Emparejamiento Automático de Servicios Web usando Programación con Restricciones

Octavio Martín Díaz

Universidad de Sevilla

Tesis Doctoral
supervisada por
Antonio Ruiz Cortés

Sevilla, 10 de Septiembre de 2007

Primera edición, Septiembre de 2007
The Distributed Group
Dpto. Lenguajes y Sistemas Informáticos
ETSI Informática - Universidad de Sevilla
Avda. de la Reina Mercedes, s/n
Sevilla, 41012. España

Copyright © MMVII *The Distributed Group*
http://www.lsi.us.es/~octavio
octavio@lsi.us.es

Categorías (ACM 1998): C.2.4 Distributed Systems; D.2 Software Engineering, D.2.1 Requirements/Specifications, D.2.11 Software Architectures, D.2.12 Interoperability, D.2.13 Reusable Software; G.1.6 Optimization; G.4 Mathematical Software; H.3.4 Systems and Software, H.3.5 Online Information Services; I.2 Artificial Intelligence, I.2.4 Knowledge Representation Formalisms and Methods, I.2.8 Problem Solving, Control Methods, and Search; K.4.4 Electronic Commerce

Financiación: Este trabajo ha sido financiado por el Ministerio de Ciencia y Tecnología del Gobierno Español (TIC2000-1106-C02-01, TIC2003-02737-C02-01 y TIN2006-00472).

Don Antonio Ruiz Cortés, Profesor Titular de Universidad del Área de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla,

HACE CONSTAR

que don Octavio Martín Díaz, Licenciado en Informática por la Universidad de Sevilla, ha realizado bajo mi supervisión el trabajo de investigación titulado

Emparejamiento Automático de Servicios Web
usando Programación con Restricciones

Una vez revisado, autorizo el comienzo de los trámites para su presentación como Tesis Doctoral al tribunal que ha de juzgarlo.

Fdo. Antonio Ruiz Cortés
Área de Lenguajes y Sistemas Informáticos
Universidad de Sevilla
Sevilla, 10 de Septiembre de 2007

Yo, Octavio Martín Díaz, con DNI número 52.237.152-N,

DECLARO BAJO JURAMENTO

Ser el autor del trabajo que se presenta en la memoria de esta Tesis Doctoral que tiene por título:

Emparejamiento Automático de Servicios Web
usando Programación con Restricciones

Lo cual firmo en Sevilla, 10 de Septiembre de 2007.

Fdo. Octavio Martín Díaz

UNIVERSIDAD DE SEVILLA

Reunido el tribunal de evaluación de la memoria de tesis doctoral presentada por Octavio Martín Díaz para la obtención del grado de Doctor por la Universidad de Sevilla, tras votación se acordó otorgar la calificación de _____. A juicio de este tribunal y habiendo obtenido un total de ____ votos de sus miembros, se otorga a la tesis la mención de _____.

Miguel Toro Bonilla
Catedrático de Universidad
Universidad de Sevilla

Joaquín Ezpeleta Mateo
Titular de Universidad
Universidad de Zaragoza

Pablo Javier Tuya González
Titular de Universidad
Universidad de Oviedo

Vicente Pelechano Ferragud
Titular de Universidad
Universidad Politécnica de Valencia

Juan Miguel Gómez Berbis
Profesor Visitante
Universidad Carlos III de Madrid

Para que conste al interesado donde estime oportuno, firmamos la presente acta en Sevilla, a _____ .

A Nerea, la niña de mis ojos,
y sus abuelitos

Índice General

Agradecimientos ..... **XIII**

Resumen ..... **XV**

I Prefacio

1 Introducción **3**
- 1.1 Contexto de la Investigación 4
  - 1.1.1 Emparejamiento de Servicios 5
  - 1.1.2 Emparejamiento Sensible a la Calidad 6
  - 1.1.3 Programación con Restricciones 8
- 1.2 Contribuciones 9
  - 1.2.1 Resumen de Contribuciones 9
  - 1.2.2 Publicaciones en Orden Cronológico 10
  - 1.2.3 Citas de Otros Autores 13
- 1.3 Estructura de la Memoria de Tesis 16

II Antecedentes y Estado del Arte

2 Acuerdos de Calidad de Servicio **21**
- 2.1 Introducción 22
- 2.2 Acuerdos Básicos en WS-Agreement 23
  - 2.2.1 Mapa Conceptual 24
  - 2.2.2 Contexto 26
  - 2.2.3 Términos de Servicio 26
  - 2.2.4 Términos de la Garantía 27

2.3 Características de los Acuerdos Avanzados 30
2.3.1 Expresividad de los Objetivos 31
2.3.2 Simetría de los Objetivos 35
2.3.3 Controlabilidad de los Atributos 36
2.3.4 Caracterización de Atributos No-Controlables 36
2.3.5 Funciones de Utilidad Complejas 38
2.3.6 Consciencia Temporal 39
2.4 Resumen y Reseña Bibliográfica 42

3 Emparejamiento de Servicios Sensible a la Calidad 43
3.1 Introducción 44
3.2 Consistencia 45
3.3 Conformidad 48
3.3.1 Objetivos Asimétricos 48
3.3.2 Atributos Controlables 49
3.3.3 Atributos No-Controlables 50
3.3.4 Atributos Controlables y No-Controlables 50
3.3.5 Conformidad con Consciencia Temporal 53
3.3.6 Conformidad de Doble-Vía 55
3.4 Optimalidad 55
3.4.1 Grados de Similaridad 57
3.4.2 Funciones de Utilidad 59
3.4.3 Atributos Controlables y No-Controlables 61
3.4.4 Consciencia Temporal 64
3.5 Soporte Automático al Emparejamiento 65
3.5.1 Lógicas Descriptivas 65
3.5.2 Lógicas de Primer Orden 66
3.5.3 Programación Lineal 66
3.5.4 Programación con Restricciones 67
3.5.5 Métodos *Ad Hoc* 67
3.5.6 Otras Propuestas 67
3.6 Resumen y Reseñas Bibliográficas 68

4 Programación con Restricciones 69
4.1 Introducción 70
4.2 Definiciones Básicas Asociadas a los CSP 71
4.3 Resolutores 74
4.3.1 Satisfacción de Restriciones 74

4.3.2 Resolución de Restricciones 75
4.3.3 Cuantificación Universal en los Resolutores 76
4.3.4 Resolutores Declarativos e Imperativos 77
4.4 Resumen 78

III Nuestra Propuesta

5 Motivación 81
5.1 Introducción 82
5.2 Problemas 82
5.3 Análisis de las Soluciones Actuales 83
5.3.1 Abstracción 83
5.3.2 Semántica Formal 84
5.3.3 Soporte para Emparejamiento Avanzado 84
5.4 Discusión 85
5.5 Resumen y Reseña Bibliográfica 92

6 Marco de Trabajo HDM 93
6.1 Introducción 94
6.1.1 Niveles de Abstracción de HDM 94
6.2 Modelo Abstracto de HDM 95
6.2.1 Espacio de Acuerdos y Puntos de Acuerdo 96
6.2.2 Regiones de Acuerdo 97
6.2.3 Criterios de Preferencias 97
6.2.4 Probabilidades de Ocurrencia 98
6.2.5 Ejemplo Ilustrativo 98
6.3 Consistencia de una Oferta de Acuerdo 98
6.3.1 Operaciones Auxiliares 98
6.3.2 Noción de Consistencia 100
6.3.3 Ejemplos Ilustrativos 101
6.4 Conformidad 102
6.4.1 Operaciones Auxiliares 102
6.4.2 Noción de Conformidad 103
6.4.3 Ejemplos Ilustrativos 105
6.5 Optimalidad 106
6.5.1 Utilidad de un Acuerdo 107
6.5.2 Funciones de Utilidad de Puntos de Acuerdo 109

6.5.3 Distribuciones de Probabilidad de Ocurrencia 111
6.5.4 Ejemplo Ilustrativo 111
6.6 Resumen y Reseña Bibliográfica 116

7 Modelo Operacional de HDM 117
7.1 Introducción 118
7.2 Esquema General de Transformaciones 119
7.2.1 Consistencia 120
7.2.2 Conformidad 123
7.2.3 Optimalidad 124
7.3 Esquemas de Transformación Temporales 126
7.3.1 Aplanamiento de una Oferta de Calidad 126
7.3.2 Recorrido Paralelo de Ofertas de Calidad 130
7.4 Esquemas de Tranformación a CSP / CSOP 132
7.4.1 Consistencia 132
7.4.2 Conformidad 135
7.4.3 Utilidad de un Acuerdo 135
7.5 Resumen y Reseña Bibliográfica 136

8 Modelo de Implementación de HDM 139
8.1 OPL–Studio de ILOG 140
8.1.1 Características de OPL 140
8.1.2 Satisfacción de Restricciones con OPL 141
8.2 Descripción de CSP en OPL 142
8.2.1 Esquemas Básicos 142
8.2.2 Esquemas para Cuantificación Universal 142
8.2.3 Esquemas para Conformidad 146
8.2.4 Otras Consideraciones 146
8.3 Tareas de Emparejamiento en OPL 147
8.4 Resultados Experimentales 149
8.4.1 Descripción General del Experimento 150
8.4.2 Conformidad 150
8.4.3 Selección Óptima 156
8.5 Resumen 158

IV Epílogo

9 Conclusiones y Trabajo Futuro 163

9.1 Conclusiones 163
9.2 Discusión sobre Decisiones Tomadas 164
9.3 Extensiones y Trabajo Futuro 165
9.3.1 Marco de Trabajo 166
9.3.2 Nuevas Características Avanzadas 167
9.3.3 Integración con FAST 169
9.3.4 Semantizar HDM 170

V Apéndices

A Drammatis Formulae 175
A.1 Programación con Restricciones 176
A.2 Modelo del Emparejamiento de Servicios 177
A.2.1 Definición de Ofertas de Acuerdo 177
A.2.2 Operaciones sobre Ofertas de Acuerdo 178
A.2.3 Tareas del Emparejamiento de Servicios 179

B Algoritmo de Proyección Temporal 181
B.1 Introducción 182
B.2 Definición del Algoritmo 183
B.2.1 Estructuras de Datos 183
B.2.2 Transiciones y Configuraciones 183
B.2.3 Transiciones debidas a los Intervalos Temporales 184
B.2.4 Transiciones hacia Estados Terminales 188
B.3 Propiedades del Algoritmo 189
B.3.1 Determinista 189
B.3.2 Corrección 190
B.4 Complejidad del Algoritmo 190

Bibliografía 193

Índice de Figuras

1.1 Proceso de adquisición de servicios y contratación de Ludwig [115] . . 4
1.2 Resumen de líneas de investigación y publicaciones 11

2.1 Mapa conceptual de ofertas de acuerdo 23
2.2 Ejemplo de ofertas de acuerdo 24
2.3 Ejemplo de acuerdo de servicio 25
2.4 Ejemplo de oferta de acuerdo con valores de negocio 30
2.5 Mapa conceptual de ofertas de acuerdo avanzadas 31
2.6 Interpretación geométrica de objetivos de nivel de servicio 32
2.7 Grados de expresividad para describir condiciones en los objetivos . 34
2.8 Simetría en ofertas de acuerdo con términos bilaterales 35
2.9 Ejemplo de oferta de acuerdo con atributos controlables 37
2.10 Ejemplo de funciones de utilidad 38
2.11 Ejemplo de ofertas de acuerdo con consciencia temporal 39
2.12 Ejemplo de períodos globales de vigencia 40
2.13 Ejemplo de oferta de acuerdo con períodos de vigencia complejos . . 40
2.14 Ejemplo de funciones de utilidad con consciencia temporal 41

3.1 Diferentes representaciones de una oferta de acuerdo 46
3.2 Función de utilidad que no es consistente 47
3.3 Conformidad en emparejadores de única vía 49
3.4 Conformidad con atributos controlables y no-controlables 51
3.5 Conformidad con consciencia temporal 54
3.6 Selección de la mejor oferta 56
3.7 Selección óptima con atributos controlables y no-controlables. 63

6.1 Niveles del marco de trabajo HDM 95
6.2 Conceptos básicos de ofertas de acuerdo 96
6.3 Representación abstracta de una oferta de acuerdo 99

6.4 Proyecciones y filtros para obtener regiones de conformidad 104

7.1 Ejemplo de transformaciones para comprobar la consistencia 120
7.2 Ejemplo de transformaciones para comprobar la conformidad 121
7.3 Transformaciones para comprobar la consistencia 122
7.4 Transformaciones para comprobar la conformidad 123
7.5 Transformaciones para obtener la utilidad de un acuerdo 125
7.6 Traza del aplanamiento de una oferta de acuerdo 129
7.7 Traza del recorrido paralelo de dos ofertas de acuerdo 131
7.8 Transformaciones desde modelo abstracto a CSP 133
7.9 Interpretación de tareas de emparejamiento como CSP 134

8.1 Ejemplos de mapas CSP o CSOP a modelos OPL 143
8.2 Mapeando CSP cuantificado universalmente a OPL 144
8.3 Traza de resolución de un CSP cuantificado 145
8.4 Esquema de modelos y *scripts* de OPL para la conformidad 148
8.5 Esquema de modelos y *scripts* de OPL para la optimalidad 148
8.6 Datos experimentales para conformidad / objetivos asimétricos 151
8.7 Resultados del estudio de conformidad / objetivos asimétricos 152
8.8 Datos experimentales para conformidad / objetivos simétricos 153
8.9 Resultados del estudio de conformidad / objetivos simétricos 155
8.10 Datos experimentales para la selección óptima 157
8.11 Resultados del estudio de selección óptima 158

9.1 Niveles de abstracción de HDM en propuestas futuras 167
9.2 Interpretación geométrica con niveles de iso-calidad 168

B.1 Relaciones primitivas entre intervalos temporales de Allen 182

Índice de Tablas

1.1 Resumen de contribuciones agrupadas por tema 11
1.2 Resumen de contribuciones por categorías 14
1.3 Resumen de citas a nuestras publicaciones 16

3.1 Distancias obtenidas del ejemplo de la figura §3.6 58
3.2 Utilidades del ejemplo de la figura §3.6 60

5.1 Resumen comparativo de propuestas de la Web Tradicional 85
5.2 Resumen comparativo de propuestas de la Web Semántica 86
5.3 Resumen detallado de propuestas de la Web Tradicional 88
5.4 Resumen detallado de propuestas de la Web Semántica 89

8.1 Rangos de tratabilidad 159

9.1 Estado de desarrollo de las características avanzadas en HDM 166

Índice de Algoritmos

7.1 Algoritmo de aplanamiento de una oferta de acuerdo 127

Agradecimientos

Largo es el camino recorrido que ha llevado a esta tesis doctoral. Muchas experiencias que incluyen trabajo, revisiones, comentarios o simplemente el apoyo de personas que se han cruzado durante este tiempo. Entre todas ellas, quiero dar un especial agradecimiento a Antonio, antes compañero de estudios y amigo que director de tesis, con quien he compartido muchísimos ratos de buenas discusiones sobre el tema de investigación, quien no cabe duda me ha ayudado a dar forma y corregir mis erráticos, a veces tozudos, hábitos investigadores. También quiero agradecer a Miguel su sabia ayuda en los aspectos fundamentales de mi investigación.

Esencial ha sido la relación con los miembros de mi grupo de investigación. En particular, quiero expresar mi gratitud a Amador, David, Carlos y José María, con quienes he trabajado codo a codo durante el desarrollo de esta tesis. Tampoco puedo dejar de recordar al resto de compañeros del grupo, Joaquín, Pepelu, Trini, Manolo, Pablo y Sergio, con quienes de una manera u otra he tenido la oportunidad de colaborar ocasionalmente y particularmente a Benavides, por su ayuda en la gestión de los trámites de la tesis. También quiero agradecer a Corchu, J y Jesús su guía y ayuda en mis primeros pasos como investigador.

También quiero expresar mi gratitud a todos los miembros del Departamento de Lenguajes y Sistemas Informáticos, con quienes he compartido aventuras y desventuras docentes, en particular a Manolo M., Isabel y Víctor. Tampoco puedo dejar de dar las gracias a Rafael y al grupo Quivir por el soporte ofrecido para utilizar OPL-Studio, herramienta que he necesitado durante mi trabajo de investigación.

Después de tantos años, con algunos de mis compañeros ha surgido una bonita relación de amistad y confianza. No podré agradecer lo suficiente a Mayte, Rafa, Bea, Antonio T. y, sobre todo, a Toñi, su continuo apoyo en los momentos difíciles. Ellos han hecho posible que no me desanimara y aguantara hasta el final.

Muchos amigos de siempre me han ayudado durante estos años, Pepe y Mari Carmen, Paco y Rocío, Pepe y Ana Mari, Diego y Merche, Antonio N., Marisa, Mariola, Carmen, Loli, Salvador, Blas, Víctor, Miguel y Laura, Rocío N., Mari Loly, Paqui, Juan José y Mónica, Fran y Alcor, Agustín y Salud, Santos, Mari Carmen G., Juani, Miguel, Marisa M. y muchísima buena gente con las que he compartido momentos inolvidables. A todos ellos les he ido afirmando muchísimas veces de manera inequívoca que acababa "antes del verano", después les decía que "antes de navidad" o "antes de acabar el curso". Adivino que ya empiezan a respirar aliviados sabiendo que, por lo menos, no seré más pesado con la tesis y cambiaré de tema de conversación.

Finalmente, no puedo olvidar a toda mi familia y, especialmente, a mis padres, quienes con su cariño y esfuerzo han hecho posible que haya podido culminar mis estudios universitarios, a mi hermano José Antonio y Ana, cuyo mejor regalo ha sido Nerea, la niña de mis ojos, y a mis primas y primos con quienes más he compartido mis inquietudes, Pili, Tomás, Emilia, Isa, Antonio, Mari Carmen y Ana María.

Resumen

El aumento de la demanda de aplicaciones basadas en servicios web y las perspectivas sobre su crecimiento, han propiciado la aparición de nuevos desafíos en el desarrollo de este tipo de sistemas. Tal y como señala la hoja de ruta de investigación que en esta área ha marcado la Comisión Europea, el emparejamiento automático de servicios web es una de las áreas en las que aún se requieren esfuerzos de investigación. No obstante, si bien el emparejamiento funcional ha recibido tradicionalmente más atención que el emparejamiento extrafuncional o de calidad de servicio, este último ha cobrado gran interés recientemente. Una de las razones que pueden explicar esta situación está en el hecho de que son los aspectos de calidad de servicio los que marcan la diferencia y constituyen en gran medida, el valor añadido diferenciador entre servicios que ofrecen funcionalidad similar.

Por otra parte, dado que el emparejamiento extrafuncional es una actividad nuclear en el contexto de la obtención automática de acuerdos de nivel de servicio, en esta tesis doctoral hemos decidido abordar la extensión de los actuales modelos de ofertas de acuerdo de calidad de servicio con características que hacen posible abordar escenarios de emparejamiento más complejos, a saber: (**1**) la posibilidad de disponer de un lenguaje muy expresivo para descubrir objetivos de nivel de servicio (expresividad); (**2**) la posibilidad de describir los objetivos que se garantizan o se requieren con el mismo lenguaje (simetría); (**3**) la posibilidad de indicar que los clientes pueden asumir que los atributos tomen un valor concreto cuando utilicen un determinado servicio (controlabilidad); (**4**) la posibilidad de definir objetivos que ligan atributos controlables y no-controlables; (**5**) la posibilidad de caracterizar estadísticamente los atributos no-controlables; (**6**) la posibilidad de describir los valores de negocio mediante funciones de utilidad complejas y (**7**) la posibilidad de describir objetivos, funciones de utilidad y probabilidades de ocurrencias, para caracterizar los atributos no-controlables, dependientes del tiempo (consciencia temporal).

La incorporación de estas características ha obligado a extender la semántica de las actuales tareas básicas de emparejamiento (consistencia, conformidad o descubrimiento y optimalidad o selección). Dicha extensión se ha llevado a cabo sobre un marco de trabajo descrito con rigor, con estructura multinivel y que cuenta con una implementación eficiente. Este marco de trabajo lo hemos denominado HDM, acrónimo de *Holistic Decision Maker*, pues por una parte su funcionalidad esencial se corresponde con la de un tomador de decisiones y por otra, ha sido necesario utilizar una perspectiva holística para poder integrar adecuadamente las características necesarias para soportar los escenarios de emparejamiento avanzados. De no haber tomado esta perspectiva holística se habrían quedado sin descubrir algunas de las características, concretamente las que han surgido de la interrelación entre características más básicas. Por ejemplo, el uso conjunto de atributos controlables y no controlables y la inclusión de la consciencia temporal en los valores de negocio y en la caracterización de los atributos no-controlables.

Parte I

Prefacio

Capítulo 1

Introducción

A long time ago in a galaxy far, far away...

"Star Wars", 1977

En el presente capítulo, presentamos las materias que constituyen el contexto del trabajo de investigación que se ha realizado y las principales contribuciones que hemos obtenido. En la sección §1.1 introducimos el emparejamiento de servicios sensible a la calidad. Añadimos una breve reseña sobre programación con restricciones, que es el paradigma empleado para dotar de semántica operacional a las tareas de emparejamiento. En la sección §1.2 sigue un resumen de nuestras contribuciones y mostramos las publicaciones realizadas y citas recibidas durante el transcurso del presente trabajo de investigación. Finalmente, en la sección §1.3 comentamos la estructura de esta memoria de tesis.

1.1. Contexto de la Investigación

La adquisición de paquetes software en general (Finkelstein *et al.* [54, 57]) es, desde un punto de vista operativo, una actividad que engloba varias tareas, a saber: la elicitación y especificación de requisitos, el estudio del software disponible a partir de las descripciones dadas por sus proveedores, la valoración de dicho software respecto a los requisitos y la selección del mejor paquete software a partir de dichas valoraciones.

En el contexto de los servicios web, la adquisición de software sigue el proceso que se muestra en la figura §1.1 propuesto por Ludwig [115], cuya automatización es clave para que las organizaciones participen en un mercado dinámico y flexible de servicios electrónicos [51, 160, 199].

En este proceso hay implicados varios actores, a saber: los clientes que necesitan utilizar servicios, los proveedores que los proporcionan y el intermediario. Éste último se encarga de recibir las demandas y ofertas de servicio por parte de clientes y proveedores para descubrir y seleccionar la mejor oferta de servicio que satisface a cada demanda. De manera general, estas ofertas y demandas de servicio se denominan *ofertas de acuerdo.*

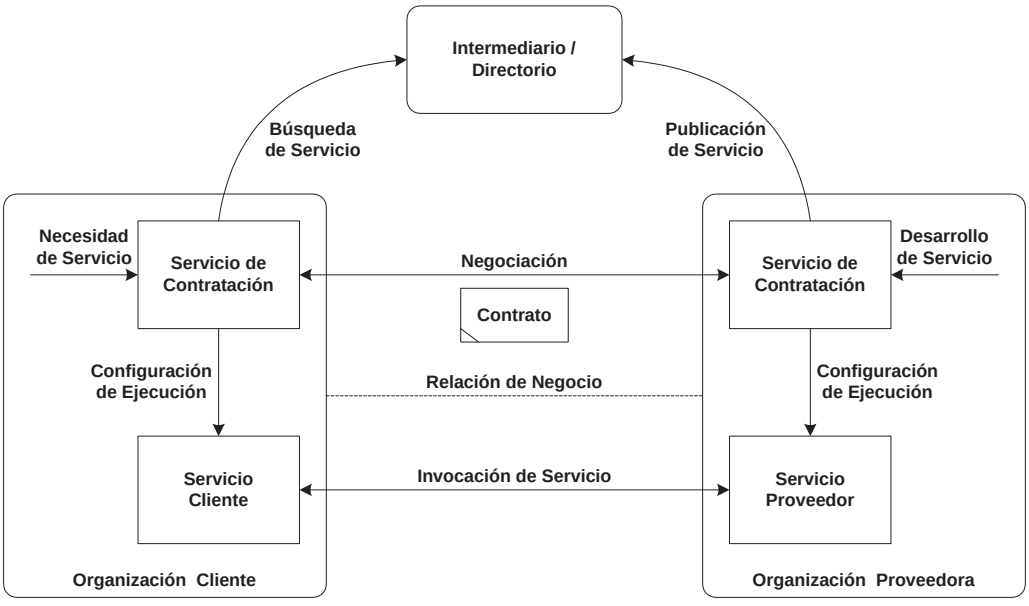


Figura 1.1: *Proceso de adquisición de servicios y contratación de Ludwig [115].*

El descubrimiento y selección se conoce como "emparejamiento de servicios" (*matchmaking*) y se puede interpretar como un problema de toma de decisión. De hecho, los componentes que dan soporte al emparejamiento de servicios suelen denominarse tomadores de decisión (*decision makers*). Según Papazoglou *et al.* el emparejamiento de servicios es un área que aún requiere esfuerzos de investigación, como se destaca en la hoja de ruta de investigación que en esta área ha marcado la Comisión Europea [158].

Una vez que un cliente y un proveedor entran en contacto, negocian los términos de su relación, hasta que finalmente se obtiene un contrato o acuerdo de nivel de servicio (SLA, *Service Level Agreement*) [119]. Los SLA establecen las obligaciones de ambas partes. Tras su firma, ambas partes tienen que configurar su sistema para darle cumplimiento efectivo, lo que podría variar de un sistema a otro según la plataforma de ejecución de cada una de las partes y del tipo de servicio contratado.

Entre las muchas propuestas para describir SLA, podemos destacar WS-Agreement [10] que es una recomendación del grupo de trabajo de Object Grid Forum[†1]. Desde el punto de vista descriptivo, proporciona una plantilla de acuerdos y deja libre la elección del lenguaje de descripción de los objetivos. También se incorporan diferentes protocolos para que proveedores y clientes lleguen a acuerdos a partir de sus ofertas de acuerdo.

1.1.1. Emparejamiento de Servicios

Según Sycara *et al.* [182], una oferta se empareja con una demanda cuando la capacidad funcional de la oferta es "suficientemente similar" a las capacidades funcionales requeridas por la demanda. En las primeras propuestas de emparejamiento, la búsqueda estaba basada en comparaciones de *palabras claves* o categorías que describían funcionalmente a los servicios. Los primeros resultados eran pobres y la búsqueda se realizaba con limitaciones, bien porque podía haber servicios que compartiesen palabras claves aunque no necesariamente ofrecieran la misma funcionalidad, o porque los servicios tenían diferentes palabras claves pero similar funcionalidad.

Solventar estas limitaciones requiere más información y extender las nociones de emparejamiento para que, por ejemplo, reconozcan *grados de similaridad* entre ofertas y demandas de servicio. Para ello, se utilizan técnicas

[†1]En esta memoria, hemos tomado la licencia de considerar "servicio web" y "servicio" como sinónimos; hay que matizar que servicio web hace referencia a la entidad computacional, mientras que servicio hace referencia al beneficio real proporcionado por el servicio web [100, 163, 165].

basadas en diferentes tipos de métricas que pretenden medir la "distancia" entre ofertas de acuerdo para determinar si hay emparejamiento.

En la Web Semántica [23], numerosos autores han argumentado la necesidad de que los servicios sean anotados semánticamente para mejorar los resultados de las búsquedas [17, 79, 138]. En este sentido, destacan Bernstein y Klein [24, 103] en la utilización de técnicas de similaridad [196] aplicadas a descripciones semánticas como se propone en [9]. En este campo, el mayor avance en el emparejamiento recae en la aplicación de técnicas de razonamiento basadas en lógicas descriptivas [14, 16, 88] o en lógicas de primer orden [42, 101]. En estas propuestas, se comprueba la existencia de ciertas relaciones semánticas entre las descripciones de servicio para determinar diversos *grados de emparejamiento semántico* entre ofertas y demandas. A su vez, estos grados permiten establecer criterios de preferencia, para seleccionar la oferta que semánticamente se aproxima más a una demanda dada, como han propuesto di Noia *et al.* [39, 45].

