la existencia de diferentes escenarios de intercambio de datos.

Cada punto tiene ligado el conjunto de conceptos de dominio que es alma-

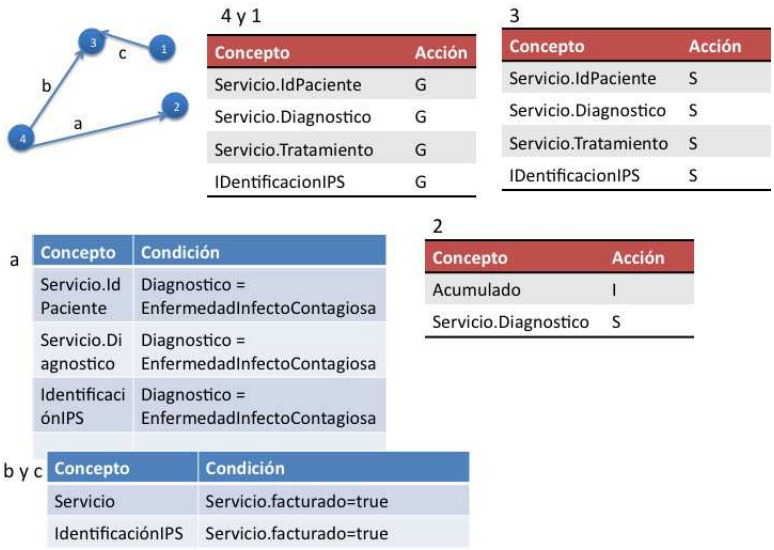

Figura 6.10: Mapa de Replicación para la OV en Salud

cenado en la fuente de datos. Cada concepto almacenado tiene asociado una de las siguientes opciones:
- Si el valor del concepto es creado por primera vez en esta fuente de datos. (G)
- Si el valor del concepto es generado por primera vez en la fuente de datos a partir de la integración de valores recibidos de otras fuentes (I)
- Si el valor del concepto es proveniente de otra fuente y permanece almacenado sin sufrir modificaciones (S)
- Si el valor del concepto proviene de otra fuente pero luego es transformado (siempre o bajo ciertas condiciones) (T)

El Algoritmo 3 permite interpretar los procesos de negocio y generar el mapa de replicación. Además de las ventajas visibles de conocer el mapa de replicación como saber cuáles fuentes pueden ser reemplazadas por otras debido a su replicación, el mapa de replicación también divulga qué fuentes son especialistas y autoridades. El rol Autoridad de una fuente DS_i se determina si todas las fuentes tienen un enlace a una misma fuente usando una misma condición. Este rol aplica únicamente a las instancias con las características descritas en la condición. Por ejemplo, en la Figura 6.10 la fuente 2 recibe las instancias de pacientes con diagnóstico de enfermedades infecto-contagiosas, si existieran otras fuentes generadoras y tuvieran este mismo enlace con la fuente 2, la fuente 2 se consideraría Autoridad para pacientes con enfermedades infecto-contagiosas. De forma similar, El rol Especialista se asigna a las fuentes que reciben enlaces de otras fuentes, pero no necesariamente de todas, usando una misma condición.

Algoritmo 3 Algoritmo para ObtenciOn del Mapa de replicaciOn

```
Entrada: Proceso de Negocio p
Salida: Mapa de replicaciOn
Inicio
 Mapa=null
 j=0
 Mientras p.existTask() <> [] //Mientras hayan tareas en el
                                                          proceso
    tarea = p.getTask(j)
    Si tarea.getdataResource() <> [] //Si la tarea tiene asociados
                                     recursos de datos
      mapa.addResource(tarea.getDataResource(),
                                         tarea.getDataKnowledge())
    Fin_Si
    Si tarea.type =SendTask // Si la tarea fue definida como
                                                          SendTask
      i=0
      Mientras tarea.ExistAssociations() //Mientras la tarea tenga
                                  asociaciones con otras tareas
        asociacion = tarea.selectAssociacion(i)
        destino = asociacion.selectTarget()
        contenido = tarea.getDataKnowledge()
        Si (asociacion.hasConditionalEvent) //Si existe un evento
                              condicional ligado a la asociaciOn
          evento = asociacion.getEvent()
          idConector =
                 mapa.addConnector(tarea,destino,contenido,evento)
        Si_No
          idConector = mapa.addConnector(tarea,destino,contenido)
        Fin_Si
        Si (destino.getDataResource() <> []) // Si la tarea asociada
           como target de la  asociaciOn tiene por lo menos un
           recurso de datos asociado
          tipoAccion = target.getActionType()
          mapa.updateResource(destino.getDataResource(),contenido,
                                                          tipoAccion)
        Fin_Si
        i++
       Fin_Mientras
      Fin_Si
      j++
   Fin_Mientras
   Return mapa
Fin
```

6.7.2. Identificación de Roles Usando Minería de Datos

Aplicar técnicas de minería de datos sobre cada una de las fuentes para obtener sus roles con respecto a los conceptos de dominio no es viable dado el contexto de gran escala de las OV. En este tipo de contextos no es posible seguir las metodologías de minería de datos que requieren de la participación de especialistas del dominio, al igual que de expertos en procesos de extracción de conocimiento específicos a cada una de las fuentes. Para superar estos retos se propone utilizar un enfoque inductivo que consiste en dividir las fuentes por grupos de acuerdo a los roles de las organizaciones que las proveen. Por cada segmento se elige una fuente representante sobre la cual se aplica una técnica de minería descriptiva [HMS01]. El proceso de minería realizado en cada representante es caracterizado y generalizado de modo que éste pueda ser replicado a otras fuentes del mismo segmento reduciendo la intervención humana.

El objetivo de la segmentación inicial es identificar grupos de fuentes que pueden ser diferenciadas mediante un conjunto de características comunes. La segmentación se realiza de acuerdo al criterio dado por expertos del dominio y a los requerimientos de consulta de la OV; por ejemplo, en una OV de salud en Colombia, el resultado de una segmentación podría establecer como grupos las Entidades Gubernamentales, las Entidades Promotoras del Servicio de Salud (EPS) y las Instituciones Prestadoras del Servicio de Salud (IPS). La selección de las fuentes representantes de cada segmento puede hacerse de manera aleatoria o buscando las fuentes que cumplan con ciertos parámetros de tamaño o de disponibilidad. Las técnicas de minería que pueden ser aplicadas pueden variar de acuerdo al grupo. Las técnicas de segmentación *(clustering)* o de reglas de asociación permiten perfilar el tipo de información contenida en cada fuente. Por ejemplo, para un grupo de fuentes en el que se encuentran la historia clínica de pacientes, la aplicación de la técnica de segmentación permite distinguir cada uno de los posibles grupos de historias clínicas existentes.

Finalmente, se realiza la conceptualización con el fin de determinar cómo puede ser replicada la generación de conocimiento de forma automática en otras fuentes. Esto equivale a formalizar de manera explícita cómo se llevaron a cabo los procesos de limpieza de datos, de transformación y de aplicación de las técnicas. La conceptualización es utilizada como un insumo para la construcción de un sistema que automatice el proceso [AVAdPV10].

6.7.3. Interpretación del Procesamiento de Consultas

Considerando una consulta Q enviada a ARIBEC y suponiendo que no hay conocimiento de los roles que pueden jugar las fuentes con respecto al tipo de VDO definido en la consulta Q, la planeación de la consulta utilizará únicamente la información intencional que se tenga de las fuentes. En este escenario todas las fuentes que sean capaces de evaluar las propiedades restringidas en la consulta serán tenidas en cuenta y sólo serán seleccionadas en la primera iteración aquellas que pueden evaluar la mayor cantidad de condiciones del predicado. La ejecución de la consulta obtendrá las instancias que coinciden con Q crea-

das a partir de los datos que aportaron todas o parte de las fuentes de datos seleccionadas. El principio de la interpretación del procesamiento de consultas es registrar durante el proceso de ejecución de consultas las fuentes que recibieron subconsultas y de éstas, aquellas que entregaron respuestas al sistema. El análisis de estas ejecuciones previas no sólo permite conocer qué fuentes tienen rol contenedor con respecto a un tipo de VDO, sino también qué fuentes tienen mayor tendencia a ser especialistas en cierto tipo de VDO.

6.8. Síntesis

Este capítulo presenta OptiSource una estrategia de selección de fuentes de datos creada para OVs de gran escala en las que las fuentes de datos están solapadas intencionalmente y extensionalmente, en donde no hay certeza acerca de la localización de los datos y en donde pueden presentarse escenarios de replicación entre fuentes. El principio de OptiSource es evitar la consulta de fuentes que no aportarán instancias que coincidan con la consulta seleccionando únicamente las fuentes de datos dominantes para cada una de las condiciones de la consulta. Una fuente de datos es dominante si su contribución en términos de instancias es mayor que la de las otras fuentes de datos que pueden evaluar la misma condición.

Para identificar las fuentes dominantes con respecto a una consulta OptiSource actúa en dos fases. La primera prioriza las fuentes de acuerdo al rol que pueden jugar dentro de la consulta, y la segunda optimiza la asignación de condiciones a las fuentes de tal forma que se maximice la contribución al integrar las respuestas de múltiples fuentes. De esta manera OptiSource evita consultar fuentes que no son relevantes para la consulta y previene la integración de respuestas de las fuentes que no tienen instancias en común.

Capítulo 7

Selección de Fuentes MultiEscala para Organizaciones Virtuales

El Capítulo 5 ilustra las principales decisiones de diseño de ARIBEC, una arquitectura de mediación creada para OV. Como se mencionó, el nivel de mediación de ARIBEC incluye un proceso de selección que busca reducir las fuentes que participan durante la ejecución. El principio de reducción de fuentes varía de acuerdo a la estrategia de selección elegida, la estrategia OptiSource presentada en el Capítulo 6 es una de ellas. Este capítulo presenta la manera a través de la cual ARIBEC elige dinámicamente la estrategia de selección más apropiada de acuerdo a la relación entre el contexto de datos y la consulta que se va a ejecutar. El objetivo es favorecer la eficiencia en contextos complejos, eligiendo estrategias cuyo aumento en precisión sea considerable en relación al aumento en costo de planeación, así como también en contextos simples, evitando algoritmos de selección exhaustivos en contextos en donde el número de fuentes seleccionadas durante la planeación es similar al total de fuentes identificadas analizando únicamente su esquema.

La organización del capítulo es la siguiente. La Sección 7.1 ilustra la problemática de tener una única estrategia para seleccionar las fuentes de datos que participarán en la ejecución de una consulta. Posteriormente, la Sección 7.2 presenta la abstracción propuesta del contexto de datos de las OV a través de una representación llamada el perfil de datos que permite definir el nivel de detalle en el que se desea conocer el contexto. Así mismo, esta sección ilustra cuál es el principio del procesamiento multiescala de consultas. La Sección 7.3 describe la manera como las estrategias de selección son clasificadas de acuerdo al contexto de datos. La Sección 7.4 ilustra el algoritmo utilizado para elegir dinámicamente la estrategia. Finalmente, la Sección 7.5 concluye el capítulo.

7.1. Problemática: Desequilibrio entre el Tiempo de Planeación y el Beneficio Obtenido

Las estrategias de selección de fuentes mencionadas en el Capítulo 4, representan importantes avances en el procesamiento de consultas en contextos de datos distribuidos. Los sistemas que las implementan las utilizan durante la planeación para reducir el número de fuentes consultadas. Sin embargo, el hecho de tener disponible sólo una estrategia en un sistema de mediación para OV puede introducir desequilibrio entre el tiempo de planeación y el beneficio realmente obtenido (e.g. Alto esfuerzo-baja precisión).

Por ejemplo, en los casos en donde se va a evaluar una consulta cuyas condiciones sólo pueden ser evaluadas en un conjunto pequeño de fuentes de todas las disponibles no se justifica el uso de una estrategia de selección que analice en detalle el contenido de las fuentes, pues el tiempo de planeación puede ser mayor al tiempo de consultar a todas las fuentes involucradas intencionalmente (a nivel de esquema). Contrariamente, la evaluación de consultas que involucran gran cantidad de fuentes solapadas, en donde las propiedades están distribuidas en múltiples fuentes requiere estrategias de selección más exhaustivas que analicen la relevancia extensional de las fuentes y que eviten el sobre-costo de evaluar la consulta en fuentes irrelevantes. En general, los sistemas de mediación de las OV necesitan mecanismos que permitan adaptar la estrategia de selección de fuentes de acuerdo al contexto de datos y a la consulta que se va a evaluar. Usar una estrategia inapropiada en los sistemas de mediación de las OV y en general de los sistemas de datos distribuidos puede generar escenarios no deseados como se presenta a continuación.

-Alto esfuerzo-baja ganancia: este escenario se puede dar por tres razones. (1) usar una estrategia compleja en contextos con pocas fuentes, donde el costo (por ejemplo en tiempo) de consultar el 100 % de las fuentes es menor al costo de planear exhaustivamente. (2) aplicar una estrategia compleja para resolver consultas que usando estrategias simples, como filtrado por esquema, obtienen una ganancia similar. (3) aplicar una estrategia compleja en contextos donde la distribución de las instancias que resuelven la consulta es tan aleatorio que para obtener el número de respuestas requerido es necesario consultar un número de fuentes igual o similar al número de fuentes que pueden resolver la consulta a nivel intencional.

-Bajo esfuerzo-baja ganancia: se da al aplicar estrategias que reducen el número de fuentes de acuerdo a un conjunto de características que tienen valores similares para la mayoría de fuentes, haciendo que la reducción de fuentes sea mínima. Por ejemplo, si en un contexto todas las fuentes tienen un esquema similar y la estrategia de selección sólo utiliza el esquema para decidir quien debe o no participar en la consulta, la reducción será insignificante frente al número total de fuentes. A pesar de que el esfuerzo de planeación en la mayoría de estas estrategias es bajo, la reducción de fuentes de datos es mínimo, generando ineficiencia durante la ejecución.

-Incompatibilidad estrategia-contexto: Se da al aplicar una estrategia a un

contexto donde el tipo de datos o los requerimientos de usuario son incompatibles. Por ejemplo, aplicar una estrategia orientada a capacidades (ver Sección 4.2) cuando todas las fuentes de datos tienen las mismas limitaciones al ejecutar el tipo de consulta que se va a evaluar. En este caso, la estrategia orientada a capacidades no filtraría las fuentes pues no le es posible diferenciar las fuentes.

Escenarios ideales como bajo esfuerzo-alta utilidad, alto esfuerzo-alta utilidad y compatibilidad estrategia-contexto, sólo pueden ser alcanzados si, previo a la aplicación de una estrategia de selección de fuentes, se evalúa la relación de la consulta con el contexto de datos, y de acuerdo a esto se elige la estrategia de selección adecuada. El sentido multiescala de ARIBEC hace referencia a su capacidad de elegir dinámicamente la estrategia de selección adecuada de acuerdo a la relación del contexto con la consulta. Esta capacidad está diseñada para hacer parte del nivel de mediación dentro del componente de Gestión de Consultas, que a partir del análisis del contexto y la consulta puede eligir la estrategia más indicada de todas las disponibles.

Las siguientes secciones presentan la manera como ARIBEC obtiene la información del contexto y cómo la utiliza para elegir la estrategia.

7.2. Contexto de Datos de la OV

El contexto de datos de las OVs involucra los diferentes recursos que influyen durante la evaluación de una consulta. Este contexto está compuesto por los recursos físicos y lógicos que se utilizan desde que se recibe una consulta hasta que se retorna la respuesta a la consulta al usuario. A nivel lógico el principal recurso son las fuentes de datos y a nivel físico los recursos de cómputo y de almacenamiento que se tienen disponibles para apoyar la evaluación de consultas, por ejemplo para almacenar los metadatos; así como también las redes de comunicación existentes que permiten tener acceso a todos los recursos. Como se mencionó en el Capítulo 5 la información de los recursos de datos, de almacenamiento y de cómputo queda registrada en la base de conocimiento de la OV. Sin embargo, los cambios que sufren estos recursos en cuanto a disponibilidad, velocidad de respuesta y capacidad de procesamiento o almacenamiento varían en el tiempo, por lo cual obtener su valor requiere una interacción directa con la infraestructura física que da soporte a la OV. Así mismo, el estado de las redes de comunicación es variable en el tiempo y obtenerlo requiere consultar permanentemente el sistema de información de la infraestructura física de la OV. Estas características variables del contexto sumadas al volumen de recursos con los que puede contar una OV hacen que abstraer toda la información del contexto de datos de la OV para usarla durante el proceso de elección de la estrategia de selección no sea una labor trivial. Por esta razón, se propone el uso de una representación del contexto a través de una abstracción llamada el *Perfil de Datos*. El objetivo del perfil de datos es representar el contexto a través de un conjunto de medidas que pueden ser obtenidas en diferentes niveles de detalle o usando diferentes perspectivas del contexto. La configuración adecuada del perfil evita la obtención de la información de todo el contexto cada vez que se

va a evaluar una consulta, que puede ser costoso en tiempo y en procesamiento, y en la mayoría de los casos innecesario, pues en ciertas ocasiones basta con conocer una sola propiedad del contexto para hacer una buena elección.

7.2.1. Perfil de Datos: Una Abstracción del Contexto de Datos de la OV

El perfil de datos se puede visualizar utilizando tres elementos llamados operadores: Lente, Filtro y Zoom. El Lente es el operador básico que permite capturar los valores de un conjunto de atributos del contexto durante un periodo de tiempo determinado, el filtro determina la zona del contexto de datos que se analizará y el zoom define el nivel de detalle con el que se verá el contexto.

La definición de cada operador es el resultado del análisis de diferentes estrategias de procesamiento de consultas en los sistemas distribuidos (véase el Capítulo 4). Los niveles de detalle de información fueron caracterizados de acuerdo con los metadatos que requieren las estrategias y con los atributos que permiten discriminar mejor una estrategia de otra. Por ejemplo, las estrategias se pueden diferenciar en el número de fuentes de datos que son capaces de manejar, o el tipo de fuentes de datos que soportan, la capacidad (o no) para planear cuando hay fuentes dependientes o la capacidad (o no) para identificar la calidad de las fuentes de datos. A continuación se describen los dispositivos propuestos que contribuyen a la definición del nivel de detalle del perfil de datos.