En general, las mejores propuestas definen esquemas híbridos que integran las técnicas hasta aquí mencionadas, a saber, palabras claves, grados de similaridad y grados de emparejamiento semántico. Destacamos el algoritmo que proponen Sycara, Paolucci *et al.* [155, 156, 183] para servicios web descritos con OWL-S [37], lenguaje que está basado en OWL (*Web Ontology Language*) [88, 137], que es una recomendación de W3C. Se describen algunos prototipos en [98, 104]. Esta propuesta está inspirada en LARKS, una de las primeras propuestas de emparejamiento en el campo de los agentes software [181].

OWL ha inspirado en mayor o menor medida otras propuestas que comparten la descripción semántica de capacidades funcionales de los servicios. Por ejemplo, METEOR-S [2, 33], WSMO [100] y MAIS [27].

1.1.2. Emparejamiento Sensible a la Calidad

Tradicionalmente, el emparejamiento funcional ha recibido más atención que el emparejamiento extrafuncional o de *calidad de servicio*. En la norma ISO–8402 *Quality Management and Quality Assurance, Vocabulary*, incorporado posteriormente al ISO-9000, la calidad de servicio se describe como la "totalidad de caracteristicas de un producto o servicio que son relevantes para su capacidad de satisfacer las necesidades ya establecidas o implícitas".

En la actualidad, éste empieza a ser un tema de interés presente en casi todos los foros internacionales, principalmente porque son los aspectos de calidad de servicio los que marcan la diferencia y constituyen en gran medida

el valor añadido diferenciador entre servicios que ofrecen similar funcionalidad [123, 206]. Tal y como señala la hoja de ruta de investigación que en esta área ha marcado la Comisión Europea, el emparejamiento de servicios web sensible a la calidad se destaca como uno de los principales retos [158].

Entre los pioneros en el emparejamiento de servicios sensible a la calidad podemos encontrar los trabajos propuestos por Mani y Nagarajan [123], Lee *et al.* [111] y Ran [164]. En estos modelos de emparejamiento, los atributos de calidad se reducían a unos cuantos, a saber, la seguridad, la fiabilidad, el rendimiento, entre otros. Las ofertas de acuerdo estaban basadas en pares atributo / valor que determinaban una única posibilidad de acuerdo.

Posteriormente, han surgido lenguajes específicos basados en XML que han tenido gran impacto en la comunidad. WSLA (*Web Service Level Agreement*) [119], WSML (*Web Service Management Language*) [169] y WSOL (*Web Services Offerings Language*) [189] son las propuestas más prominentes. Sus modelos de emparejamiento no restringen los atributos de calidad como hacían las primeras y las ofertas de acuerdo se pueden describir con mayor expresividad. En estas propuestas ya empieza a hablarse de descubrimiento y selección de servicios. Por otro lado, Ruiz-Cortés *et al.* [167] introducen la consistencia, conformidad y optimalidad como las tareas básicas del emparejamiento. Podemos identificar la conformidad con el descubrimiento de las ofertas de acuerdo que satisfacen una demanda, y la optimalidad con la selección de la mejor oferta de acuerdo para una demanda. A lo que conocemos, la consistencia para detectar errores en la descripción de una oferta de acuerdo no ha sido considerada por la mayor parte de estas propuestas.

En la Web Semántica también hay propuestas de emparejamiento sensible a la calidad. Destacamos los trabajos de Trastour *et al.* [70, 190] y Li y Horrocks [112] que extienden OWL-S, y los trabajos de Wang *et al.* [203] y Vu *et al.* [201, 202] que hacen lo propio con WSML (*Web Service Modeling Language*) [43]. Éste es una familia de lenguajes basados en WSMO [165], una interesante propuesta de ontologías para la descripción y gestión de servicios web.

Todas las propuestas mencionadas abordan la extensión de los modelos de ofertas de calidad de servicio con características útiles para abordar escenarios de emparejamiento más complejos, que particularmente tienen en cuanta la calidad de servicio. Principalmente, nos referimos a características tales como:

- *Expresividad*. Posibilidad de disponer de un lenguaje muy expresivo para describir objetivos de nivel de servicio (SLO, *Service Level Objective*).
- *Simetría*. Posibilidad de describir los SLO que se garantizan o requieren con el mismo lenguaje.

- *Controlabilidad* Posibilidad de indicar que los clientes puedan asumir que los atributos tomen un valor concreto cuando utilicen un determinado servicio. Aquí podemos añadir además:
  - Posibilidad de definir SLO que ligan atributos controlables y no-controlables.
  - Posibilidad de caracterizar estadísticamente los atributos no-controlables.
- *Funciones de Utilidad*. Posibilidad de describir los criterios de preferencias mediante funciones de utilidad complejas.
- *Consciencia Temporal*. Posibilidad de que diferentes partes de las ofertas de acuerdo tengan dependencia temporal, a saber: los SLO, las funciones de utilidad y las probabilidades de ocurrencia que caracterizan estadísticamente a los atributos no-controlables.

1.1.3. Programación con Restricciones

La programación con restricciones [193] ha sido campo de investigación de la inteligencia artificial desde hace décadas, y está reconocida como una de las direcciones de investigación estratégicas por la ACM (*Association for Computing Machinery*) desde 1996 [77].

En este contexto, un problema de satisfacción de restricciones (CSP, *Constraint Satisfaction Problem*) viene dado por un conjunto de variables, dominios finitos y restricciones que establecen relaciones entre dichas variables, de manera que se restrigen los valores que pueden tomar. Toda solución de un CSP consiste en la asignación de un valor a cada variable, dentro de sus dominios y satisfaciendo todas las restricciones.

Los resolutores de restricciones permiten comprobar si un CSP es satisfactible, es decir, si tiene alguna solución; también permiten obtener todas las soluciones o las soluciones óptimas según una función-objetivo. En este último caso, hablamos de problemas de optimación.

La programación con restricciones es nuevo paradigma con un firme fundamento teórico que ha demostrado una gran capacidad para resolver problemas complejos y se ha aplicado en múltiples áreas. De esta manera, podemos interpretar las tareas de emparejamiento de servicios como CSP, como mostramos en el capítulo §7.

1.2. Contribuciones

1.2.1. Resumen de Contribuciones

Nuestro trabajo se ha centrado en el emparejamiento automático de servicios sensible a la calidad. Tras el estudio realizado sobre las propuestas actuales, hemos llegado a la conclusión de que existen problemas que deben ser resueltos. Al objeto de resolver estos problemas, hemos definido un marco de trabajo HDM (*Holistic Decision Maker*). Su denominación es debida a que, por una parte su funcionalidad esencial se corresponde con la de un tomador de decisiones y por otra, ha sido necesario utilizar una perspectiva holística para poder integrar adecuadamente las características necesarias para soportar los escenarios de emparejamiento avanzados. De no haber tomado esta perspectiva holística se habrían quedado sin descubrir algunas de las características, concretamente las que han surgido de la interrelación entre características más básicas. Por ejemplo, el uso conjunto de atributos controlables y no controlables y la inclusión de la consciencia temporal en los valores de negocio y en la caracterización de los atributos no-controlables.

Este marco de trabajo se ha dividido en niveles de abstracción, a saber: modelo abstracto, modelo operacional y modelo de implementación. Agrupamos nuestras contribuciones por dichos niveles:

- *Modelo Abstracto*:
  - Hemos definido un modelo abstracto del emparejamiento que alcanza todas las características avanzadas descritas en la sección §1.1.2.
  - Hemos sido pioneros en hacer una interpretación geométrica de las ofertas de acuerdo, que permite una metáfora visual que facilita la mejor comprensión del emparejamiento.
- *Modelo Operacional*:
  - Hemos definido un modelo operacional mediante la interpretación de las tareas de emparejamiento como problemas de satisfacción con restricciones. La utilización de este paradigma para el emparejamiento también es novedosa.
  - Hemos definido las transformaciones entre el modelo abstracto y operacional. De manera particular, destacamos el algoritmo de proyección temporal necesario para dar soporte a la consciencia temporal.

- *Modelo de Implementación*:
  - Hemos definido un modelo de implementación donde se ha utilizado OPL-Studio de ILOG como resolutor de restricciones. Hemos definido los esquemas de transformación necesarios para obtener los CSP de consistencia, conformidad y optimalidad.
  - Hemos definido un tratamiento específico para resolver los problemas de satisfacción de restricciones con cuantificadores universales, cuestión necesaria para resolver la conformidad en contextos de emparejamiento avanzado.
  - Hemos mostrado la viabilidad de la resolución de restricciones en el emparejamiento de servicios mediante un estudio empírico sobre nuestra prueba de conceptos.
  - Hemos implementado una prueba de conceptos, aunque sólo incluye algunas características avanzadas como la expresividad, la simetría y las funciones complejas.
- *Marco de Trabajo HDM*:
  - Hemos definido un marco de trabajo para el emparejamiento avanzado de servicios, que nos permite resolver los problemas que se plantean en el capítulo §5.
  - Hemos propuesto HDM como referencia útil para el estudio de propuestas similares y futuras revisiones del estado del arte.
  - Hemos presentado una revisión actualizada del estado del arte, que nos ha permitido conocer los problemas que posteriormente hemos resuelto.

Estas contribuciones se han publicado parcialmente en revistas y congresos, como se muestra en la tabla-resumen §1.1. Sin embargo, el marco de trabajo HDM completo aún no ha sido publicado. A continuación, detallamos cada una de las publicaciones.

1.2.2. Publicaciones en Orden Cronológico

Nuestra primera publicación relevante fue en el TES'03 [127], taller de trabajo internacional en el que presentábamos los aspectos de implementación del modelo de emparejamiento básico propuesto por Ruiz-Cortés en su tesis doctoral [166]. Algunos resultados preliminares fueron publicados en sendos talleres de trabajo nacionales, ZOCO'02 [128] y WEBE'02 [132]:

Contribución	Publicaciones	Citas
Nivel Abstracto	ICSOC'03, JISBD'03, ICSOC'05, IJCIS'05	15
Nivel Operacional	JSWEB'07, ICSOC'07 (x2)	–
Nivel Implementación	WEBE'02, ZOCO'02, TES'03	1
Marco de Comp./Clasif.	WQW'02, ZOCO'06	2
Total	12 Publicaciones	**18**

Tabla 1.1: *Resumen de contribuciones agrupadas por tema.*

TES'03 O. Martín-Díaz, A. Ruiz-Cortés, D. Benavides, A. Durán, M. Toro, A Quality-Aware Approach to Web Services Procurement, *4th Intl. VLDB Workshop Technologies for E-Services*, Springer Verlag LNCS 2819, 42–53, Berlín, Alemania, Septiembre 2003.

ZOCO'02 O. Martín-Díaz, A. Ruiz-Cortés, R. Corchuelo, A. Durán, A Management and Execution Environment for Multi-Organisational Web-based Systems, *JISBD ZOCO: Métodos y Herramientas para el Comercio Electrónico*, 79–88, San Lorenzo del Escorial, España, Noviembre 2002.

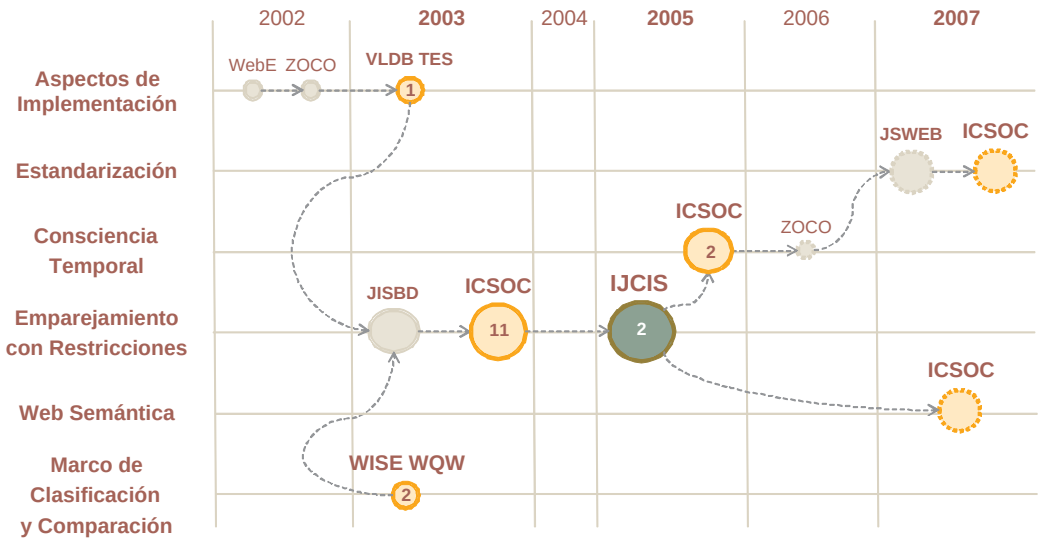


Figura 1.2: *Resumen de líneas de investigación y publicaciones.*

WEBE'02 O. Martín-Díaz, A. Ruiz-Cortés, M. Toro, Un Entorno de Administración y Ejecución de Sistemas Multi-Organizacionales basados en la Web, *JISBD Workshop on Web Engineering*, 27–40, San Lorenzo del Escorial, España, Noviembre 2002.

En el ICSOC'03, primera edición de esta emergente conferencia internacional sobre computación orientada a servicios, presentamos nuestro soporte operacional del emparejamiento de servicios basado en resolutores de restricciones [130]. Fue seleccionado entre los mejores para su extensión, revisión y publicación en la revista con índice de impacto IJCIS [168], donde se añadió un estudio de la viabilidad de los resolutores de restricciones en el contexto del emparejamiento. Este trabajo se presentó previamente en la JISBD'03 [126]:

ICSOC'03 O. Martín-Díaz, A. Ruiz-Cortés, A. Durán, D. Benavides, M. Toro, Automating the Procurement of Web Services, *1st Intl. Conf. on Service-Oriented Computing*, Springer-Verlag LNCS 2910, 91–103, Trento, Italia, Diciembre 2003.

IJCIS'05 A. Ruiz-Cortés, O. Martín-Díaz, A. Durán, M. Toro, Improving the Automatic Procurement of Web Services using Constraint Programming, *Intl. Journal of Cooperative Information Systems*, 14(4): 439–467, Diciembre 2005.

JISBD'03 O. Martín-Díaz, D. Benavides, J. Peña, M. Toro, Un Tratamiento Sensible a la Calidad para la Adquisición de Servicios Web, *VIII Jornadas de Ingeniería del Software y Bases de Datos*, 209–220, Alicante, España, Noviembre 2003.

En WQW'03, taller de trabajo internacional, se presentó nuestro estudio preliminar del estado del arte [129]; las características de interés que se propusieron conformaban un marco de clasificación y comparación preliminar:

WQW'03 O. Martín-Díaz, A. Ruiz-Cortés, R. Corchuelo, M. Toro, A Framework for Classifying and Comparing Web Services Procurement Platforms, *1st IEEE Intl. WISE Web Services Quality Workshop*, 156–164, Roma, Italia, Diciembre 2003.

En el ICSOC'05, presentamos una mejora de nuestro modelo de solución que incorporaba consciencia temporal [131]. Posteriormente, presentamos un trabajo sobre grados de descripción de consciencia temporal en el ZOCO'06 [142]. Este trabajo constituye el comienzo de una línea de investigación centrada en la mejora de WS-Agreement con consciencia temporal, que se ha presentado en el JSWEB'07 [143] y en el ICSOC'07 [144]:

ICSOC'05 O. Martín-Díaz, A. Ruiz-Cortés, A. Durán, C. Müller, An Approach to Temporal-Aware Procurement of Web Services, *3rd Intl. Conf. on Service-Oriented Computing*, Springer-Verlag LNCS 3826, 170–184, Amsterdam, Holanda, Diciembre 2005.

ZOCO'06 C. Müller, O. Martín-Díaz, A. Ruiz-Cortés, J.M. García, Consistencia y Conformidad en un Contexto Temporal, *JISBD ZOCO: Métodos y Herramientas para el Desarrollo de Aplicaciones*, 15–24, Sitges, España, Octubre 2006.

JSWEB'07 C. Müller, O. Martín-Díaz, M. Resinas, P. Fernández, y A. Ruiz-Cortés, A WS–Agreement Extension for Specifying Temporal Properties in SLA's, *III Jornadas Científico-Técnicas en Servicios Web y SOA*, Zaragoza, España, Septiembre 2007.

ICSOC'07 C. Müller, O. Martín-Díaz, A. Ruiz-Cortés, M. Resinas, y P. Fernández, Improving Temporal–Awareness of WS–Agreement, *5th Intl. Conf. on Service-Oriented Computing*, Springer-Verlag LNCS 4749, 193–206, Viena, Austria, Septiembre 2007.

En el ICSOC'07 también hemos presentado nuestro trabajo sobre emparejadores híbridos [66]; constituye otra línea de investigación paralela en la que estamos estudiando la integración del emparejamiento funcional de servicios web semánticos y el emparejamiento extra-funcional basado en resolutores de restricciones:

ICSOC'07 J.M. García, D. Ruiz, A. Ruiz-Cortés, O. Martín-Díaz, y M. Resinas, An Hybrid, QoS-Aware Discovery of Semantic Web Services using Constraint Programming, *5th Intl. Conf. on Service-Oriented Computing*, Springer-Verlag LNCS 4749, 69–80, Viena, Austria, Septiembre 2007.

La figura §1.2 y la tabla §1.2 resumen la línea temporal de la investigación desarrollada y las publicaciones que se han realizado.

1.2.3. Citas de Otros Autores

Nuestras publicaciones ya han sido citadas en los principales foros de la computación orientada a servicios:

Categoría	**Número**	**Acrónimos**
Revistas con factor de impacto	1	IJCIS'05
Conferencias internacionales	4	ICSOC'03 (LNCS), ICSOC'05 (LNCS), ICSOC'07 (LNCS) (x2)
Talleres de trabajo internacionales	2	TES'03 (LNCS), WQW'03
Conferencias nacionales	2	JISBD'03, JSWEB'07
Talleres de trabajo nacionales	3	ZOCO'02, WEBE'02, ZOCO'06
Total	**12**	

Tabla 1.2: *Resumen de contribuciones por categorías.*

- Nuestro trabajo en ICSOC'03 [130] puede considerarse como la contribución más influyente porque es la publicación más citada hasta el momento. De hecho, es el segundo trabajo más citado del ICSOC según consta en el *ISI Web of Knowledge*[†2] y está indexado en el *citeSeer*. En concreto, ha sido citado por:

  1. P.A. Bonnati, P. Festa, On Optimal Service Selection, *14th Intl. World Wide Web Conference*, 530–538, Chiba, Japón, Mayo 2005.
  2. M. Aiello, P. Giorgini, Applying the Tropos Methodology for Analysing Web Services Requirements and Reasoning about Qualities of Services, *UPGRADE*, 5(4): 20–26, Agosto 2004.
  3. M. Aiello, G. Frankova, D. Malfatti, What's in an Agreement? An Analysis and an Extension of WS-Agreement, *3st Intl. Conf. on Service-Oriented Computing*, Springer-Verlag LNCS 3826, 424–436, Amsterdam, Holanda, Diciembre 2005.
  4. D. Bianchini, V. de Antonellis, M. Melchiori, QoS in Ontology-Based Service Classification and Discovery, *3rd Intl. DEXA Workshop on Web Semantics*, 145–150, Zaragoza, España, Agosto / Septiembre 2004.
  5. G. Swart, B. Aziz, S.N. Foley, J. Herbert, Trading Off Security in a Service-Oriented Architecture. *19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, Springer-Verlag LNCS 3654, 295–309, Storrs, CT, Agosto 2005.
  6. N. Thio, S. Karunasekera, Automatic Measurement of a QoS Metric for Web Service Recommendation, *16th Australian Software En-*

[†2]Esta información data del 18 de julio de 2007.

gineering Conference, 202–211, Brisbane, Australia, Marzo / Abril 2005.

7. K. Kritikos, Extending OWL for QoS-based Web Service Description and Discovery, *ICSOC IBM PhD Student Symposium*, CEUR-WS 169, Amsterdam, Holanda, Diciembre 2005.
8. G. Frankova, Web Service Quality Composition Modelling, *ICSOC IBM PhD Student Symposium*, CEUR-WS 169, Amsterdam, Holanda, Diciembre 2005.

- Nuestro trabajo en WQW'2003 [129] ha sido citado por:

9. V. Tosic, *Service Offerings for XML Web Services and Their Management Application*, Tesis Doctoral, Carleton University, Canadá, Agosto 2004.
10. V. Tosic, WSOL versus Related Work, Informe Técnico SCE-04-07, Carleton University, Canadá, Junio 2004.

- Nuestro trabajo en TES'2003 [127] ha sido citado por:

11. K. Kritikos, D. Plexousakis, Semantic QoS Metric Matching, *4th IEEE European Conference on Web Services*, 265–274, Zurich, Suiza, Diciembre 2006.

- Nuestro trabajo en ICSOC'2005 [131] ha sido citado por:

12. N. Kokash, A Service Selection Model to Improve Composition Reliability, *4th Intl. ECAI Workshop on AI for Service Composition*, 9–14, Riva del Garda, Italia, Agosto 2004.
13. N. Kokash, A Dynamic Model for QoS-driven Web Service Composition, Informe técnico, Universidad de Trento, 2006.

- Nuestro trabajo en IJCIS [168] ha sido citado por:

14. P. Fernández-Montes, M. Resinas-Arias de Reyna, R. Corchuelo-Gil, Towards an Automated Trading Process, *UPGRADE*, 7(5):29–33, Octubre 2006.
15. K. Kritikos and D. Plexousakis, A Semantic QoS-based Web Service Discovery Engine for Over-Constrained QoS Demands, *1st Intl. ICSOC Workshop on Non Functional Properties and Service Level Agreements*, Vienna, Austria, Septiembre 2007.

La figura §1.3 muestra un resumen de las citas que hemos recibido, según la categoría de la fuente.

Categorías	Número
Revistas	2
Conf. internacionales indexadas	3
Otras conferencias y talleres internacionales	8
Libros y tesis doctorales	1
Informes técnicos y otras publicaciones	4
Total	**18**

Tabla 1.3: *Resumen de citas a nuestras publicaciones.*

1.3. Estructura de la Memoria de Tesis

Esta memoria de tesis doctoral se ha estructurado de la siguiente manera:

Prefacio Esta parte sólo comprende el presente capítulo introductorio.

Antecedentes y Estado del Arte Esta parte tiene como objetivo presentar una base de conocimientos para la comprensión de la memoria de tesis. El capítulo §2 presenta el conjunto de características avanzadas para la descripción de ofertas de calidad. A su vez, el capítulo §3 presenta las tareas de emparejamiento junto a los diferentes tipos de emparejadores. Posteriormente, el capítulo §4 muestra una breve introducción a la programación con restricciones, que es la técnica empleada para describir el modelo operacional de HDM.

Nuestra Propuesta Esta parte tiene como objetivo presentar los niveles de abstracción que conforman HDM. El capítulo §5 justifica la necesidad de HDM y su arquitectura para superar los problemas de las propuestas actuales, e introduce las características avanzadas que, hasta lo que conocemos, no han sido abordadas por dichas propuestas.

El capítulo §6 presenta nuestra definición rigurosa del modelo abstracto de HDM. El capítulo §7 muestra el modelo de operación, donde interpretamos las tareas de emparejamiento como problemas de satisfacción de restricciones. El capítulo §8 expone la utilización de un resolutor de restricciones para resolver las tareas de emparejamiento y ofrece los resultados experimentales de las pruebas empíricas a las que hemos sometido a nuestro prototipo, para demostrar la viabilidad de nuestra propuesta.

Epílogo Esta parte sólo comprende el capítulo conclusivo donde presentamos las líneas de trabajo futuro abiertas con esta tesis doctoral.

Apéndices Esta última parte tiene como objetivo presentar información complementaria a la memoria. El apéndice §A muestra un cuadro que resume las notaciones utilizadas en nuestro modelo del problema. El apéndice §B muestra una definición rigurosa del algoritmo de proyección temporal, junto a la demostración de que es determinista, correcto y su comportamiento computacional es de orden lineal.

Parte II

Antecedentes y Estado del Arte

Capítulo 2

Acuerdos de Calidad de Servicio

‘ The first part of the party of the first part
shall be known in this contract
as the first part of the party of the first part
Look! why should we quarrel about a thing like this?
We’ll take it right out, eh?
Yeah! It’s a too long, anyhow ...’

Groucho & Chico Marx, Driftwood & Fiorello
“A Night at the Opera”, 1935

En este capítulo presentamos los acuerdos de calidad de servicio para un modelo de emparejamiento avanzado. En la sección §2.1 se hace una introducción a los lenguajes de descripción de acuerdos. A continuación, en la sección §2.2 se presenta WS-Agreement, recomendación de estándar que hemos tomado como referencia para introducir qué entendemos por acuerdo básico. Posteriormente, en la sección §2.3 se detallan las características para un modelo de acuerdos avanzado, a saber: la expresividad y simetría de objetivos, la controlabilidad de los atributos, la caracterización estadística de los atributos no-controlables, los valores de negocio con funciones de utilidad complejas y la consciencia temporal. Estas características se han tomado como marco de clasificación y comparación para el estudio del estado del arte que hemos realizado. En particular, resaltamos la interpretación geométrica de los objetivos de nivel de servicio que nos permite comparar entre diferentes grados de expresividad para describirlos. Finalmente, en la sección §2.4 se muestran nuestras conclusiones.

2.1. Introducción

El uso comercial de los servicios web suele estar regulado por los que se han dado en llamar acuerdos de nivel de servicio (SLA, *Service Level Agreement*). Según Ludwig *et al.* un SLA recoge las obligaciones de cada una de las partes, los métodos para medir su grado de cumplimiento, su modelo de facturación y las medidas sancionadoras en caso de violación de los términos del contrato [119]. En el contexto del emparejamiento de servicios sólo estamos interesados en la descripción de obligaciones de cada una de las partes.

Existen numerosos modelos para describir acuerdos en general, y descripción de obligaciones en particular, siendo pioneros los propuestos por Mani y Nagarajan [123], Lee *et al.* [111] y Ran [164]. Existen algunas propuestas anteriores para describir lo que podrían entenderse como proto-SLA. Por ejemplo, QML (*QoS Modeling Language*) [63, 64] y QDL (*QoS Description Language*) [217], ambos en el contexto de CORBA y los sistemas de objetos distribuidos; y QRL (*Quality Requirements Language*) [166, 167] en el ámbito de los requisitos no-funcionales, que también se ha aplicado en el contexto de la Web.

Posteriormente, han surgido lenguajes específicos basados en XML que han tenido gran impacto en la comunidad. WSLA (*Web Service Level Agreement*) fue propuesto por Ludwig *et al.* [99, 119], permite a clientes y proveedores describir objetivos de nivel de servicio y se centra fundamentalmente en la descripción y monitorización de acuerdos de calidad. WSML (*Web Service Management Language*) propuesto por Sahai *et al.* [169, 170], también dedicado a la descripción y monitorización de acuerdos. WSOL (*Web Services Offerings Language*) propuesto por Tosic *et al.* [186, 189], extiende WSDL para definir ofertas de un mismo servicio con diferentes niveles de acuerdo.

En el ámbito de la Web Semántica, destacamos OWL-S (*OWL for Services*) [37], lenguaje basado en OWL para describir los servicios web. Existen extensiones que incorporan la descripción de acuerdos de calidad, por ejemplo, Trastour *et al.* [70, 190] y Li y Horrocks[112]. Por otra parte, WSML (*Web Service Modeling Language*) [43] es una familia de lenguajes de descripción semántica de servicios basados en WSMO [165] que, aunque también está centrado en aspectos funcionales, hay extensiones para describir aspectos de calidad, por ejemplo, Wang *et al.* [203] y Vu *et al.* [201, 202].

2.2. Acuerdos Básicos en WS-Agreement

WS-Agreement [10] es una recomendación propuesta por el grupo de trabajo del Object Grid Forum, que proporciona una estructura para describir ofertas de acuerdo y un protocolo para obtener acuerdos de nivel de servicio. Su diseño ha estado fuertemente influenciado por WSLA.

WS-Agreement ha tenido mucha aceptación. Entre otras propuestas que la han utilizado, podemos destacar CREMONA, propuesta por Ludwig *et al.* [116, 117] que incluye una arquitectura para la gestión del ciclo de vida de los acuerdos. También Aiello *et al.* [3, 4] proponen una extensión de la recomendación para incluir términos de negociación. Por último, Oldham *et al.* [149] lo aplican en METEOR-S [140], referente del emparejamiento en la Web Semántica.

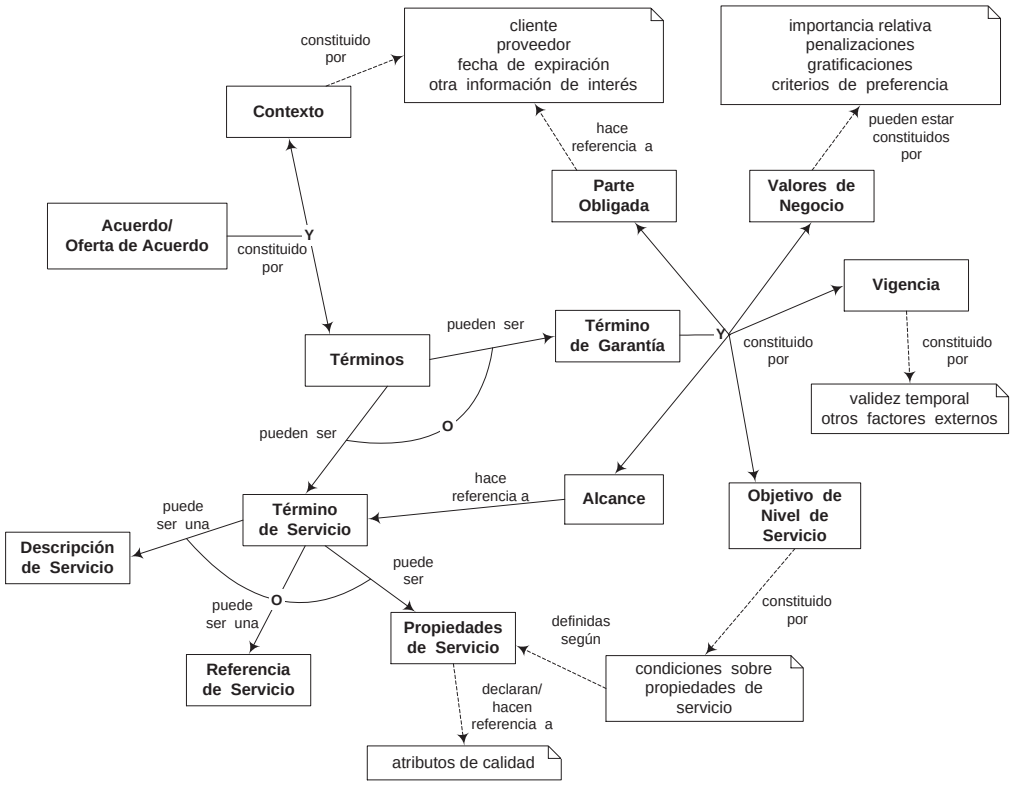


Figura 2.1: *Mapa conceptual de ofertas de acuerdo.*

2.2.1. Mapa Conceptual

La figura §2.1 muestra el mapa conceptual para describir ofertas de acuerdo en WS-Agreement, tomado como referencia para un modelo de acuerdos básico. En esta recomendación, una oferta de acuerdo tiene dos partes claramente diferenciadas: el contexto y los términos. A su vez, los términos pueden distinguirse entre los que describen los servicios objeto del acuerdo (*términos de servicio*) y los que describen las obligaciones (*términos de la garantía*) respecto a dichos servicios.

La figura §2.2 muestra un ejemplo de dos ofertas de acuerdo. La figura §2.3 muestra el acuerdo que finalmente podría establecerse tras seguir un proceso de emparejamiento. En WS-Agreement, los acuerdos siguen un esquema simi-

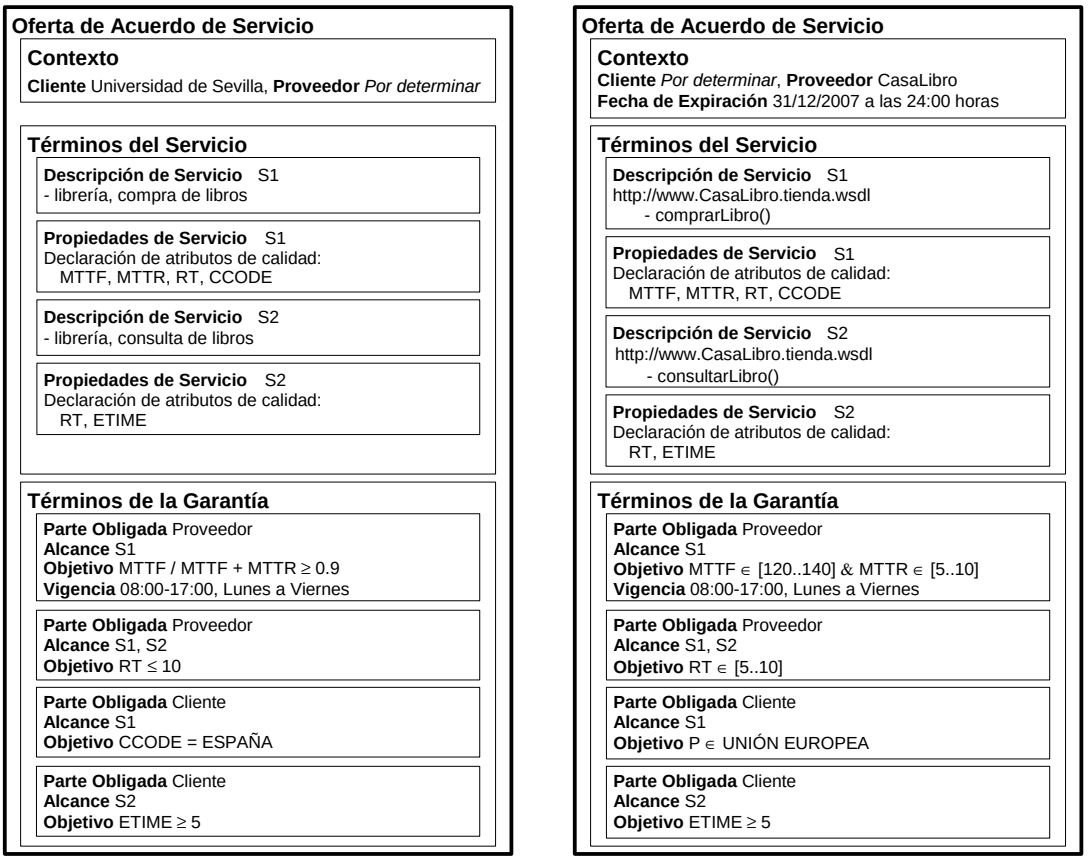


Figura 2.2: *Ejemplo de ofertas de acuerdo.*

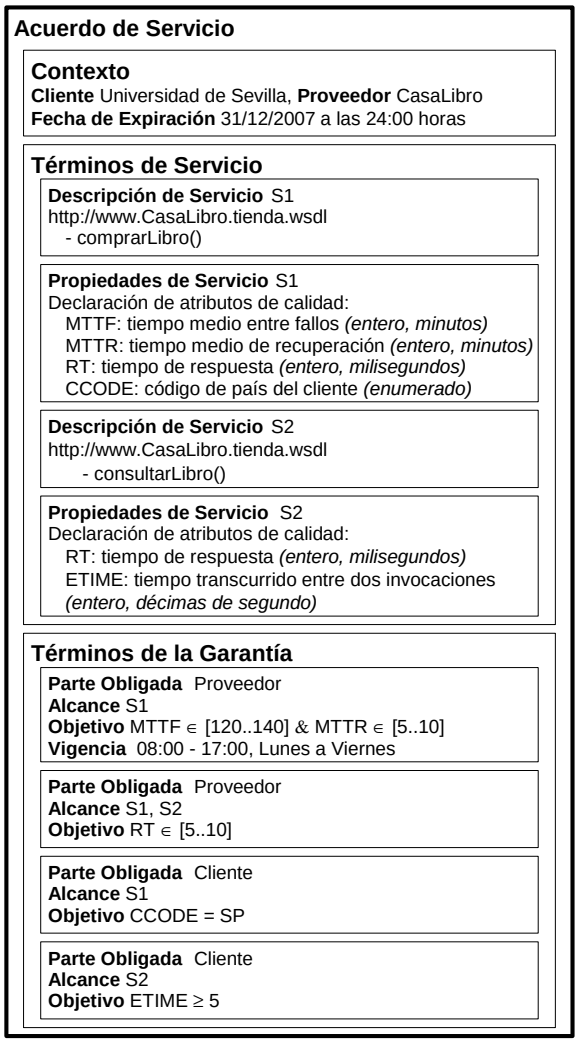


Figura 2.3: *Ejemplo de acuerdo de servicio.*

lar a las ofertas de acuerdo[†1]. En este caso, tanto el acuerdo como las ofertas de acuerdo son *bilaterales*, es decir, recogen obligaciones de ambas partes. Hay que decir que también es habitual en numerosas propuestas que las ofertas de acuerdo por parte de los clientes sólo requieran obligaciones a los proveedores, mientras que las ofertas de acuerdo por parte de los proveedores sólo garanticen obligaciones a los clientes. Éstas se denominan *unilaterales*.

[†1] Nótese que estos ejemplos no se muestran con la notación XML de WS-Agreement, sino que se utiliza una sintaxis concreta más fácil de entender para ilustrar cómo utilizar esta recomendación de una forma clara y concisa.

Hay que agregar que, en contextos diferentes a WS-Agreement, a las ofertas de acuerdo propuestas por los clientes también se les conoce como *demandas,* mientras que a las ofertas de acuerdo propuestas por los proveedores también se les conoce simplemente como *ofertas*.

2.2.2. Contexto

El contexto contiene información general de la oferta de acuerdo, por ejemplo, la identificación de las partes y la fecha de expiración. En el ejemplo de la figura §2.2, las partes del acuerdo son la `Universidad de Sevilla` como cliente y una empresa `CasaLibro` dedicada a la venta de libros, como proveedor. El acuerdo expira el `31/12/2007`.

2.2.3. Términos de Servicio

Describen los servicios que son objeto de las ofertas de acuerdo. Por regla general, un proveedor puede hacer referencia a los interfaces WSDL de los servicios web que oferta. En nuestro ejemplo, el proveedor incluye las referencias a sendos servicios, `comprarLibro()` y `consultarLibro()`, etiquetados con `S1` y `S2`, respectivamente. Por otro lado, un cliente puede describir los servicios en los que está interesado mediante, por ejemplo, palabras claves que indican una cierta funcionalidad que se requiere. En nuestro ejemplo, se utilizan palabras claves como `librería`, `compra de libros` y `consulta de libros`.

También podemos distinguir las propiedades del servicio, donde pueden describirse, por ejemplo, los atributos de calidad que serán utilizados para definir los objetivos de nivel de servicio. En nuestro ejemplo, se declaran los siguientes atributos de calidad: MTTF, MTTR, RT, CCODE y ETIME.

Descripción de Atributos de Calidad Respecto a los atributos de calidad, Ran [164] sostiene la ausencia de consenso en la terminología usualmente empleada, a pesar de los estándares. Por simplicidad, en esta memoria de tesis doctoral hablamos de “atributos de calidad”, como sinónimo de propiedades, parámetros o características no-funcionales o extra-funcionales.

En la norma ISO–8402 *Quality Management and Quality Assurance, Vocabulary*, incorporado posteriormente al ISO-9000, la calidad de servicio (QoS, *Quality of Service*) se describe como la “totalidad de caracteristicas de un producto o servicio que son relevantes para su capacidad de satisfacer las necesi-

dades ya establecidas o implícitas". Esto es, un atributo de calidad es cualquier característica medible de un servicio que puede influir sobre la percepción que pueda tenerse del mismo [166]. De obligada referencia son los estándares IEEE 610.12 *Standard Glossary of Software Engineering Terminology*, IEEE–1061 *Standard for a Software Quality Metrics Methodology* e ISO–9126 *Information Technology - Software Product Evaluation - Quality Characteristics and Guidelines for Their Use*.

O'Sullivan *et al.* [152, 153] argumentan la necesidad de una descripción de atributos de calidad que incluya nombre, tipo, medidas, unidades, etc. Existen diferentes maneras de describir y organizar los atributos de calidad. OASIS propone WSQM (*Web Services Quality Model*) [53], un modelo de calidad que incluye diferentes vistas para agrupar atributos relacionados, definiendo una jerarquía similar a ISO–9126. Estos modelos de calidad también se conocen como catálogos. A su vez, OMG propone *UML Profile for Modelling Quality of Service and Fault Tolerance Characteristics and Mechanisms* [150]. Entre otras propuestas específicas, [111, 123, 164] proponen modelos que incluyen atributos de relevancia para los servicios web.

2.2.4. Términos de la Garantía

Los términos de la garantía están constituidos principalmente por los objetivos de nivel de servicio a los que cliente y proveedor se comprometen respecto a unos servicios concretos incorporados al acuerdo u oferta de acuerdo. Un término de la garantía está constituido por los siguientes elementos:

- **Alcance** Identifica a los términos de servicio sobre los que se establece el término de garantía.
- **Parte Obligada** Identifica a la parte que se obliga a cumplir el término de la garantía.
- **Objetivo de Nivel de Servicio** (SLO, *Service Level Objective*) Describe las obligaciones respecto a los términos de servicio, que suelen expresarse como condiciones sobre las propiedades de servicio que tengan asociadas[†2]. Podemos distinguir entre *objetivos garantizados* por la parte que establece la oferta de acuerdo y *objetivos requeridos* a la parte contraria[†3].

[†2]Si las propiedades de servicio incluyen descripciones de atributos de calidad, abusando de la terminología se denominan como "condiciones de calidad".

[†3]Por extensión, también hablamos de términos requeridos o garantizados, o bien de condiciones requeridas o garantizadas.

- **Vigencia** Indica la condición que debe cumplirse para que el objetivo se considere vigente. Por ejemplo, pueden indicar períodos temporales que determinan la validez del objetivo en el tiempo.

- **Valores de Negocio** Indica la pérdida o aumento de valor que a la parte obligada de cumplir el objetivo le supone su cumplimiento o su violación. En este sentido, WS-Agreement contempla los siguientes mecanismos para describirlos:

  - La *importancia* expresa la importancia relativa del objetivo respecto a otros, usualmente de una manera cualitativa (alta, media o baja).
  - La *gratificación* y *penalización* por satisfacer el objetivo o no hacerlo, respectivamente.
  - Las *preferencias* expresan prioridades entre los términos de servicio a los que hace referencia el término de garantía, en forma de valores de utilidad que permiten la selección del mejor acuerdo.

En el ejemplo de la figura §2.2, tenemos los siguientes conjuntos de términos de garantía:

- En la demanda del cliente, los términos de la garantía establecen que:

  1. El proveedor debe comprometerse a suministrar el servicio `comprarLibro()` satisfaciendo que $\frac{MTTF}{MTTF+MTTR} \geq 0.9$[†4] de 8am a 5pm, de lunes a viernes.
  2. El proveedor debe comprometerse a suministrar ambos servicios satisfaciendo que $RT \leq 10$.
  3. El cliente se compromete a consumir el servicio `comprarLibro()` satisfaciendo que se encuentra en España.
  4. El cliente se compromete a consumir el servicio `consultarLibro()` satisfaciendo que $ETIME \geq 5$.

  Esta oferta de acuerdo no tiene fecha de expiración.

- En la oferta del proveedor, los términos de la garantía establecen que:

  1. El proveedor se compromete a suministrar el servicio `comprarLibro()` satisfaciendo que $MTTF \in [120\,..\,140]$ y $MTTR \in [5\,..\,10]$ de 8am a 5pm, de lunes a viernes.

[†4]Esta expresión corresponde a la disponibilidad de servicio, según la definición de Trivedi [192].

2. El proveedor se compromete a suministrar todas los servicios satisfaciendo RT $\in [5 \,..\, 10]$.

3. El cliente debe comprometerse a consumir el servicio `comprarLibro()` satisfaciendo que se encuentra en la Unión Europea.

4. El cliente debe comprometerse a consumir el servicio `consultarLibro()` satisfaciendo que ETIME ≥ 5.

Esta oferta de acuerdo tiene fecha de expiración, a saber, `31 de diciembre de 2007`.

- Los términos de la garantía del acuerdo finalmente establecido a partir de las ofertas de acuerdo anteriores, que se muestran en la figura §2.3, establecen que:

1. El proveedor se compromete a suministrar el servicio `comprarLibro()` satisfaciendo que MTTF $\in [120 \,..\, 140]$ y MTTR $\in [5 \,..\, 10]$ de 8am a 5pm, de lunes a viernes.

2. El proveedor se compromete a suministrar ambos servicios satisfaciendo que RT $\in [5 \,..\, 10]$.

3. El cliente se compromete a consumir el servicio `comprarLibro()` satisfaciendo que se encuentra en España.

4. El cliente se compromete a consumir el servicio `consultarLibro()` satisfaciendo que ETIME ≥ 5.

Todos ellos tienen validez hasta la fecha de expiración del acuerdo, que en este caso coincide con la fecha de expiración de la demanda.

La figura §2.4 muestra otro ejemplo de oferta de acuerdo cuyos términos de garantía incluyen valores de negocio para valorar las diferentes alternativas entre la oferta de acuerdo establece. Estas alternativas aparecen si la composición de términos está basada en operadores de disyunción, de manera que el acuerdo definitivo se puede constituir con un conjunto de términos u otro.

En este caso, la oferta de acuerdo establece que opcionalmente el servicio pueda utilizarse con un mecanismo de autentificación. La oferta de acuerdo asigna mayor valoración de utilidad para la descripción de servicio con autentificación pero a un coste mayor. Además, hay una penalización para el proveedor, que consiste en ofrecer hasta 10 invocaciones al servicio a un coste mucho más bajo en caso de que el objetivo del término no se cumpla.

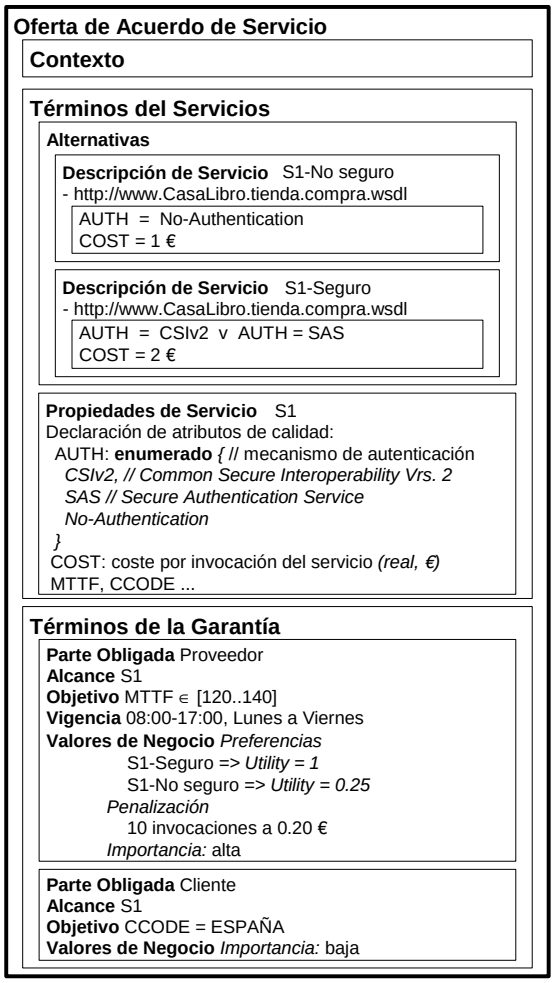


Figura 2.4: *Ejemplo de oferta de acuerdo con valores de negocio.*

2.3. Características de los Acuerdos Avanzados

En esta sección, presentamos las características para un modelo de acuerdos avanzado. Estas características se han tomado como marco de clasificación y comparación de la revisión del estado del arte que hemos realizado. La figura §2.5 muestra el mapa conceptual de las ofertas de acuerdo, inspirado en WS-Agreement, que indica en cursiva las extensiones debidas a dichas características.

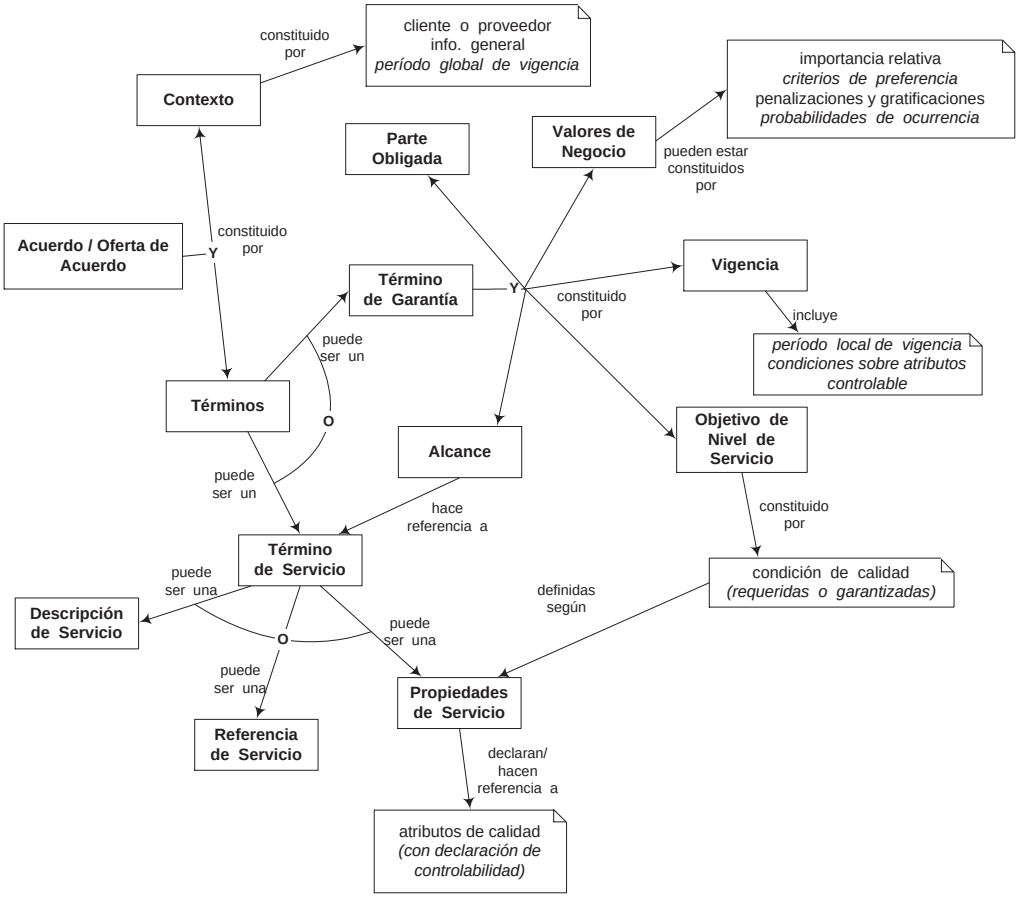


Figura 2.5: *Mapa conceptual de ofertas de acuerdo avanzadas.*

2.3.1. Expresividad de los Objetivos

No existe concenso para describir SLO, y WS-Agreement tampoco propone un lenguaje de descripción específico, aunque sugiere WS-Policy [18]. Esta reciente especificación es asumida por cada vez más propuestas. Propone una representación XML de las políticas de servicio. Una política define una colección de alternativas en forma normal disyuntiva. A su vez, cada alternativa es una colección de aseveraciones (*assertions*), donde cada aseveración contiene una condición particular sobre el servicio web. Hay que añadir que se han definido algunos estándares de políticas. Por ejemplo, IETF RFC-3060 y RFC-3460 [52, 145] proponen un modelo de políticas basados en reglas, que permite describir políticas mediante formas normales disyuntiva y conjuntiva. Nótese que, de hecho, las descripciones de SLO basadas en políticas ha inspirado la manera de componer los términos en WS-Agreement.

Dada la numerosa y variada relación de propuestas para describir SLO, se hace aconsejable disponer de un instrumento que permita clasificarlas y compararlas. Para tal fin, proponemos una extensión de la interpretación geométrica de SLO propuesta por Ruiz-Cortés en su tesis doctoral [166].

2.3.1.1. Interpretación Geométrica

Utilizamos diagramas de Venn para representar las ofertas de acuerdo. La figura §2.6 muestra una oferta de acuerdo que sólo tiene un término de garantía con su objetivo. Este objetivo está basado en un conjunto de n atributos de calidad $x_1 \ldots x_n$ que determinan un espacio n-dimensional de acuerdos. Un *punto de acuerdo* es una combinación de valores de los atributos tal que satisfacen las m condiciones $\Theta_1 \ldots \Theta_m$ que constituyen el objetivo. En este espacio de acuerdos, la región de acuerdos R contiene todos los puntos de acuerdo de la oferta de acuerdo.

Añadir que, por motivos de simplicidad y comprensión, los puntos de acuerdo de una región se omiten cuando no son relevantes. Igualmente, un objetivo tal que su región de acuerdos contenga un único punto de acuerdo posible, se representa mediante un punto en el espacio de acuerdos.

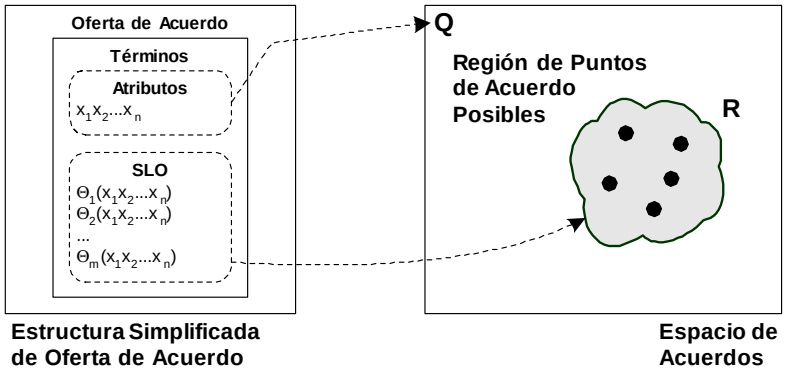


Figura 2.6: *Interpretación geométrica de objetivos de nivel de servicio.*

Al tratar de emparejar una demanda y una oferta, los diagramas de Venn permiten representar sus regiones de acuerdo, de manera que podemos determinar el espacio común de emparejamiento. De esta manera, la interpretación geométrica permite además la interpretación visual de las tareas de emparejamiento, como mostramos en el capítulo §3.