Lentes, este operador permite capturar un conjunto de medidas generales del contexto de datos como: las fuentes de datos disponibles, la carga de trabajo de los recursos de cómputo, etc. Una lista de estas medidas se ilustran en la Figura 7.1, estos atributos pueden extenderse de acuerdo a la necesidad y al tipo de OV. Los lentes son capaces de capturar la información del contexto de datos durante un período de tiempo y luego proporcionar el valor de sus atributos. Existen dos tipos de lentes que se diferencian por la restricción de tiempo que tienen para capturar el valor de las medidas. El **lente no comprehensivo** puede ver el contexto por un período de tiempo establecido después del cual debe entregar el valor obtenido de cada una de las medidas. De acuerdo a la limitación de tiempo la información entregada por este lente corresponde más a una muestra del contexto que a una visión completa de él. El **lente comprehensivo** puede ver el contexto durante un periodo "indeterminado" de tiempo [1] que le permite obtener una visión completa (no necesariamente coherente por el tiempo transcurrido) de las medidas que debe proporcionar. La información obtenida usando este operador permite obtener una visión completa o casi completa del contexto; sin embargo, el tiempo que utiliza puede ser muy costoso en la mayoría de los casos.
En ambos tipos de lentes la expresión ver el contexto significa en términos reales solicitar la información del contexto a la base de conocimiento o al sistema de

[1] Para asegurar la viabilidad del lente comprehensivo es necesario utilizar un tiempo de espera definido en una escala diferente al tiempo definido para los lentes no comprehensivos

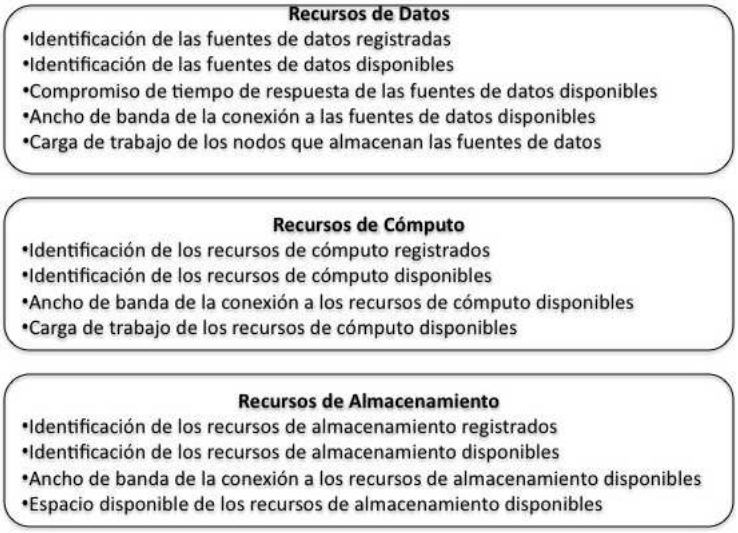

Figura 7.1: Visibilidad de los Lentes

información de la infraestructura física sobre la que esté implementada la OV.

Filtros controlan el conjunto de fuentes de datos visibles durante la captura del perfil de datos. Los posibles filtros existentes son: **Filtro de Fuente de Datos Única** que restringe la captura de información de sólo una fuente de datos declarando su identificación; y **filtro de múltiples fuentes de datos** que restringe el grupo de fuentes de datos de acuerdo a una condición de la forma $caracterstica_1 OP valor_i, ..., caracterstica_n OP valor_k$ donde $caracterstica_j$ es una característica simple o compleja de las fuentes de datos, OP es un operador $=, <=, >=, !=$, y $valor_m$ puede ser un valor un numérico o un booleano. Características simples son: disponibilidad y tiempo de respuesta. Características complejas son: concordancia intencional con respecto a una consulta, que es un booleano que determina si el esquema de una fuente de datos es capaz o no de resolver una consulta completamente o parcialmente; y concordancia extensional con respecto a una consulta, que es un booleano que determina si una fuente de datos es capaz o no de resolver una consulta de acuerdo a su contenido. Un ejemplo de condición de un filtro es:

$$Disponibilidad = true, ConcordanciaIntencional(VDOquery) = true.$$

Zooms, permiten profundizar en el conocimiento de las fuentes de datos. Los zooms son los dispositivos más ricos, debido a su habilidad de ver las fuentes de datos como cajas blancas en términos de su estructura, contenido y restricciones. Los zooms propuestos agrupan las características de las fuentes de datos de acuerdo a sus características a nivel intencional, a nivel de duplicación y a nivel extensional.

- Zoom Intencional captura metadatos detallados del esquema de las fuentes de datos. Dado un VDO, este zoom presenta el identificador de cada fuente y las propiedades del VDO que es capaz de resolver. Adicionalmente, este zoom presenta las posibles restricciones que tienen las fuentes para consultar las propiedades.
- Zoom de Duplicación este zoom captura la manera a través de la cual se intercambia información entre las fuentes. Este zoom describe los fenómenos de duplicación determinando qué fuente es generadora, que fuente es receptora y qué fuente es concentradora. Una fuente es considerada generadora si es posible comprobar que la información que contiene es creada por primera vez en ella. Es receptora, cuando es posible comprobar que recibe información de otra fuente. Es concentradora si es receptora de más de una fuente.
- Zoom Extensional este zoom captura el conocimiento en términos del tipo de instancias que contiene y la calidad de estas instancias. La caracterización presentada por este zoom determina qué fuentes contienen qué tipo de instancias y bajo que parámetros de calidad. Para describir el tipo de instancias que contienen se utiliza el mismo concepto de roles presentado en el Capítulo 6. En este caso una fuente puede ser definida como Autoridad, Especialista o Contenedora de un tipo de instancia. Así mismo, para describir la calidad de la fuente con respecto a la consulta se utiliza la función de densidad utilizada en [NLF99, NFL04] que corresponde al promedio de la densidad de los atributos que están involucrados en la consulta. Un atributo tiene densidad del 100 % si tiene valor y de 100 % si su valor es vacío o nulo.

7.2.2. Configuración para Obtener el Contexto

El proceso de obtención del contexto de datos involucra un proceso de configuración inicial utilizando el perfil de datos. Este proceso define qué lente se va a utilizar y opcionalmente qué zooms o qué filtro se va a emplear. Una configuración es representada por una tupla *(l,f,Z)* donde l es el lente elegido, f es el filtro elegido y Z es el conjunto de zooms. l no puede ser vacío, y si Z no es vacío, f no puede ser vacío. Un ejemplo de configuración es la siguiente:

$$(l^{NC}_{100ms}, f^{multifuente}_{disponibilidad=true}, z_{intencional})$$

l^{NC}_{100ms} representa un lente no comprehensivo de 100ms. $f^{multifuente}_{disponibilidad=true}$ representa un filtro que sólo ve las fuentes de datos disponibles. $z_{intencional}$ es un zoom intencional. Cuando la configuración es ejecutada, se obtendrán las medidas del contexto de datos visibles para estos elementos.

Si la captura de información de un contexto de datos revela un posible escenario complejo, éste puede ser detallado incorporando nuevos zooms. La Figura 7.2 ilustra los operadores del perfil de datos y cómo cada uno de ellos es utilizado dentro de una clase que configura la manera a través de la cual se obtendrá el contexto de datos.

El método *capturarContexto* de las clases que implementan la interfaz *Lente*, el método *definirEnfoque* de las clases que implementan la interfaz *Zoom* y el

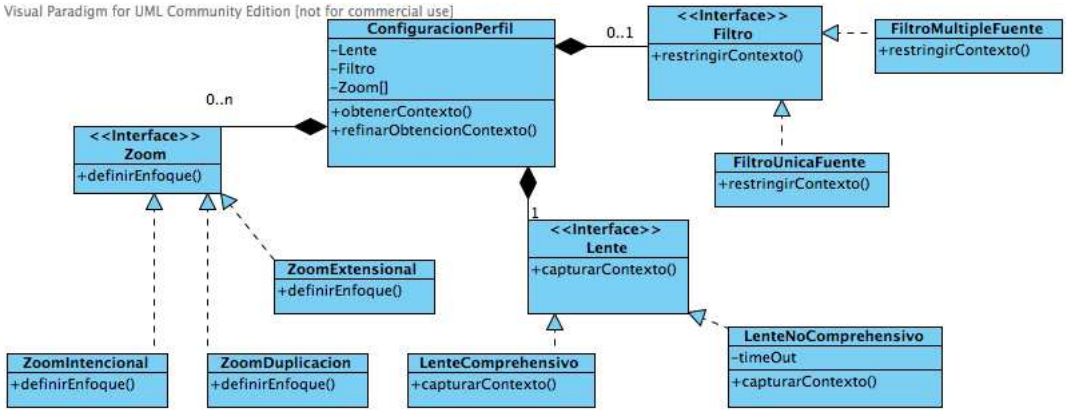

Figura 7.2: Configuración del Perfil de Datos

método *restringirContexto* de las clases que implementan la interfaz *Filtro*, son implementados de acuerdo al tipo de elemento elegido. Por ejemplo, el método *definirEnfoque* de la clase *ZoomIntencional*, solicita la información del esquema de las fuentes a la base de conocimiento de la OV y la compara con la consulta para identificar qué fuentes están asociadas a la consulta y en qué porcentaje.

El método *obtenerContexto* de la clase *ConfiguracionPerfil* obtiene el perfil de datos de acuerdo a la configuración definida, es decir a los operadores elegidos. Para profundizar en el contexto una vez que ya se ha hecho una captura, se utiliza el método *refinarObtenciónContexto*, que utiliza un nuevo zoom para observar más en detalle el contexto. El zoom menos detallado es el intencional, luego se encuentra el zoom extensional y posteriormente, el zoom de duplicación. La manera como fue implementado el componente que selecciona dinámicamente la estrategia de selección será presentado en el Capítulo 8.

7.2.3. Atributos Derivados

Las medidas obtenidas a partir de una configuración del perfil de datos pueden ser enriquecidas a través de la aplicación de funciones matemáticas, a través de la combinación entre distintas medidas o a través de su discretización. El hecho de enriquecerlas hace que la información que provean sea más significativa durante el proceso de elección. Por ejemplo, es más relevante conocer cuál es la propiedad más distribuida entre las fuentes que saber cuál propiedad está en cada fuente. El resultado de enriquecer las medidas es llamado un **Atributo Derivado** y es utilizado para hacer una mejor interpretación del perfil de datos. Algunos ejemplos de este tipo de atributos son:
- Fuentes de datos completas: Dado un conjunto de propiedades de VDO, una fuente de datos es completa si conoce intencionalmente todas estas propiedades.
- Contexto restringido por capacidades: El valor de este atributo es verdadero si al menos una fuente de datos tiene restricciones de ejecución de consultas.

- Número de fuentes de datos activas: El valor de esta variable discretiza la sumatoria de fuentes de datos disponibles en tres valores Alto, Medio, Bajo. Donde Alto significa que existen 200 o más fuentes, Medio significa que hay entre 26 y 199 fuentes y bajo significa que hay entre 1 y 25 fuentes.
- Concentración de instancias de VDO: Teniendo en cuenta el número de receptores y concentradores obtenidos a través del zoom de duplicación, el conocimiento intencional obtenido a través del zoom intencional y el número de fuentes activas, este atributo derivado define el número mínimo de fuentes de datos que deben ser consultadas para obtener todas las respuestas disponibles.
- Porcentaje de fuentes con solapamiento extensional: Utilizando los atributos provistos por el zoom extensional y de duplicación, este atributo calcula cuántas fuentes tienen solapamiento.
- Porcentaje de fuentes con solapamiento intencional: Utilizando los atributos provistos por el zoom intencional este atributo calcula cuántas fuentes tienen solapamiento en el esquema.

7.2.4. Patrones del Perfil de Datos

Un patrón de perfil de datos identifica un grupo de perfiles de datos capturados que tienen en común los valores para un conjunto de atributos. Un patrón puede involucrar atributos básicos o atributos derivados. Cada patrón se identifica con un nombre, un conjunto de atributos y un valor para cada atributo. El valor que toma un atributo puede ser un valor exacto o un conjunto de valores. La Figura 7.3 ilustra un conjunto de patrones detectados en los contextos de datos de las OVs.

El patrón *OVCompleja* por ejemplo, describe todos los contextos que tienen un alto número de fuentes (200 o más), con alto volumen de datos (más de 5 mil registros para bases de datos relacionales) y distribuidos en una red de área amplia, y que adicionalmente, tienen niveles altos de solapamiento intencional y extensional entre fuentes y alta distribución de instancias. El contexto de datos de una OV cumple con un patrón si los valores de las medidas de su perfil de datos se encuentran dentro de los rangos establecidos por un patrón. Adicionalmente, los patrones son útiles para representar de forma concisa los estados a través de los cuales la OV evoluciona en el tiempo, lo que puede dar insumos para conocer qué tipo de estrategia debe estar disponible en la OV. En este caso, cada captura del perfil debe ser almacenada para poder evaluar dicha evolución.

7.2.5. Principio del Proceso de Selección de Fuentes Multiescala

El principio de la elección multiescala de la estrategia de selección de fuentes de datos es el uso de patrones de perfil de datos para reconocer cuál es la mejor estrategia. Los patrones son utilizados con dos propósitos:

Figura 7.3: Patrones de Perfil de Datos

1. Asociar las estrategias de selección de fuentes de datos disponibles al patrón que mejor describa el contexto al cuál están dirigidas.

2. Identificar los patrones más cercanos al perfil de datos obtenido de una OV en un lapso de tiempo.

En términos generales la característica multiescala del proceso de selección se lleva a cabo en dos fases. Una fase inicial de alimentación en donde se asocian todas las estrategias de selección disponibles al patrón de perfil de datos en el cual tienen un mejor desempeño. Y una fase de elección, en donde se obtiene el perfil de datos de la OV cuando llega una consulta. El perfil obtenido es comparado con los patrones existentes con el fin de identificar cuál es el patrón más cercano al perfil. De las estrategias asociadas al patrón más cercano es elegida la estrategia menos costosa en tiempo de planeación. Las siguientes dos secciones ilustran estas fases.

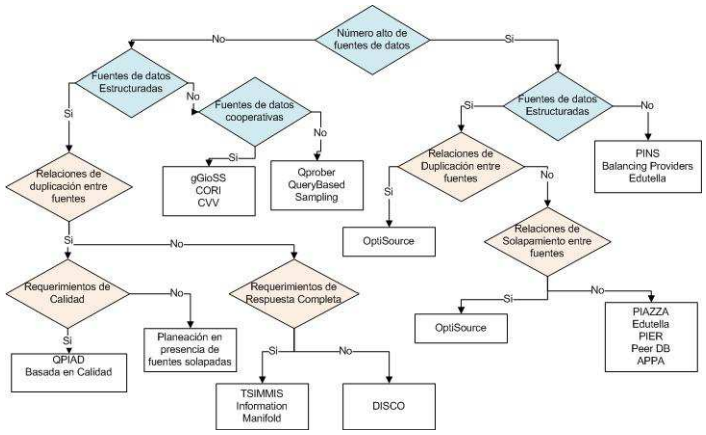

Figura 7.4: Árbol de Decisión de las Estrategias de Mediación

7.3. Clasificación de Estrategias de Acuerdo a los Patrones del Perfil de Datos

Saber cuál es la estrategia indicada para un contexto de datos determinado requiere en primer lugar hacer un análisis de para qué tipo de contexto fueron creadas las estrategias actualmente existentes. La Figura 7.4 ilustra un árbol de decisión, que relaciona las estrategias de selección de fuentes descritas en el Capítulo 4 con las dimensiones asociadas al contexto y a la consulta. La figura ilustra en azul las variables puras del contexto y en rosa las que dependen de la relación de la consulta con el contexto.

La vinculación de una estrategia a uno o más patrones de perfil de datos consiste en tomar cada punto de decisión del árbol como una variable y comparar los valores que puede tomar con los atributos presentes en los patrones. Cada vez que haya una coincidencia entre una variable o un conjunto de variables de una rama del árbol y un patrón se asocian las estrategias que se encuentran en la rama con el patrón. Por ejemplo, todas las estrategias que están en el lado izquierdo de la variable de decisión *Número alto de fuentes de datos*, se asocian al patrón *OVPequeñaEscala*; de manera contraria, todos los que están en el lado derecho se asocian al patrón *OVGranEscala*.

Así mismo, siguiendo el camino del árbol *Número alto de fuentes de datos = No y Fuentes de datos estructuradas = Si y Fuentes de datos cooperativas = Si*, las estrategias GGioss, CORI y CVV quedan asociadas al patrón *OVPequeñaEstructuradosCooperativas*.

Al finalizar el recorrido por el árbol de decisión se habrán vinculados las estrategias de selección a uno o más patrones de perfil de datos.

7.4. Algoritmo de Elección de la Estrategia de Selección de Fuentes de Datos

La lógica del componente de selección es presentado en el algoritmo 4. El fundamento está en que cada vez que es obtenido el perfil de datos de la OV (1), es posible calcular el patrón más cercano a él. La cercanía se calcula usando una función de distancia entre los valores de las variables del perfil y las variables de los patrones disponibles (2). Si el perfil obtenido está asociado a múltiples patrones (3), se elije el patrón más preciso, es decir el que tenga asociados más número de atributos. Si incluso con esto no es posible diferenciar, es necesario obtener un perfil más detallado que permita discriminar el tipo de contexto por el que transcurre la OV, en cuyo caso el componente Capturador solicita una vista más detallada de la OV hasta encontrar un único patrón o hasta que se venza el tiempo máximo permitido para planear(4). Por otra parte, si existe más de una estrategia asociada al patrón más cercano, se elige la estrategia menos costosa, determinada por el tiempo de respuesta en ejecuciones pasadas(5).

Luego de numerizar los atributos discretos del perfil la función de distancia utilizada para obtener el patrón más cercano se calcula como se presenta en la Definición 12.

Definición 12 ***Función de Distancia.*** *Considere el conjunto de atributos P que se capturan del perfil de datos en el nivel de detalle l $P_l = (A1, ..., An)$ con atributos Ai de tipo real. Entonces, dada una captura del perfil $p = (a1, ..., an)$ y un patrón $t = (t1, ..., tm)$, los atributos tj que no pertenecen a P_l son descartados, produciendo $t' = (t1, ..., tm)$ donde $t'j \in P, \forall t'j$ y los atributos que se encuentran en p y no en t' son descartados produciendo $p' = (a1, ..., an)$ donde $a'i \in t', \forall a'i$, se define la distancia entre p' y t' usando una de las siguientes funciones de distancia.*

$$Sum(a, t) = \sum_{i=1}^{n} |ai - ti|. \tag{7.1}$$

$$Eucl(a, t) = \sqrt{\sum_{i=1}^{n} \langle ai - ti \rangle^2}. \tag{7.2}$$

El método multiescala hace parte del nivel de mediación de ARIBEC y permite elegir la mejor estrategia de las disponibles en la OV. Para que la evaluación de consultas sea eficiente se recomienda tener una estrategia orientada a capacidades que por su simplicidad es liviana dentro del proceso de planeación, y una estrategia orientada a conocimiento que permite reducir considerablemente las fuentes cuando tienen esquemas similares.

Algoritmo 4 Algoritmo de Selección de Estrategia

```
Entrada:  query - consulta a resolver
        MAXTime - máximo tiempo para seleccionar estrategia
Salida: strategy - Estrategia más apropiada de acuerdo al contexto
                                                        -consulta
Inicio
    int level = 1 // Determina el nivel de detalle de los atributos
                  que son capturados en el data profile
    long initialTime = SystemCurrentTime()
    long time = 0
    Pattern nearestPattern = null //Contiene el patrOn mAs cercano
                                                        al Perfil
(1) dataProfile = CaptureDataProfile(query, level)
(2) Pattern[] nearestPatterns = ObtainNearestPattern(dataProfile)
    // Arreglo para almacenar los patrones mAs cercanos al perfil
    Strategy[] strategies = null
    // Arreglo para almacenar las estrategias asociadas a un patrOn
    Strategy strategy = null
(3) Si (nearestPatterns.length > 1)
    y (SystemCurrentTime() - initialTime<MAXTime)
        Pattern[] accuratePattern =
                         SelectMostAccuratePattern (nearestPatterns)
        Si(accuratePattern.length = 1)
           nearestPattern =  accuratePattern[0]
Sino
           Hacer
              level = level+1
              dataProfile = CaptureDataProfile(query, level)
              nearestPatterns = ObtainNearestPattern(dataProfile)
              time = SystemCurrentTime() - initialTime
(4)        Hasta Que (nearestPatterns.length<>1 and time < MAXTime)
           nearestPattern = nearestPatterns[0]
        FinSi
    Sino
        nearestPattern = nearestPatterns[0]
    FinSi
    Stategies = ObtainRelatedStrategies(nearestPattern)
    Si (strategies.length > 1)
(5)     strategy = SelectCheaperStrategy(strategies)
    Sino
        strategy = strategies[0]
    FinSi
    retorne strategy
Fin
```

7.5. Síntesis

Este capítulo presenta un método, llamado MultiEscala, que permite elegir de forma dinámica la estrategia de selección de fuentes más apropiada para resolver una consulta de VDOs en una OV. El método multiescala parte del principio de que cada consulta tiene una relación diferente con el contexto de datos y puede ser resuelta de forma más eficiente si se evalúa esta relación. Si las propiedades del VDO que se solicitan en un consulta sólo involucran un número reducido de fuentes, basta con usar una estrategia de selección que evalúe la parte intencional de las fuentes para planear eficientemente esta consulta. Por el contrario, si la consulta involucra a un porcentaje alto de fuentes, es necesario usar una estrategia que permita filtrar de forma más precisa las fuentes que deben participar. Elegir la estrategia indicada hace necesario evitar usar estrategias de selección costosas para planear consultas que no involucran un gran número de fuentes a nivel intencional, y evitar usar estrategias simples en contextos complejos, como aquellos que tienen solapamiento.

Para hacer esta elección de forma inteligente el método multiescala captura diferentes atributos del contexto de datos cuando llega una consulta al sistema, usando un modelo que facilita capturar el nivel de detalle indicado del contexto, y posteriormente determina cuál es la estrategia que mejor puede hacer la selección de fuentes para esa consulta. La asociación entre consulta, estrategia y contexto de datos se realiza en dos fases. Una fase inicial de alimentación en donde se asocian todas las estrategias de selección disponibles a un tipo de contexto de datos llamado patrón en el cual tienen un mejor desempeño. Y una fase de elección, en donde se obtiene el contexto de datos de la OV cuando llega una consulta. El contexto obtenido es comparado con los patrones existentes con el fin de identificar cuál es el patrón más cercano. De las estrategias asociadas al patrón más cercano es elegida la estrategia menos costosa en tiempo de planeación.

Parte III

Implementación y Validación de la Propuesta

Capítulo 8

Análisis y Evaluación de OptiSource

Los capítulos 5, 6 y 7 presentaron nuestra propuesta para mediar las consultas que involucran múltiples fuentes en OV. Inicialmente se presentó la arquitectura general del sistema de mediación llamada ARIBEC y posteriormente, se describió en detalle OptiSource, la estrategia de selección de fuentes creada para OV de gran escala. Este capítulo presenta el análisis y la evaluación realizados sobre OptiSource como estrategia de selección de fuentes de ARIBEC. El énfasis de la evaluación es sobre esta estrategia pues de su buen funcionamiento depende el comportamiento de ARIBEC.

La estructura del capítulo es la siguiente. La Sección 8.1 presenta la evaluación de precisión y exhaustividad de la selección de fuentes realizada por OptiSource. Posteriormente, la Sección 8.2 presenta los resultados del análisis de sensibilidad que permitió establecer el impacto que tiene en OptiSource el nivel de fiabilidad de la función de beneficio utilizada. Posteriormente, la Sección 8.3 compara OptiSource con otras estrategias. La Sección 8.4 presenta los detalles del prototipo implementado sobre el cual se realizaron los experimentos. Finalmente, la Sección 8.5 concluye el capítulo.

8.1. Evaluación de Precisión y Exhaustividad de OptiSource

Como se mencionó en el Capítulo 6, OptiSource es una estrategia de selección de fuentes de datos diseñada para OV de gran escala. Cuando el procesador multiescala de ARIBEC detecta características de solapamiento, alto número de fuentes y replicación entre fuentes, OptiSource es elegida como estrategia de selección. Para poder medir en qué grado la selección realizada por OptiSource es satisfactoria, es necesario evaluar si el conjunto de fuentes seleccionadas realmente son relevantes para la consulta. Adicionalmente, es necesario evaluar si

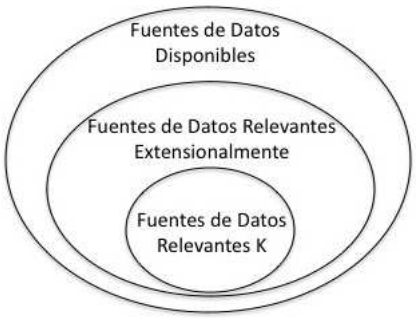

Figura 8.1: Esquema de Recuperación de Información

OptiSource elige las fuentes en orden de relevancia para la consulta. La evaluación fue realizada usando las medidas de evaluación que típicamente se aplican en los sistemas de recuperación de información, aplicando ciertas adaptaciones para el contexto. Esta sección presenta en la primera parte cuáles son estas medidas; posteriormente, cómo fueron obtenidas y finalmente, qué resultados se obtuvieron.

8.1.1. Métricas Evaluadas

La recuperación de información es un área de estudio cuyo objetivo es "encontrar material (generalmente documentos) de naturaleza no estructurada que satisface una necesidad, de una colección amplia de materiales (almacenada usualmente en computadores)"[MRS08]. La aproximación estándar de evaluación empleada en los sistemas de recuperación de información utiliza el concepto de documentos relevantes o no relevantes para poder medir la efectividad del sistema. Bajo este concepto de relevancia se calculan un conjunto de medidas como la precisión y la exhaustividad (*recall*) a través de las cuales se puede establecer si el algoritmo de recuperación de información es efectivo o no.

Para evaluar la estrategia OptiSource, ésta es vista como un sistema de recuperación de información que recupera fuentes de datos y en donde la relevancia de una fuente puede evaluarse desde dos puntos de vista: la relevancia extensional y la relevancia k. Desde la relevancia extensional, una fuente de datos es considerada relevante si contiene instancias que satisfacen por lo menos una condición de la consulta y no relevante en caso contrario. Desde la relevancia k, una fuente de datos es considerada relevante si pertenece al conjunto de las k fuentes más relevantes extensionalmente. Dada una consulta Q, una fuente A es más relevante extensionalmente que otra fuente B si A es capaz de entregar más instancias que resuelvan las condiciones de Q, que la fuente B. Las definiciones formales de estos dos puntos de vista son presentadas en la Definición 13 y 14. La Figura 8.1 ilustra el esquema de recuperación de fuentes de datos en OptiSource.

Definición 13 ***Relevancia Extensional***
Una fuente de datos DS_i es considerada relevante extensionalmente con respecto a una consulta $Q(VDO_A, propiedades(p_1, p_2, ..., p_n), condiciones(c_i, ..., c_m))$ donde $c_k = (p_j = val)$ ssi $\exists c_k \mid \sigma_{c_k}(DS_i) \neq \phi$, donde $\sigma_{c_k}(DS_l)$ representa los registros (tuplas,instancias, etc.) que satisfacen la condición c_k y $c_k \in$ condiciones.

Definición 14 ***Relevancia k***
Una fuente de datos DS_i es considerada relevante k con respecto a una consulta $Q(VDO_A, propiedades(p_1, p_2, ..., p_n), condiciones(c_i, ..., c_m))$ donde $c_k = (p_j = val)$ ssi dado un conjunto $K\{DS_r, ...DS_l\}$ donde K está conformado por las k fuentes más relevantes extensionalmente con respecto a Q, $DS_i \in K$.

El cálculo de las medidas tradicionales usadas en la evaluación de los sistemas de recuperación[MRS08] fue adaptado para evaluar OptiSource. Las medidas utilizadas se presentan a continuación.

1. Precisión extensional: Permite evaluar la proporción de fuentes de datos seleccionadas que son relevantes extensionalmente a la consulta.

2. Precisión k: Dado un k, permite evaluar la proporción de fuentes de datos seleccionadas que son relevantes k a la consulta.

3. Exhaustividad k: Dado un k, permite evaluar la proporción de fuentes de datos relevantes k con respecto a las fuentes de datos relevantes extensionalmente que fueron seleccionadas.

4. *Fall Out*[1] extensional: Permite evaluar la proporción de fuentes de datos no relevantes extensionalmente a la consulta que fueron seleccionadas.

5. *Fall Out* k: Dado un k, permite evaluar la proporción de fuentes de datos no relevantes k a la consulta que fueron seleccionadas.

6. Medida F: Permite comprobar que la precisión extensional y la exhaustividad k están compensadas, ya que un sistema con una exhaustividad muy alta pero con baja precisión o viceversa no será adecuado.

La Exhaustividad extensional, que permite evaluar la proporción de fuentes de datos relevantes extensionalmente que fueron seleccionadas, no fue tenida en cuenta en la experimentación, pues el objetivo de OptiSource no es seleccionar todas las fuentes relevantes extensionalmente, sino elegir primero las más relevantes.

Para calcular el conjunto de medidas descritas, se obtuvieron experimentalmente los valores de las siguientes variables dada una consulta de prueba y diferentes valores de k:

[1]Debido a que no existe una traducción exacta del término Fall Out en español, éste será presentado en inglés.

- R_{ext}**:** conjunto de fuentes relevantes extensionalmente para la consulta Q.
- $-R_{ext}$**:** conjunto de fuentes no relevantes extensionalmente para la consulta Q.
- R_k**:** conjunto de las k primeras fuentes relevantes extensionalmente para la consulta Q.
- $-R_k$**:** conjunto de fuentes no relevantes k para la consulta Q.
- A_{ext}**:** conjunto de fuentes de datos seleccionadas por OptiSource.
- A_k**:** k primeras fuentes de datos seleccionadas por OptiSource.

Las fórmulas empleadas para obtener cada medida se ilustran en las Fórmulas 8.1, 8.2, 8.3, 8.4, 8.5 y 8.6.

$$PrecisionExtensional = \frac{|A_{ext} \cap R_{ext}|}{A_{ext}}. \tag{8.1}$$

$$ExhaustividadK = \frac{|A_{ext} \cap R_k|}{|A_{ext} \cap R_{ext}|}. \tag{8.2}$$

$$FallOutExtensional = \frac{|A_{ext} \cap -R_{ext}|}{-R_{ext}}. \tag{8.3}$$

$$MedidaFBalanceada = \frac{2P_{ext} * E_{ext}}{P_{ext} + E_{ext}},$$
$$donde\, P_{ext}\, es\, la\, PrecisiOn\, Extensional\, y\, E_{ext}\, es\, la\, Exhaustividad\, Extensional. \tag{8.4}$$

$$PrecisionK = \frac{|A_k \cap R_k|}{A_k}. \tag{8.5}$$

$$FallOutK = \frac{|A_k \cap -R_k|}{-R_k}. \tag{8.6}$$

8.1.2. Metodología de Experimentación

La evaluación del comportamiento de OptiSource en cuanto a precisión, exhaustividad y *Fall Out* fueron llevados a cabo de forma experimental utilizando el prototipo de OptiSource. El objetivo de los experimentos realizados fue validar cómo se ven afectadas estas medidas de acuerdo a tres características del contexto de datos: el número de fuentes, el nivel de conocimiento extensional que se tiene de las fuentes, y la relación entre la consulta y el contexto de datos. En cada uno de los experimentos se definieron como parámetros la consulta a evaluar y el número de fuentes k a seleccionar. Posteriormente se capturaron las k fuentes seleccionadas para la consulta y éstas fueron comparadas frente a

las fuentes relevantes extensionalmente y frente a las fuentes relevantes k, que se conocían de antemano.

La carga de hechos a la base de conocimiento se realizó a través de una herramienta construida en el proyecto que, a partir de documentos descriptivos de las fuentes en XML, cargaba las relaciones entre las fuentes de datos y los diferentes conceptos de dominio alrededor de la salud según los requerimientos de cada experimento a la base de conocimiento de la OV. Para poder realizar pruebas más significativas se variaron los niveles de solapamiento entre las fuentes de datos. En el 30 % de los experimentos no se manejó solapamiento entre fuentes y en el 70 % restantes se manejaron niveles de solapamiento para un grupo de condiciones de la consulta.

Para asegurar que los experimentos fueran significativos, se siguió un protocolo de experimentación en donde cada experimento tiene diferentes características del contexto de datos. A continuación se describe cómo las características del contexto fueron variadas en cada experimento.

1. Número de fuentes: Los experimentos fueron ejecutados variando el número de fuentes disponibles. El posible número de fuentes en cada experimento puede estar en los siguientes rangos: 31-100, 101-300, 301-500 y 501-1000. Al interior de cada rango el número de fuentes puede ser aleatorio. Por ejemplo, si el rango elegido es entre 101-300, el número de fuentes puede ser 205.

2. Nivel de conocimiento extensional de las fuentes: Esta característica permite conocer qué tanto influye tener conocimiento detallado del rol que pueden jugar las fuentes con respecto a una propiedad del VDO. En los experimentos se manejaron tres niveles de conocimiento de las fuentes disponibles que se explican a continuación.

   - Nivel Bajo: Significa que la base de conocimiento sólo contiene el conocimiento intencional de las fuentes. Por ejemplo, el posible saber que una fuente DS1 puede evaluar la propiedad diagnóstico.
   - Nivel Medio: Significa que la base de conocimiento contiene el conocimiento intencional de las fuentes pero adicionalmente conoce el rol que las fuentes de datos pueden jugar con respecto a las propiedades utilizadas en las consultas. Sin embargo, este conocimiento está declarado en un nivel superior del que se encuentra la condición de la consulta. Por ejemplo, el rol de las fuentes está asociado a *cáncer* y las consultas solicitan instancias de pacientes con *cáncer de hígado*.
   - Nivel Alto: Significa que la base de conocimiento contiene el conocimiento intencional de las fuentes y los roles que ellas juegan con respecto a las propiedades de la consulta. Estos roles se encuentran en el mismo nivel en el que se declara la consulta. Por ejemplo, la consulta solicita pacientes con *cáncer de hígado* y los roles están asociados a *cáncer de hígado*.

3. Tipo de Consulta: La manera como están distribuidas las instancias que dan respuesta a una consulta afecta en gran medida la efectividad de un proceso de selección de fuentes. Para poder evaluar el comportamiento de OptiSource en contextos con diferentes distribuciones de instancias se definieron tres tipos de consultas que se diferencian en la distribución de las instancias que dan respuesta a ellas. Los tipos de consulta se describen a continuación:

   - Concentración de instancias: Consultas en donde las instancias que cumplen con las condiciones están concentradas en menos del 20 % de todas las fuentes que pueden evaluar la condición a nivel intencional (es decir que contiene la propiedad) y la distribución de las propiedades asociadas al predicado de la consulta están distribuidas en más del 70 % de las fuentes.
   - Dispersión de instancias: Consultas en donde las instancias que cumplen con las condiciones están dispersas en más del 70 % de todas las fuentes que pueden evaluar la(s) condición(es) a nivel intencional; es decir que contiene la propiedad asociada a las condiciones de la consulta.
   - Concentración de propiedades: Consultas en donde las instancias que pueden evaluar la condición a nivel intencional (es decir que contiene la propiedad) es menor al 20 % de todas las fuentes disponibles.

8.1.3. Resultados Obtenidos

Para ilustrar los resultados éstos fueron agrupados por la característica que fue variada durante la experimentación. Las otras características se mantuvieron constantes. A continuación se presenta cada grupo de experimentos.

Variación del Tipo de Consulta

Este grupo de experimentos varía la distribución de las instancias que dan respuesta a la consulta que se realiza y compara las medidas con respecto a una estrategia que sólo utiliza el conocimiento intencional de las fuentes. La configuración del experimento fue la siguiente:

- Número de Fuentes: 501-1000
- Nivel de Conocimiento: Medio

El análisis en este grupo de experimentos se concentró en analizar la relación entre las medidas k y las medidas extensionales, pues esta comparación permite identificar qué tan efectivo es OptiSource para seleccionar primero las mejores fuentes de datos de acuerdo al tipo de distribución de las instancias.

Los resultados obtenidos se pueden observar en las Figuras 8.2, 8.3, 8.4 que ilustran la precisión k y la precisión extensional obtenidas usando OptiSource y

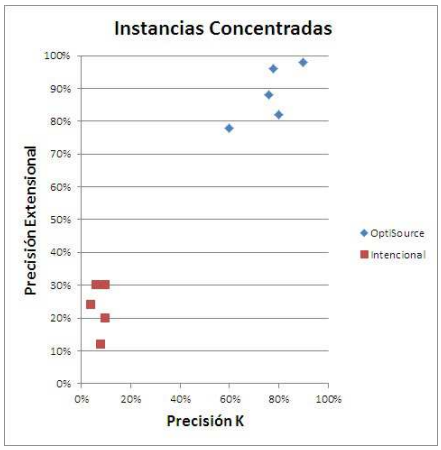

Figura 8.2: Precisión caso instancias concentradas. Número de fuentes: 501-1000. Nivel de conocimiento: Medio

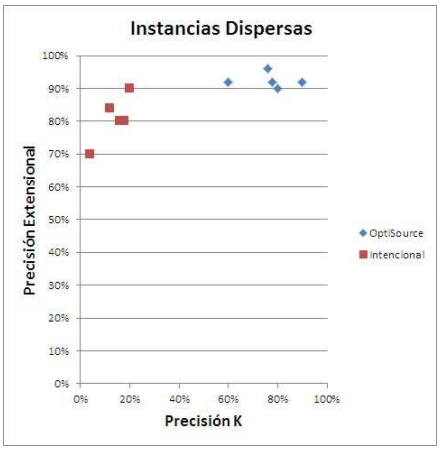

Figura 8.3: Precisión caso instancias dispersas. Número de fuentes: 501-1000. Nivel de conocimiento: Medio

una estrategia intencional. Para el caso intencional se eligieron las k primeras fuentes de los planes de ejecución generados.

Como puede observarse en la Figura 8.2, cuando las instancias están concentradas OptiSource sobrepasa en más del 50 % la precisión k y la precisión extensional de la estrategia intencional. Este resultado es consecuencia de la reducción de fuentes que realiza OptiSource utilizando los roles y el tamaño de las fuentes. Como es de esperarse la precisión k y la precisión extensional en OptiSource se acerca pero nunca alcanza el 100 % pues al tener un nivel de conocimiento medio los roles contenidos en la base no necesariamente están en el mismo nivel de la consulta, afectando la selección.

La Figura 8.3 muestra el comportamiento de las estrategias cuando las instancias están dispersas. OptiSource mantiene su alta precisión e incluso en ciertos casos la mejora. Esta mejora se debe principalmente a que la probabilidad de encontrar fuentes relevantes es mayor gracias a la alta dispersión de instancias relevantes. Por las mismas razones en este tipo de contextos la estrategia intencional se ve favorecida en más del 20 % en la precisión extensional.

La Figura 8.4 ilustra los resultados cuando las propiedades que permiten evaluar las condiciones de la consulta están concentradas en un número reducido de fuentes. En este caso el comportamiento de OptiSource y de la estrategia intencional es muy similar debido a que al ser un número tan reducido de fuentes las que pueden resolver las consultas, estas fuentes tienen mayor probabilidad de pertenecer al grupo de las k mejores fuentes. En este caso evaluar sólo el esquema es suficiente para identificar las fuentes más relevantes extensionalmente.