2.3.1.2. Grados de Expresividad

La forma geométrica que adopta la superficie correspondiente a la región de acuerdos viene determinada por el tipo de operadores lógicos, relacionales y aritméticos, así como sus operandos, que una propuesta admita para describir los objetivos. De esta manera, la forma geométrica puede servir como indicador del grado de expresividad, que nos permita clasificar y comparar las notaciones empleadas, según mostramos en la figura §2.7:

Operador de Igualdad La expresividad está restringida a condiciones basadas en pares atributos / valor $x = v$ de manera que la región de acuerdos consiste en un único punto en el espacio de acuerdos.

Operadores Relacionales La expresividad está restringida a condiciones simples basadas en operadores relacionales, de manera que un operando es un atributo de calidad y el otro operando es un valor en su dominio que restringe los valores que dicho atributo puede tomar, como $x = v$, $x < v$, $x > v$, $x \leq v$ o $x \geq v$. Las expresiones para determinar si el valor de un atributo pertenece a un rango, $x \in [v_1 \,..\, v_2]$, también se consideran como una forma particular de condición relacional. La región de acuerdos es convexa con límites paralelos a los ejes.

Operadores Lógico-Relacionales La expresividad está restringida a expresiones relacionales ligadas por operadores lógicos como & y ||. Es habitual que adquieran la forma normal disyuntiva o conjuntiva, como por ejemplo, en las descripciones basadas en políticas. La región de acuerdos puede contener varias sub-regiones convexas y con límites paralelos a los ejes. Hay mayor expresividad si se permiten operadores lógicos avanzados tales como la negación lógica, el implica lógico, cuantificadores existenciales y universales, etc. La región de acuerdos en este caso puede contener varias sub-regiones convexas con límites paralelos a los ejes, que pueden contener huecos y / o representar alternativas.

Operadores Aritmético-Relacionales La expresividad viene dada por expresiones relacionales, de manera que los operandos contienen a su vez expresiones aritméticas o relacionales. Según los operadores y operandos permitidos en las expresiones aritméticas, éstas pueden ser lineales o no-lineales, así que la región de acuerdos puede no ser convexa y sus límites no necesariamente lineales o paralelos a los ejes.

Operadores Aritmético-Lógico-Relacionales La expresividad viene dada por disyunciones y conjunciones de expresiones aritmético-relacionales. La región de acuerdos puede contener varias sub-regiones, que pueden

ser no-convexas, cuyos límites no son necesariamente lineales ni paralelos a los ejes. En caso de que se permitan operadores lógicos avanzados, la región de acuerdos puede contener varias sub-regiones que pueden no ser convexas, cuyos límites no son necesariamente lineales ni paralelos a los ejes, que pueden contener huecos y / o representar alternativas.

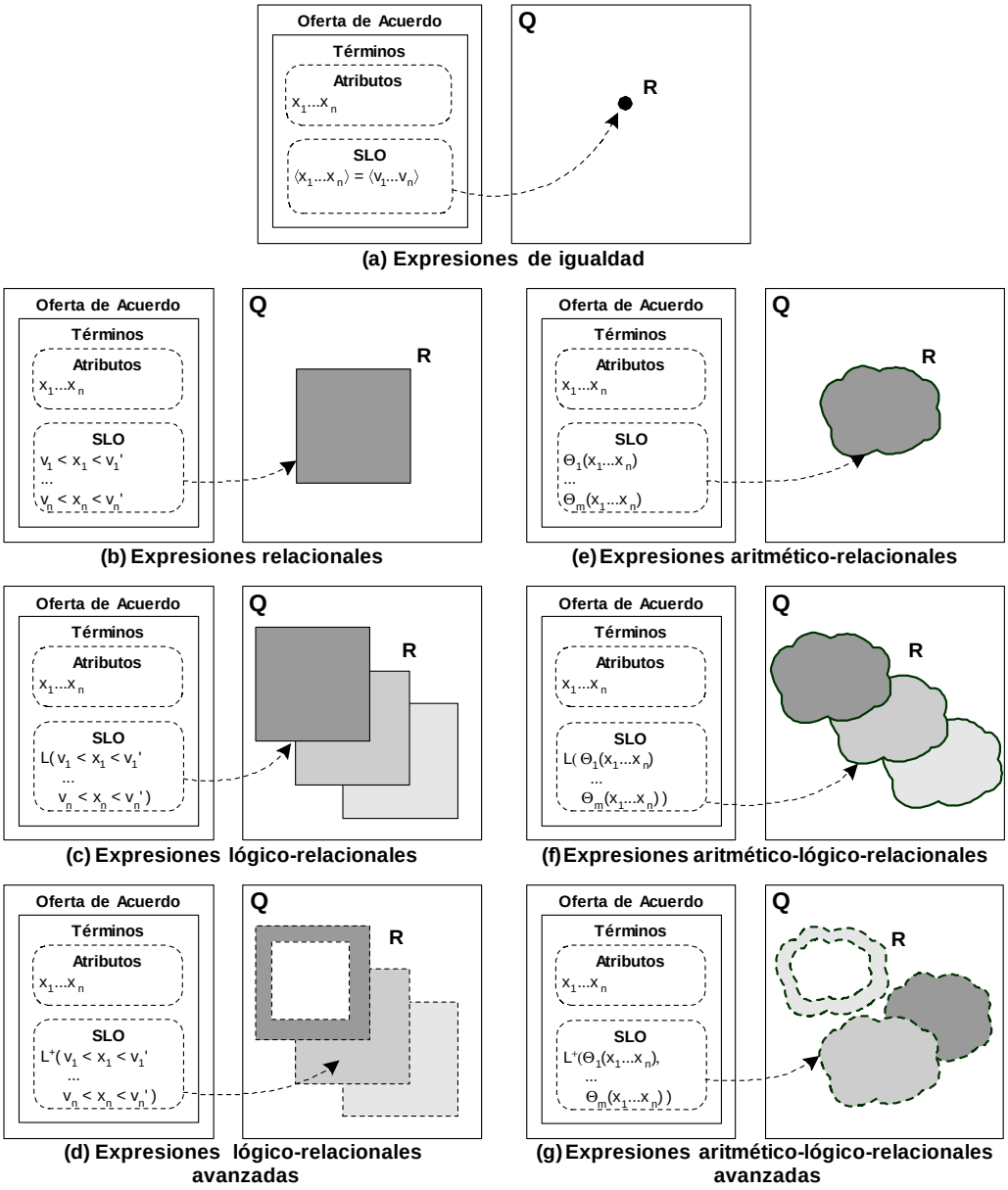


Figura 2.7: *Grados de expresividad para describir condiciones en los objetivos.*

2.3.2. Simetría de los Objetivos

En la descripción de SLO, distinguimos si son requeridos o garantizados por alguna de las partes. Es habitual que la expresividad del lenguaje / notación esté limitada al operador de igualdad para los objetivos garantizados, pero no lo esté para los objetivos requeridos. Podría decirse que los objetivos son *asimétricos* en este caso, y *simétricos* cuando se permitan describirlos con la misma expresividad. En este último caso, podemos distinguir además entre *simétricos puntuales* y *simétricos no-puntuales*, ya que puede darse el caso de que tanto objetivos requeridos como garantizados estén restringidos a pares atributo / valor.

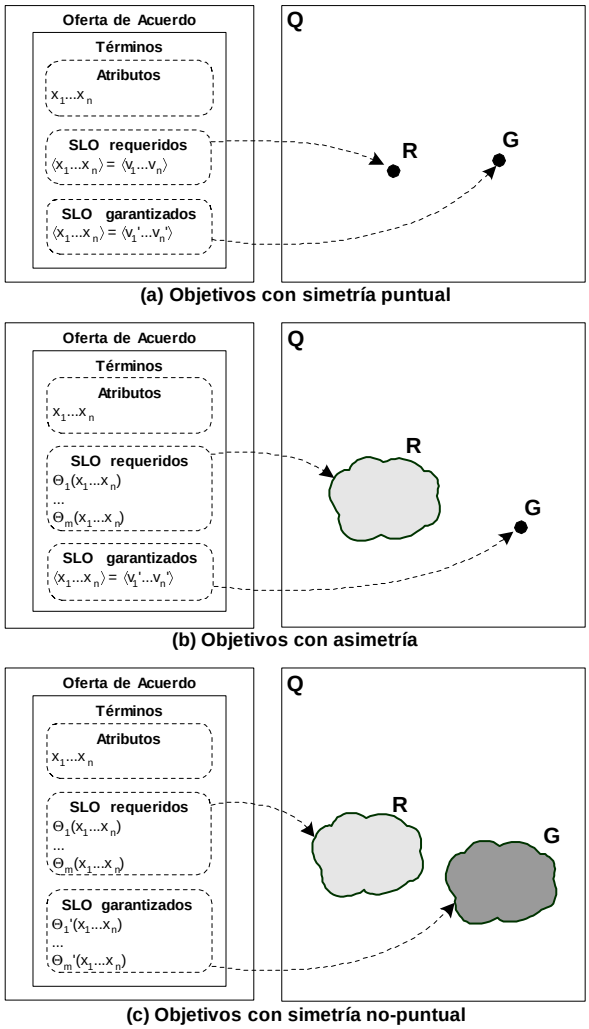


Figura 2.8: *Simetría en ofertas de acuerdo con términos bilaterales.*

La interpretación geométrica de los SLO también nos permite distinguir entre ambos casos, como mostramos en la figura §2.8[†5].

2.3.3. Controlabilidad de los Atributos

En una oferta de acuerdo, los clientes o proveedores pueden establecer que un atributo es controlable para expresar a la parte contraria que puede contar que dicho atributo tome un valor concreto, que satisface los términos garantizados, cuando vaya a utilizar el servicio objeto del acuerdo. Los atributos controlables pueden determinar qué objetivos son válidos, por lo que en WS-Agreement las condiciones sobre estos atributos determinan la vigencia de los términos de garantía. Por ejemplo, en la figura §2.9 mostramos una oferta de acuerdo en la que el proveedor garantiza la controlabilidad del atributo MEDIA, de manera que el cliente podrá contar con cualquiera de los medios de acceso al servicio que aparecen especificados entre los objetivos, teniendo en cuenta que, aparte de su período local de vigencia, la vigencia de cada objetivo también viene determinada por el valor que toma dicho atributo.

La controlabilidad está relacionada con la adaptabilidad de la infraestructura para gestionar los recursos disponibles y proporcionar la calidad de servicio garantizada, es decir, garantizar el cumplimiento del acuerdo. Entre otras propuestas que han trabajo en la adaptibilidad, destacamos WSOL [186, 188], GlueQoS [207], MAIS [27, 124, 133] y AMPol-Q [1]. La adaptabilidad es una condición necesaria para que las partes puedan garantizar la controlabilidad de los atributos de calidad durante el emparejamiento. Sin embargo, no hay propuestas a lo que conocemos que declare explícitamente si los atributos son controlables. A excepción de [64, 166, 168], las propuestas asumen implícitamente la controlabilidad de todos los atributos, aunque esto no sea muy acertado ya que sólo las propuestas mencionadas dan soporte a la adaptabilidad.

2.3.4. Caracterización de Atributos No-Controlables

La controlabilidad está ligada al concepto de incertidumbre, puesto que si no se establece si un atributo es controlable, no hay certeza del valor concreto que dicho atributo pueda tomar. Para paliar esta incertidumbre, el proveedor puede dar información sobre la probabilidad de ocurrencia de los atributos no-controlables, es decir, la probabilidad de que un atributo no-controlable tome un determinado valor.

[†5] Nótese que en la figura §2.8 hemos utilizado una nube para representar una región de acuerdos correspondiente a cualquier grado de expresividad que sea no-puntual.

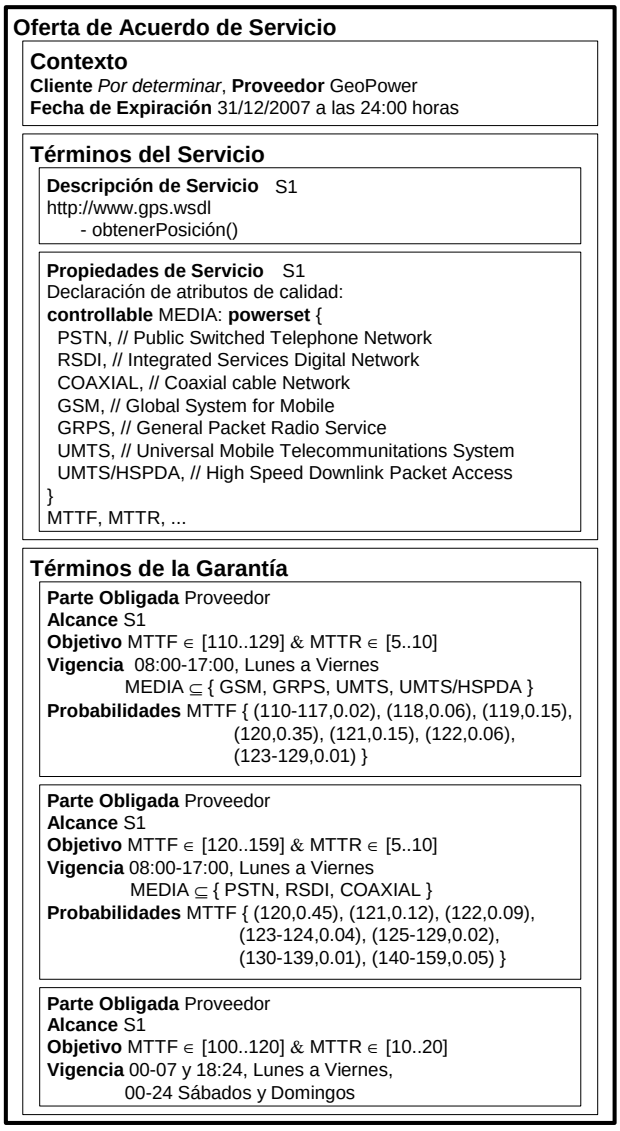


Figura 2.9: *Ejemplo de oferta de acuerdo con atributos controlables.*

Nótese que estas probabilidades pueden depender de los valores que tomen los atributos controlables a los que están ligados los atributos no-controlables.

En la figura §2.9, el proveedor de la oferta de acuerdo también infoma sobre las probabilidades de ocurrencia del atributo no-controlable MTTF que, además, son diferentes según el valor que tome el atributo controlable MEDIA al que está ligado.

2.3.5. Funciones de Utilidad Complejas

La recomendación WS-Agreement esta algo limitada para describir los valores de negocio, sobre todo respecto a la descripción de los criterios de preferencias. Una forma más avanzada es expresar los valores de negocio mediante *funciones de utilidad.*

Estas funciones asignan valores que denotan una valoración subjetiva o preferencia sobre los valores que puede tomar un atributo de calidad. La figura §2.10 muestra algunos ejemplos correspondientes a atributos de calidad habituales. Suelen ser funciones monótonas crecientes o decrecientes, definidas punto a punto. Las funciones de utilidad se agregan para obtener la utilidad de los puntos de acuerdo entre una demanda y una oferta, utilizando los grados de importancia relativa para ponderar el peso de cada atributo de calidad.

Estos agregados suelen asumir la independencia de los atributos, pero en nuestro modelo de emparejamiento avanzado hay que tener en cuenta otros factores, a saber, la dependencia entre atributos ya que distinguimos entre controlables y no-controlables, las probabilidades de ocurrencia que describen estadísticamente a los atributo no-controlables y las funciones de utilidad no han de ser necesariamente monótonas y adoptar formas mucho más complejas.

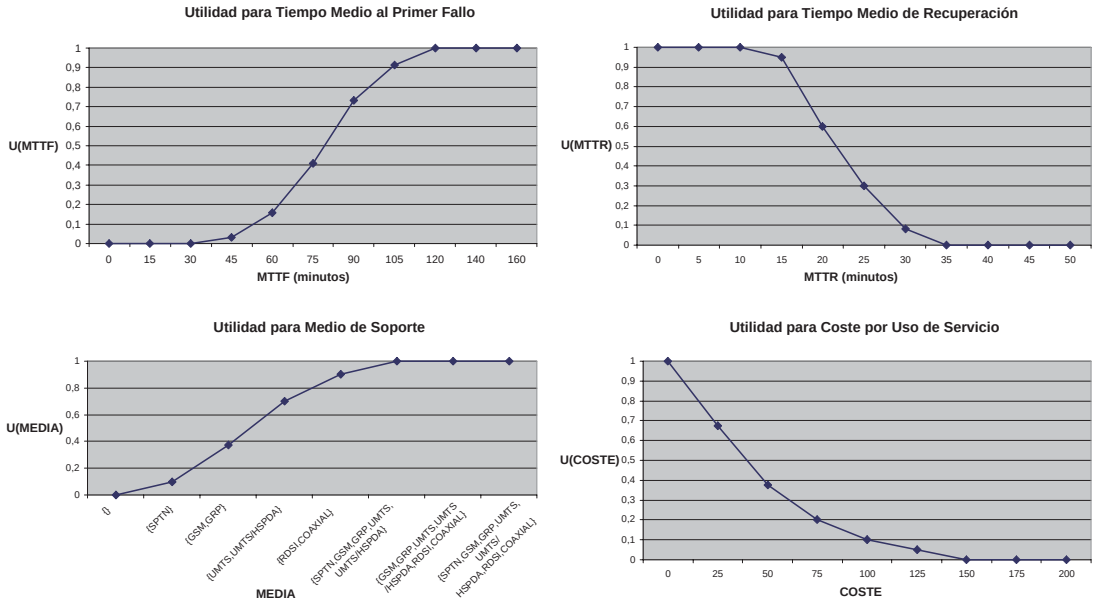


Figura 2.10: *Ejemplo de funciones de utilidad.*

2.3.6. Consciencia Temporal

Algunas propuestas permiten indicar el período que determina la vigencia de una oferta de acuerdo y por extensión de todos sus términos. En [131] propusimos denominar estos períodos como "períodos globales", en contraposición a los períodos de vigencia que en ocasiones se necesita indicar para un objetivo concreto, a los cuales denominamos "períodos locales". La figura §2.11 ilustra los períodos de vigencia correspondientes a fragmentos de las ofertas de acuerdo de las figuras §2.3 y §2.2.

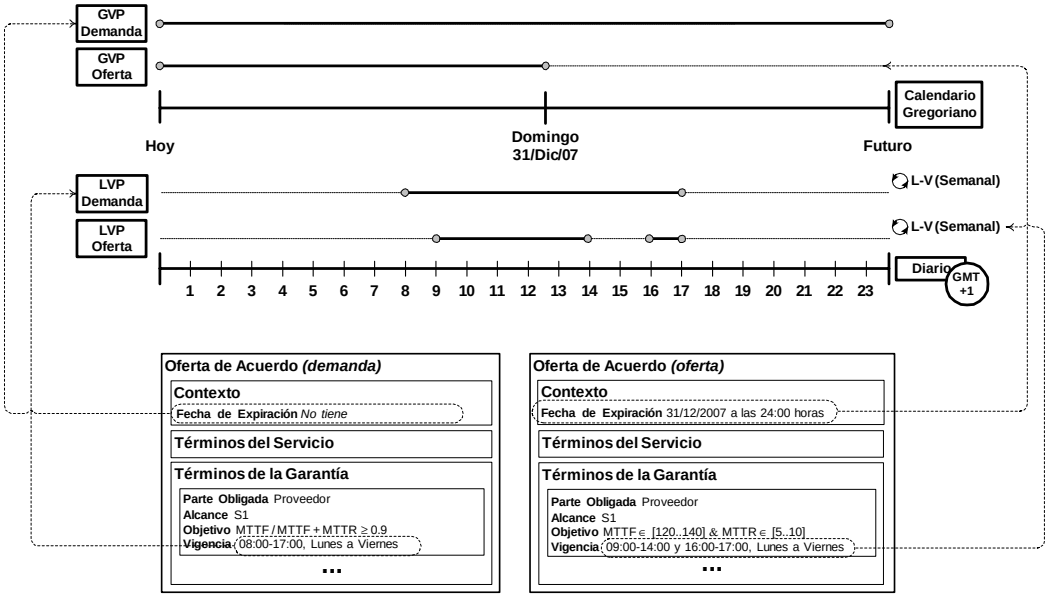


Figura 2.11: *Ejemplo de ofertas de acuerdo con consciencia temporal.*

En cuanto a la forma de expresar un período de vigencia, las propuestas podrían clasificarse según el tipo de intervalos empleados para definir dicho período. En [142] identificamos cinco grupos que posteriormente hemos refinado en tres, a saber:

- *Intervalo único.* Definen el período de vigencia a partir de un único intervalo, que determina la fecha inicial y la final.
- *Múltiples intervalos periódicos.* Definen el período a partir de un intervalo y una periodicidad.
- *Múltiples intervalos no periódicos.* Definen el período a partir de múltiples intervalos que no han de ser periódicos necesariamente.

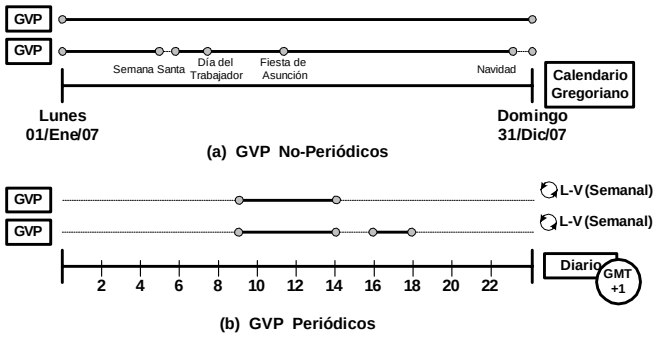


Figura 2.12: *Ejemplo de períodos globales de vigencia.*

En la figura §2.12(a) se muestra un período global de vigencia que denota que una fecha de expiración, en este caso la vigencia alcanza al año 2007. En la figura §2.12(a) también se muestra un período global de estas características, que puede interpretarse como un período global de vigencia con excepciones correspondientes a las fiestas. En la figura §2.12(b) se muestra un período global de vigencia que denota que el acuerdo u oferta de acuerdo sólo es válido de 9 a 14 horas, de lunes a viernes. También pueden combinarse varias periocidades, para denotar que sólo es válido de 9 a 14 horas y de 16 a 18 horas, de lunes a viernes.

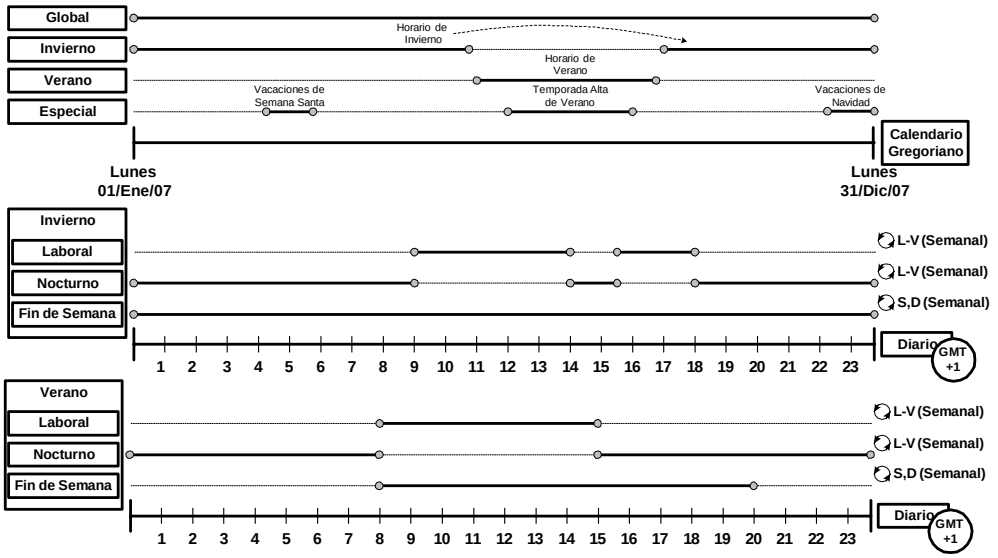


Figura 2.13: *Ejemplo de oferta de acuerdo con períodos de vigencia complejos.*

La figura §2.13 representa un acuerdo con diferentes tipos de períodos locales de vigencia. El acuerdo sólo está vigente durante 2007. El horario habitual o de invierno distingue el horario laboral (de 9am a 2pm y 4pm a 6pm, de lunes a viernes), el horario nocturno (de 0am a 8am, de 2pm a 4pm y de 6pm a 12pm, de lunes a viernes) y el horario de fin de semana (las 24 horas, sábados y domingos). En cambio, el horario de verano distingue el horario laboral (de 8am a 3pm, de lunes a viernes), el horario nocturno (de 0am a 8am y de 3pm a 12pm, de lunes a viernes) y el horario de fin de semana (de 8am a 8pm, sábados y domingos). Además, hay partes del acuerdo que sólo están vigentes durante Navidad, Semana Santa y los meses de julio y agosto (temporada alta de verano).

Nótese que en estas figuras se definen algunas peculiaridades del dominio temporal como, por ejemplo, el calendario al que están referidas las expresiones temporales, a la vez que la referencia horaria, aspectos muy importantes en un contexto globalizado como Internet. En [166] se hace un estudio más detallado de las mismas.

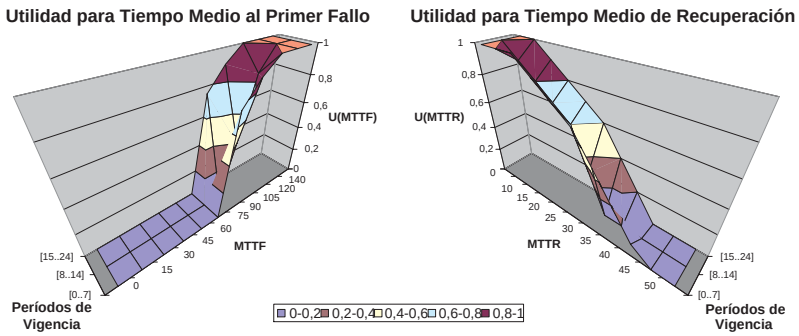


Figura 2.14: *Ejemplo de funciones de utilidad con consciencia temporal.*

Hay que agregar que las funciones de utilidad y las probabilidades de ocurrencia pueden depedender del tiempo. En [131] fuimos los primeros en proponer funciones de utilidad dependientes del tiempo en el ámbito del emparejamiento de servicios. Por ejemplo, la figura §2.14 muestra dos funciones de utilidad con consciencia temporal.

En [144] también hemos incorporado consciencia temporal a las restricciones sobre la creación de un acuerdo, pero esta característica ya queda fuera del alcance de esta memoria de tesis.

2.4. Resumen y Reseña Bibliográfica

En este capítulo hemos presentado una revisión de las notaciones para la descripción de acuerdos de calidad de servicio. Hemos tomado como referencia la recomendación WS-Agreement, que proporciona una estructura básica para describir ofertas de acuerdo y un protocolo para obtener acuerdos. Este modelo avanzado incluye características tales como la simetría y expresividad de los objetivos, la controlabilidad de los atributos de calidad, la caracterización estadística de los atributos no-controlables, los valores de negocio con funciones de utilidad complejas y la consciencia temporal respecto a diferentes partes de una oferta de acuerdo.

Estas características avanzadas conforman un corpus que proporciona un marco de clasificación y comparación de propuestas relacionadas. En [129, 168] ya propusimos una versión preliminar de dicho marco, que ha sido utilizada como referencia por Tosic en sus trabajos [187]. En esta tesis doctoral, hemos extendido el conjunto de características y actualizado la revisión del estado del arte.