Dentro de este grupo de experimentos también se evaluó el *Fall Out* obtenido en cada tipo de consulta. La Figura 8.5 ilustra el *Fall Out* promedio obtenido usando cada estrategia en los experimentos para cada tipo de consulta. Como

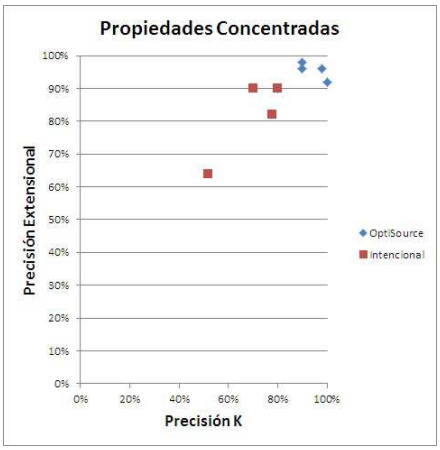

Figura 8.4: Precisión caso propiedades concentradas. Número de fuentes: 501-1000. Nivel de conocimiento: Medio

puede observarse el *Fall Out* obtenido en OptiSource es considerablemente menor en todos los casos. Aunque el *Fall Out* extensional mejora sustancialmente para la estrategia intencional en los casos en donde hay dispersión de instancias o concentración de propiedades, el *Fall Out* k se mantiene alto pues la aleatoriedad que beneficia la obtención de fuentes relevantes extensionalmente, no necesariamente beneficia la obtención de las k más relevantes.

Conclusión

Este grupo de experimentos permitió validar el buen comportamiento de OptiSource en los contextos para los que fue diseñado, es decir contextos con alto número de fuentes en donde hay solapamiento intencional y extensional y en donde la distribución extensional (de instancias) es menor con respecto a la intencional (propiedades). Así mismo, se pudo constatar que no es conveniente usar OptiSource en contextos en donde las propiedades están concentradas en un conjunto reducido de fuentes, pues en estos casos el esfuerzo de planeación realizado por OptiSource es innecesario cuando la elección de las fuentes que contienen las propiedades asociadas al predicado de la consulta son suficientes.

Variación del Nivel de Conocimiento

En este conjunto el nivel de conocimiento fue variado en cada experimentación y se mantuvieron constantes las otras características del contexto de la siguiente manera:

- Número de Fuentes: 501-1000
- Tipo de Consulta: Concentración de instancias.

El objetivo de este grupo de experimentos es validar qué tanto afecta el nivel de conocimiento la efectividad en la selección de OptiSource. Aunque el nivel

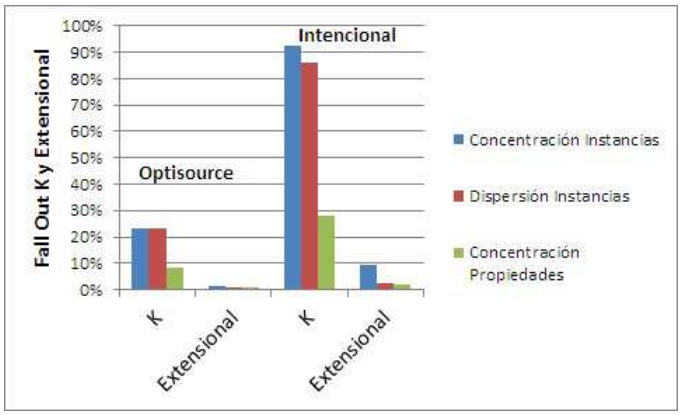

Figura 8.5: Comparación *Fall Out* OptiSource y Estrategia Intencional. Número de fuentes: 501-1000. Nivel de conocimiento: Medio

de conocimiento bajo es similar al intencional, se diferencia en que OptiSource además del conocimiento intencional de las fuentes utiliza el número de instancias de VDO contenidas en las fuentes para determinar la contribución esperada de cada fuente.

Para realizar este grupo de experimentos el valor de k fue definido como el 70 % del número de fuentes que son relevantes extensionalmente. Por ejemplo, si para un experimento el número de fuentes relevantes extensionalmente es 100 se evaluará la relevancia extensional de las 70 primeras fuentes elegidas por OptiSource.

En la Figura 8.6 se ilustra la variación en la precisión de OptiSource de acuerdo al nivel de conocimiento que se tenga disponible en la base de conocimiento. Como era de esperarse el nivel de precisión cuando el conocimiento contenido en la base es únicamente de tipo intencional es bajo; sin embargo, la precisión es un poco más elevada que una estrategia intencional al utilizar el número de instancias para predecir la contribución. Por otra parte, cuando el nivel de conocimiento es medio o alto la precisión es superior al 60 %. Esta apreciación corrobora el hecho de que no es necesario un conocimiento alto para que OptiSource tenga un buen comportamiento.

La Figura 8.7 ilustra la variación en la exhaustividad k de OptiSource. El comportamiento de la exhaustividad k fue similar al de la precisión extensional. Entre más conocimiento, más exhaustiva la selección. Sin embargo, no es necesario tener un conocimiento completo para tener una exhaustividad superior al 80 %. La Figura 8.8 permite visualizar más claramente la relación entre exhaustividad k y precisión extensional.

Finalmente, la Figura 8.9 presenta la proporción de fuentes irrelevantes extensionalmente obtenidas con respecto al total de las fuentes en el sistema. Como puede observarse, nuevamente la diferencia entre la proporción de fuentes no relevantes obtenidas cuando hay un nivel de conocimiento medio o uno

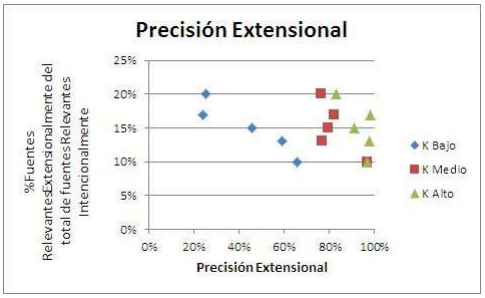

Figura 8.6: Precisión extensional variando el nivel de conocimiento. Número de fuentes: 501-1000. Tipo de Consulta: Concentración de instancias.

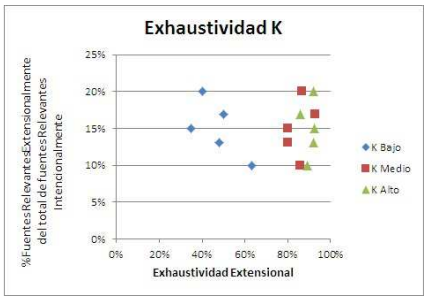

Figura 8.7: Exhaustividad K variando el nivel de conocimiento. Número de fuentes: 501-1000. Tipo de Consulta: Concentración de instancias.

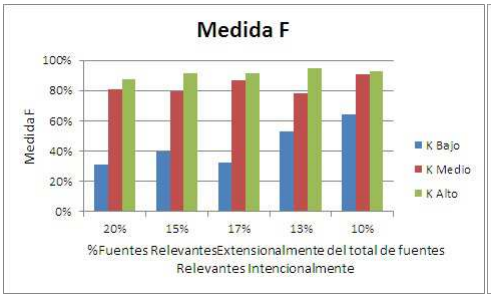

Figura 8.8: MedidaF variando el nivel de conocimiento

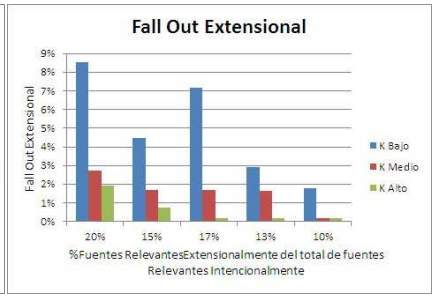

Figura 8.9: *Fall Out* extensional variando el nivel de conocimiento. Número de fuentes: 501-1000. Tipo de Consulta: Concentración de instancias.

alto es despreciable (menos del 2 %).

Conclusión

Si bien el nivel de conocimiento afecta el comportamiento de la estrategia de selección de OptiSource, los experimentos presentados demuestran que el nivel de conocimiento no debe estar necesariamente en el mismo nivel de la consulta para obtener buenos resultados. En la mayoría de los casos basta con tener un nivel de conocimiento estable para obtener una precisión y exhaustividad alta. Un nivel estable significa combinar hechos de conocimiento en niveles superiores con hechos en niveles detallados con respecto a las consultas. Por ejemplo, si las consultas usualmente solicitan información de pacientes con diagnóstico *Glaucoma Inflamatorio*, y el 90 % de las fuentes que contienen pacientes con diagnóstico *Glaucoma* (superclase de *Glaucoma Inflamatorio*) contienen pacientes con diagnóstico *Glaucoma Inflamatorio*, el nivel de conocimiento puede estar al nivel de *Glaucoma* (nivel medio). Por el contrario, si sólo el 20 % de las fuentes contienen pacientes con *Glaucoma Inflamatorio* sería necesario tener el nivel

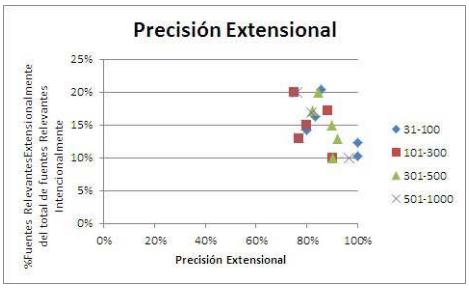

Figura 8.10: Precisión extensional variando el número de fuentes. Nivel de conocimiento: Medio. Tipo de consulta: Concentración de instancias.

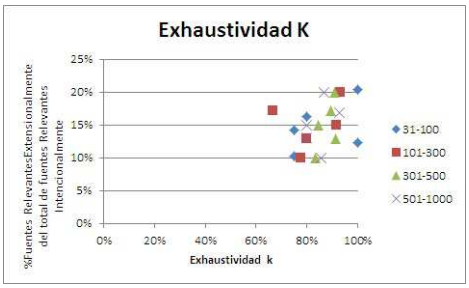

Figura 8.11: Exhaustividad K variando el número de fuentes. Nivel de conocimiento: Medio. Tipo de consulta: Concentración de instancias.

de conocimiento más detallado para lograr diferenciar las fuentes.

Adicionalmente, se puede concluir que al tener un nivel de conocimiento medio aún cuando la precisión k disminuye, por seleccionar un grupo de fuentes que no están dentro de las k más relevantes, en la mayoría de los casos estas fuentes son relevantes extensionalmente, por lo cual no es completamente irrelevante consultarlas.

Variación del Número de Fuentes de Datos

En este conjunto de experimentos el número de fuentes de datos fue variado en cada experimentación y se mantuvieron constantes las otras características. La configuración del experimento fue la siguiente:

- Nivel de Conocimiento: Medio
- Tipo de Consulta: Concentración de instancias.

Los resultados obtenidos de precisión extensional y exhaustividad k se pueden observar en las Figuras 8.10 y 8.11.

A pesar de que OptiSource está enfocado a OV con gran número de fuentes, su precisión y exhaustividad es igualmente alta en contextos con un bajo número de fuentes como puede observarse. Sin embargo, aplicarlo en estos contextos sólo es útil cuando las consultas realizadas son del tipo concentración de instancias, pues de otra forma el costo de usar OptiSource no se vería justificado frente a la ganancia en precisión y exhaustividad.

Adicionalmente, este grupo de experimentos también demostraron que OptiSource escala correctamente cuando se tienen contextos con rangos entre 501-1000. Además de mantener valores elevados de precisión y exhaustividad el tiempo empleado para realizar el proceso de optimización de la selección de fuentes tiene un crecimiento casi lineal. Esto puede observarse en la Figura 8.12, en donde fueron variadas las consultas en el número de condiciones, y

Figura 8.12: Escalabilidad en tiempo de acuerdo al número de fuentes y de condiciones

en el número de fuentes extensionalmente relevantes, al aplicar el modelo de optimización que utiliza OptiSource.

Conclusión

OptiSource es una estrategia creada para OV de gran escala pero su comportamiento en cuanto a precisión y exhasutividad se mantiene estable para contextos con diferentes números de fuentes. Es importante aclarar que los experimentos se realizaron para contextos en donde la distribución de las instancias que contribuyen a la consulta no es alto, pues es el contexto al que va dirigido OptiSource, esta característica beneficia los valores de precisión y exhaustividad así como también la capacidad de escalabilidad de la estrategia, debido a que OptiSource descarta desde el inicio un porcentaje alto de fuentes.

Conclusión de la Evaluación de Precisión y Exhaustividad

Los experimentos muestran que si bien es deseable un alto nivel de conocimiento del contexto, OptiSource tiene un buen comportamiento en cuanto a la precisión y la exhaustividad con un nivel medio de conocimiento. Esto se debe al hecho de que un nivel medio de conocimiento es suficiente para descartar un gran grupo de fuentes y de esta forma poder dirigir las consultas a las fuentes con mayor probabilidad de tener instancias coincidentes. Por el contrario, la métrica del *fall out* mejora (es decir, es menor), cuando aumenta el nivel de conocimiento. Las pruebas también muestran que en los contextos de datos con menor número de fuentes de datos, la mejora en la precisión de acuerdo con el nivel de conocimiento no es significativo. Esto confirma nuestra hipótesis de que en contextos de datos pequeños no son necesarios grandes esfuerzos en la planeación de consulta para la reducción de fuentes seleccionadas. Por otro lado, la mejora observada cuando hay un gran número de fuentes nos llevó a concluir que OptiSource es especialmente útil en estas circunstancias.

Por último, los experimentos muestran que en el peor de los casos (bajo nivel

de conocimiento) OptiSource selecciona el mismo conjunto de fuentes que las estrategias de selección que sólo tienen en cuenta el conocimiento de las fuentes a nivel intencional (esquema).

8.2. Análisis de Sensibilidad de OptiSource

La entrada más importante al modelo de optimización es el beneficio de utilizar una fuente de datos para evaluar una condición de consulta. Como fue presentado en el Capítulo 6, el valor del beneficio es estimado utilizando una función de estimación que utiliza el rol que puede desempeñar la fuente en la consulta y el tamaño relativo de la fuente con respecto a las otras fuentes que pueden resolver la misma condición. Si la estimación realizada es precisa, la asignación realizada por el modelo de optimización va a ser muy precisa. Sin embargo, si no hay certeza sobre la validez de la estimación, el modelo probablemente producirá resultados inexactos. Para evaluar qué tanto afecta la imprecisión durante la estimación del beneficio al resultado final de asignación de fuentes a condiciones de la consulta, esta sección presenta el análisis de sensibilidad realizado sobre el modelo optimización con respecto a los valores de la matriz de entrada de beneficio. Para presentar este análisis se ilustrará inicialmente la metodología utilizada para evaluar el impacto del valor de beneficio en la asignación final, y posteriormente se presentan los resultados obtenidos y un análisis sobre ellos.

8.2.1. Metodología de Análisis

El análisis de sensibilidad de los modelos de optimización es una tarea bien conocida que permite evaluar el impacto de los cambios en los parámetros de entrada de los modelos de programación lineal. Aunque actualmente existen métodos para evaluar la sensibilidad de los resultados de los modelos de optimización, debido a la alta degeneración del problema de asignación [LW03], estos métodos tradicionales no son prácticos para evaluar la sensibilidad en el modelo de optimización de OptiSource. Además, aunque hay algunas propuestas para determinar el rango de sensibilidad de los parámetros en los problemas de asignación [LW03], éstos no pueden ser aplicados al modelo de OptiSource, ya que suponen el uso del problema clásico de asignación, y el modelo de OptiSource difiere del problema clásico en sus matrices de restricciones y en el cálculo de la función objetivo. Adicionalmente, estos métodos de análisis de sensibilidad suponen la propiedad de unimodularidad de la matriz de restricciones, que no es validada en nuestro modelo. En consecuencia, con el fin de evaluar el impacto del modelo, se desarrolló un estudio experimental que permitió analizar el impacto del valor del beneficio en la asignación obtenida a través del modelo. La intención de este estudio fue medir los cambios en la asignación cuando el valor del beneficio varía con respecto al valor real. Para poder evaluar experimentalmente la sensibilidad del modelo se desarrolló una herramienta que a partir de una matriz de beneficio base con valores fiables permite generar matrices que reducen la fiabilidad aumentando o disminuyendo el beneficio de usar

una fuente para evaluar una condición. El porcentaje en que la fiabilidad de beneficio puede ser reducida o aumentada son parámetros de configuración de la herramienta, así como también el número de matrices que serán generadas a partir de una matriz base.

Para asegurar que los experimentos realizados fueran significativos, el modelo de optimización fue aplicado a diferentes matrices base que varían en cuanto al número de fuentes de datos, al número de condiciones en el predicado y a los valores de beneficio de cada relación fuente-condición. Estos valores de beneficio fueron generados aleatoriamente para simular base de conocimiento con poco o mucho conocimiento de las fuentes y para tener diferentes roles con respecto a cada fuente. Se hicieron experimentos con 25, 50, 100, 200, 400, 800 fuentes de datos y con 5, 10, 20 número de condiciones. En total el modelo fue ejecutado sobre 18 matrices base y los resultados de asignación obtenidos en cada caso son considerados la asignación base.

Para analizar la sensibilidad, por cada una de las matrices base se generaron con la herramienta un conjunto de matrices con valores menos fiables. El nivel de fiabilidad fue reducido en un valor entre 0 y 0.66. En el primer grupo de experimentos los valores podían variar hasta en un 100 %, es decir que un valor que inicialmente era 0.99, podía perder fiabilidad de hasta 0.66 y quedar en 0.33, en los otros grupos de pruebas lo valores podía variar hasta en un 50 %, 20 %, 15 %, 10 %, 5 %. En resumen, por cada matriz base se generaron 5 matrices por cada valor de reducción de fiabilidad. En total para cada matriz base fueron generadas 30 matrices, dando un total de 540 pruebas.

8.2.2. Resultados de la Experimentación

La Figura 8.13 ilustra el resumen de los resultados de experimentación. El eje de las x representa el porcentaje de asignaciones que cambiaron en cada experimento. El eje y representa el porcentaje de experimentos para los cuales fue obtenido un cambio en la asignación. Cada línea representa el número de fuentes de datos utilizadas en el experimento. Por ejemplo, con 100 fuentes de datos, 11 % de experimentos no tuvieron ningún cambio en la asignación del beneficio. De forma similar, con 800 fuentes de datos 22 % de los experimentos tuvieron 60 % de cambios en la asignación cuando el beneficio cambió.

Como puede observarse, los cambios en la asignación están directamente relacionados con el número de fuentes de datos. Este resultado era esperado debido a que entre mayor cantidad de fuentes de datos, mayor será la probabilidad de encontrar una fuente de datos con una mejor contribución a la condición, lo que obliga al modelo a cambiar la asignación. El estudio también demostró que la mayor parte de los experimentos tuvieron cambios entre el 40 % y el 70 % en la asignación principal. Además, las pruebas mostraron que el 25 % de los experimentos tenían cambios del 60 %, independientemente del número de fuentes de datos.

La Figura 8.14 ilustra el promedio de experimentos agrupados por el porcentaje de cambios. Por ejemplo, 27 % de los experimentos tuvieron cambios cercanos al 60 %, y 18 % de los experimentos tuvieron cambios cercanos al 40 %.