En [129, 166, 168] hemos propuesto la expresividad y simetría como características avanzadas del emparejamiento. En estos trabajos se propone la interpretación geométrica como vía a la identificación de diferentes grados de expresividad. En [166, 168] también incluimos un estudio de la descripción de valores de negocio mediante funciones de utilidad complejas.

Finalmente, en [166] se hace una introducción de la consciencia temporal en el ámbito de los requisitos no-funcionales. Este estudio se ha actualizado en el contexto de los servicios web, primero en [131, 142] y posteriormente de un modo mucho más detallado en [143, 144]. En [131, 166] incluimos las funciones de utilidad dependientes del tiempo. En [142] describimos los grados de expresividad para la descripción de períodos de vigencia y en [143, 144] hemos propuesto un esquema para agregar expresiones temporales en acuerdos descritos con WS-Agreement, además de añadir temporalidad a otros aspectos de los acuerdos, como son las restricciones de creación.

Capítulo 3

Emparejamiento de Servicios Sensible a la Calidad

That's not fair! – You say that so often, I wonder what your basis for comparison is?

Jennifer Connelly & David Bowie
Sarah & Jareth The Goblin King, "Labyrinth", 1986

En este capítulo presentamos las tareas de emparejamiento de servicios sensible a la calidad. En la sección §3.1 presentamos la noción de emparejamiento. A continuación, las secciones §3.2, §3.3 y §3.4 presentan las tareas de emparejamiento, a saber, la comprobación de la consistencia, la comprobación de la conformidad y la selección del mejor acuerdo, respectivamente. En la definición de estas tareas se han tenido en cuenta todas las características del modelo de emparejamiento avanzado, identificadas en el capítulo anterior. Posteriormente, en la sección §3.5 se muestran los diferentes tipos de formalismos subyacentes que se han utilizado para dotar de soporte automático al emparejamiento de servicios. Finalmente, en la sección §3.6 concluimos con un resumen.

3.1. Introducción

Según Sycara *et al.* [182], una oferta se empareja con una demanda cuando la capacidad funcional de la oferta es "suficientemente similar" a las capacidades funcionales requeridas por la demanda. Si particularizamos o concretamos esta definición teniendo en cuenta la calidad de servicio como el aspecto diferenciador entre los servicios que ofrecen una misma funcionalidad [123, 206], nos quedaría que *una oferta se empareja con una demanda cuando la calidad de servicio de la oferta es "suficientemente similar" a la calidad de servicio requerida por la demanda.*

El emparejamiento de servicios puede ser diferente según entendamos qué significa "suficientemente similar". En cualquier caso, las tareas que son necesarias en el emparejamiento de servicios son básicamente la *consistencia* para comprobar que las ofertas de acuerdo no contengan errores; la *conformidad* para comprobar que es posible encontrar un acuerdo entre dos ofertas de acuerdo; y la *optimalidad* para encontrar el mejor acuerdo de entre todos los posibles. Numerosos autores también se refieren a estas dos últimas tareas como descubrimiento y selección, respectivamente.

Actualmente, el emparejamiento es una actividad que en la mayoría de los casos se realiza manualmente. Sin embargo, desde los trabajos de Sycara *et al.* [181, 182] a finales de los 90 y posteriormente por Casati y Shan [34], entre otros, se ha identificado la necesidad de automatizar dicha actividad. Las principales razones que justifican esta necesidad son:

- El número de ofertas de acuerdo, así como el número de sus objetivos y atributos involucrados pueden ser muy elevados, por lo que existe una mayor probabilidad de realizar búsquedas y / o selecciones incorrectas y finalmente obtener acuerdos defectuosos.

- Las descripciones de los objetivos pueden llegar en ocasiones a ser expresiones muy complejas, que no hacen posible que se pueda realizar manualmente el emparejamiento de servicios.

- Las ofertas de acuerdo suelen estar sujetas a cambios, de manera que las tareas de emparejamiento deben volver a llevarse a cabo con relativa frecuencia. En el caso más desfavorable, puede ser necesario revisar los acuerdos toda vez que se suministra y / o se consume un servicio.

Son numerosas las propuestas que, hasta la fecha, se han realizado para automatizar algunas de las tareas de emparejamiento. Sin embargo, hasta lo

que nosotros sabemos, son escasas las propuestas que describen con rigor estas tareas de emparejamiento y siempre con un modelo de oferta de acuerdo que cubre sólo unas pocas de las características del modelo avanzado descrito en el capítulo §2.

3.2. Consistencia

Por regla general, la elaboración de una oferta de acuerdo, salvo cuando está asistida por una herramienta que garantice la ausencia de errores, es una actividad propensa a errores. Entre los errores que no son detectables por un analizador sintáctico podemos destacar:

- Los objetivos de nivel de servicio pueden ser inalcanzables. Los motivos pueden ser varios:
  - Inalcanzables por sí mismos, es decir, el objetivo en sí contiene contradicciones. Por ejemplo, el objetivo $\mathrm{MTTF} \leq 100 \wedge \mathrm{MTTF} > 120$ es inconsistente.
  - Inalcanzables porque si se cumple uno, es a costa de que otro objetivo no se cumpla. Por ejemplo, si el objetivo de un término es $A \geq 0.9$ y el objetivo de otro $\mathrm{MTTF} \leq 100 \wedge \mathrm{MTTR} > 15$, donde $A = \frac{\mathrm{MTTF}}{\mathrm{MTTF}+\mathrm{MTTR}}$ [192], entonces la conjunción de ambos es contradictoria, ya que no hay par de valores $\mathrm{MTTF} \leq 100$ y $\mathrm{MTTR} > 15$ que satisfaga $A \geq 0.9$.
  - Inalcanzable en algún momento, debido a que los términos válidos durante un instante temporal son contradictorios. Por ejemplo, consideremos la oferta de acuerdo de la figura §3.1[†1]. Es inconsistente porque en la franja horaria $[8\,..\,10]$ la conjunción de los objetivos resulta la expresión $90 \leq \mathrm{MTTF} \leq 100 \wedge 110 \leq \mathrm{MTTF} \leq 120$ que es contradictoria.
- Los intervalos temporales que conforman los períodos de vigencia también pueden contener errores:
  - Los períodos locales de vigencia no cubren completamente el período global de vigencia de la oferta de acuerdo. Por ejemplo, el período global es `de lunes a viernes, de 9 a 14` y el único término

[†1]En dicha figura, mostramos una expresión matemática para representar de manera compacta los términos de garantía de dicha oferta. Esta notación se emplea a lo largo de esta tesis doctoral por motivos de brevedad. También se muestra su representación geométrica

tiene como período local `de lunes a jueves, de 9 a 14`, dejando el intervalo temporal correspondiente al viernes sin definir.

- Un período de vigencia puede ser ambiguo, por ejemplo, una expresión temporal como `de lunes a jueves de 8 a 17 y de 15 a 20 y viernes de 8 a 15`. Los intervalos temporales que lo componen no son disjuntos, porque tienen una intersección no vacía durante el período $15 - 17$ de lunes a jueves.
- Un período de vigencia puede estar mal formado, por ejemplo, `de lunes a jueves de 17pm a 8am`. En este caso, debería estar formado de esta manera `de lunes a jueves de 0 a 8 y 17 a 24`.

- Las funciones de utilidad se consideran consistentes si, en primer lugar, las valoraciones de utilidad que asignan están entre 0 y 1. Además, puede haber otros tipos de errores. Por ejemplo, si están definidas por trozos, hay que asegurarse de que a cada valor en el dominio del atributo sólo le corresponde una valoración de utilidad. Por ejemplo, la figura §3.2 es una función de utilidad que no es consistente porque al valor $MTTF = 60$ se le han asignado dos valores de utilidad diferentes, 0.15 y 0.3.

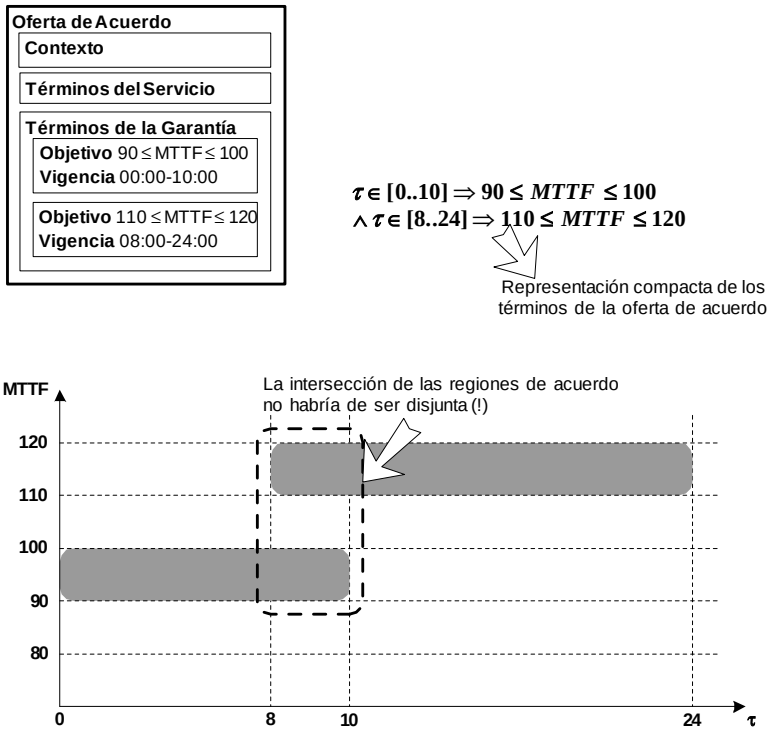


Figura 3.1: *Diferentes representaciones de una oferta de acuerdo.*

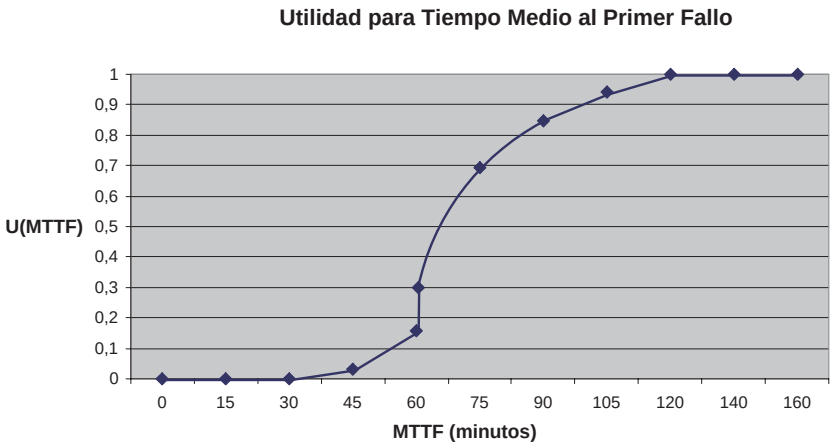


Figura 3.2: *Función de utilidad que no es consistente.*

- Las probabilidades de ocurrencia suelen tomar la forma de funciones de distribución en el caso de atributos discretos, o funciones de densidad en el caso de atributos continuos. En cualquier caso, hay que comprobar que el sumatorio de probabilidades de los valores que puede tomar un atributo debe ser 1.

- Tanto las funciones de utilidad como las probabilidades de ocurrencias pueden ser dependientes del tiempo, por lo que hay que comprobar que no se definan diferentes utilidades o probabilidades en un mismo instante de tiempo para un mismo conjunto de valores de los atributos.

Algunos de los errores anteriores son detectables sintácticamente, por ejemplo, comprobar el rango de los valores de utilidad en una función de utilidad definida por puntos. No es una tarea complicada definir reglas que permitan detectar los anteriores errores semánticos, sobre todo respecto a las funciones de utilidad. No obstante, en la Web Tradicional hay pocos trabajos que describan ni siquiera informalmente la detección de estos errores. En el lenguaje WSOL propuesto Tosic y Pagurek [186, 188] la comprobación de la consistencia se guía por la sintaxis. De manera similar, el lenguaje QML propuesto por Frølund y Koistinen [63, 64] tiene su semántica guiada por la sintaxis y por ende farragosa, dependiente de la sintaxis concreta y difícil de extender. Ruiz-Cortés [166] describe en detalle las limitaciones de QML y propone una semántica basada en problemas de satisfacción de restricciones que salva estas limitaciones. Por otro lado, las propuestas de Web Semántica interpretan la consistencia como problemas de inferencia lógica o lógicas descriptivas, pero aún no se ha abordado, a lo que conocemos, la consciencia temporal.

3.3. Conformidad

Descubrir si una oferta de acuerdo cumple los objetivos de otra, en otras palabras, determinar si entre ambas ofertas de acuerdo es posible alcanzar un acuerdo de nivel de servicio, es una tarea que de un modo más o menos riguroso se lleva haciendo desde hace décadas[†2]. Uno de los trabajos pioneros más referidos que abordaba esta problemática en el contexto del servicio de *trading* de CORBA fue propuesto por Frølund y Koistinen [63, 64]. En dicho trabajo se empleaba la expresión "un acuerdo A es conforme a un acuerdo B, si el cumplimiento de todas las cláusulas de A implica el cumplimiento de B". Este término lo siguió utilizando Ruiz-Cortés en [166] para la obtención automática de SLA en el contexto de los servicios web.

La complejidad computacional del algoritmo necesario para determinar la conformidad viene dada fundamentalmente por el tipo de las ofertas de acuerdo intervinientes en el acuerdo. A su vez, el tipo de una oferta de acuerdo viene determinado por el grado de simetría, la expresividad del lenguaje para describir sus objetivos, el grado de consciencia temporal y por su carácter bilateral (véase el capítulo §2). El número de tipos de ofertas de acuerdo es muy numeroso[†3] y salvo el algoritmo propuesto por Ruiz-Cortés *et al.* en [168] no conocemos ningún otro algoritmo que pueda ser utilizado en todas las posibles situaciones. Incluso éste, no es capaz de determinar la conformidad en ofertas de acuerdo donde hay atributos controlables y no-controlables al mismo tiempo.

3.3.1. Objetivos Asimétricos

El algoritmo de conformidad más habitual está diseñado para ofertas de acuerdo con objetivos asimétricos en emparejadores de única vía[†4]. La figura §3.3(a) muestra esquemáticamente este caso, donde la oferta del proveedor (ω) se corresponde a un único punto de acuerdo, de manera que la comprobación de la conformidad equivale a comprobar si dicho punto pertenece a la región de acuerdos de la demanda (δ). Por ejemplo, la oferta $\{\mathrm{MTTF} = 120\}$ es conforme a la demanda $\{\mathrm{MTTF} \geq 100\}$ porque el punto de acuerdo de la oferta satisface esta condición, por lo que pertenece a la región de puntos de acuerdos de la demanda.

[†2] Habitualmente, una de las ofertas de acuerdo se corresponde a la demanda de servicio por parte de un cliente, y la otra a la oferta propiamente dicha por parte de un proveedor.

[†3] No hemos considerado valioso determinar el número exacto de combinaciones posibles.

[†4] Un emparejador es de única-vía cuando sólo comprueba la conformidad entre ofertas de acuerdo unilaterales, por lo que también hablamos de conformidad de única-vía.

Por otro lado, hay que decir que hay propuestas donde las demandas también están restringidas a pares atributo-valor, lo que se trata como un caso particular del emprejamiento con objetivos asimétricos.

3.3.2. Atributos Controlables

A continuación, incrementando la dificultad y siguiendo con los emparejadores de única-vía, encontramos el algoritmo de conformidad diseñado para las ofertas de acuerdo con objetivos simétricos no-puntuales.

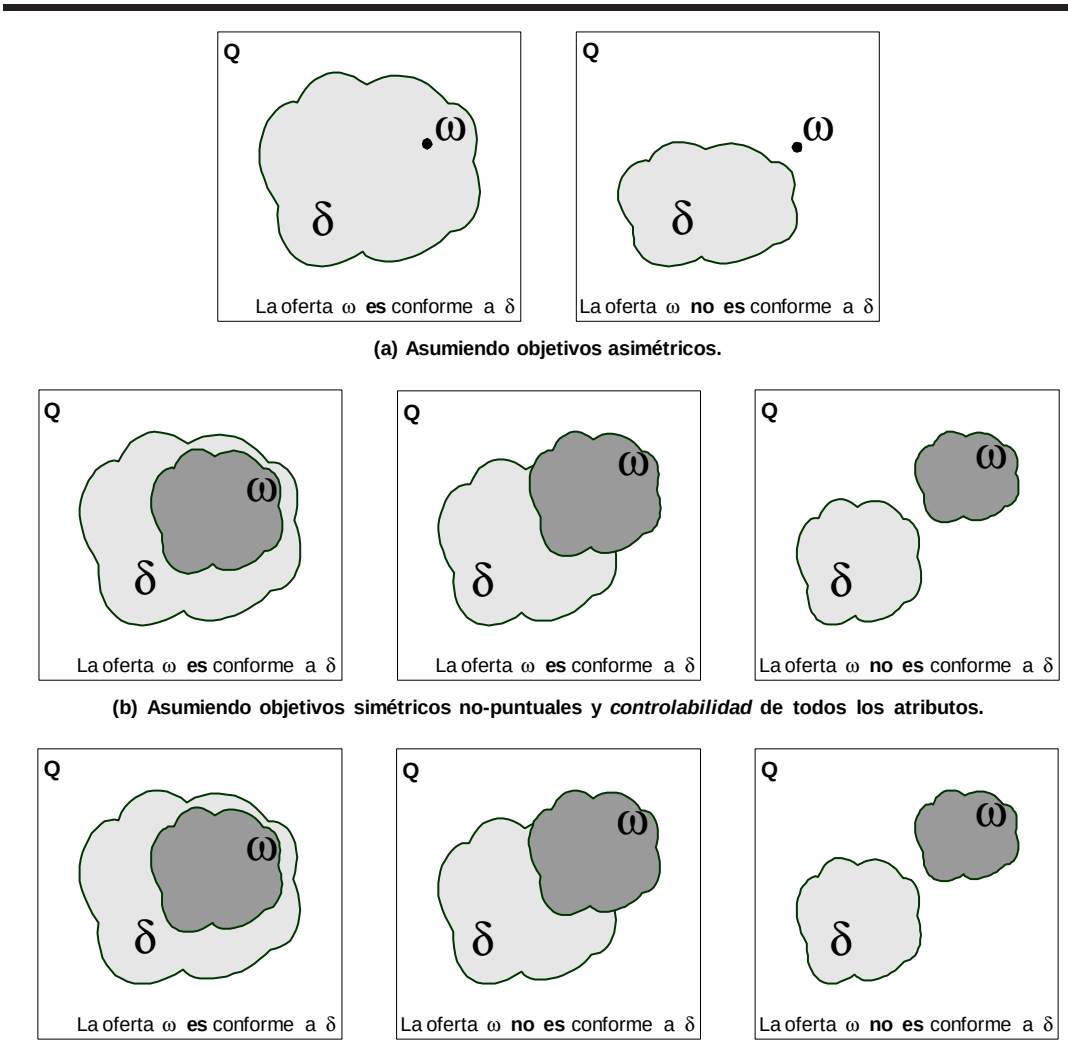


Figura 3.3: *Conformidad en emparejadores de única vía.*

La figura §3.3(b) muestra esquemáticamente este caso, donde la oferta del proveedor también se corresponde a una región de puntos de acuerdo posibles, de manera que la comprobación de la conformidad equivale a determinar la intersección entre ambas regiones. Por ejemplo, la oferta $\{90 \leq \text{MTTF} \leq 120\}$ es conforme a la demanda $\{\text{MTTF} > 100\}$ porque la intersección de regiones no está vacía, incluye los valores de MTTF en $[101\,..\,120]$. Esta noción optimista de conformidad asume la controlabilidad de todos los atributos, que es el enfoque que asumen la mayor parte de las propuestas estudiadas.

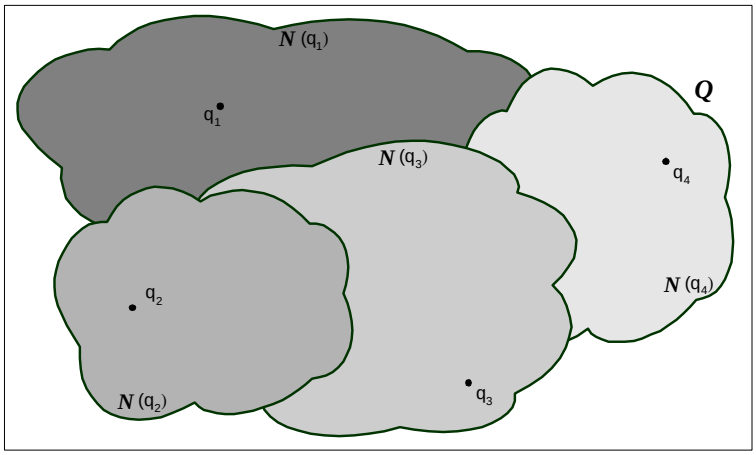
3.3.3. Atributos No-Controlables

En cambio, si asumimos que todos los atributos son no-controlables, entonces el algoritmo de conformidad debe comprobar que cada punto de acuerdo de la oferta satisface los objetivos de la demanda, es decir, que la región de la oferta se incluye en (o iguala a) la región de acuerdos de la demanda, como mostramos en la figura §3.3(c). En este contexto, la ofertas de acuerdo del párrafo anterior no son conformes porque hay valores de MTTF, aquéllos en $[90\,..\,100]$, que no satisfacen la demanda. Esta noción pesimista de conformidad sólo se ha asumido en las propuestas de QML [63, 64] y QRL [166].

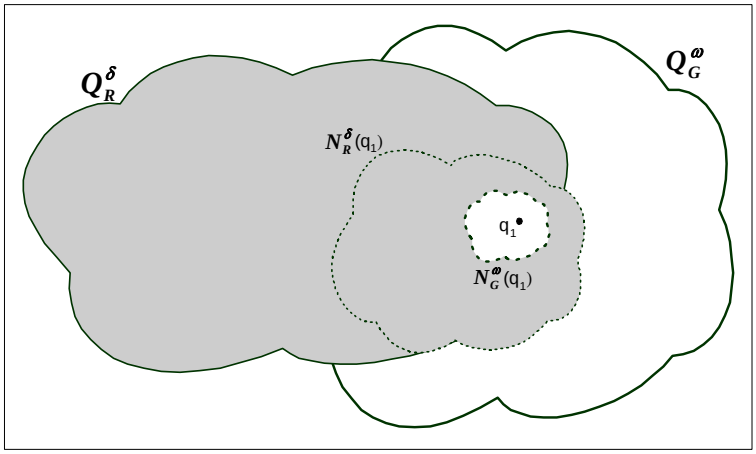
En el caso general, la inclusión de conjuntos es un problema difícil y su complejidad computacional va a depender de la expresividad conque se hayan descrito los objetivos de las ofertas de acuerdo. Nótese que no es lo mismo comparar sendas regiones de acuerdo cuyos límites sean paralelos a los ejes, que es el caso de la expresividad lógico-relacional simple, frente a comparar regiones de acuerdo con límites no necesariamente lineales ni paralelos a los ejes, que es el caso que puede surgir cuando hay expresividad aritmética. Esta es la razón por la que muchas propuestas no tengan una expresividad para describir objetivos demasiado alta ni tengan en cuenta la controlabilidad de los atributos de calidad

3.3.4. Atributos Controlables y No-Controlables

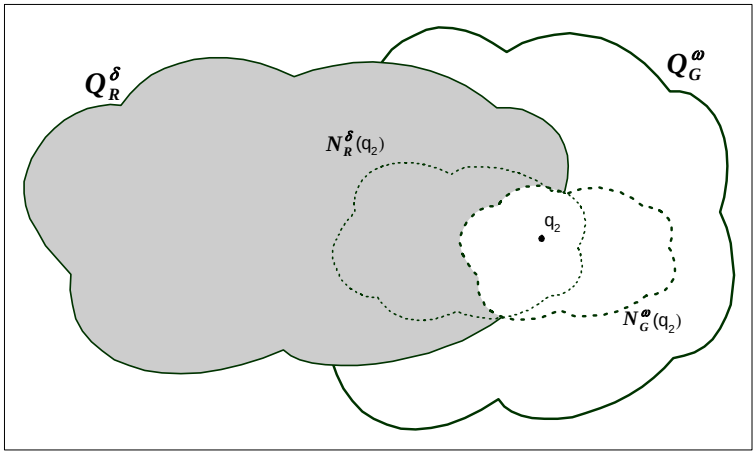
En el caso general, podemos encontrar que haya atributos tanto controlables como no-controlables. A lo que conocemos, no hay ninguna propuesta que lo trate de manera conjunta, aunque la sintaxis si lo permite. Si tuvieran una semántica formal se habría identificado el problema, es decir, con la cláusula de vigencia es posible expresar una condición sobre un atributo controlable referida a un objetivo definido sobre un atributo no-controlable.



(a) Regiones no-controlables ligadas a puntos controlables en una región de acuerdos.



(b) Punto de acuerdo controlable que es conforme



(c) Punto de acuerdo controlable que no es conforme

Figura 3.4: *Conformidad con atributos controlables y no-controlables.*

En este contexto, hablamos de un *punto de acuerdo controlable* para hacer referencia a la parte controlable de un punto de acuerdo, es decir, la combinación de valores asignados a los atributos controlables de dicho punto de acuerdo. A su vez, la *región no-controlable ligada a un punto de acuerdo controlable* incluye todas las combinaciones de valores asignados a atributos no-controlables que forman parte de los puntos de acuerdo que comparten la misma parte controlable.

La región de acuerdos de una oferta de acuerdo está formada por todos los puntos de acuerdo controlables y las regiones no-controlables ligadas a ellos. Las regiones no-controlables no forman necesariamente una partición de la región de acuerdos, sino que puede haber intersecciones entre ellas. Por ejemplo, la figura §3.4(a) muestra una oferta de acuerdo con cuatro puntos de acuerdo controlables, q_1 a q_4, de manera que la región de acuerdos incluye a sus regiones no-controlables, $N(q_1)$ a $N(q_4)$, respectivamente.

Para comprobar la conformidad entre dos ofertas de acuerdo hay que comparar sus correspondientes regiones de acuerdo. Hay conformidad si encontramos un punto de acuerdo controlable que sea conforme. A su vez, un punto de acuerdo controlable es conforme si (**1**) satisface todos los objetivos requeridos y (**2**) la región no-controlable ligada a dicho punto que satisface los objetivos garantizados se incluye en (o iguala a) la región no-controlable también ligada a dicho punto que satisface los objetivos requeridos.

Las figuras §3.4(b) y §3.4(c) muestran los ejemplos correspondientes a un caso de punto de acuerdo conforme y otro no-conforme. En ambas figuras se asume que las ofertas de acuerdo son unilaterales para hacer más fácil la comprensión.

Sea Q_R^δ que representa a la región de acuerdos requerida por la demanda y Q_G^ω representa a la región de acuerdos garantizada por la oferta. En ambos casos hemos encontrados dos puntos de acuerdo controlables q_1 y q_2 que pertenecen a Q_R^δ, pero uno es conforme y el otro no. Explicamos cada caso:

- *Caso positivo*. En la figura §3.4(b) el punto de acuerdo controlable q_1 es conforme porque la región no-controlable ligada a q_1 que cumple los objetivos garantizados, $N_G^\omega(q_1)$, se incluye en la región no-controlable ligada a q_1 que satisface los objetivos requeridos, $N_R^\delta(q_1)$.

- *Caso negativo*. En la figura §3.4(c) el punto de acuerdo controlable q_2 es conforme porque la región no-controlable ligada a q_2 que cumple los objetivos garantizados, $N_G^\omega(q_2)$, no se incluye completamente en la región no-controlable ligada a q_2 que satisface los objetivos requeridos, $N_R^\delta(q_2)$.

Por ejemplo, considere que el proveedor declara que el atributo MEDIA es controlable y MTTF es no-controlable en la oferta con los siguientes términos garantizados:

$$\begin{aligned} &\text{MEDIA} = \{\text{SPTN}\} \Rightarrow 100 \leq \text{MTTF} \leq 120 \\ &\wedge\ \text{MEDIA} = \{\text{COAXIAL}\} \Rightarrow 140 < \text{MTTF} \leq 160 \end{aligned}$$

El cliente propone la demanda con los siguientes términos requeridos:

$$\begin{aligned} &\text{MEDIA} = \{\text{SPTN}\} \Rightarrow \text{MTTF} > 100 \\ &\wedge\ \text{MEDIA} = \{\text{COAXIAL}\} \Rightarrow \text{MTTF} \geq 140 \end{aligned}$$

La oferta es conforme porque existe un punto de acuerdo controlable, $\text{MEDIA} = \{\text{COAXIAL}\}$, que satisface que todo valor de MTTF garantizado cumple el requisito de que sea mayor-igual que 140. El otro punto de acuerdo controlable, $\text{MEDIA} = \{\text{SPTN}\}$, no debe tenerse en cuenta puesto que hay un punto de acuerdo posible en la garantía, $\text{MTTF} = 100$, que no cumple la condición en el requisito de que sea mayor-estricto que 100.

3.3.5. Conformidad con Consciencia Temporal

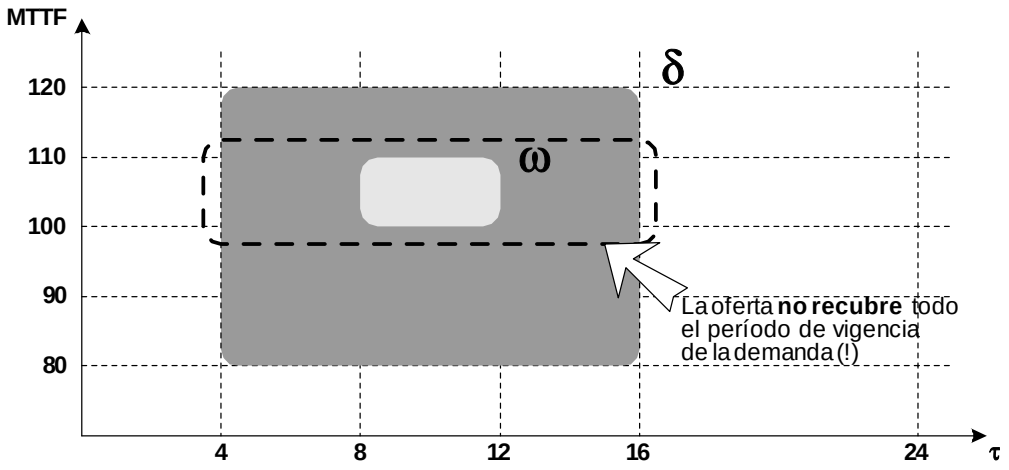
Dando un paso adelante, si las ofertas de acuerdo tienen consciencia temporal, entonces hay que comprobar su conformidad para cada instante temporal del período de vigencia de la demanda. Para ello, necesitamos conocer en cada momento los términos que tienen validez por cada una de las partes. Por ejemplo, consideremos los términos de la oferta y demanda siguientes:

$$\begin{aligned} \omega &\equiv \tau \in [8\,..\,12] \Longrightarrow 100 \leq \text{MTTF} \leq 110 \\ \delta &\equiv \tau \in [4\,..\,16] \Longrightarrow 80 \leq \text{MTTF} \leq 120 \end{aligned}$$

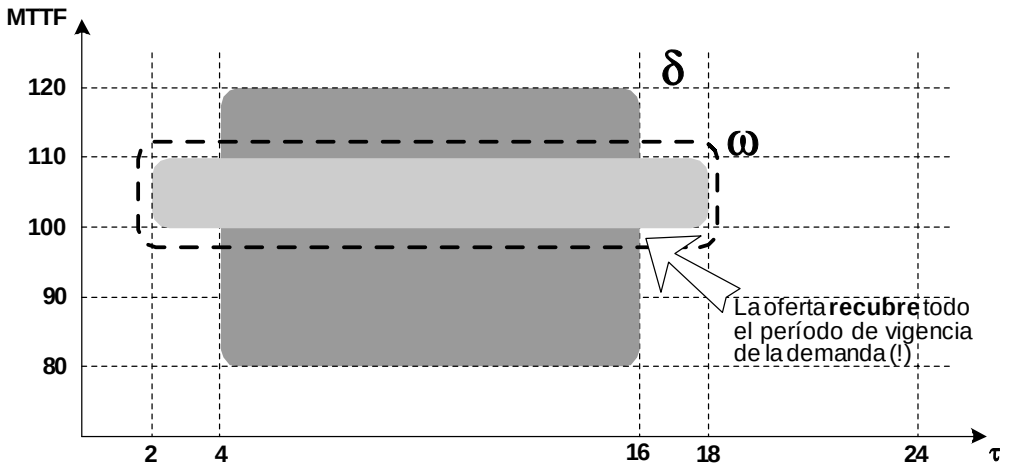
La figura §3.5(a) muestra una representación geométrica de sus regiones de puntos de acuerdo. A primera vista, parece que son conformes, pero sin embargo, el período de vigencia de la oferta no cubre completamente el período de vigencia de la demanda, mas que sólo un pequeño intervalo temporal durante $[8\,..\,12]$.

En cambio, consideremos ahora los términos de la oferta y demanda siguientes:

$$\begin{aligned} \omega &\equiv \tau \in [2\,..\,18] \Longrightarrow 100 \leq \text{MTTF} \leq 110 \\ \delta &\equiv \tau \in [4\,..\,16] \Longrightarrow 80 \leq \text{MTTF} \leq 120 \end{aligned}$$



(a) La oferta *no es* conforme a la demanda.



(b) La oferta *es* conforme a la demanda.

Figura 3.5: *Conformidad con consciencia temporal.*

La figura §3.5(b) muestra su representación geométrica. Nótese que, en esta ocasión, la oferta cubre completamente el período de vigencia de la demanda. En este caso, asumiendo que los atributos son no-controlables, durante el intervalo temporal $[4\,..\,16]$ la región de puntos de acuerdo de la oferta está incluida en la región de puntos de acuerdo de la demanda, por lo que podemos concluir que son conformes.

3.3.6. Conformidad de Doble-Vía

Por último, hay que indicar que el carácter bilateral de las ofertas de acuerdo no añade complejidad computacional al algoritmo, pues simplemente se trata de comprobar la conformidad en ambos sentidos[†5], aunque conceptualmente puede complicar la semántica si abordamos la conformidad en ambas direcciones de manera conjunta. En este contexto, hay conformidad entre dos ofertas de acuerdo si los términos que garantizan ambas partes satisfacen los términos que requieren la parte contraria, respectivamente. Por ejemplo, si un servicio ofrece un conjunto de operaciones criptográficas de 128 bits y está hospedado en un servidor localizado en EE.UU., entonces todas las demandas procedentes del exterior de dicho país no pueden atenderse (sin permiso) porque así lo determinan sus actuales leyes de exportación. Así pues, la oferta debe requerir que todas las demandas de servicio satisfagan el término $\mathrm{CCODE} = \mathrm{US}$.

3.4. Optimalidad

Elegir la oferta de acuerdo óptima de entre todas las ofertas que son conformes es una tarea que, de manera general, se puede interpretar como un problema de toma de decisión multi-criterio (MCDM, *Multiple Criteria Decision Making*) [91, 141]. Por regla general, todas las propuestas revisadas, con independencia de que usen o no de manera explícita técnicas específicas de MCDM, siguen el esquema general de una toma de decisión multicriterio:

- *Establecer quién y cómo toma la decisión.* Lo habitual es que decida el que realiza la búsqueda de los acuerdos, por ejemplo, el intermediario. En cualquier caso, no es una cuestión que influya en la esencia del problema.
- *Establecer los criterios de decisión.* Hasta la fecha, los criterios más habituales consisten en preferencias que el cliente establece sobre sus requisitos, aunque también el proveedor puede indicar criterios similares respecto a sus propios requisitos sobre las demandas que reciba.
- *Ponderar los criterios.* Por regla general, no todos los criterios tienen que tener la misma importancia y es habitual que a cada criterio se le asigne un peso indicador de dicha importancia.

[†5] Por extensión, también hablamos de emparejamiento de doble-via. Añadir que Grefen *et al.* [72] lo ha denominado *modelo de emparejamiento simétrico*, aunque hemos evitado este término para que no haya ambigüedad respecto a la simetría de objetivos.

- *Valorar las alternativas.* Para determinar el grado de cumplimiento de cada criterio por parte de una oferta de acuerdo se suelen emplear técnicas basadas en el cálculo del *grado de similaridad* y del valor respecto de *funciones de utilidad.*

- *Ordenar las alternativas.* Para cada alternativa se determina un valor que suele obtenerse mediante la agregación de las valoraciones de los diferentes criterios.

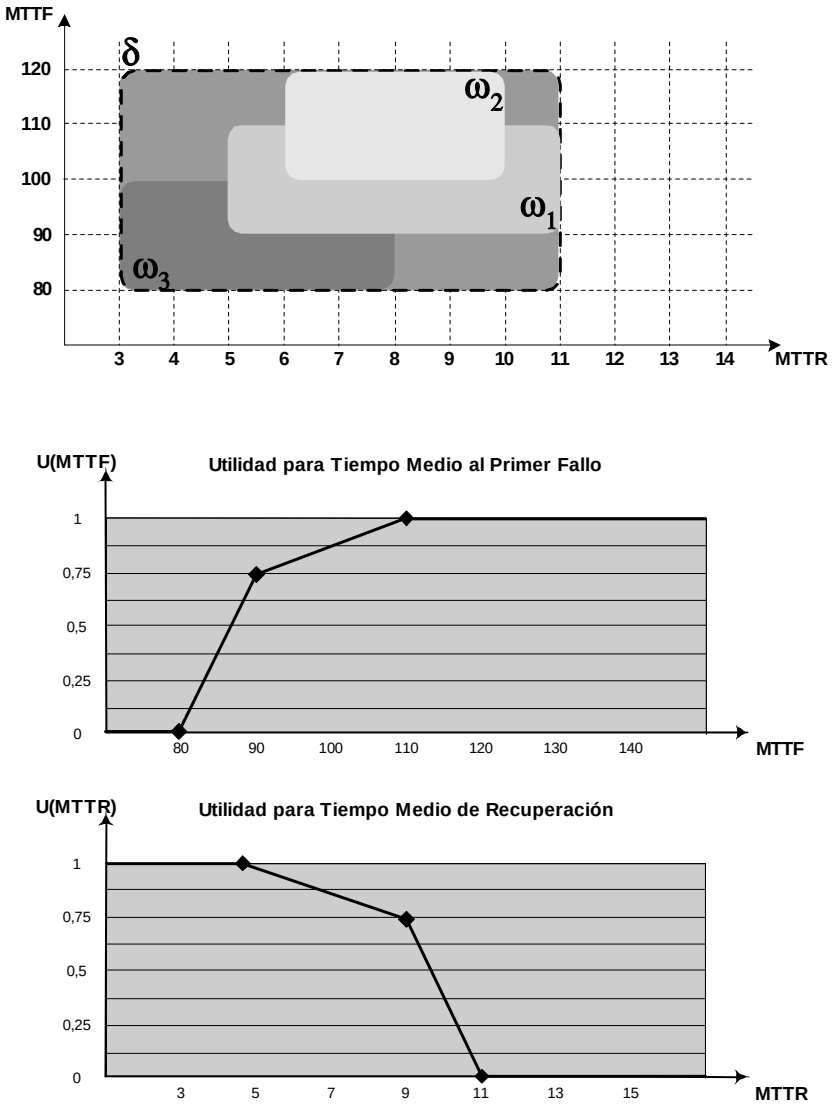


Figura 3.6: *Selección de la mejor oferta.*

Al igual que en el caso de la conformidad, la complejidad computacional del algoritmo necesario para determinar la optimalidad (valorar y ordenar alternativas) viene dada fundamentalmente por el tipo de las ofertas de acuerdo intervinientes en el acuerdo (grado de simetría, expresividad del lenguaje para describir objetivos, grado de consciencia temporal y bilateralidad). No obstante, en este caso, hay factores adicionales que afectan a la complejidad, a saber: el método para calcular la similaridad, el tipo de función de utilidad y el tipo de agregado. A continuación, hacemos una revisión de cada uno de ellos. Hay que decir que, a lo que conocemos, el único modelo de emparejamiento que tiene en cuenta la consciencia temporal para la selección óptima es propuesto por Ruiz-Cortés [131, 166].

3.4.1. Grados de Similaridad

Un camino sencillo para determinar la valoración del emparejamiento entre ofertas de acuerdo viene dada por su *grado de similaridad*, que básicamente consiste en obtener la distancia entre sus regiones de acuerdo. Por regla general, esta distancia viene dada por un agregado que pondera aritméticamente los valores que toman los atributos, o valores de calidad:

$$\mathcal{D}(a_1, \ldots, a_n) = \sum_{i=1}^{n} k_i a_i \quad \text{con} \quad \sum_{i=1}^{n} k_i = 1 \quad \text{y} \quad k_i > 0, i = 1, \ldots, n$$

donde k_i es el peso o importancia relativa del i-ésimo atributo, y a_i es el valor de calidad.

En la figura §3.6 mostramos un ejemplo ilustrativo. En la parte superior hay una demanda (δ) y tres ofertas (ω_1, ω_2 y ω_3). Si seguimos el esquema anterior, obtenemos diferentes distancias según tengamos como referencia los valores máximos y mínimos de cada atributo de calidad, que mostramos en la tabla §3.1(a).

Sin embargo, este esquema tiene sus limitaciones. En primer lugar, los dominios de los atributos no necesariamente tienen que ser iguales y un atributo con un dominio muy grande puede dominar la valoración global del emparejamiento. Una posibilidad para resolver este aspecto es normalizar los valores de los atributos:

$$\mathcal{D}(a_1, \ldots, a_n) = \sum_{i=1}^{n} k_i \frac{a_i}{N_{a_i}}$$

donde N_{a_i} representa la amplitud del dominio del i-ésimo atributo de calidad.

Oferta de Acuerdo	*Peso de MTTF:* 60 % *Peso de MTTR:* 40 %	**Distancia**
δ	Máx: $120 \times 0.6 + 11 \times 0.4 = 76.4$	
	Mín: $80 \times 0.6 + 3 \times 0.4 = 49.2$	
	Promedio: **62.8**	
ω_1	Máx: $110 \times 0.6 + 11 \times 0.4 = 70.4$	6.0
	Mín: $90 \times 0.6 + 5 \times 0.4 = 56.0$	6.8
	Promedio: **63.2**	**0.4**
ω_2	Máx: $120 \times 0.6 + 10 \times 0.4 = 76.0$	0.4
	Mín: $100 \times 0.6 + 6 \times 0.4 = 62.4$	13.2
	Promedio: **69.2**	**6.4**
ω_3	Máx: $100 \times 0.6 + 8 \times 0.4 = 63.2$	13.2
	Mín: $80 \times 0.6 + 3 \times 0.4 = 49.2$	0.0
	Promedio: **56.2**	**6.2**

(a) Distancias entre demanda y ofertas.

Oferta de Acuerdo	*Peso de MTTF:* 60 % *Peso de MTTR:* 40 %	**Distancia**
δ	Máx: $120 \times 0.6/40 + 11 \times 0.4/8 = 2.35$	
	Mín: $80 \times 0.6/40 + 3 \times 0.4/8 = 1.35$	
	Promedio: **1.85**	
ω_1	Máx: $110 \times 0.6/40 + 11 \times 0.4/8 = 2.2$	0.15
	Mín: $90 \times 0.6/40 + 5 \times 0.4/8 = 1.6$	0.25
	Promedio: **1.9**	**0.05**
ω_2	Máx: $120 \times 0.6/40 + 10 \times 0.4/8 = 2.3$	0.05
	Mín: $100 \times 0.6/40 + 6 \times 0.4/8 = 1.8$	0.45
	Promedio: **2.05**	**0.2**
ω_3	Máx: $100 \times 0.6/40 + 8 \times 0.4/8 = 1.9$	0.45
	Mín: $80 \times 0.6/40 + 3 \times 0.4/8 = 1.35$	0.0
	Promedio: **1.625**	**0.225**

(b) Distancias normalizadas entre demanda y ofertas.

Tabla 3.1: *Distancias obtenidas del ejemplo de la figura §3.6.*

En el ejemplo anterior, si normalizamos los valores que toman los atributos de calidad, obtenemos las distancias de la tabla §3.1(b). La oferta óptima es ω_1, a pesar de que ω_2 y ω_3 aparentemente ofrecen mejor calidad de servicio. Nótese que en el promedio, se prefiere aquella cuyo centro más se aproxima a la demanda, independientemente de que en valores absolutos las ofertas que no

se escogen ofrezcan mejor calidad de servicio. Sin embargo, precisamente aquí recae la mayor limitación de estas propuestas, ya que no se puede incorporar la noción de qué valor del dominio del atributo se prefiere, o si las mejores valoraciones se ordenan de manera creciente o decreciente.

3.4.2. Funciones de Utilidad

Las limitaciones arriba mencionadas se pueden solucionar con *funciones de utilidad*. Éstas asignan una valoración o utilidad a cada punto de acuerdo del emparejamiento, de manera que, cuanto mayor sea, mayor es la consideración que tiene desde el punto de vista del cliente y proveedor implicados. La utilidad de un punto de acuerdo suele definirse a partir de un agregado de las funciones de utilidad de cada atributo, asumiendo la independencia entre atributos y ponderadas por su importancia, siguiendo habitualmente un modelo de valoración aritmético [50, 105]:

$$U(a_1, \ldots, a_n) = \sum_{i=1}^{n} k_i U_i(a_i) \quad \text{con} \quad \sum_{i=1}^{n} k_i = 1 \quad \text{y} \quad k_i \geq 0, i = 1, \ldots, n$$

donde k_i es el peso o importancia relativa del i-ésimo atributo, a_i es el valor que tiene asignado y $U_i(a_i)$ es la función de utilidad del atributo a_i, que asigna una utilidad a cada valor en el dominio del atributo, según una escala porcentual.

Las funciones de utilidad pueden presentar diversas formas, aunque por regla general son monótonas crecientes si se prefieren los valores mayores del dominio, o bien monótonas decrecientes si se prefieren los valores menores del dominio. Por ejemplo, el coste de utilizar un servicio podría tener una función de utilidad creciente desde el punto de vista del proveedor (quiere que el cliente le pague más) o decreciente desde el punto de vista del cliente (quiere ahorrarse dinero).

En la tabla §3.2(a) mostramos las valoraciones de utilidad correspondientes al ejemplo de la figura §3.6. Podemos observar que en esta ocasión, la mejor oferta es ω_2, lo que resulta acorde puesto que es la oferta que presenta mejores valores del atributo MTTF, que tiene mayor peso. A esta oferta le sigue ω_1 que tiene peor valoración que MTTR respecto a ω_3, pero este atributo tiene menos peso.

Este modelo presenta limitaciones en contextos donde puede haber un gran número de atributos, o también que los haya obligatorios y opcionales. Por ejemplo, si hay muchos atributos entonces la contribución de cada uno de

Oferta de Acuerdo	*Peso de MTTF:* 60 % *Peso de MTTR:* 40 %
ω_1	Máx: $1 \times 0.6 + 0 \times 0.4 = 0.600$ Mín: $0.75 \times 0.6 + 1 \times 0.4 = 0.850$ Promedio: **0.725**
ω_2	Máx: $1 \times 0.6 + 0.375 \times 0.4 = 0.750$ Mín: $0.875 \times 0.6 + 0.9375 \times 0.4 = 0.900$ Promedio: **0.825**
ω_3	Máx: $0.875 \times 0.6 + 0.8125 \times 0.4 = 0.850$ Mín: $0 \times 0.6 + 1 \times 0.4 = 0.4$ Promedio: **0.625**

(a) Utilidades siguiendo el modelo aritmético.

Oferta de Acuerdo	*Peso de MTTF:* 60 % *Peso de MTTR:* 40 %
ω_1	Máx: $1 \times 0.6 \times 0 \times 0.4 = 0.000$ Mín: $0.75 \times 0.6 \times 1 \times 0.4 = 0.180$ Promedio: **0.090**
ω_2	Máx: $1 \times 0.6 \times 0.375 \times 0.4 = 0.090$ Mín: $0.875 \times 0.6 \times 0.9375 \times 0.4 = 0.197$ Promedio: **0.1435**
ω_3	Máx: $0.875 \times 0.6 \times 0.8125 \times 0.4 = 0.171$ Mín: $0 \times 0.6 \times 1 \times 0.4 = 0.000$ Promedio: **0.085**

(b) Utilidades siguiendo el modelo geométrico.

Tabla 3.2: *Utilidades del ejemplo de la figura §3.6.*

éstos en la función de utilidad puede resultar insignificante; o bien si hay un atributo obligatorio, la utilidad puede que no sea nula si su peso es cero, basta que cualquier otro atributo tenga algún peso aunque sea mínimo. Esto puede afrontarse siguiendo un modelo de valoración geométrico:

$$\mathcal{U}(a_1, \ldots, a_n) = \prod_{i=1}^{n} k_i U_i(a_i)$$

En la tabla §3.2(b) mostramos las valoraciones de utilidad correspondientes. En esta ocasión, nótese que las ofertas ω_1 y ω_3 tienen utilidades muy bajas precisamente porque alguno de sus valores de referencia tiene utilidad cero.

Podemos decir que es un modelo *elitista* en el sentido de que rechaza cualquier oferta en la que alguno de sus atributos tenga un valor sin utilidad.

Sin embargo, este modelo no resuelve el problema respecto a la obligatoriedad de los atributos, ya que en caso de que la valoración de un atributo sea cero, la utilidad global es nula, como se muestra en el ejemplo. Entre otras, estas limitaciones motivaron el modelo de valoración lógica [50]:

$$\mathcal{U}(a_1, \ldots, a_n) = \sum_{i=1}^{n} (k_i U_i(a_i)^r)^{1/r}$$

donde el valor de r depende del grado de conjunción (c) o disyunción (d) que pretendamos en la función de utilidad. Este valor puede obtenerse de múltiples maneras; aquí sólo indicamos que si optamos por una actitud conservadora donde el resultado no sea mayor que el mínimo de las utilidades entonces la conjunción debe ser máxima ($c = 1$), mientras que si optamos por una actitud optimista para que el resultado se corresponda con la máxima de las utilidades entonces la disyunción debe ser máxima ($d = 1$ ó $c = 0$).

3.4.3. Atributos Controlables y No-Controlables

Hay que indicar que las propuestas que hemos estudiados asumen la controlabilidad de los atributos. El tratamiento de la optimalidad cuando hay atributos controlables y no-controlables requiere un tratamiento difícil puesto que implica la existencia de ligaduras entre las variables, por lo que no podemos asumir que los atributos son independientes unos de otros como hacen las propuestas actuales.

Más aún, si hay atributos no-controlables, la optimalidad del emparejamiento puede mejorarse si las partes participantes proporcionasen información sobre las *probabilidades de ocurrencia* de los valores de calidad garantizados cuando los atributos no son controlables. Sin embargo, sólo hemos encontrado la propuesta de Vu *et al.* [201, 202] que tenga en cuenta una estimación de los valores de calidad más probables a partir de estudios estadísticos de las evaluaciones de ejecución del servicio [200].

En general, el problema se trata de encontrar una valoración para cada punto de acuerdo controlable, es decir, una asignación de valores de los atributos controlables, teniendo en cuenta que cada punto de acuerdo controlable va a tener ligada una región no-controlable que posiblemente lleva asociadas probabilidades de ocurrencias. En cada caso podemos seguir diferentes heurísticas:

- Respecto a los atributos controlables podemos maximizar o promediar para comparar los mejores puntos de acuerdo posibles, siguiendo un criterio optimista.

- Por el contrario, respecto a los atributos no-controlables, podemos minimizar para comparar los peores casos, siguiendo un criterio pesimista.

Estas heurísticas puede ofrecer resultados más ajustados si evitamos los criterios pesimistas. Por ejemplo, podemos apostar por los valores no-controlables con mayor utilidad ponderada por la probabilidad de que ocurran efectivamente cuando el servicio se utiliza. O también apostar directamente por aquellos valores no-controlables que tienen mayor probabilidad de ocurrir. Esto es posible siempre que las ofertas de acuerdo incluyan información respecto a la probabilidad de ocurrencia de los valores no-controlables.

Por ejemplo, considere las ofertas de acuerdo de la figura §3.7. En la oferta propiamente dicha se asume que en el caso de $\text{MEDIA} \subseteq \{\text{PSTN}\}$ todos los valores de MTTF siguen una distribución uniforme, por lo que todos ellos son equiprobables. Las funciones de utilidad que expresan los criterios de preferencias del cliente respecto a MTTF y MEDIA se muestran en la parte inferior de la figura.

En este contexto, el punto de acuerdo controlable $\text{MEDIA} \subseteq \{\text{PSTN}\}$ queda descartado porque la oferta no es conforme a la demanda en este caso. De esta manera, la obtención de la utilidad del emparejamiento entre ambas ofertas de acuerdo se restringe al resto de puntos de acuerdos controlables:

- Si $\text{MEDIA} \subseteq \{\text{GSM}, \text{GRPS}\}$ entonces la peor valoración respecto a MTTF es 0.95, que corresponde a $\text{MTTF} = 100$ y su probabilidad de ocurrencia es $0,02$. El punto de acuerdo más probable es $\text{MTTF} = 120$ con probabilidad 0.35 y valoración 1, también corresponde al punto de acuerdo que maximiza el producto entre la probabilidad y su valoración.

- Si $\text{MEDIA} \subseteq \{\text{RSDI}, \text{COAXIAL}\}$ entonces la peor valoración respecto a MTTF es 1, corresponde a $\text{MTTF} = 120$ y su probabilidad es $0,45$, corresponde al más probable y también al punto de acuerdo que maximiza el producto entre probabilidad y valoración.

Las valoraciones de $\text{MEDIA} \subseteq \{\text{GSM}, \text{GRPS}\}$ y $\text{MEDIA} \subseteq \{\text{RDSI}, \text{COAXIAL}\}$ son 0.4 y 0.9, respectivamente.

Considere que los pesos que denotan la importancia asignados a los atributos MEDIA y MTTF son 80 % y 20 %, respectivamente.