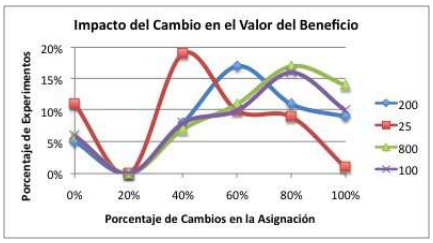

Figura 8.13: Impacto General de Beneficio

Figura 8.14: Promedio de Cambios en el Conjunto de Experimentos

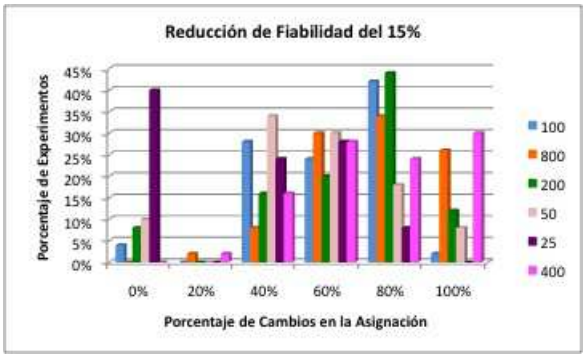

Figura 8.15: Impacto del Beneficio en la Asignación - 15 %

La Figura 8.15 y la Figura 8.16 presentan los resultados de sensibilidad del modelo cuando el beneficio tiene un cambio máximo del 15 % o del 50 %, respectivamente. Como en el caso general, la mayoría de experimentos tuvieron cambios entre el 40 % y 70 %, independientemente del grado de disminución de la fiabilidad del valor del beneficio.

Conclusión

Los resultados en los experimentos de sensibilidad demuestran que, como es natural en los problemas de asignación, cambios en el valor del beneficio afectan la asignación final. Sin embargo, un porcentaje entre el 20 % y el 60 % de las asignaciones siguen manteniéndose, incluso si la reducción en la fiabilidad del valor del beneficio es del 50 % del valor real. Este hecho, demuestra un buen nivel de flexibilidad de nuestro modelo de optimización, que no requiere una predicción exacta del valor del beneficio de usar una fuente para hacer una asignación cercana al valor óptimo.

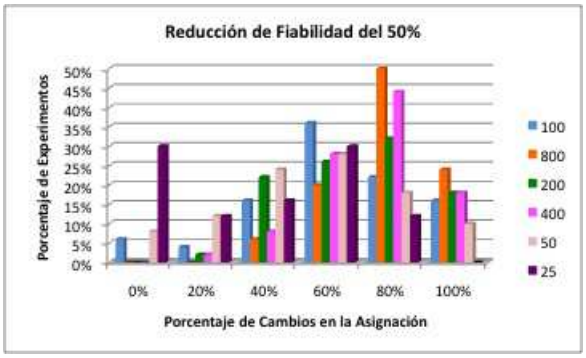

Figura 8.16: Impacto del Beneficio en la Asignación - 50 %

8.3. Comparación de OptiSource con Otras Estrategias

Como fue presentado en el Capítulo 4 las estrategias de selección disponibles en la literatura fueron creadas para contextos diferentes al de las OV de gran escala o funcionan bajo supuestos de información que no está disponible en las OV. Aunque OptiSource pudo ser comparado con una estrategia puramente intencional (ver Sección 8.1), para poder comparar equitativamente la capacidad de elegir las mejores fuentes primero es necesario evaluar el comportamiento de otra estrategia que tenga esta intención, aún cuando haya sido creada para otro contexto. Por esta razón esta sección presenta la comparación realizada entre OptiSource y una estrategia llamada iDrips que por su naturaleza flexible pudo ser adaptada para ser utilizada como estrategia de selección en OV. Así mismo, se presenta el análisis realizado sobre dos estrategias más que buscan elegir las mejores fuentes llamadas QPIAD [WKC+07] y Caminos de Navegación [BKN+06])

8.3.1. iDrips

La intención de iDrips es, dada una consulta, encontrar primero los planes más relevantes y mientras éstos son ejecutados encontrar los demás planes. El algoritmo iDrips fue elegido porque su funcionamiento está basado en una función de utilidad genérica que, de acuerdo a cada contexto, puede ser adaptada. Su funcionamiento general fue presentado en la 4.2.4. En resumen, iDrips funciona bajo el supuesto de similaridad entre fuentes que le permite al procesador de consultas obtener los planes de ejecución más relevantes de forma eficiente. La comparación de iDrips con Optisource se enfoca en medir qué tantas fuentes de datos debe consultar cada estrategia para obtener todas las instancias disponibles y relevantes para la consulta.

Las variables que fueron evaluadas para analizar la eficiencia de las estrategias fueron las siguientes:

$NInstancias_i^{est}$ Número de instancias relevantes obtenidas al consultar la i-ésima fuente de datos usando la estrategia est, donde $1 \leqslant i \leqslant TFuentes$ y TFuentes es el número total de fuentes disponibles.

$QFuentes^{est}$ Número total de fuentes de datos seleccionadas usando la estrategia est para obtener todas las instancias relevantes disponibles en el sistema.

Para poder calcular la eficiencia de cada estrategia se mantuvieron constantes los siguientes valores:

$TFuentes$: número total de fuentes disponibles.

$TInstanciasRelevantes$: número total de instancias relevantes disponibles en el sistema.

El cálculo de la eficiencia se hizo aplicando la Fórmula 8.7 que permite saber qué tantas instancias relevantes de las disponibles entrega una estrategia al consultar la i-ésima fuente seleccionada.

$$Eficiencia(est) = \text{máx}\{\frac{NInstancias_i^{est}}{TInstanciasRelevantes} * \frac{1}{i}\}|1 \leqslant i \leqslant n, \tag{8.7}$$

donde n es el número de fuentes disponibles. Al elegir el máximo se busca darle mayor valor de eficiencia a las estrategias que encuentran mayor número de instancias relevantes primero. Por ejemplo, si para una consulta hay 100 instancias relevantes en el sistema, si usando una estrategia E1 se pueden entregar 20 instancias relevantes luego de consultar la cuarta fuente de datos seleccionada la eficiencia de E1 ($Eficiencia(E1) = 20/4$) será mejor que otra estrategia E2 que logra entregar 20 instancias luego de consultar la quinta fuente de datos seleccionada ($Eficiencia(E2) = 20/5$).

Para aplicar iDrips se utilizó la medida de utilidad llamada cobertura del plan, en donde se estima cuántas instancias aportará la ejecución de un plan. Entre mejor sea la cobertura del plan mayor prioridad le da iDrips a este plan. Para calcular la cobertura de un plan se utilizaron tres fórmulas diferentes que permitieron analizar los niveles de conocimiento requeridos del contexto de datos de la OV. Dado un plan $P\{DS_1, ..., DS_n\}$ donde las DS_i son las fuentes que serán consultadas en el plan y una consulta $Q\{p(p_1, ..., p_m)\}$ con m condiciones en el predicado, las tres fórmulas utilizadas para calcular la cobertura del plan son las siguientes:

$$Cov(P) = min(ext(DSi, VDOj))\ donde\ DS_i \in P. \tag{8.8}$$

$$Cov(P) = min(overlap(DS_i, DS_k))\ donde\ DS_i\ y DS_k \in P, \tag{8.9}$$

$$Cov(P) = \frac{ext(DSi, VDOj)^p - (ext(DSk, VDOj)^p ... ext(DSl, VDOj)^p)}{ext(U, VDOj)^p} \tag{8.10}$$

Se utilizaron tres fórmulas diferentes para calcular la utilidad con el fin de analizar el comportamiento de iDrips con diferentes niveles de conocimiento del contexto. La Fórmula 8.8 supone el conocimiento del número de instancias de VDO que cada fuente contiene. La Fórmula 8.9 por su parte, supone el mismo conocimiento que la Fórmula 8.8, pero adicionalmente hace necesario conocer

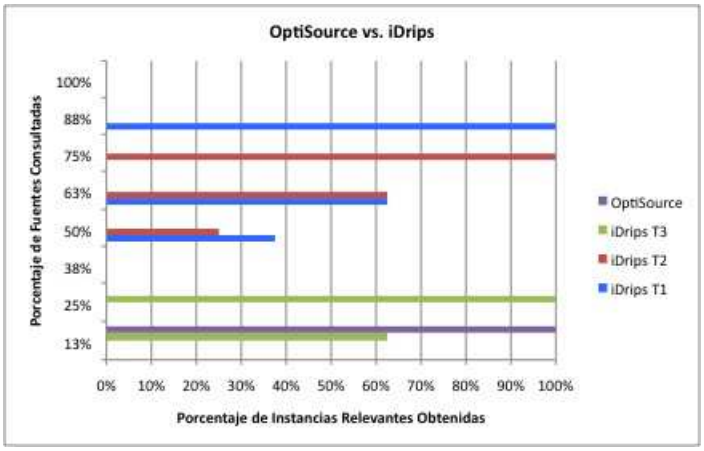

Figura 8.17: Comparación del comportamiento de OptiSource e iDrips

el nivel de solapamiento que existe entre cada fuente. En la Fórmula 8.10, las fuentes de datos $\{DS_i, ..., DS_k\}$ son aquellas que han sido consultadas previas al plan que se está evaluando. Esta fórmula (8.10) supone el conocimiento de las fórmulas 8.8 y 8.9. Adicionalmente, supone conocer el número de instancias que satisfacen el predicado de la consulta en cada fuente (o en cada plan) y de estas instancias cuáles de ellas también pueden ser proporcionadas por otro plan. Aunque esta información es difícil y costosa de obtener en los contextos de las OV, fue utilizada pues los creadores de iDrips suponen tenerla en el algoritmo diseñado.

El análisis del comportamiento de iDrips frente al comportamiento de OptiSource fue realizado en los contextos para los cuales fue creado OptiSource, contextos en donde las fuentes están solapadas extensionalmente e intencionalmente, y en donde la distribución intencional es alta; aunque la distribución extensional puede ser alta debido a que las instancias relevantes están distribuidas en múltiples fuentes, por fenómenos de replicación, éstas pueden ser obtenidas en su totalidad sin necesidad de consultar todas las fuentes con instancias relevantes extensionalmente. En otras palabras, en este contexto existe un grupo de fuentes de datos especialistas en una o más de las condiciones de la consulta. Como consecuencia, consultarlas es suficiente para obtener casi todas las instancias. OptiSource fue utilizado con un nivel de conocimiento medio con respecto a las consultas.

La Figura 8.17 ilustra la relación entre porcentaje de fuentes consultadas e instancias relevantes obtenidas entre iDrips y OptiSource para este contexto. El eje de las x representa el porcentaje de instancias relevantes obtenidas y el eje de las y el porcentaje de fuentes de consultadas.

Por su parte, la Figura 8.18 ilustra el comportamiento del valor de la eficiencia y la eficiencia máxima de cada estrategia usando la Fórmula 8.7. Los resultados demuestran la eficiencia de OptiSource en este tipo de contextos don-

Figura 8.18: Eficiencia de OptiSource e iDrips

de consultando en promedio el 13 % de las fuentes relevantes se obtienen el 100 % de las instancias relevantes. Por el contrario iDrips requiere consultar el 25 % de las fuentes relevantes para obtener el 100 % de las instancias relevantes en el mejor de los casos (usando la Fórmula (8.10)).

La Figura 8.17 muestra que la proporción de fuentes seleccionadas por OptiSource es considerablemente menor a aquellas seleccionadas usando iDrips empleando la Fórmula 8.8 y 8.9 para calcular la cobertura del plan. Aunque esto podría haberse inferido considerando la diferencia en el nivel de conocimiento, permite concluir que las soluciones de ranqueo de planes de ejecución no pueden ser aplicadas en las OV si no se utiliza una función de utilidad capaz de representar las relaciones entre las fuentes y la probabilidad de que una fuente tenga instancias relevantes para la consulta. Sin embargo, aun cuando se tenga una función de utilidad poderosa, la dificultad de representar en una función de utilidad por fuente (o por plan) la replicación dificulta identificar las fuentes que no contribuirán con nuevas instancias. Esta ausencia afecta el comportamiento de iDrips cuando utilizó la Fórmula 8.10. Valdría la pena explorar funciones de utilidad que permitan, a través de un valor, reflejar la contribución individual y la contribución grupal de las fuentes. Si bien OptiSource maneja estos conceptos no los condensa en una sola medida.

El siguiente grupo de análisis se llevó a cabo en un contexto donde las fuentes están solapadas intencionalmente pero no extensionalmente, y un número reducido de fuentes están especializadas en una o más de las condiciones de los predicados. Como era de esperarse, el comportamiento de ambas estrategias fue similar debido a que fue necesario consultar casi todas las fuentes relacionadas con la consulta para obtener un número considerable de instancias.

8.3.2. Estrategias de selección aplicadas a OV de gran escala

Caminos de Navegación

Al igual que OptiSource, la estrategia de Caminos de Navegación [BKN+06] tiene por objetivo seleccionar el mejor grupo de fuentes de datos que permitan resolver una consulta, donde el mejor camino es el que posea el mejor beneficio a menor costo. A diferencia de OptiSource, esta estrategia supone la disponibilidad de un grafo que describe los objetos contenidos en cada fuente de datos. La relación entre los objetos contenidos en las fuentes se refleja a través de la relación de los nodos que representan cada fuente. El resumen de esta estrategia puede consultarse en la Sección 4.2.2.

A pesar de que el algoritmo de selección utilizado en esta estrategia es útil si es posible conocer y mantener en los grafos los objetos de cada fuente y las relaciones entre ellos, en el caso de las OV de gran escala esto se dificulta debido a la autonomía de las fuentes, a su tamaño y a la confidencialidad de los datos. La propuesta para poder adaptar este algoritmo de selección a un contexto de OV es organizando las fuentes de datos disponibles con información de un VDO en el grafo de caminos, como se ilustra en la Figura 8.19 y se explica a continuación. Cada nodo contiene las fuentes que incluyen el mismo fragmento intencional de un VDOs. En otras palabras, las fuentes solapadas intencionalmente quedan en el mismo nodo. Como los objetos (instancias) que están al interior de cada fuente no se conocen en las OV, cada fuente es vista como una caja gris que presenta únicamente los roles que la fuente puede jugar con respecto al fragmento intencional que conoce. Cada fuente se conecta con las fuentes que pertenecen a los mismos conjuntos de integración a los que ella pertenece, que no pertenecen al mismo nodo. Por ejemplo, si las fuentes de datos DS2 y DS3 contienen *DatosDemograficos* del VDO Paciente, estas fuentes estarán en el mismo nodo, y si DS2 está en al menos un conjunto de integración con DS4 estas dos fuentes estarán relacionadas. Si es conocido que dos fuentes del mismo nodo están solapadas extensionalmente, esta relación de solapamiento permitirá que todas las relaciones con una de ellas sean igualmente válidas para la otra, como lo muestra en líneas punteadas la Figura 8.19.

Para aplicar el algoritmo de Caminos de Navegación con este nuevo grafo el beneficio será calculado a partir de los roles asociados a cada fuente que hace parte del camino. El mejor camino será el que tenga roles más relevantes. La Figura 8.19 ilustra en rojo el camino más importante para una consulta que busca todos los pacientes niños con diagnóstico Cancer.

Como puede observarse, OptiSource combinado con la estrategia de Caminos de Navegación puede mejorar la aplicación de este tipo de estrategias en OV. En general, diferentes estrategias que se apoyen en el uso de una medida de beneficio para seleccionar las mejores fuentes puede usar el concepto de roles y de predicción de beneficio introducido por OptiSource.

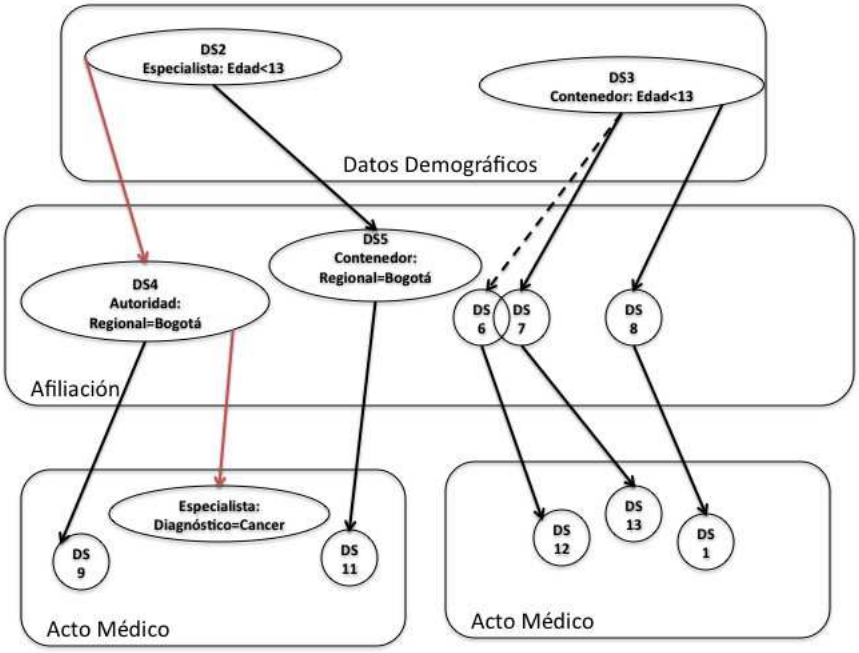

Figura 8.19: Adaptación de la estrategia de caminos de navegación para OV

QPIAD

QPIAD [WKC+07] es una estrategia creada para contextos en donde las fuentes de datos pueden tener en nulo el valor de una propiedad dentro del predicado de la consulta. QPIAD evalúa las consultas en todas las fuentes que la pueden evaluar a nivel intencional y posteriormente genera nuevas consultas creadas para obtener los resultados que tienen alta probabilidad de ser relevantes para la consulta, incluso si unas de las propiedades del predicado se encuentra en nulo. El resumen de esta estrategia puede consultarse en la Sección 4.2.2.

A pesar de que el principio de QPIAD es muy atractivo para ser usado en OV cuyas fuentes tienen problemas de calidad, la estrategia no podría escalar cuando el número de fuentes es muy alto. El problema de escalamiento se puede ver a través de un ejemplo. Considere una OV con 20 fuentes de datos, que son capaces de evaluar una consulta Q en términos intencionales. QPIAD reescribe y envía la consulta original en términos de cada fuente y obtiene a cambio respuestas certeras, es decir las resultantes después de evaluar todas las condiciones de la consulta. Para obtener las respuestas posibles (pero no certeras), QPIAD analiza las dependencias entre atributos y valores en las respuestas ciertas recibidas y genera nuevas reescrituras de la consulta original. Si por cada una de las fuentes QPIAD genera 10 reescrituras de la consulta para obtener las

respuestas posibles, serían ejecutadas 100 consultas más a las originales. Este número de ejecuciones es viable en un contexto con pocas fuentes y con alta disponibilidad, pero no lo es en las OV.

Más que adaptar QPIAD para ser aplicado en los contextos de datos de las OV, la propuesta es utilizar la estrategia de QPIAD únicamente para obtener respuestas posibles del conjunto de las k fuentes más relevantes dentro de una ejecución. Por ejemplo, si k es 3, OptiSource determina que las fuentes DS1, DS2 y DS3 deben ser consultadas en primer lugar. Una vez se obtengan las instancias ciertas provenientes de estas fuentes, sería conveniente usar QPIAD para obtener la nuevas instancias posibles, únicamente de este grupo de fuentes. De esta manera, QPIAD sería de utilidad en las OV cuyas fuentes tienen problemas de calidad y al mismo tiempo se garantizaría escalabilidad al reducir el número de posibles reescrituras.

8.4. Prototipo

Esta sección presenta los componentes del prototipo implementado para validar la propuesta de esta investigación. Inicialmente se describirán las responsabilidades e interfaces de los componentes que fueron implementados. Posteriormente, se presentará las elecciones de herramientas y lenguajes tomadas durante la implementación.

8.4.1. Implementación de Componentes

La Figura 8.20 ilustra los componentes de ARIBEC que fueron implementados en el prototipo. La función de cada uno de estos componentes fue descrita en la Sección 5.4 del Capítulo 5. Las interfaces que proveen estos componentes son

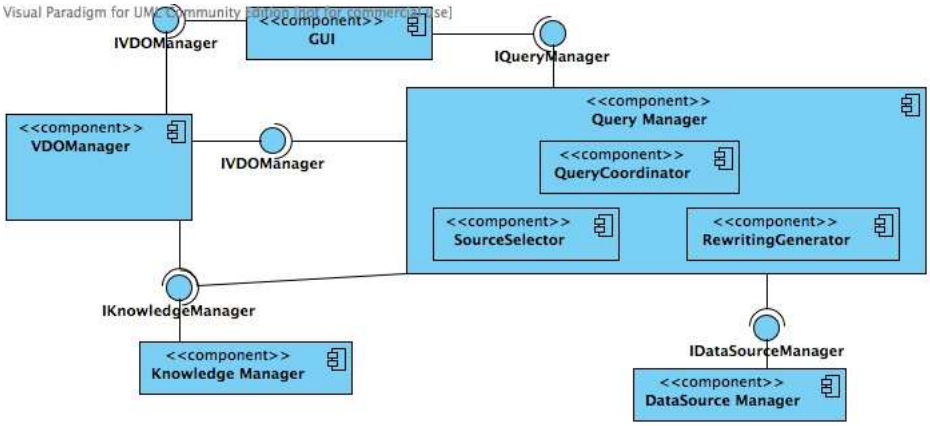

Figura 8.20: Componentes Implementados de ARIBEC

presentados en la Figura 8.21. A continuación se presentan sus responsabilidades

y su interacción:

- *GUI*(Interfaz Gráfica): Permite la interacción del usuario con el nivel de mediación. A través de este componente el usuario puede crear consultas y lanzar su ejecución. Utiliza las interfaces *loadVDO()* del componente *VDOManager* y *executeQuery()* del componente *QueryManager*.

- *VDOManager*(Gestión de VDO): Es el responsable del registro de nuevos VDOs en el sistema de mediación. Interactúa con el componente *KnowledgeManager* usando la interfaz *queryKnowledgeBase()*.

- *KnowledgeManager*(Gestión de Conocimiento): Es el responsable de la administración de la base de conocimiento de la OV. Provee las interfaces para consultar, agregar y remover hechos conocimiento.

- *Query Manager*(Gestión de Consultas): Es responsable de todos los servicios de consulta de ARIBEC. Se encarga de planear las consultas y coordinar su ejecución. Utiliza una estrategia de selección de fuentes diferente de acuerdo al tipo de consulta. Su implementación actual incluye dos estrategias de selección: OptiSource e Intencional. Esta última utiliza el esquema de las fuentes para hacer el proceso de selección. Su componente fachada llamado *QueryCoordinator* provee la interfaz *executeQuery()* cuya implementación se encarga de iniciar la lógica de evaluación de consultas. Interactúa con el componente *KnowledgeManager* a través de sus interfaces *queryKnowledgeBase()*, *insertKnowledgeFact()* y *removeKnowledgeFact()*. Así mismo, utiliza la interfaz *queryDataSource()* del componente *DataSourceManager*.

- *DataSource Manager*(Gestión de Fuentes): Es responsable de registrar físicamente las fuentes disponibles en el sistema. Interactúa con los componentes del nivel de adaptación.

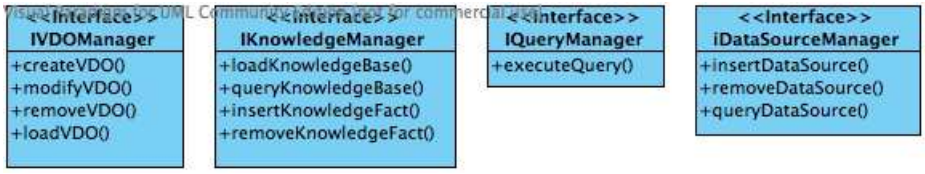

Figura 8.21: Interfaz de los componentes

8.4.2. Elección de Lenguajes y Marcos de Trabajo

La programación de los componentes se realizó en el lenguaje Java 1.6. Adicionalmente, para el desarrollo de los componentes KnowledgeManager y QueryManager se utilizaron componentes externos que brindaron parte de las funcionalidades requeridas.

La definición de la base de conocimiento se hizo utilizando el lenguaje OWL DL [Hor05]. Para poder administrar esta base se utilizó el marco de trabajo Jena[CDD+04], diseñado para construir aplicaciones semánticas. Este marco de trabajo incluye un ambiente completo para administrar ontologías en OWL y para consultarlas. Jena fue elegida porque permite utilizar directamente Pellet [SPG+07], un poderoso motor de inferencia sobre el lenguaje OWL. Este motor nos permitió realizar las consultas sobre la base de conocimiento con tiempos de respuesta satisfactorios. Adicionalmente, Jena es de código abierto y su comunidad de usuarios es muy activa lo que promovió su rápido aprendizaje.

Para implementar el componente de optimización de OptiSource, se definió el modelo usando el lenguaje GNU MathProg [Mak09b] el cual puede ser consultado en el Apéndice B. Para ejecutar éste modelo desde la lógica de selección de fuentes de OptiSource se utilizó GNU Linear Programming Kit (GLPK) [Mak09b] y su enlace con java llamado GLPK-Java [Mak09a].

GLPK es un conjunto de herramientas creadas para resolver programación lineal de gran escala, programación entera mixta y otros problemas relacionados. Por su carácter de código abierto GLPK fue elegido para realizar la implementación; sin embargo, para realizar las pruebas de sensibilidad también se utilizó una herramienta comercial llamada CPLEX.

8.5. Síntesis

Este capítulo presenta la evaluación llevada a cabo sobre OptiSource como estrategia de selección de ARIBEC. La evaluación experimental llevada a cabo sobre Optisource permitió evaluar diferentes características como precisión, exhaustividad, sensibilidad a la fiabilidad de la función de beneficio. Así mismo, la evaluación permitió analizar en detalle otras estrategia y compararlas con OptiSource.

La experimentación llevada a cabo con OptiSource permitió identificar que la precisión y exhaustividad durante la selección de fuentes se incrementa ampliamente (más de un 80 %) frente a las estrategias basadas en el esquema teniendo un nivel medio de conocimiento del contexto en ambientes con un alto número de fuentes. Éste era uno de los principales objetivos de la investigación: lograr hacer más eficiente el proceso de selección sin requerir tener toda la información detallada del contenido de las fuentes. Así mismo, la relación entre precisión y exhaustividad demostró ser equilibrada en la gran mayoría de los contextos, excepto en los contextos con un alto número de fuentes y con un nivel de información sólo de tipo intencional. En este caso OptiSource se comporta de forma similar a una estrategia intencional.

Por otra parte, el análisis de sensibilidad llevado a cabo con OptiSource demostró la flexibilidad del modelo de optimización con respecto al parámetro de entrada de beneficio. Esta flexibilidad permite que el modelo entregue respuestas cercanas a la óptima incluso cuando el valor del beneficio estimado para una fuente no es tan preciso.

Finalmente, la comparación de OptiSource con otras estrategias existentes

ilustró cómo OptiSource puede ser utilizada en otras estrategias para estimar el beneficio de usar una fuente determinada y cómo OptiSource puede ser enriquecida a través de estrategias cuyo enfoque sea el de mejorar la completitud de las respuestas obtenidas.

Capítulo 9

Conclusiones y Perspectivas

Este capítulo concluye el trabajo de investigación realizado alrededor de la mediación de consultas y en particular de la selección de fuentes de datos, en el contexto de las OVs. La Sección 9.1 presenta las principales contribuciones y conclusiones. La Sección 9.2 explora las perspectivas de trabajo futuro.

9.1. Contribuciones

La creación de infraestructuras de comunicación y colaboración que soporten el nuevo modelo organizacional llamado Organización Virtual es un área de estudio aún en desarrollo que requiere de investigación desde diferentes frentes. Este trabajo de investigación se enfocó en fortalecer la infraestructura de información de las OVs a través de un sistema de mediación flexible que permite a la red de organizaciones autónomas compartir información. Los principales aportes de esta tesis en esta área serán resumidos a continuación.

1. **Caracterización del contexto de datos de las organizaciones virtuales de gran escala.** Para poder diseñar una infraestructura de información que de soporte a las OVs es necesario conocer las características que las hacen diferentes de otros contextos distribuidos. Estas diferencias pueden dificultar el uso de algoritmos, estrategias o métodos ya conocidos, y a la vez pueden incorporar nuevos elementos que permiten ver los problemas clásicos desde otra perspectiva. Esta investigación identificó cuáles son las principales características de los contextos de datos de las OVs de gran escala que impactan el procesamiento de consultas (ver Capítulo 2) y cuáles son los nuevos elementos disponibles en las OVs producto de su carácter organizacional que enriquecen la planeación de consultas distribuidas. Estos últimos elementos fueron claves en la propuesta de mediación y de selección para OVs (ver Capítulo 6).

   La investigación identificó que las características del contexto que más dificultan el procesamiento de consultas en las OVs son la fragmentación

vertical y horizontal no disyunta, que genera solapamiento entre las fuentes a nivel intencional y extensional. Estas características, sumadas a la existencia de copias difusas y a la replicación en contextos con alto número de fuentes y ampliamente distribuidas vuelven ineficientes o inutilizables las estrategias de selección actuales.

Por otra parte, las peculiaridades organizacionales de la OV hacen que su contexto de datos tenga ciertas características que en otros contextos no es posible suponer y que fueron identificadas en esta investigación. En primer lugar conocer el rol que cada participante juega dentro de la OV permite relacionar su "saber hacer" con la información que contienen las fuentes de datos que provee. La asociación de los roles organizacionales a los roles que pueden jugar las fuentes en las OVs es una importante contribución de esta investigación, pues permite incorporar a las estrategias de selección la capacidad de diferenciar y de priorizar las fuentes de datos en fuentes autoridades, especialistas o contenedoras con respecto a la consulta. Así mismo, las relaciones de las organizaciones participantes al interior de la OV incorporan un nuevo conocimiento que fue utilizado como regla heurística para integrar las instancias de las fuentes de datos provenientes de participantes con cierto nivel de afinidad.

2. **Análisis de las estrategias de selección de fuentes de datos desde el ángulo de las OVs.** El análisis del estado del arte realizado durante esta tesis permitió contribuir con la clasificación de las estrategias de selección de fuentes de datos existentes (ver Capítulo 4) de acuerdo a las características del contexto para la cual fueron creadas. Además de proporcionar un panorama más claro de las estrategias disponibles actualmente, el análisis realizado permite a futuras investigaciones identificar cuál es la estrategia de selección de datos más apropiada de acuerdo a la aplicación que se le vaya a dar. Adicionalmente, a través de la clasificación se detectó que no existía ninguna estrategia adecuada para los contextos con un gran número de fuentes estructuradas con situaciones de solapamiento intencional y extensional y con fenómenos de replicación, permitiendo dirigir la investigación a llenar este vacío con OptiSource.

3. **ARIBEC: Sistema de mediación adaptable para OVs**. ARIBEC (ver capítulos 5 y 7) es una arquitectura de mediación creada específicamente para OVs. Esta nueva arquitectura enriquece el nivel de mediación de la arquitectura de referencia para proveer un servicio de evaluación de consultas escalable, adaptable y fácilmente utilizable por los usuarios de la OV.

   La facilidad de uso de ARIBEC se logra a través de la incorporación de una nueva interfaz del nivel de mediación que presenta a los usuarios la información disponible en la OV a través de un conjunto de objetos virtuales de datos (VDO). Un VDO es un concepto de interés en la OV que agrupa diferentes subconceptos y propiedades. La nueva noción de VDO no sólo permite brindar transparencia a los usuarios durante la definición

de consultas, sino que también es utilizado como elemento núcleo para optimizar las consultas.

ARIBEC basa su estrategia de planeación en la identificación de la cartografía de las consultas sobre VDOs. La cartografía es una estructura que contiene las fuentes que deben ser consultadas para obtener las instancias de VDO que correspondan con la consulta. La flexibilidad y escalabilidad proporcionados por OptiSource están en la manera dinámica de identificar cuál es la cartografía de una consulta usando la estrategia de selección más apropiada de acuerdo al tipo de consulta. El principio de ARIBEC, llamado multiescala, parte del principio de que cada consulta tiene una relación diferente con el contexto de datos y puede ser resuelta de forma más eficiente si se evalúa esta relación. Elegir la estrategia indicada para identificar la cartografía significa evitar usar estrategias de selección costosas para planear consultas que no involucran un gran número de fuentes a nivel intencional, y evitar usar estrategias que emplean únicamente reducción intencional en contextos complejos, como aquellos que tienen solapamiento a nivel intencional y extensional.

4. **OptiSource: Estrategia de selección de fuentes para OVs**. OptiSource es una estrategia de selección de fuentes cuyo principio es evitar la consulta de fuentes que no aportarán instancias que coincidan con la consulta. Para lograr esto selecciona únicamente las fuentes de datos dominantes para cada una de las condiciones de la consulta. Una fuente de datos es dominante si su contribución en términos de instancias es mayor que la de las otras fuentes de datos que pueden evaluar la misma condición. OptiSource puede ser utilizado dentro de ARIBEC pues dada una consulta proporciona su cartografía. Sin embargo, también puede ser utilizada por otro sistema de mediación, siempre y cuando le proporcione una interfaz para obtener el conocimiento de las fuentes.

   Para identificar las fuentes dominantes con respecto a una consulta OptiSource prioriza las fuentes de acuerdo al rol que pueden jugar dentro de la consulta, y optimiza la asignación de condiciones a fuentes usando un modelo de optimización combinatoria. Este modelo es un importante aporte al área de integración de datos pues permite simplificar el problema de selección de fuentes utilizando un modelo matemático ya comprobado. Experimentos realizados sobre este modelo permitieron validar su precisión y su buen desempeño incluso con un número alto de fuentes de datos y de condiciones.

5. **Prototipo y Experimentación**. Además del diseño realizado sobre OptiSource y ARIBEC, durante la investigación se desarrolló un prototipo funcional que permite realizar la selección de fuentes de datos siguiendo los principios de OptiSource. Este prototipo fue utilizado para validar la propuesta a través de diferentes tipos de pruebas enfocadas a validar su precisión, sensibilidad y desempeño. Los resultados de la experimentación demostraron que la precisión y exhaustividad durante la selección de fuen-

tes usando OptiSource se incrementa ampliamente (más de un 80 %) frente a las estrategias basadas en el esquema. t Así mismo, los experimentos evidenciaron que el modelo de optimización usado en OptiSource es capaz de entregar respuestas cercanas a la óptima incluso cuando el valor del beneficio estimado para una fuente es cercano pero no igual al real.

9.2. Conclusiones y Perspectivas de Investigación

El trabajo alrededor del manejo de datos en las OVs es aún muy incipiente. El esfuerzo realizado en esta investigación es un paso para desarrollar arquitecturas de integración dinámicas, flexibles y semi-automáticas. ARIBEC y OptiSource demostraron ser útiles como parte de la infraestructura de información de las OVs. Esta sección presenta los lineamientos que se creen convenientes para continuar con esta investigación a corto, mediano y largo plazo,.

A corto plazo.

Validación en diferentes infraestructuras. Como se mencionó en el Capítulo 2 las OVs pueden ser implementadas sobre diferentes infraestructuras físicas, como por ejemplo en infraestructuras de mallas de cómputo, redes nodo a nodo, infraestructuras de computación en nube (*cloud computing*)[BMQ+07], de acuerdo a las necesidades, presupuesto y posibilidad tecnológicas de las organizaciones participantes. El sistema de mediación de fuentes que se utilice en la OV debe ser eficiente en cualquiera de estas infraestructuras. Una buena continuación de este trabajo es validar la eficiencia de OptiSource en diferentes infraestructuras tecnológicas comparando en cuál de estas es más eficiente.

Finalización de los componentes de ARIBEC. El prototipo sobre el cual se realizaron los diferentes experimentos de selección incluye los componentes de OptiSource y el conjunto de componentes necesarios de ARIBEC para poder validar OptiSource. Un trabajo importante es terminar de implementar los componentes restantes de ARIBEC como el caché y el de coordinación de ejecución.

A mediano plazo.

Obtención de Roles. Como se ilustró en la descripción de OptiSource una de las entradas más importantes al proceso de estimación del beneficio son los roles que las fuentes pueden jugar durante la ejecución de una consulta. Si bien, esta investigación propone tres enfoques para capturarlos es necesario profundizar más en este aspecto. En particular, es necesario profundizar en la manera como serán interpretados los resultados obtenidos del procesamiento de consultas.

Extensión de la estimación del beneficio La estimación del beneficio de una fuente para una consulta realizada en OptiSource está determinada por el rol que ésta puede jugar en una consulta y por su tamaño relativo a otras fuentes relacionadas con la misma condición de la consulta. Para enriquecer aún más esta estimación se propone como extensión de OptiSource incorporar nuevas variables que permitan adaptar el cálculo del beneficio a las necesidades del usuario en términos de tiempo, calidad y especialidad de las fuentes. Por ejemplo, algunos usuarios podrían estar más interesados en la cantidad de res-

puestas que en la calidad, otros están más interesados en que el tipo de fuentes que provea las respuestas sea una fuente especialista en cierta área, otros sólo quieren las respuestas de las fuentes con mayores estándares de calidad en sus datos.

A largo plazo.

Enlace de componentes de OptiSource con los componentes de otras estrategias. Analizar el comportamiento de otras estrategias de selección adaptadas para ser usadas en OVs permitió identificar diferentes puntos en los cuales OptiSource puede ser enriquecido utilizando componentes de otras estrategias. En particular, OptiSource podría entregar respuestas más completas si es combinado con estrategias como la propuesta en el QPIAD [KFCK07]. Adicionalmente, el componente de estimación de beneficio de OptiSource puede ser incorporado a otras estrategias como *iDrips*[DH02] y *Caminos de Navegación*[BKN+06] para proporcionar los valores de utilidad o beneficio que requieren para su funcionamiento.