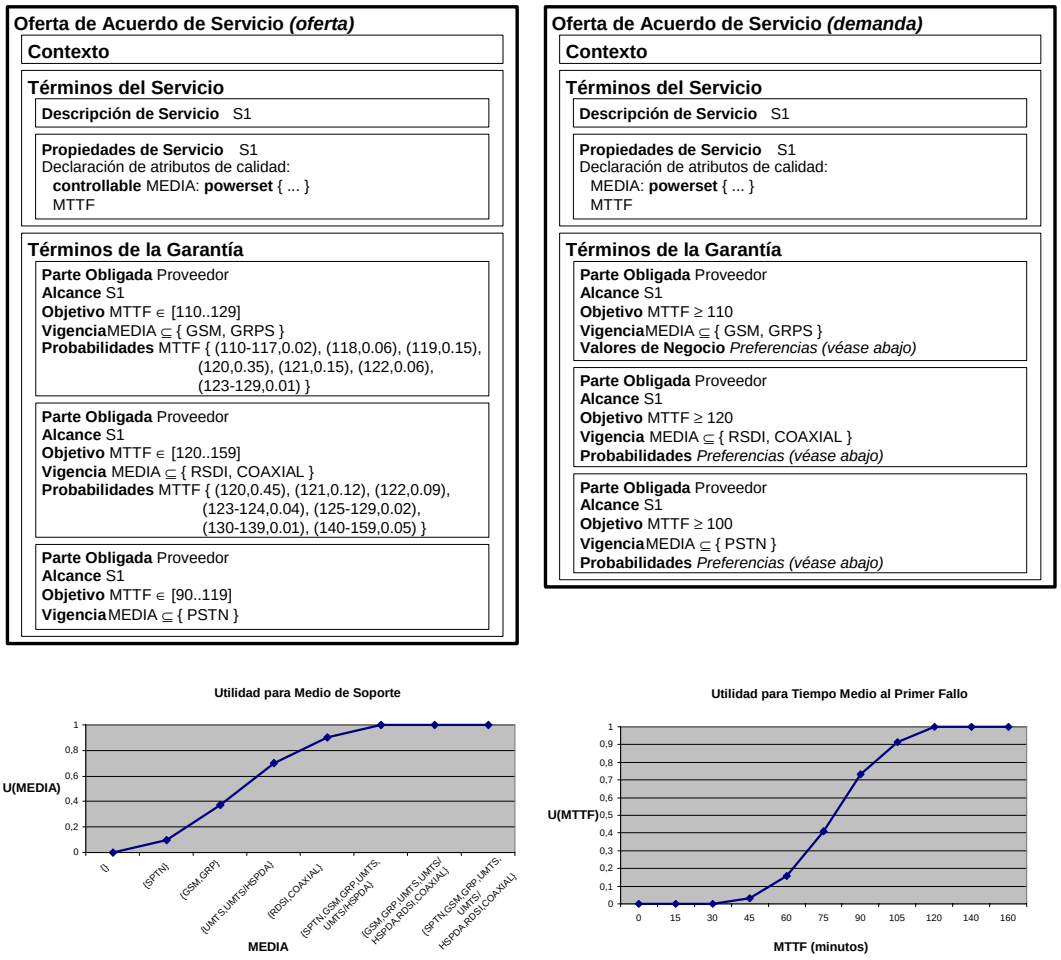


Figura 3.7: *Selección óptima con atributos controlables y no-controlables..*

Esta información es suficiente para proponer una heurística para obtener la utilidad del acuerdo o emparejamiento entre ambas ofertas de acuerdo. Por ejemplo, podemos proponer la expresión:

$$\max_{\text{MEDIA}} \{ 0.80 * \text{U}(\text{MEDIA}) + 0.20 * \min_{\text{MTTF}} \text{U}(\text{MTTF}) \}$$

donde la heurística es optimista frente al atributo controlable y pesimista respecto al atributo no-controlable. En este caso, la optimalidad del emparejamiento corresponde al punto de acuerdo controlable $\text{MEDIA} \subseteq \{\text{RDSI}, \text{COAXIAL}\}$ cuya valoración de utilidad es $0.9 * 0.8 + 1 * 0.2 = 0.92$, frente al punto de acuerdo controlable $\text{MEDIA} \subseteq \{\text{GSM}, \text{GRPS}\}$ y con valoración $0.4 * 0.8 + 0.95 * 0.2 = 0.51$.

Siguiendo otra heurística, también podemos proponer la siguiente expresión:

$$\max_{\text{MEDIA}} \{ 0.8 * \mathcal{U}(\text{MEDIA}) + 0.20 * \max_{\text{MTTF}} \{ \mathcal{U}(\text{MTTF}) * \mathcal{P}(\text{MEDIA}, \text{MTTF}) \} \}$$

donde la heurística es optimista respecto a ambos atributos. En este caso, la utilidad del emparejamiento corresponde al promedio de las valoraciones de sendos puntos de acuerdo conformes. Respecto a $\text{MEDIA} \subseteq \{\text{RDSI}, \text{COAXIAL}\}$, su valoración de utilidad es $0.9 * 0.8 + 1 * 0.45 * 0.2 = 0.81$; respecto a $\text{MEDIA} \subseteq \{\text{RDSI}, \text{COAXIAL}\}$, su valoración es $0.4 * 0.8 + 1 * 0.35 * 0.2 = 0.79$. Así pues, la utilidad del emparejamiento tiene como promedio 0.8.

Nótese que, de manera implícita, los valores de MTTF en cada agregado están ligados al valor actual del atributo MEDIA.

3.4.4. Consciencia Temporal

La temporalidad introduce una complejidad adicional puesto que debe obtenerse la utilidad del mejor punto de acuerdo en cada instante temporal del período de vigencia de la demanda. En este contexto, tanto las funciones de utilidad como las funciones de distribución pueden variar a lo largo de los períodos de vigencia de las ofertas de acuerdo implicadas.

Por ejemplo, consideremos la siguiente oferta y demanda. El proveedor no garantiza la controlabilidad de MTTF y los criterios de preferencias vienen dados por las funciones de utilidad de la figura §2.14. Los términos garantizado y requerido, respectivamente, son:

$$\begin{aligned} \omega &\equiv \tau \in [8\,..\,14] \Longrightarrow 100 \leq \text{MTTF} \leq 140 \\ &\wedge\ \tau \in [15\,..\,20] \Longrightarrow 100 \leq \text{MTTF} \leq 120 \\ \delta &\equiv \tau \in [8\,..\,15] \Longrightarrow \text{MTTF} \geq 100 \end{aligned}$$

En este caso, durante el intervalo temporal $[8\,..\,14]$ la utilidad del emparejamiento es 0.4 correspondiente al punto de acuerdo con peor valoración $\text{MTTF} = 100$; durante el instante temporal $[15\,..\,15]$ la utilidad del emparejamiento es 0.85 correspondiente al mismo punto de acuerdo. Así pues, si promediamos durante el período de vigencia de la demanda, la utilidad del emparejamiento es $(0.4 * 7 + 0.85 * 1)/8 = 0.46$.

3.5. Soporte Automático al Emparejamiento

Las tareas de emparejamiento se afrontan desde múltiples perspectivas. Hasta ahora, en muchas ocasiones estas tareas se han realizado con aproximaciones *ad hoc*, que no son demasiado prácticas por sus limitaciones y escasa capacidad de extensión. Las propuestas actuales ofrecen un soporte automático a estas tareas basado en algún formalismo subyacente que, a su vez, influye en las características del correspondiente modelo de emparejamiento. Esto es debido a que, en la mayor parte de las ocasiones, se identifica el modelo de emparejamiento con el soporte automático empleado para llevar a cabo las tareas de consistencia, conformidad y optimalidad.

A continuación, hacemos una revisión de los formalismos subyacentes más habituales entre las propuestas que hemos estudiado.

3.5.1. Lógicas Descriptivas

En la Web Semántica, el tratamiento de la mayor parte de lenguajes semánticos utilizados para describir acuerdos de calidad está basado en lógicas descriptivas [16, 88], que son una familia de formalismos de representación del conocimiento. Los emparejadores semánticos utilizan razonadores de lógicas descriptivas para llevar a cabo las tareas de emparejamiento. Las principales operaciones que estos razonadores permiten son:

- *Comprobar la consistencia de una descripción semántica*. Esta operación se utiliza para comprobar la consistencia de una oferta de acuerdo.
- *Comprobar que una descripción semántica subsume o es equivalente a otra*. La subsunción es una operación por la que se comprueba si una descripción semántica particular se incluye en una descripción semántica más general. Esta operación se utiliza para comprobar la conformidad, pues intuitivamente los objetivos garantizados deben satisfacer los objetivos requeridos, lo que se puede interpretar como un problema de subsunción. La relación de subsunción permite inferir un grado de emparejamiento semántico, lo que a su vez da soporte a la optimalidad.

El mayor problema de los razonadores es que si el lenguaje de lógicas descriptivas es muy expresivo, su tratamiento puede tener una complejidad computacional muy alta e incluso puede ser no-decidible. De hecho, muchas propuestas restringen la expresividad de estos lenguajes para poder utilizar razonadores eficientes en el emparejamiento de servicios [156, 181, 182].

Una descripción más detallada de las lógicas descriptivas está fuera del alcance de la presente memoria de tesis. El lector interesado puede consultar Baader *et al.* [14] para una visión panorámica, o bien la bibliografía relativa a los principales razonadores, a saber, RACER [74, 75], FACT [86, 87] y FACT++ [89, 194].

3.5.2. Lógicas de Primer Orden

En la Web Semántica, las tareas de emparejamiento también pueden llevarse a cabo mediante motores de inferencia de lógica de primer orden u otros tipos de lógicas. En estas propuestas, el espacio de emparejamiento se representa como una teoría lógica que se utiliza para formalizar tanto las descripciones semánticas de las ofertas de acuerdo como las obligaciones de prueba que deben establecerse para determinar si hay conformidad [101]. Una obligación de prueba se define como una consecuencia lógica entre las fórmulas lógicas que representan a las ofertas de acuerdo, para así comprobar la conformidad. En estas propuestas también hay que enfrentarse al problema de la complejidad computacional versus la expresividad de las descripciones lógicas.

Una descripción más detallada está fuera del alcance de la presente memoria de tesis. El lector interesado puede consultar de Bruijn *et al.* [42, 101] que ofrece un estudio de diferentes razonadores de lógicas descriptivas y motores de inferencia lógica que se utilizan para el tratamiento de las diferentes variantes del lenguaje WSML en el contexto de WSMO.

3.5.3. Programación Lineal

La propuesta de Zeng *et al.* es pionera en el contexto de la composición de servicios. En este trabajo, los autores modelan el problema de encontrar los servicios óptimos para un flujo de trabajo basado en servicios como un problema de programación lineal [212, 213].

Por regla general, la optimalidad se resuelve como un problema de decisión de múltiples criterios. Se define un agregado para obtener una valoración global del acuerdo entre una demanda y una oferta según los criterios de preferencia disponibles. Este agregado se utiliza como función-objetivo.

3.5.4. Programación con Restricciones

Ruiz-Cortés *et al.* [166, 168] son pioneros en la interpretación de las tareas de emparejamiento como problemas de satisfacción de restricciones, permitiendo obtener un modelo de emparejamiento que incluye la mayor parte de características de interés.

El capítulo §4 se dedica a una breve introducción al tema, ya que se trata precisamente del formalismo subyacente que proponemos para resolver las tareas de emparejamiento en esta memoria de tesis doctoral, que presentamos en los capítulos §6 al §8.

3.5.5. Métodos *Ad Hoc*

Muchas propuestas resuelven las tareas de emparejamiento mediante sus propios métodos. En general, desarrollan algoritmos basándose en estructuras de datos tradicionales. Entre ellos, destacan los que plantean algoritmos de optimación mediante cálculo de matrices y similares, para obtener diferentes grados de emparejamiento.

Por regla general, la expresividad de estas propuestas es mucho menor comparada con aquéllas que utilizan formalismos subyacentes más avanzados.

3.5.6. Otras Propuestas

Mencionamos brevemente otros formalismos subyacentes que proporcionan visiones alternativas que pueden resultar de interés:

- Benatallah *et al.* [20, 21] introduce el concepto de cobertura funcional, que consiste en encontrar una agrupación de servicios semánticos que cubren todos los requisitos funcionales. Los algoritmos para realizar la cobertura óptima están basados en recorridos de hipergrafos y reescritura de conceptos ontológicos.

- Canfora *et al.* [32] propone resolver la composición de servicios mediante algoritmos genéticos. El algoritmo de búsqueda sigue una heurística que optimiza una función de ajuste para obtener una valoración global del servicio compuesto, basado en los valores de calidad normalizados.

- Chen *et al.* [36] propone un algoritmo de composición de servicios basado en programación multi-objetivo [91], un esquema de toma de decisiones para obtener un valor que representa una valoración global del servicio compuesto, que permite la selección óptima.

- McIlraith *et al.* [138] presentan uno de los primeros trabajos sobre emparejamiento funcional de servicios semánticos. El algoritmo de emparejamiento se describe con programación dirigidas por modelos y el lenguaje de programación de agentes Golog, basado en cálculo de situaciones para razonar sobre acciones y cambios que éstas provocan.

3.6. Resumen y Reseñas Bibliográficas

En este capítulo hemos presentado las principales tareas que se llevan a cabo durante el emparejamiento, a saber, consistencia, conformidad y optimalidad, en el contexto de un modelo de emparejamiento avanzado. Destacamos la dificultad de la conformidad y la optimalidad cuando se tienen en cuenta el uso conjunto de atributos controlables y no-controlables.

Con este capítulo queda completo el marco de clasificación y comparación con su vista operacional, al que hemos agregado los diferentes soportes operacionales que se han utilizado en el emparejamiento de servicios. La revisión del estado del arte nos ha permitido conocer en qué grado las propuestas actuales son capaces de ofrecer las características avanzadas. Este estudio se muestra en el capítulo §5.

En [129, 166, 168] hemos dado soporte operacional a las tareas de emparejamiento, siendo pioneros en hacerlo con programación con restricciones. En estos trabajos interpretamos las tareas de emparejamiento como problemas de satisfacción de restricciones (véanse los capítulos §4 y §7), en parte debido a la cercanía entre la expresividad de las restricciones y la interpretación geométrica que hemos adoptado de las ofertas de acuerdo. Se abarcan parte de las características avanzadas como la expresividad, simetría y valores de negocio mediante funciones de utilidad complejas.

Estos trabajos han sido extendidos en [131, 166], donde hemos re-definido las tareas de emparejamiento para tener en cuenta la consciencia temporal de objetivos y funciones de utilidad. En [142] mostramos algunos aspectos de implementación de las tareas de emparejamiento según los diferentes grados de expresividad para describir períodos de vigencia.

Capítulo 4

Programación con Restricciones

It's against my programming to impersonate a deity!

Anthony Daniels, C-3PO
"Star Wars: Episode VI - Return of the Jedi", 1983

En este capítulo nuestro objetivo es ofrecer una breve introducción a la programación con restricciones (CP, Constraint Programming). En la sección §4.1 presentamos conceptos básicos de la CP. A continuación, en la sección §4.2 describimos de manera rigurosa algunas definiciones básicas asociadas a los problemas de satisfacción de restricciones que usamos en otros capítulos. Posteriormente, en la sección §4.3 introducimos brevemente algunas técnicas de satisfacción y resolución de restricciones, el problema de la cuantificación universal en los resolutores y una breve presentación de los resolutores comerciales y académicos que hemos valuado y/o utilizado durante el desarrollo de nuestra investigación. Finalmente, en la sección §4.4 concluimos con un resumen.

4.1. Introducción

La programación con restricciones (CP, *Constraint Programming*) ha sido una importante área de investigación de la inteligencia artificial en las últimas décadas [77]. Ha demostrado ser de gran utilidad para resolver problemas complejos del mundo real. Entre otros, problemas de optimación, de planificación y de organización en diversos campos de la ciencia, ingeniería, economía y otros muchos[†1].

El esquema general de la CP consiste en expresar un problema como un conjunto de restricciones entre variables, de manera que un componente software denominado *resolutor* es el responsable de encontrar las soluciones que satisfacen dichas restricciones. En la CP se distinguen dos grandes áreas, que suelen compartir terminología pero difieren en sus orígenes y, sobre todo, en la manera de establecer estos problemas y las técnicas aplicadas para obtener las soluciones. Hablamos de la satisfacción de restricciones (*Constraint Satisfaction*) y de la resolución de restricciones (*Constraint Solving*).

En el primer grupo, los problemas se constituyen de un conjunto finito de variables con dominios finitos y un conjunto finito de restricciones sobre dichas variables. Un problema así expresado se conoce como problema de satisfacción de restricciones (CSP, *Constraint Satisfaction Problem*). Una solución de un CSP está formada por una asignación a cada variable de un valor en su dominio, de manera que todas las restricciones se satisfacen simultáneamente. Para resolver un CSP hay un algoritmo trivial que genera todas las posibles combinaciones de valores y entonces verifica si cada combinación satisface o no todas las restricciones. Si todas las restricciones se satisfacen, entonces la combinación se dice que es una solución. La complejidad computacional de este algoritmo en el peor de los casos es exponencial.

Por ejemplo, la siguiente tupla denota un CSP:

$$\Big(\ \underbrace{\{x, y\}}_{\text{variables}}\ ,\ \underbrace{\{[0\,..\,2], [0\,..\,2]\}}_{\text{dominios}},\ \underbrace{\{x + y < 4, x - y \geq 1\}}_{\text{restricciones}}\ \Big)$$

Por regla general, cuando un problema se expresa como un CSP se puede querer:

- *Encontrar sólo una solución*. En el ejemplo anterior una solución sería:
$$\{x = 2, y = 1\}$$

[†1]El lector interesado puede encontrar numerosas fuentes que introducen la programación con restricciones. Aquí proporcionamos algunas de las que, a nuestro parecer, son más interesantes [11, 60, 62, 77, 125, 177, 193].

- *Encontrar todas las soluciones*. En el ejemplo anterior:
$$\{\{x = 1, y = 0\}, \{x = 2, y = 0\}, \{x = 2, y = 1\}\}$$

- *Encontrar una solución óptima o buena*. A menudo no queremos encontrar una solución, sino la mejor, o siendo prácticos, una que sea lo suficientemente buena. La bondad de una solución se determina por la *función de optimación u objetivo* asociada al CSP en cuestión, que mapea cada solución a un valor numérico. Estos problemas se denominan problemas de optimación (CSOP, *Constraint Satisfaction Optimization Problems*). En estos casos, el resolutor trata de encontrar soluciones que maximizan o minimizan la función-objetivo.

La manera de encontrar estas soluciones depende de las estrategias de búsqueda que puedan aplicarse, dándose el caso de que un problema pueda resolverse utilizando diferentes estrategias.

Por otro lado, en el segundo grupo, los problemas se establecen con variables cuyos dominios son infinitos, tales como los números reales. También es común encontrar restricciones más complicadas en este tipo de problemas, tales como las restricciones no-lineales. Para resolver estos problemas se suelen utilizar métodos algebraicos y numéricos en vez de combinatorios.

4.2. Definiciones Básicas Asociadas a los CSP

En general, un CSP viene dado por un conjunto finito de variables cuyos valores quedan restringidos por sus dominios y restricciones, que determinan las asignaciones de valores a las variables que son soluciones.

Definición 4.1 (CSP) *Un problema de satisfacción de restricciones, denotado con* ψ*, es una tupla de la forma* (V, D, C) *donde:*

- $V = \{v_1, \ldots, v_n\}$ *es el conjunto finito, no vacío, de variables.*
- $D = d_1 \times \ldots \times d_n$ *es el dominio del espacio de soluciones, que viene dado por el producto cartesiano de los dominios de las variables, donde* d_i *es el dominio finito que corresponde a la variable* v_i *con* $i \in [1 \,..\, n]$*.*
- $C = \{C_1, \ldots, C_m\}$ *es un conjunto de* m *restricciones sobre los valores que pueden tomar las variables de* V *en el dominio* D*.*

Respecto a los diferentes lenguajes para describir los CSP, suelen permitir variables de tipo entero y enumerado, a veces conjunto, utilizadas en restricciones basadas en operadores lógicos, relacionales y aritméticos, que dan lugar a expresiones tanto lineales como no-lineales.

Aunque los CSP en sentido estricto están restringidos a variables de dominio finito, se suele usar el mismo acrónimo para hacer referencia a problemas de restricciones que incluyen alguna variable de dominio infinito, por ejemplo los números reales. En estos casos, las restricciones suelen estar restringidas a expresiones lineales. El uso de dominios infinitos como es el caso de los reales implica la utilización de métodos de resolución diferentes.

Definición 4.2 (Asignación y Solución) *Sea ψ un CSP que viene dado por la tupla (V, D, C), una asignación válida v se define asignando a cada variable $v_i \in V$ un valor en su dominio d_i. Considere el conjunto de restricciones C como una función $C : D \longrightarrow \mathbb{B}$ que recibe como entrada una asignación v y devuelve si v satisface el conjunto de restricciones. Si el resultado es cierto, v es una solución del CSP y se denota como s.*

Por ejemplo, sea el siguiente CSP:

$$\psi_1 = (\{x, y\}, \{[0\,..\,2], [0\,..\,2]\}, \{x + y < 4\})$$

El conjunto de todas las asignaciones posibles es $\{(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1), (2,2)\}$. Nótese que sólo una de ellas, la asignación $(2,2)$ no verifica que $2 + 2 < 4$, por lo que decimos que no satisface la restricción y no forma parte del espacio de soluciones.

Respecto a la notación, $C(v)$ denota que la asignación v satisface las restricciones en C, mientras que $\neg C(v)$ denota que no las satisface.

Definición 4.3 (Espacio de Soluciones) *Sea ψ un CSP que viene dado por la tupla (V, D, C), decimos que su espacio de soluciones está conformado por todas las asignaciones posibles que son solución:*

$$\mathrm{sol}(\psi) = \{v \in D \mid C(v)\}$$

Siguiendo el ejemplo anterior, el espacio de soluciones de ψ_1 es:

$$\mathrm{sol}(\psi_1) = \{(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1)\}$$

Definición 4.4 (Satisfactibilidad) *Sea ψ un CSP que viene dado por la tupla (V, D, C), decimos que es satisfactible si y sólo si, su espacio de soluciones no está vacío:*

$$\mathrm{sat}(\psi) \Leftrightarrow \mathrm{sol}(\psi) \neq \emptyset$$

Alternativamente, también decimos que un CSP es satisfactible si y sólo si, tiene *al menos* una solución:

$$\text{sat}(\psi) \Leftrightarrow \exists \upsilon \in D \cdot C(\upsilon)$$

En el ejemplo anterior, ψ_1 es satisfactible ya que su espacio de soluciones no está vacío. En cambio, considere ψ_2 definido como:

$$\psi_2 = \{\{x, y\}, \{[0\,..\,5], [0\,..\,3]\}, \{x \geq 3, x \leq 2, y \leq 1\}\}$$

No es satisfactible puesto que su espacio de soluciones está vacío. Podemos observar que la restricción contiene una contradicción al incluir $x \geq 3$ y $x \leq 2$.

Cuando se está interesado en conocer el espacio de soluciones de un CSP, pero limitado a un subconjunto de variables, se puede hacer uso de la operación de proyección.

Definición 4.5 (Proyección) *Sea ψ un CSP que viene dado por la tupla (V, D, C), siendo V_1 y V_2 sendos conjuntos de variables que constituyen una partición de V, D_1 y D_2 sus dominios respectivos, entonces definimos la proyección de ψ sobre V_1, denotándose Π_{V_1}, como un subespacio de* $\text{sol}(\psi)$ *donde para toda asignación υ_1 en el dominio D_1 hay alguna solución de ψ que la contiene:*

$$\Pi_{V_1}(\psi) = \{\upsilon_1 \in D_1 \mid \exists \upsilon_2 \in D_2 \cdot C(\upsilon_1, \upsilon_2)\}$$

Considérese el siguiente CSP:

$$\psi_3 = \{\{x, y, z\}, \{[0\,..\,5], [0\,..\,3], [\text{'p'}, \text{'q'}]\}, \{x \geq 3, y \leq 1, z = \text{'p'}\}\}$$

donde $\text{sol}(\psi_3) = \{(3, 0, \text{'p'}), (3, 1, \text{'p'}), (4, 0, \text{'p'}), (4, 1, \text{'p'}), (5, 0, \text{'p'}), (5, 1, \text{'p'})\}$. Se tienen las siguientes proyecciones respecto a las variables x, y y z:

$$\begin{aligned}
\Pi_x(\psi_3) &= \{3, 4, 5\} \\
\Pi_y(\psi_3) &= \{0, 1\} \\
\Pi_z(\psi_3) &= \{\text{'p'}\} \\
\Pi_{x,y}(\psi_3) &= \{(3, 0), (3, 1), (4, 0), (4, 1), (5, 0), (5, 1)\} \\
\Pi_{x,z}(\psi_3) &= \{(3, \text{'p'}), (4, \text{'p'}), (5, \text{'p'})\} \\
\Pi_{y,z}(\psi_3) &= \{(0, \text{'p'}), (1, \text{'p'})\} \\
\Pi_{x,y,z}(\psi_3) &= \text{sol}(\psi_3)
\end{aligned}$$

Definición 4.6 (Espacio de Soluciones Óptimas) *Sea ψ un CSP que viene dado por la tupla (V, D, C) y F una función de optimación u objetivo, con signatura $F : D \longrightarrow \mathbb{R}$ que asocia a cada asignación en el dominio D una valoración*

numérica, entonces los espacios de soluciones óptimos están formados por todas las soluciones de ψ *que maximizan o minimizan* F*:*

$$\begin{aligned} \text{máx}(\psi, F) &= \{s \in \text{sol}(\psi) \mid \neg\exists s' \neq s \in \text{sol}(\psi) \cdot F(s) \leq F(s')\} \\ \text{mín}(\psi, F) &= \{s \in \text{sol}(\psi) \mid \neg\exists s' \neq s \in \text{sol}(\psi) \cdot F(s) \geq F(s')\} \end{aligned}$$

Un CSP y una función-objetivo conforman un problema de optimación. Considere el siguiente ejemplo:

$$\begin{aligned} \psi_1 &= (\{x, y\}, \{[0\,..\,2], [0\,..\,2]\}, \{x + y < 4\}) \\ F_1(x, y) &= x \end{aligned}$$

Los espacios de soluciones óptimos de ψ_1 respecto a F_1 son:

$$\begin{aligned} \text{máx}(\psi_1, F_1) &= \{(2, 0), (2, 1)\} \\ \text{mín}(\psi_1, F_1) &= \{(0, 0), (0, 1), (0, 2)\} \end{aligned}$$

4.3. Resolutores

Una vez que un problema se ha modelado con restricciones, dependiendo de su naturaleza, es decir, si se trata de un CSP en sentido estricto o bien se trata de un problema de restricciones con dominios infinitos, podemos utilizar diferentes tipos de resolutores. De éstos, hay un gran número, tanto comerciales como académicos en diferentes lenguajes y plataformas. A continuación, hacemos una breve revisión a las técnicas y métodos utilizados.

4.3.1. Satisfacción de Restriciones

Si se trata de un CSP en sentido estricto, entonces los resolutores tratan de determinar si es satisfactible. Si lo es, pueden devolver una solución, todas las soluciones o la solución óptima, según sea el caso. Hentenryck ofrece una perspectiva de las técnicas para construir el árbol de búsqueda y las estrategias de exploración utilizadas [76, 198]. El árbol de búsqueda es la estructura que toma el espacio de soluciones en las técnicas de satisfacción de restricciones, que tienen dos pasos fundamentales, etiquetado y búsqueda:

- Primero, el *procedimiento de etiquetado estándar* establece el orden de asignación de valores a las variables para encontrar una solución. La ordenación de variables es crítica, pues determina el tamaño y forma del árbol de búsqueda. Usualmente, los resolutores lo construyen mediante

un procedimiento que, por defecto, ordena las variables seleccionando aquélla que tiene el dominio más pequeño, para reducir el tamaño del árbol de búsqueda y descubrir los fallos anticipadamente.