Asociación de ARIBEC con sistemas de colaboración de OVs. ARIBEC fue creado para fortalecer la infraestructura de información de las OVs. Sin embargo, para lograr un estado estable de esta infraestructura es necesario vincular el sistema de mediación a los sistemas de colaboración que actualmente soportan la interacción entre los participantes. A largo plazo se busca integrar ARIBEC como el nivel de datos de un sistema de trabajo colaborativo para OVs. Para lograr esto es necesario en primer lugar agregar la funcionalidades que permiten adicionar y actualizar datos a ARIBEC, e incorporar nuevas interfaces que satisfagan las necesidades de información de los sistemas de trabajo colaborativo. Así mismo, se pudo constatar que no es conveniente usar OptiSource en contextos en donde las propiedades están concentradas en un conjunto reducido de fuentes, pues en estos casos el esfuerzo de planeación realizado por OptiSource es innecesario cuando la elección de las fuentes que contienen las propiedades asociadas al predicado de la consulta son suficientes.

Índice alfabético

bGIOSS, 57

Caminos de Navegación, 52
Capacidad Extensional de una Fuente, 67
Capacidad Intencional de una Fuente, 67
Cartografía, 68
Conjunto de Integración, 107
Conocimiento extensional, 65
Contexto de Datos, 17
Copias difusas, 17
CORI, 57
CVV, 57

Estrategia Liviana, 77
Estrategias Orientadas a Calidad, 50
Estrategias Orientadas a Capacidades, 47
Estrategias orientadas a fuentes cooperativas, 56
Estrategias orientadas a fuentes no cooperativas, 57
Estrategias orientadas a oferta y demanda, 53

Fragmentación horizontal no disyunta, 17
Fragmentación vertical no disyunta, 17
Fuente Dominante, 96
Fuentes Compatibles, 66
Función de Distancia, 133

gGIOSS, 57

iDrips, 54, 55
Information Manifold, 47

Objeto Virtual de Datos, 66
Organización Virtual, 11
Organización Virtual en el Sector Salud, 12

QProber, 57

Relevancia Extensional, 138
Relevancia K, 139
Rol Autoridad, 84
Rol Contenedor, 85
Rol Especialista, 85

SQLB, 53
START, 57
Streamer, 54, 55

TSIMMIS, 48

VDO, 66

Bibliografía

[AGR07] Philippe Adjiman, François Goasdoué, and Marie-Christine Rousset. Somerdfs in the semantic web. *J. Data Semantics*, 8:158–181, 2007.

[AHV95] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[AHY86] Peter M. G. Apers, Alan R. Hevner, and S. Bing Yao. Optimization algorithms for distributed queries. pages 262–273, 1986.

[AKM+03] Harith Alani, Sanghee Kim, David E. Millard, Mark J. Weal, Wendy Hall, Paul H. Lewis, and Nigel R. Shadbolt. Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems*, 18(1):14–21, 2003.

[AM07] Reza Akbarinia and Vidal Martins. Data management in the appa system. *Journal of Grid Computing*, 5(3):303–317, 2007.

[AP09] Serge Abiteboul and Neoklis Polyzotis. Searching shared content in communities with the data ring. *IEEE Data Eng. Bull.*, 32(2):44–51, 2009.

[ATS04] Stephanos Androutsellis-Theotokis and Diomidis Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv.*, 36(4):335–371, 2004.

[AVAdPV10] Diego Alvarez, Natalia Valencia, José Abásolo, and María del Pilar Villamil. Generación de metadatos extensionales en organizaciones virtuales. *Proyectos de grado Ingeniería de Sistemas-Redis*, (113):36–45, 2010.

[Bea03] I. Bilykh and et. al. Can grid services provide answers to the challenges of national health information sharing? In *CASCON '03*, pages 39–53. IBM Press, 2003.

[Ber07] Freie Universitat Berlin. The d2rq plattform, http://sites.wiwiss.fu-berlin.de/suhl/bizer/d2rq/, 2007.

[Biz] Company Bizagi. Bizagi process modeler. Electronic Source. Available at: http://www.bizagi.com/.

[BKN+06] Jens Bleiholder, Samir Khuller, Felix Naumann, Louiqa Raschid, and Yao Wu. Query planning in the presence of overlapping sources. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, pages 811–828, 2006.

[BLR97] Catriel Beeri, Alon Y. Levy, and Marie-Christine Rousset. Rewriting queries using views in description logics. In *PODS '97: Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 99–108, New York, NY, USA, 1997. ACM.

[BMQ+07] Greg Boss, Padma Malladi, Dennis Quan, Linda Legregni, and Harold Hall. Cloud computing. Technical report, IBM, Octubre 2007.

[BRBT07] David Budgen, Michael Rigby, Pearl Brereton, and Mark Turner. A data integration broker for healthcare systems. *Computer Magazine*, 40(4):34–41, 2007.

[CBS+09] Jean Charlet, Audrey Baneyx, Olivier Steichen, Iulian Alecu, Christel Daniel-Le Bozec, Cédric Bousquet, and Marie-Christine Jaulent. Utiliser et construire des ontologies en médecine. le primat de la terminologie. *Technique et Science Informatiques*, 28(2):145–171, 2009.

[CC01] Jamie Callan and Margaret Connell. Query-based sampling of text databases. *ACM Trans. Inf. Syst.*, 19(2):97–130, 2001.

[CDD+04] Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena: implementing the semantic web recommendations. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 74–83, New York, NY, USA, 2004. ACM.

[CER] CERN. European organization for nucelar research, cern project web page:http://public.web.cern.ch , 2008.

[CFK+99] Ann Chervenak, Ian Foster, Carl Kesselman, Charles Salisbury, and Steven Tuecke. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. *Journal of Network and Computer Applications*, 23:187–200, 1999.

[CG89] Nicholas Carriero and David Gelernter. Linda in context. *Communications of the ACM*, 32(4):444–458, 1989.

[CG08] Jonathan-Javier Córdoba-González. Esquema de seguridad para un sistema integrado de información basado en una infraestructura grid. Master's thesis, Universidad de los Andes, http://biblioteca.uniandes.edu.co, 2008.

[CHS+95] M. J. Carey, L. M. Haas, P. M. Schwarz, M. Arya, W. F. Cody, R. Fagin, M. Flickner, A. W. Luniewski, W. Niblack, D. Petkovic, J. Thomas, J. H. Williams, and E. L. Wimmers. Towards heterogeneous multimedia information systems: the garlic approach. In *RIDE '95: Proceedings of the 5th International Workshop on Research Issues in Data Engineering-Distributed Object Management (RIDE-DOM'95)*, page 124, Washington, DC, USA, 1995. IEEE Computer Society.

[CHZ05] Kevin Chen-Chuan Chang, Bin He, and Zhen Zhang. Toward large scale integration: Building a metaquerier over databases on the web. In *CIDR '05: Proceedings of the Second Biennial Conference on Innovative Data Systems Research*, pages 44–55, 2005.

[CLB04] James Caverlee, Ling Liu, and David Buttler. Probe, cluster, and discover: Focused extraction of qa-pagelets from the deep web. In *ICDE '04: Proceedings of the 20th International Conference on Data Engineering*, page 103, Washington, DC, USA, 2004. IEEE Computer Society.

[CLC95] James P. Callan, Zhihong Lu, and W. Bruce Croft. Searching distributed collections with inference networks. In *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–28, New York, NY, USA, 1995. ACM.

[CR10] l'Union Régionale des Médecins Libéraux et l'Agence Régionale de l'Hospitalisation Conseil Régional, L'Union Régionale des Caisses d'Assurance Maladie. Dossier patient partage et reparti, 2010. http://www.sante-ra.fr/DPPR.htm.

[CXWL05] Shaowu Cheng, Xiaofei Xu, Gang Wang, and Quanlong Li. An agile method of modeling business process simulation for virtual enterprises. In *ICEBE '05: Proceedings of the IEEE International Conference on e-Business Engineering*, pages 87–93, Washington, DC, USA, 2005. IEEE Computer Society.

[DH02] AnHai Doan and Alon Y. Halevy. Efficiently ordering query plans for data integration. In *ICDE '02: Proceedings of the 18th International Conference on Data Engineering*, page 393, Washington, DC, USA, 2002. IEEE Computer Society.

[dJD+07] Laurent d'Orazio, Fabrice Jouanot, Yves Denneulin, Cyril Labbé, Claudia Roncancio, and Olivier Valentin. Distributed semantic caching in grid middleware. In *DEXA*, pages 162–171, 2007.

[dJLR07] Laurent d'Orazio, Fabrice Jouanot, Cyril Labbé, and Claudia Roncancio. Caches sémantiques coopératifs pour grilles de données. In *BDA*, 2007.

[DL02] Anhai Doan and Alon Levy. Efficiently ordering query plans for data integration. In *ICDE '02: Proceedings of the 18th International Conference on Data Engineering*, page 393, Washington, DC, USA, 2002. IEEE Computer Society.

[DNMN09] Antonio De Nicola, Michele Missikoff, and Roberto Navigli. A software engineering approach to ontology building. *Inf. Syst.*, 34(2):258–275, 2009.

[EDo06] EDonkey. Edonkey project, http://www.edonkey2000.com, 2006.

[EMH+06] Martin Eisenhardt, Wolfgang Muller, Andreas Henrich, Daniel Blank, and Soufyane El Allali. Clustering-based source selection for efficient image retrieval in peer-to-peer networks. In *ISM '06: Proceedings of the Eighth IEEE International Symposium on Multimedia*, pages 823–830, Washington, DC, USA, 2006. IEEE Computer Society.

[EP07] Andy Seaborne Eric Prud. Sparql query language for rdf, http://www.w3.org/tr/rdf-sparql-query/, 2007.

[ESIT07] Proyecto EGEE EGEE SA3 Integration Team. glite: Lightweight middleware for grid computing, http://glite.web.cern.ch/glite/, 2007.

[FKT01] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *Int. J. High Perform. Comput. Appl.*, 15(3):200–222, August 2001.

[GCGMP97] Luis Gravano, Chen-Chuan K. Chang, Héctor García-Molina, and Andreas Paepcke. Starts: Stanford proposal for internet metasearching. In *SIGMOD '97: Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, pages 207–218, New York, NY, USA, 1997. ACM.

[Gea06] Anastasios Gounaris and et al. A novel approach to resource scheduling for parallel query processing on computational grids. *Distrib. Parallel Databases*, 19(2-3):87–106, 2006.

[GGM95] Luis Gravano and Héctor García-Molina. Generalizing GlOSS to vector-space databases and broker hierarchies. In *VLDB '95: Proceedings of 21th International Conference on Very Large Data Bases*, pages 78–89, 1995.

[GGMT99] Luis Gravano, Héctor García-Molina, and Anthony Tomasic. Gloss: text-source discovery over the internet. *ACM Transactions on Database Systems*, 24(2):229–264, 1999.

[GGT96] Georges Gardarin, Jean-Robert Gruser, and Zhao-Hui Tang. Cost-based selection of path expression processing algorithms in object-oriented databases. In *VLDB '96: Proceedings of the 22th International Conference on Very Large Data Bases*, pages 390–401, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.

[GM05] Bora Gazen and Steven Minton. Autofeed: an unsupervised learning system for generating webfeeds. In *K-CAP '05: Proceedings of the 3rd international conference on Knowledge capture*, pages 3–10, New York, NY, USA, 2005. ACM.

[GMPQ+97] Hector Garcia-Molina, Yannis Papakonstantinou, Dallan Quass, Anand Rajaraman, Yehoshua Sagiv, Jeffrey D. Ullman, Vasilis Vassalos, and Jennifer Widom. The tsimmis approach to mediation: Data models and languages. *Journal of Intelligent Information Systems*, 8(2):117–132, 1997.

[Gnu06] Gnutella. Gnutella project, http://www.gnutella.com/, 2006.

[GPFLCG03] Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Corcho-Garcia. *Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

[Gri09] Earth System Grid. http://www.earthsystemgrid.org/, 2009.

[GWJD03] Leonidas Galanis, Yuan Wang, Shawn R. Jeffery, and David J. DeWitt. Locating data sources in large distributed systems. In *VLDB '03: Proceedings of the 29th international conference on Very large data bases*, pages 874–885. VLDB Endowment, 2003.

[HCH+05] Ryan Huebsch, Brent N. Chun, Joseph M. Hellerstein, Boon Thau Loo, Petros Maniatis, Timothy Roscoe, Scott Shenker, Ion Stoica, and Aydan R. Yumerefendi. The architecture of pier: an internet-scale query processor. In *CIDR '05: Proceedings of the Second Biennial Conference on Innovative Data Systems Research*, pages 28–43, 2005.

[HF01] Ramzi A. Haraty and Roula C. Fany. Query acceleration in distributed database systems. *Revista Comlombiana de Computación*, 2(1):19–34, 2001.

[HHL+03] Ryan Huebsch, Joseph M. Hellerstein, Nick Lanham, Boon Thau Loo, Scott Shenker, and Ion Stoica. Querying the internet with

pier. In *VLDB '03: Proceedings of 29th International Conference on Very Large Data Bases*, pages 321–332, 2003.

[HL05] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operations Research*. McGraw-Hill, 8th edition, 2005.

[HMS01] David J. Hand, Heikki Mannila, and Padhraic Smyth. *Principles of Data Mining*. Bradford, MIT Press, Cambridge, MA, 2001.

[HMYW04] Hai He, Weiyi Meng, Clement Yu, and Zonghuan Wu. Automatic integration of web search interfaces with wise-integrator. *The VLDB Journal*, 13(3):256–273, 2004.

[Hor05] Ian Horrocks. Owl: A description logic based ontology language. In *ICLP '05: Proceedings of the 21st Logic Programming International Conference*, pages 5–8, 2005.

[HT99] David Hawking and Paul Thistlewaite. Methods for information server selection. *ACM Trans. Inf. Syst.*, 17(1):40–76, 1999.

[IG02] Panagiotis G. Ipeirotis and Luis Gravano. Distributed search over the hidden web: hierarchical database sampling and selection. In *VLDB '02: Proceedings of the 28th international conference on Very Large Data Bases*, pages 394–405, 2002.

[IGS01] Panagiotis G. Ipeirotis, Luis Gravano, and Mehran Sahami. Probe, count, and classify: Categorizing hidden web databases. In *SIGMOD Conference*, pages 67–78, 2001.

[Int08] Health Level Seven International. Reference Information Model HL7, http://www.hl7.org/implement/standards/rim.cfm, 2008.

[JKK07] Sung-won Jung, Mi-young Kang, and Hyuk-chul Kwon. Constructing domain ontology using structural and semantic characteristics of web-table head. In *IEA/AIE'07: Proceedings of the 20th international conference on Industrial, engineering, and other applications of applied intelligent systems*, pages 665–674, Berlin, Heidelberg, 2007. Springer-Verlag.

[KAA+05] Konstantinos Karasavvas, Mario Antonioletti, Malcolm Atkinson, Neil C. Hong, Tom Sugden, Alastair Hume, Mike Jackson, Amrey Krause, and Charaka Palansuriya. *Introduction to OGSA-DAI Services*, volume 3458. June 2005.

[KC04] Ralph Kimball and Joe Caserta. *The Data Warehouse ETL Toolkit*. Wiley Publishing, Inc., 2004.

[KFCK07] Hemal Khatri, Jianchun Fan, Yi Chen, and Subbarao Kambhampati. Qpiad: Query processing over incomplete autonomous databases. In *ICDE*, pages 1430–1432, 2007.

[KLSS95] Thomas Kirk, Alon Y. Levy, Yehoshua Sagiv, and Divesh Srivastava. The information manifold. In *Proceedings of the AAAI 1995 Spring Symp. on Information Gathering from Heterogeneous, Distributed Enviroments*, pages 85–91, 1995.

[KMN+02] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):881–892, 2002.

[Kos00a] Donald Kossmann. The state of the art in distributed query processing. *ACM Comput. Surv.*, 32(4):422–469, 2000.

[Kos00b] Donald Kossmann. The state of the art in distributed query processing. *ACM Comput. Surv.*, 32(4):422–469, 2000.

[KPP04] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*, volume 1. Springer, 2004.

[LC96] Chengwen Liu and Hao Chen. A hash partition strategy for distributed query processing. In *Extending Database Technology*, pages 373–387, 1996.

[Let01] Nick Lethbridge. An i-based taxonomy of virtual organisations and the implications for effective management. *International Journal of Emerging Transdiscipline*, pages 17–24, 2001.

[LMSS95] Alon Levy, Alberto Mendelzon, Yehoshua Sagiv, and Divesh Srivastava. Answering queries using views. In *PODS '95: Proceedings of the fourteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 95–104, New York, NY, USA, 1995. ACM Press.

[LRO96] Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Querying heterogeneous information sources using source descriptions. In *VLDB '96: Proceedings of the 22th International Conference on Very Large Data Bases*, pages 251–262, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.

[LW03] Chi-Jen Lin and Ue-Pyng Wen. Sensitivity analysis of the optimal assignment. *European Journal of Operational Research*, 149(1):35–46, August 2003.

[LWZ06] Jia Liu, Yongwei Wu, and Weimin Zheng. Grid enabled data integration framework for bioinformatics research. In *GCCW '06: Proceedings of the Fifth International Conference on Grid and Cooperative Computing Workshops*, pages 401–406, 2006.

[Mak09a] Andrew Makhorin. Gnu project, glpk for java. http://glpk-java.sourceforge.net/, 2009.

[Mak09b] Andrew Makhorin. Gnu project, gnu linear programming kit. http://www.gnu.org/software/glpk/, 2009.

[MdPS08] Colombia Ministerio de Protección Social. Sistema Integral de la Protección Social SISPRO, http://www.minproteccionsocial.gov.co, 2008.

[Mow03] Abbe Mowshowitz. Virtual organization: toward a theory of societal transformation stimulated by information technology. *Ubiquity*, 2003(May):2–2, 2003.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[MSMC06] Zbigniew Michalewicz, Martin Schmidt, Matthew Michalewicz, and Constantin Chiriac. *Adaptive Business Intelligence*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[Nap06] Napster. Napster project, http://free.napster.com, 2006.

[Nea07] Tiezheng Nie and et al. Sla-based data integration on database grids. In *COMPSAC (2)*, pages 613–618, 2007.

[NEE08] NEESGrid. Nees consortium, project web page:http://neesgrid.ncsa.uiuc.edu/, 2008, 2008.

[NFL04] Felix Naumann, Johann-Christoph Freytag, and Ulf Leser. Completeness of integrated information sources. *Information Systems Journal*, 29(7):583–615, 2004.

[NLF99] Felix Naumann, Ulf Leser, and Johann Christoph Freytag. Quality-driven integration of heterogenous information systems. In *VLDB '99: Proceedings of the International Conference on Very Large Databases*, pages 447–458, Edinburgh, UK, 1999.

[OMG09] Object Management Group OMG. Bpmn specification releases, http://www.bpmn.org, 2009.

[Org07] BIRN Organization. Biomedical informatics research network (birn), 2007.

[OTZ+03] Beng Chin Ooi, Kian-Lee Tan, Aoying Zhou, Chin Hong Goh, Yingguang Li, Chu Yee Liau, Bo Ling, Wee Siong Ng, Yanfeng Shu, Xiaoyu Wang, and Ming Zhang. Peerdb: peering into personal databases. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 659–659, New York, NY, USA, 2003. ACM.

[OV99] Tamer M. Ozsu and Patrick Valduriez. *Principles of Distributed Database Systems (2nd Edition)*. Prentice Hall, January 1999.

[PAR08] Alexandra Pomares, Jose Abasolo, and Claudia Roncancio. Virtual objects in large scale health information systems. In *Studies in Health Technology and Informatics*, pages 80–89. IOS Press, 2008.

[PH01] Rachel Pottinger and Alon Y. Halevy. Minicon: A scalable algorithm for answering queries using views. *VLDB Journal*, 10(2-3):182–198, 2001.

[Pro10] EGSO Project. http://www.egso.org/, 2010.

[QRLV07] Jorge-Arnulfo Quiané-Ruiz, Philippe Lamarre, and Patrick Valduriez. Sqlb: A query allocation framework for autonomous consumers and providers. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, pages 974–985, 2007.

[RM06] John Risson and Tim Moors. Survey of research towards robust peer-to-peer networks: search methods. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 50(17):3485–3521, 2006.

[ROH99] Mary Tork Roth, Fatma Ozcan, and Laura M. Haas. Cost models do matter: Providing cost information for diverse data sources in a federated system. In *VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases*, pages 599–610, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[RRHS04] Sriram Ramabhadran, Sylvia Ratnasamy, Joseph M. Hellerstein, and Scott Shenker. Prefix hash tree: An indexing data structure over distributed hash tables. Technical report, http://berkeley.intel-research.net/sylvia/pht.pdf, 2004.

[RS97a] M. Roth and P. Schwarz. A wrapper architecture for legacy data sources. In *VLDB '97: Proceedings of 23th International Conference on Very Large Data Bases*, pages 266–275. Morgan Kaufman, 1997.

[RS97b] Mary Tork Roth and Peter M. Schwarz. Don't scrap it, wrap it! a wrapper architecture for legacy data sources. In *VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases*, pages 266–275, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[SL90] Amit P. Sheth and James A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv.*, 22(3):183–236, 1990.

[SMM+08] Pierre Senellart, Avin Mittal, Daniel Muschick, Rémi Gilleron, and Marc Tommasi. Automatic wrapper induction from hidden-web sources with domain knowledge. In *WIDM '08: Proceeding of the 10th ACM workshop on Web information and data management*, pages 9–16, New York, NY, USA, 2008. ACM.

[SPG+07] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *Web Semant.*, 5(2):51–53, 2007.

[TIM+03] Igor Tatarinov, Zachary Ives, Jayant Madhavan, Alon Halevy, Dan Suciu, Nilesh Dalvi, Xin (Luna) Dong, Yana Kadiyska, Gerome Miklau, and Peter Mork. The piazza peer data management project. *SIGMOD Rec.*, 32(3):47–52, 2003.

[TRV98] Anthony Tomasic, Louiqa Raschid, and Patrick Valduriez. Scaling access to heterogeneous data sources with DISCO. *Knowledge and Data Engineering*, 10(5):808–823, 1998.

[VRL06] M. Villamil, C. Roncancio, and C. Labb. Range Queries in Massively Distributed Data. In *Proc. Int'l WS on Grid and Peer-to-Peer Computing Impacts on Large Scale Heterogeneous Distributed Database Systems*, Krakow, Poland, September 2006.

[WBT05] Alexander Wöhrer, Peter Brezany, and A. Min Tjoa. Novel mediator architectures for grid information systems. *Future Gener. Comput. Syst.*, 21(1):107–114, 2005.

[Wie92] Gio Wiederhold. Mediators in the architecture of future information systems. *Computer Journal*, 25(3):38–49, 1992.

[WKC+07] Garrett Wolf, Hemal Khatri, Bhaumik Chokshi, Jianchun Fan, Yi Chen, and Subbarao Kambhampati. Query processing over incomplete autonomous databases. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, pages 651–662, 2007.

[WKK+09] Garrett Wolf, Aravind Kalavagattu, Hemal Khatri, Raju Balakrishnan, Bhaumik Chokshi, Jianchun Fan, Yi Chen, and Subbarao Kambhampati. Query processing over incomplete autonomous databases: query rewriting using learned data dependencies. *The VLDB Journal*, 18(5):1167–1190, 2009.

[XC98] Jinxi Xu and Jamie Callan. Effective retrieval with distributed collections. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 112–120, New York, NY, USA, 1998. ACM.

[YL96] Budi Yuwono and Dik Lun Lee. Search and ranking algorithms for locating resources on the world wide web. In *ICDE '96: Proceedings of the Twelfth International Conference on Data Engineering*, pages 164–171, Washington, DC, USA, 1996. IEEE Computer Society.

[YLGMU99] Ramana Yerneni, Chen Li, Hector Garcia-Molina, and Jeffrey Ullman. Computing capabilities of mediators. *SIGMOD Rec.*, 28(2):443–454, 1999.

[ZHC05] Zhen Zhang, Bin He, and Kevin Chen-Chuan Chang. Lightweight domain-based form assistant: querying web databases on the fly. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 97–108. VLDB Endowment, 2005.

Apéndice A

Publicaciones Realizadas

1. ALEXANDRA POMARES, VAN-DAT CUNG, CLAUDIA RONCANCIO, JOSE ABASOLO, MARIA DEL PILAR VILLAMIL GIRALDO "Source Selection in Large Scale Data Contexts: An Optimization Approach" En: International Conference on Database and Expert Systems Applications, Springer Verlag Berlin Heidelberg, próximo a ser publicado, 2010.

2. ALEXANDRA POMARES, JOSE ABASOLO PRIETO, CLAUDIA RONCANCIO, MARIA DEL PILAR VILLAMIL GIRALDO, "Selección de Fuentes de Datos en Organizaciones Virtuales". En: Conference on Information Resource Management, 2010

3. ALEXANDRA POMARES, CLAUDIA RONCANCIO, JOSE ABASOLO PRIETO, MARIA DEL PILAR VILLAMIL GIRALDO, "Knowledge based query processing" En: International Conference on Enterprise Information Systems Ponencia:Knowledge based query processing Libro:ICEIS, Springer Verlag Berlin Heidelberg , p.208 - 219 , v.24 <, fasc.N/A , 2009.

4. ALEXANDRA POMARES, JOSE ABASOLO PRIETO, MARIA DEL PILAR VILLAMIL GIRALDO, "Virtual Organizations Data Sharing Profile" Mémoires Atelier sur les Systèmes d'Information des oRganisations Etendues (INFORSID), 2009

5. ALEXANDRA POMARES, JOSE ABASOLO PRIETO, CLAUDIA RONCANCIO, MARIA DEL PILAR VILLAMIL GIRALDO, "Selección de Fuentes de Datos en Organizaciones Virtuales" En: Paradigma: Revista En Construcción De Software ISSN: 2011-0065 ed: v.3 fasc.3 ,2009.

6. ALEXANDRA POMARES, JOSE ABASOLO, CLAUDIA RONCANCIO, "Virtual Objects in Large Scale Health Information Systems" En: Studies in Health Technology and Informatics, IOS Press, p. 80-89, 2008.

7. ALEXANDRA POMARES, JOSE ABASOLO, CLAUDIA RONCANCIO, MARIA DEL PILAR VILLAMIL GIRALDO,"Dynamic Source Selection

in Large Scale Mediation Systems" En: International Conference on Database and Expert Systems Applications Ponencia: Libro:Data Management in Grid and Peer-to-Peer Systems, Springer Verlag , p.58 - 69 , v.1 <, fasc.1, 2008.

8. ALEXANDRA POMARES, JAVIER MORALES, "PASTEUR: An Indexing and Localization System for Software Components" En: Conferencia LatinoAmericana de Computación del Alto Rendimiento. Volúmen 1, 2007.

Apéndice B

Modelo de Optimización

```
param  NbSites, integer, > 0;
param NbPredicates, integer, > 0;
set Sites := 1..NbSites;
set Predicates := 1..NbPredicates;

param Benefit{i in Sites, j in Predicates};
param Resource{i in Sites, j in Predicates}, >= 0;
param MaxResource{i in Sites}, >= 0;
param  MaxAssign{i in Sites}, >= 0;
param  Solve{i in Sites, j in Predicates}, >= 0;

var x{i in Sites, j in Predicates}, binary;
var asig{i in Sites, j in Predicates}, binary;
var y{i in Sites}, binary;
var BenefitX, >=0;
var BenefitAsig, >=0;

maximize obj1: sum {i in Sites, j in Predicates} (x[i,j]
               + asig[i,j]) *  Benefit[i,j];

s.t. one{j in Predicates}: sum{i in Sites} x[i,j] = 1;
s.t. lim: sum {i in Sites} y[i] <= k;
s.t. sel{i in Sites}: sum {j in Predicates} x[i,j] >= y[i];
s.t. limRes{i in Sites}: sum{j in Predicates} Resource[i,j]
                         * asig[i,j] <= MaxResource[i];
s.t. limCap{i in Sites}: sum{j in Predicates} asig[i,j] <=
                                                  MaxAssign[i];
s.t. couple2{i in Sites, j in Predicates}: x[i,j] <= asig[i,j];
s.t. couple3{i in Sites, j in Predicates}: asig[i,j] <= y[i];

end;
```

Apéndice C

Base de Conocimiento y Metadatos en OptiSource

Los metadatos de la OV se organizan a través de una base de conocimiento. La Figura C.1 presenta las tres clases núcleo de la OV. A partir de estas tres clases se crean las clases que describirán a la OV.

C.1. Clases Raíz de la Base de Conocimiento

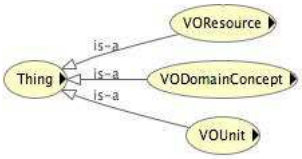

Figura C.1: Clases Núcleo de la Base de Conocimiento

1. La clase *VODomainConcept* es la superclase de las clases que describen los conceptos del dominio de la OV.
2. La clase *VOResource* es la superclase de todos los recursos de la OV.
3. El objetivo de la clase *VOUnit* es representar a las organizaciones que hacen parte de la OV.

C.2. Recursos de la Organización Virtual

De acuerdo al tipo de OV el tipo de recursos puede variar. Algunas OVs comparten recursos físicos, como recursos de cómputo o instrumentos, otras recursos

lógicos, como bases de datos. Debido a que las OVs de interés en este trabajo son las que comparten datos, los recursos más importantes en este tipo de OVs son los recursos de datos, los recursos de almacenamiento y de procesamiento. La Figura C.2 ilustra los recursos básicos de toda OV: los recursos de datos y los recursos de cómputo.

C.2.1. Clases Asociadas

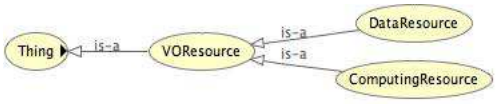

Figura C.2: Recursos de las OV

C.2.2. Propiedades *DataType*

Las clases hijas de *VOResource* tienen las siguientes propiedades:

- Identificación
- Compromiso de tiempo de respuesta
- Ubicación Física

Adicionalmente, cada clase hija tiene asociadas clases específicas. Para el caso de la clase DataResource, las propiedades asociadas son las siguientes:

- Número de instancias por VDO
- Tipo de Fuente
- Ubicación Lógica

C.3. Unidades de la Organización Virtual

Estas organizaciones, llamadas unidades de la OV, pueden ser atómicas o pueden involucrar asociaciones de organizaciones. La clase *AtomicVOUnit* representa al conjunto de participantes de la OV.

C.3.1. Clases Asociadas

La Figura C.3 presenta las clases de la OV. La clase *CompositeVOUnit* representa las asociaciones entre participantes que se realizan a lo largo del ciclo de vida de la OV.

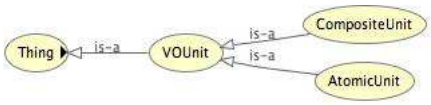

Figura C.3: Unidades de la OV

C.3.2. Propiedades *DataType*

Los individuos de la clase AtomicUnit tienen las siguientes propiedades:

- Nombre
- Rol en la OV
- Ubicación Física
- Objetivo dentro de la OV
- Fecha de Vinculación a la OV
- Fecha de Desvinculación a la OV
- Número de usuarios

Los individuos de la clase CompositeUnit tienen las siguientes propiedades:

- Nombre Asociación
- Objetivo de creación
- Fecha de Creación
- Fecha de disolución
- Proceso de Negocio

C.4. Conceptos de Dominio Organización Virtual: El Caso de la Salud

El conjunto de subclases de esta clase es el esquema global de la OV pues incluye todos los conceptos que son de interés para la OV, y sus propiedades. El número, intención y propiedades de las clases variará de una OV a otra. Estas clases representan al conjunto de individuos que se encuentran almacenados en los recursos de datos de la OV y por lo tanto no existen individuos reales de estas clases en la OV. La Figura C.4 ilustra un ejemplo de clases que heredan de la clase VODomainConcept.

Figura C.4: Subclases de VODomainConcept

C.5. Propiedades *ObjectType*

Para poder representar el conocimiento interno de los recursos o de las unidades de la OV, se declararon un conjunto de propiedades que permiten relacionar los recursos y las unidades con los conceptos de dominio. Adicionalmente, para reflejar las relaciones entre unidades y recursos fue necesario crear nuevas propiedades.

C.5.1. Propiedades que Relacionan Recursos y Unidades

- provide: Permite declarar una relación de provisión de un recurso por parte de una unidad. Sin embargo, puede tener como dominio y como rango diferentes clases.

C.5.2. Propiedades que Relacionan Unidades y Unidades

- isComposedBy: Permite declarar una relación de composición. Sin embargo, puede tener como dominio y como rango diferentes clases.

C.5.3. Propiedades que Relacionan Recursos de datos y Recursos de datos

Para declarar las relaciones de replicación entre fuentes se definieron un conjunto de propiedades que permiten expresar enq qué grado una fuente está replicada en otra.

- isReplicated: Permite declarar que una fuente está replicada en otra, pero el nivel de replicación no es conocido. Cuando este nivel es conocido se puede especializar.
- isContained: Subclase de isReplicated. Permite declarar que una fuente está contenida en otra.

C.5.4. Propiedades que Relacionan Recursos y Conceptos de Dominio

Para relacionar lo que contienen internamente los individuos de la clase VOResource, fue necesario crear un conjunto de propiedades que permitieran describir qué tanto puede aportar un recurso con respecto a un concepto de dominio. Para hacer esto se creó una propiedad básica que establece que un recurso conoce un concepto de dominio y ésta fue especializada para poder describir en qué grado existe este conocimiento.

Como se mencionó anteriormente las subclases de VODomain Concept no tienen individuos asociados en la base de conocimiento (sino en los recursos de datos), por esta razón fue necesario crear un individuo artificial por cada una de ellas que permitiera relacionar las clases con los individuos de las clases *VOResource*.

Las propiedades que permiten relacionar los individuos de estas dos clases son las siguientes:

- playsKnowledgeRole: Dominio - Recurso, Rango - VODomainConcept
  - playsAuthorityRole: Dominio - Recurso, Rango - VODomainConcept
  - playsContainerRole: Dominio - Recurso, Rango - VODomainConcept
  - playsSpecialistRole: Dominio - Recurso, Rango - VODomainConcept
  - playsContainerProbablyRole: Dominio - Recurso, Rango - VODomainConcept

La Figura C.5 ilustra un ejemplo de relación entre una subclase de *VODomainConcept* y *DataResource*.

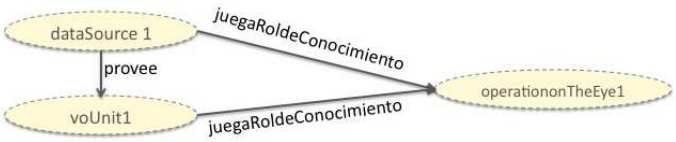

Figura C.5: Relaciones entre individuos de la clase VODomainConcept e individuos de la clase VOUnit y VOResource

C.5.5. Propiedades que Relacionan Unidades y Unidades

- isComposedBy: Dominio - CompositeUnit, Rango - AtomicUnit
- Composes: Dominio - AtomicUnit, Rango - CompositeUnit

Résumé Français

La sélection de sources de données est un des processus des plus critiques pour les systèmes de médiation dans des contextes grande échelle. C'est le cas notamment des grandes organisations virtuelles où le grand nombre de sources de données, la distribution, l'hétérogénéité, la fragmentation et la duplication des données rendent difficile l'identification des sources pertinentes à l'évaluation d'une requête. Cette thèse aborde cette problématique et propose OptiSource, une stratégie de sélection de sources de données créée pour des tels contextes. OptiSource est particulièrement performante dans des configurations où un grand nombre de sources sont susceptibles de contribuer à une requête selon leur niveau intentionnel (schéma), mais seulement un petit nombre d'entre elles peuvent effectivement le faire au niveau extensionnel (le contenu). OptiSource propose un processus itératif basé sur la sélection des sources de données dominantes pour chaque condition de la requête. Les sources dominantes sont désignées selon leur contribution attendue. Cette estimation utilise un modèle qui priorise les sources en fonction du rôle qu'elles peuvent jouer dans la requête, et optimise la répartition des sous-requêtes en utilisant un modèle d'optimisation combinatoire. OptiSource fait partie d'un système de médiation créé pour organisations virtuelles qui peut choisir dynamiquement la stratégie de sélection de sources la plus approprié au contexte. Notre domaine d'application privilégié a été le médical. Nous avons validé nos propositions sur divers types de contextes de grande taille.

English Summary

Data source selection is one of the most critical processes in mediation systems for large-scale contexts, as those found in large virtual organizations. In such contexts, the high volume of structured data sources, distribution, heterogeneity, fragmentation and replication of data difficult the identification of the relevant data sources that should evaluate a query. This thesis addresses this problem and proposes OptiSource, a strategy for selecting data sources in large scale contexts. OptiSource is particularly effective in applications where a large number of sources are likely to contribute to a query at the intentional level (schema), but only a few of them can actually do at the extensional level (content). OptiSource proposes an iterative process based on the selection of the dominant data sources for each query condition. These dominant sources are designated according to their expected contribution. In order to estimate this contribution OptiSource uses a model that prioritizes sources based on the role they can play in the query and optimizes the assignment of sub-queries using a combinatorial optimization model. OptiSource is part of a mediation system created for virtual organizations that can dynamically choose the most appropriate source selection strategy according to the context. Our domain of application was the health sector. We validated our proposals on a variety of large scale contexts.

Resumen en Español

La selección de fuentes de datos es uno de los procesos más críticos de los sistemas de mediación en los contextos de gran escala como los presentes en las organizaciones virtuales . En este nuevo modelo organizacional los contextos de datos involucran un alto número de fuentes de datos, heterogéneas, fragmentadas, distribuidas y con escenarios de replicación que dificultan la identificación de las fuentes de datos pertinentes para la evaluación de una consulta. Esta tesis aborda esta problemática y propone OptiSource, una estrategia de selección de fuentes de datos creada para este tipo de contextos. OptiSource es particularmente eficiente en contextos en donde un gran número de fuentes de datos son susceptibles de contribuir a una consulta según su nivel intencional (esquema), pero solamente una proporción reducida de ellas puede efectivamente evaluarla a nivel extensional (contenido). OptiSource propone un proceso iterativo basado en la selección de fuentes de datos dominantes para cada condición de la consulta. Las fuentes dominantes son definidas según su contribución esperada. Esta contribución es estimada utilizando un modelo que prioriza las fuentes en función del rol que pueden jugar en la consulta y asigna las sub-consultas utilizando un modelo de optimización combinatoria. OptiSource hace parte de un sistema de mediación creada para organizaciones virtuales que elige de forma dinámica la estrategia de selección de fuentes más apropiada de acuerdo al contexto. Nuestro dominio de aplicación fue alrededor del área de la salud. La validación de la propuesta se realizó sobre diferentes contextos de gran escala.