- A continuación, los resolutores aplican una búsqueda en profundidad. Usualmente, a cada variable le asigna un valor de su dominio, entonces aplican la *propagación de restricciones* para eliminar los valores que son inconsistentes y reduce el dominio del resto de variables mediante arco-consistencia. El orden de asignación de valores a la variable de turno, siempre valores en su dominio que no hayan sido eliminados previamente por arco-consistencia, también es muy importante. En caso de fallo, el resolutor vuelve un paso atrás reconsiderando la asignación de la variable actual.

En el caso de los problemas de optimación, los resolutores utilizan esquemas híbridos e incorporan técnicas de Programación Lineal (LP, *Linear Programming*) [41] o Programación Entera (IP / MIP, *Integer / Mixed Integer Programming*) [67]), procedentes de la investigación operativa [184]:

Programación Lineal Los problemas LP consisten en la optimación de una función-objetivo lineal sujeta a un conjunto de ecuaciones o inecuaciones lineales sobre variables reales. Los métodos *simplex* resuelven diversas variantes de estos problemas, que suelen tener una complejidad computacional de orden polinomial.

Programación Entera y Mixta En los problemas IP todas las variables son enteras, mientras que en los problemas de MIP pueden haber variables enteras y reales. Destacan los métodos de ramificación, *Branch & Bound* [110] y *Branch & Cut* [83]. En general, la resolución es un problema NP-completo, aunque hay soluciones con mejor comportamiento para problemas particulares.

En general, para utilizar técnicas de programación lineal que son más eficientes, los resolutores aplican previamente técnicas de relajación lineal para considerar todas las variables como reales.

4.3.2. Resolución de Restricciones

Si se trata de un problema de restricciones con variables de dominios infinitos, entonces el resolutor obtiene, por regla general, soluciones acotadas o aproximadas, basadas en métodos algebraicos o numéricos. Estos métodos

suelen estar basados en aritmética de intervalos [146], donde el espacio de soluciones del problema se restringe a un espacio convexo o caja que lo "engloba". La bondad de la solución depende en gran medida de la precisión de las variables continuas y de la granularidad de intervalos seleccionada, de manera que a mayor bondad la solución obtenida estará más cercana a la solución exacta. Se distinguen dos tipos de aproximaciones:

Externa Garantiza la completitud porque las cajas envuelven todas las soluciones, pero no garantiza la solidez del resultado porque las cajas pueden incluir asignaciones que no forman parte del conjunto de soluciones.

Interna Garantiza la solidez porque las cajas sólo incluyen soluciones, pero no garantiza la completitud porque las cajas no incluyen todas las soluciones [204, 205].

En el caso de la optimación, estos resolutores aplican *métodos de búsqueda local*, basados en diferentes heurísticas que proporcionan una aproximación a la solución óptima [109].

4.3.3. Cuantificación Universal en los Resolutores

Hay una limitación de los resolutores que afecta directamente a nuestra propuesta. En el capítulo §7 asumimos la disponibilidad de un operador de cuantificación universal en el lenguaje empleado para definir restricciones, para definir el CSP que comprueba la conformidad entre ofertas de acuerdos.

Los resolutores son capaces de resolver la satisfactibilidad de un CSP con operador existencial, ya que la propia definición de satisfacción está basada en dicho operador. De hecho, los siguientes CSP son equivalentes[†2]:

$$(\{x, y\}, \{-, -\}, \{\exists x, y \cdot C(x, y)\}) \equiv (\{x, y\}, \{-, -\}, \{C(x, y)\})$$

En cambio, los resolutores no saben resolver la satisfactibilidad del siguiente CSP cuantificado universalmente:

$$(\{x, y\}, \{-, -\}, \{\forall x, y \cdot C(x, y)\})$$

Este esquema es contrario al modelo existencial de los resolutores para resolver la satisfactibilidad. En este caso, podemos transformarlo basándonos

[†2]La notación $\{-, -\}$ en los dominios simplemente denotan que no estamos interesados en mostrarlos explícitamente.

en el hecho de que preguntar si todo par de valores (x, y) satisface una restricción $C(x, y)$ equivale a preguntar si ningún par (x, y) satisface la restricción complementaria $\neg C(x, y)$:

$$(\{x, y\}, \{-, -\}, \{\neg \exists x, y \cdot \neg C(x, y)\})$$

Sin embargo, la transformación anterior no puede hacerse en un CSP como:

$$\psi = (\{x, y, y'\}, \{[0..2], [0..2], [0..2]\}, \{x + y \geq 3, \forall y' \cdot x - y' \geq 0\})$$

que transformándolo, obtenemos:

$$\psi' = (\{x, y, y'\}, \{[0..2], [0..2], [0..2]\}, \{x + y \geq 3, \neg \exists y' \cdot \neg (x - y' \geq 0)\})$$

Este CSP no puede resolverse porque existe una contradicción en su planteamiento. Ambas partes están ligadas por la variable x, además mientras que la parte no-cuantificada universalmente ($x + y \geq 3$) está basada en el modelo existencial, la parte cuantificada universalmente ($x - y' \geq 0$) se ha transformado a un modelo no-existencial.

Este problema ocurre en el caso general donde sólo hay cuantificadas algunas variables y existen ligaduras entre la parte cuantificada y no-cuantificada universalmente:

$$(\{x, y\}, \{-, -\}, \{\exists x, y \cdot C_1(x, y) \wedge \forall y' \cdot C_2(x, y')\})$$

Precisamente este esquema es necesario para definir el CSP para comprobar la conformidad entre ofertas de acuerdo cuando hay atributos controlables y no-controlables, como mostramos en el capítulo §7.

Por todo ello, hay necesidad de un tratamiento específico de los cuantificadores universales en la programación con restricciones. Es una línea abierta en este y otros campos próximos. En concreto, se ha estudiado en el ámbito de la programación lógica [12, 40, 68], programación lineal entera [30, 44, 69, 122] y programación lineal [22, 38, 159].

4.3.4. Resolutores Declarativos e Imperativos

Podemos distinguir entre los resolutores que proponen un lenguaje de restricciones, frente a aquellos donde las restricciones se incrustan en un lenguaje huésped. Los primeros son declarativos, mientras que los segundos imperativos. Entre estos últimos, hemos estudiado CHOCO [108], JaCoP [107] e ILOG [94]. Los dos primeros proponen Java como lenguaje huésped, mientras que

ILOG tiene versiones con C++ y Java. En estas propuestas, se entremezcla el código propio para invocar al resolutor con el código que corresponde al problema de restricciones, haciendo más difícil su comprensión y mantenimiento.

Por otro lado, OPL Studio [95] de ILOG es una herramienta basada en OPL, que es un lenguaje específico para definir problemas de restricciones creado por Hentenryck [78]. Esta definición es declarativa e independiente al lenguaje de programación, siendo los modelos OPL fáciles de comprender y mantener. El resolutor OPL está diseñado como un componente COM, lo que lo hace accesible desde otros entornos de programación. El resolutor incorpora CPLEX [93], un componente que ofrece métodos de programación lineal, programación entera y mixta, que se utilizan en los problemas de optimación. Todas estas características han sido determinantes para elegir OPL Studio para implementar nuestro prototipo.

4.4. Resumen

En este capítulo hemos realizado una introducción a la programación con restricciones, incluyéndose una definición rigurosa así como algunos aspectos fundamentales de la resolución de restricciones.

En nuestra decisión para seleccionar la programación con restriciones para nuestro modelo de solución, aparte de la declaratividad con la que se permiten especificar problemas de restricciones, ha influido la opinión de Smith (1995) [177], que argumenta que la representación de un problema mediante programación con restricciones es frecuentemente más cercana al problema original, ya que las variables de los CSP se pueden hacer corresponder directamente a las entidades del problema y las restricciones se pueden expresar directamente con una formulación más simple, mientras que en la programación entera o mixta el modelado debe adaptarse a una representación matemática más estricta. También hay que decir que hay resolutores de CSP que internamente utilizan técnicas de programación lineal, entera o mixta para resolver problemas de optimación.

Sin embargo, el alto grado de declaratividad de la programación con restricciones también ha sido fuente de confusión durante el trabajo de tesis doctoral. De hecho, en trabajos preliminares propusimos la programación con restricciones como lenguaje único para describir el problema y la solución del emparejamiento [127, 168]. En esta memoria de tesis doctoral, hemos decidido separar claramente el modelo del problema del modelo de solución haciendo uso de teoría de conjuntos y programación con restricciones, respectivamente.

Parte III

Nuestra Propuesta

Capítulo 5

Motivación

You are forgetting the most important fact: motive.

Peter Sellers, The Inspector Clouseau
"A Shot in the Dark", 1964

El emparejamiento de servicios está cobrando cada vez más importancia en general y particularmente sobre las cuestiones asociadas a los aspectos no funcionales. No obstante las propuestas actuales presentan carencias para soportar las características diferenciadoras de los modelos avanzados de emparejamiento, la definición del modelo operacional suele carecer de una descripción rigurosa y la definición del modelo abstracto se suele realizar en los mismos términos que el modelo operacional.

Nuestro objetivo en este capítulo es presentar estas carencias y motivar la necesidad de buscar una solución. En las secciones §5.1 y §5.2 identificamos los problemas que han sido tenidos en cuenta, en la sección §5.3 analizamos en qué medida dichos problemas son resueltos por las propuestas actuales que hemos tenido en cuenta. Posteriormente, en la sección §5.4 las ponemos todas juntas y analizamos sus capacidades de ofrecer las características avanzadas. Finalmente, en la sección §5.5 se ofrece un resumen del capítulo y argumentamos la necesidad para de nueva propuesta.

5.1. Introducción

Recientemente, se ha prestado mucha atención al emparejamiento de servicios, lo que ha generado una gran cantidad de artículos de investigación y experiencias industriales al respecto. El emparejamiento sensible a la calidad es una de las áreas que más crecimiento está experimentando. De hecho, incluso se ha celebrado el primer taller de trabajo "NFP-SLA-SOC *Non Functional Properties and Service Level Agreements in Service-Oriented Computing*" de 2007.

De las conclusiones de dicho taller, se puede afirmar que aún quedan problemas que no están resueltos adecuadamente. Aunque algunas de las propuestas referidas en los capítulos §2 y §3 han abordado estos problemas, ninguna de ellas la consideramos suficientemente adecuada por las siguientes circunstancias:

- No separan de manera explícita el modelo abstracto del modelo operacional.
- No disponen de una definición rigurosa o formal, a excepción de la mayor parte de propuestas relacionadas con la Web Semántica aunque muy ligadas al modelo operacional.
- No soportan características avanzadas tales como:
  - El tratamiento conjunto de atributos controlables y no-controlables.
  - La caracterización estadística de los atributos no-controlables.
  - La descripción de criterios de preferencias mediante funciones de utilidad complejas.
  - La consciencia temporal de algunas partes de las ofertas de acuerdo.

Esta situación nos permite afirmar que no hay una propuesta integral que ofrezca el soporte necesario para llevar a cabo el emparejamiento automático de servicios web. Esta es la principal motivación de nuestra tesis.

5.2. Problemas

Los tres importantes problemas que determinan las limitaciones de las actuales propuestas de emparejamiento son:

Abstracción La programación con restricciones, las lógicas descriptivas y la programación lineal son sólo algunos de los paradigmas que se pueden emplear tanto para el modelo operacional de las tareas de emparejamiento como para el modelo de implementación para obtener un prototipo. No obstante, creemos que resulta muy beneficioso describir la semántica de las tareas de emparejamiento en un nivel de abstracción superior al del modelo de implementación, que permita entre otras cosas, definir un modelo de emparejamiento que:

- No esté limitado a las posibilidades del entorno tecnológico y por tanto, tenga más posibilidades de seguir resultando útil por más tiempo.
- Pueda ser tomado como referencia por los investigadores en el área a fin de comparar las posibilidades de sus respectivas propuestas.

Semántica Formal No existe consenso en cuanto a la semántica de una oferta de acuerdo de nivel de servicio, si bien WS-Agreement empieza a contar con un gran respaldo. A lo que conocemos, tampoco hay propuestas que tengan una semántica formal en cuanto a las tareas de emparejamiento. Sin embargo, las semánticas formales son necesarias para definir de manera rigurosa el emparejamiento de servicios. Sin semántica formal, pueden introducirse falsos conceptos y la construcción de prototipos se más compleja y poco rigurosa.

Soporte para Emparejamiento Avanzado El emparejamiento básico de servicios está soportado en gran medida por la mayoría de propuestas, no así en el caso del emparejamiento avanzado, que en nuestra opinión es una problemática que aún no se ha considerado.

5.3. Análisis de las Soluciones Actuales

Nuestro objetivo en esta sección es sustentar nuestra hipótesis de que ninguna de las propuestas presentadas en el capítulo §3 abordan adecuadamente los tres problemas anteriores de manera conjunta.

5.3.1. Abstracción

No hemos encontrado ninguna propuesta que considere el emparejamiento de servicios a un nivel de abstracción mayor que el que puedan ofrecer los diferentes paradigmas que usualmente se han utilizado. Aunque éstos hayan

permitido de una manera rigurosa definir el emparejamiento, dichas definiciones están ligadas al modelo operacional, lo que nos lleva a los problemas anteriormente mencionados. Creemos que la teoría de conjuntos es adecuada para abstraernos de dichos paradigmas, aunque en el futuro igual hay que utilizar teoría de conjuntos difusos o similares para incorporar nuevas características al emparejamiento avanzado.

5.3.2. Semántica Formal

A lo que conocemos, no hay propuestas con una semántica formal del emparejamiento de servicios independiente del modelo operacional, lo cual tiene sentido dado que este problema es relativamente sencillo mientras no se abordan sus características más avanzadas. No obstante, a fin de poder construir nuevas plataformas o extender las plataformas actuales para que soporten el emparejamiento avanzado y permitan la interoperabilidad, resulta muy útil disponer de una descripción rigurosa o formal de dicha semántica. En este sentido, merece al pena señalar que WSDL en su última versión ha sido especificado con el lenguaje de especificación formal Z[†1].

5.3.3. Soporte para Emparejamiento Avanzado

Se han utilizado diferentes técnicas en el modelo operacional para el emparejamiento de servicios, que hemos introducido en el capítulo §3. Los principales son:

Lógicas Descriptivas Propuestas de la Web Semántica que utilizan razonadores de lógicas descriptivas. Destacan los trabajos de González-Castillo *et al.* [70], Li y Horrocks [112] y MAIS de Bianchini *et al.* [27].

Lógicas de Primer Orden Propuestas de la Web Semántica que utilizan motores de inferencia lógica. Destacan los trabajos sobre PQL de Klein y Bernstein [25, 102], METEOR-S de Cardoso y Sheth [33], WSMO de Roman *et al.* [43, 165] y otros relacionados.

Programación Lineal Propuestas que utilizan las técnicas de programación lineal para obtener la composición óptima para un flujo de trabajo basado en servicios [114, 212, 213]. Por regla general, se utiliza en combinación con otras técnicas.

[†1] http://www.w3.org/2006/Talks/0301-z-notation.pdf

Programación con Restricciones Propuestas que interpretan las tareas de emparejamiento como problemas de satisfacción de restricciones. A lo que conocemos, es la técnica que más características avanzadas permite alcanzar, siendo Ruiz-Cortés [166, 168] pionero en proponer su aplicación en un trabajo precursor de la presente tesis doctoral. En [142] expusimos la manera de complementar la programación con restricciones con algoritmos temporales para abordar las características avanzadas relacionadas con la consciencia temporal.

5.4. Discusión

Las tablas §5.1 y §5.2 muestran sendos resúmenes comparativos de propuestas de la Web Tradicional y la Web Semántica, respectivamente. La fila **Abstr** se refiere a si la propuesta ha distinguido entre diferentes niveles de abstracción; la fila **SemFormal** se refiere a si la propuesta proporciona una semántica formal del emparejamiento, independiente del modelo operacional; y la fila **ModOp** se refiere a si la propuesta ofrece un modelo operacional. En todos los casos, el símbolo $\surd$ significa que la propuesta ha abordado dicho problema, mientras que ~ significa que, a lo que conocemos, la propuesta no ha abordado dicho problema pero sus características nos hace pensar que sí lo ha hecho aunque no se haya publicado explícitamente.

Propuesta	Gouscos [71]	Shi06 [176]	**WS-QoS** [185]	**EWSDL** [36]	Gao [65]	**DQoS** [90]	Ran [164]	**QCWS** [209]	**AgFlow** [213]	**UDDIe** [6]	**WSME** [84]	**WSML** [169]	**WSOL** [189]	Canfora[32]	**WS-CoL** [19]	Mukhi [148]	**GlueQoS** [207]	Ruiz-C. [130, 168]	**WSLA** [118]	**HDM**
Abstr																				$\surd$
SemFormal																				$\surd$
ModOp				$\surd$	$\surd$				$\surd$				$\surd$	$\surd$				$\surd$		$\surd$

Tabla 5.1: *Resumen comparativo de propuestas de la Web Tradicional.*

Propuesta	**WSMO-QoS** [203]	Tsesmetzis[195]	**PQL** [102]	Trastour [190]	**QoSOnt** [46]	**SWSQL** [28]	Yu-jie [210]	Gonzalez-C. [70]	Li03 [112]	**WSAF** [134]	Vu [201, 202]	**DAML-QoS** [216]	**MAIS** [27]	**WSMO/WSML** [165]	**Rei** [97]	**KAoS** [197]	Kolovski [106]	**AMPol-Q** [1]	**METEOR-S** [33]
Abstr																			
SemFormal																			
ModOp		√	√	~	~	~		√	√		√	√	√	√	~	~	√	√	√

Tabla 5.2: *Resumen comparativo de propuestas de la Web Semántica.*

Las tablas §5.3 y §5.4 muestran sendos resúmenes detallados de propuestas de la Web Tradicional y la Web Semántica, respectivamente. Explicamos las abreviaturas utilizadas:

- Respecto a la expresividad de los objetivos, la fila **Rel** se refiere a si la propuesta admite operadores relacionales, la fila **LogRel** si la propuesta admite operadores lógicos y relacionales, la fila **AritRel** si la propuesta admite operadores aritméticos y relacionales y, finalmente, la fila **AritLogReg** si la propuesta admite operadores aritméticos, lógicos y relacionales. Añadir que, por defecto, las propuestas sólo admiten operadores de igualdad.

- Respecto a la simetría de los objetivos, los nombres de las filas son autoexplicativos. Añadir que, por defecto, las propuestas son simétricas pero restringidas a una expresividad con operadores de igualdad.

- Respecto a la consciencia temporal, **MonoInter** se refiere a si la propuesta admite intervalos únicos para expresar períodos de vigencia de los objetivos, **MultIntPer** si admite múltiples intervalos periódicos, **MultINPer** si admite múltiples intervalos no-periódicos y, finalmente, **FUt/EstNC** si la propuesta admite funciones de utilidad y probabilidades de ocurrencias dependientes del tiempo. En las tres primeras filas, el símbolo **G** significa que la propuesta admite ese tipo de intervalos para los períodos globales de vigencia, **L** significa que admite ese tipo de intervalos para los períodos locales de vigencia y **T** significa que admite ese tipo de intervalos tanto para períodos globales como locales. En cursiva significa que la propuesta ha indicado su intención de incorporar esta característica avanzada. En la última fila, el símbolo **U** significa

que la propuesta incorpora funciones de utilidad dependientes del tiempo, y el símbolo **T** significa que a las anteriores también se añaden las probabilidades de ocurrencia dependientes del tiempo.

- Respecto a la consistencia, el nombre de su única línea hace referencia que en el estudio del estado del arte sólo se ha estudiado la consistencia a nivel de objetivos.

- Respecto a la conformidad, **AtribNC** se refiere a si la propuesta considera que todos los atributos son no-controlables y **AtribC/NC** si los atributos pueden ser controlables y no-controlables. Añadir que, por defecto, todas las propuestas asumen que todos los atributos son controlables.

- Respecto a los criterios de preferencia y optimalidad, **Dist/Simil** se refiere a que la propuesta define grados de similaridad, **FunUtil** se refiere a que la propuesta admite funciones de utilidad y **Probabil** se refiere a que la propuesta admite la caracterización estadística de los atributos no-controlables.

- Respecto al modelo operacional, **LógDescr** se refiere a que la propuesta está basada en lógicas descriptivas, **LógPrOrd** que está basada en lógica de primer orden, **ProgLineal** que está basada en programación lineal, **ProgRest** que está basada en programación con restricciones, ***Ad Hoc*** que aborda el emparejamiento con métodos propios y **Otros** que está basada en otros tipos de técnicas menos habituales.

En todos los casos, el símbolo $\surd$ significa que la propuesta tiene la característica que señala la fila correspondiente, mientras que $\sim$ significa que, a lo que conocemos, la propuesta no la tiene o no se ha publicado, o también que los autores han mencionado explicitamente su futura incorporación.

Presentamos detalladamente los resultados de nuestro estudio sobre el estado del arte, donde hemos revisado las propuestas según las características avanzadas de emparejamiento. Este estudio nos ha permitido conocer la capacidad de los diferentes técnicas para utilizar en el modelo operacional para conseguir las mencionadas características:

Expresividad de Objetivos Las propuestas tradicionales suelen ofrecer mayor expresividad. Muchas de ellas presentan el máximo grado de expresividad identificado. Hay que recordar que mayor expresividad implica mayor complejidad computacional. Respecto las propuestas semánticas, destaca la ausencia de tratamiento de las expresiones aritméticas.

Propuesta	Gouscos [71]	Shi [176]	**WS-QoS** [185]	**EWSDL** [36]	Gao [65]	**DQoS** [90]	Ran [164]	**QCWS** [208, 209]	**AgFlow** [213]	**UDDIe** [6]	**WSME** [84]	**WSML** [169]	**WSOL** [189]	Canfora [32]	**WS-CoL** [19]	Mukhi [148]	**GlueQoS** [207]	Ruiz-C. [166]	**WSLA** [118]	**HDM**
Expresividad																				
Rel				√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
LogRel					~					√	√		√		√	√	√	√	√	√
AritRel					~						√		√				√	√	√	√
AritLogRel											√		√				√	√	√	√
Simetría de Objetivos																				
Asimétrico				√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Simétrico												√	√	~	√	√	√	√	√	√
Consciencia Temporal																				
MonoInter			G	G						G		G	G					T		T
MultIntPer	L											L	L					T	L	T
MultINPer																		T		T
FUt/EstNC																		U		T
Consistencia																				
Objetivos													~					√		√
Conformidad																				
AtribNC																		√		√
AtribC/NC																				√
Criterios de Preferencia / Optimalidad																				
Dist/Simil		√		√	√	√		√	√	√				√	~			√		√
FunUtil						~		√						√				√		√
Probabil																				√
Modelo Operacional																				
ProgLineal					√				√					√						
ProgRestr																		√		√
Ad Hoc	√	√	√			√	√	√	√	√	√	√	√		√	√	√		√	√
Otros				√										√						

Tabla 5.3: *Resumen detallado de propuestas de la Web Tradicional.*

Propuesta	**WSMO-QoS** [203]	Tsesmetzis [157, 195]	**PQL** [25, 102]	Trastour [190]	**QoSOnt** [46, 47]	**SWSQL** [28]	Yu-Jie [210]	González-C. [70]	Li [112]	**WSAF** [134]	Vu [201, 202]	**DAML-QoS** [216]	**MAIS** [27]	**WSMO/WSML** [165]	**Rei** [97]	**KAoS** [197]	Kolovski [106]	**AMPol-Q** [1]	**METEOR-S** [33]
Expresividad																			
Rel		√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
LogRel			√		√	√	√				√		√	√	√	√	√	√	√
AritLogRel					√														
Simetría de Objetivos																			
Asimétrico		√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Simétrico		√						√	√	√	√	√	√	√	√	√	√	√	√
Consciencia Temporal																			
Monointer				G	*G*			G	G			G		*T*					*T*
MultIntPer					*L*														
No hay propuestas que traten MultINPer ni FUt/EstNC																			
Consistencia																			
Objetivos		~	~		~	~		√	√		~	√	~	~	~	√	√	√	√
Conformidad																			
No hay propuestas que traten AtribNC ni AtribC/NC																			
Criterios de Preferencia / Optimalidad																			
Dist/Simil	√	√			~		√	~		√	√		√	~				√	√
FunUtil	~							~		√	~			~				~	~
Probabil											√								
Modelo Operacional																			
LógDescr		√		~	~	~		√	√				√		~	~	√		
LógPrOrd			√								√	√		√				√	√
ProgLineal													√						√
ProgRest				√															
Ad Hoc	√			√		√	√			√	~								

Tabla 5.4: *Resumen detallado de propuestas de la Web Semántica.*

La principal razón de la falta de expresividad de muchas propuestas estriba en su soporte operacional, o incluso en la ausencia de dicho soporte como es el caso de los emparejadores *ad-hoc*. Al contrario, las propuestas semánticas suelen estar basadas en razonadores de logicas descriptivas o motores de inferencia de lógica de primer orden, que no son capaces de tratar relaciones aritméticas por sí mismos, por lo que hay una necesidad de utilizar razonadores híbridos.

Simetría de Objetivos La mayor parte de propuestas tradicionales están basadas en emparejadores *ad hoc*, por lo que asumen la asimetría de los objetivos para facilitar su implementación. Es muy fácil comprobar si un punto (único punto de acuerdo que representa a la oferta) pertenece a un conjunto (región de acuerdos de la demanda). Por regla general, la asimetría de objetivos está ligada a los términos unilaterales, excepción hecha de WSME [84, 92] que es una propuesta asimétrica-bilateral precursora de WSLA y WS-Agreement.

Por otro lado, las propuestas tradicionales con simetría de objetivos suelen tener términos bilaterales. Para conseguir la simetría se recomienda la utilización de técnicas que sean capaces de comparar conjuntos (regiones de acuerdo). Las propuestas más interesantes pertenecen a este grupo, destacan WSML [169–171], WSOL [186, 188, 189] y WSLA [99, 118].

Las propuestas semánticas más avanzadas son simétricas por naturaleza, ya que las técnicas en los que suelen estar basadas son capaces de darle soporte. Entre ellas, destacan WSAF [134–136], MAIS [27, 124] y las propuestas de González-Castillo *et al.* [70] y Li y Horrocks [112] con términos unilaterales, mientras que WSMO [43, 165] y METEOR-S [2, 33, 149] son bilaterales.

Consciencia Temporal Aunque algunas propuestas tradicionales han ofrecido esta característica avanzada, ninguna de ellas alcanza el mayor grado de expresividad excepto Ruiz-Cortés [166, 168]. Hay propuestas que permiten que las ofertas de acuerdo tengan asociadas una fecha de expiración, como WS-QoS [185], EWSDL [36] y UDDIe [5, 6]. También hay propuestas que permiten períodos de vigencia locales mediante intervalos temporales con períodos diario y semanal combinados, como el trabajo de Gouscos *et al.* [71] y WSLA [99, 118]. Los lenguajes WSML [169–171] y WSOL [186, 188, 189] admiten ambos tipos de períodos de vigencia.

En el contexto de la Web Semántica, sólo los trabajos de Trastour *et al.* [190], González-Castillo *et al.* [70], Li y Horrocks [112] y DAML-QoS [214–216] admiten fechas de expiración o similares.

O'Sullivan y Edmond [151] identifican las necesidades que debe satisfacer la descripción de intervalos temporales que conforman los períodos de vigencia en el contexto del emparejamiento de servicios. La consciencia temporal es tenida en cuenta por IETF RFC-3060 [145], norma que ha sido incorporada parcialmente por WSLA [99, 118] que es el precursor de WS-Agreement. Por otro lado, Hobbs y Pustejovsky proponen OWL-TIME, una ontología temporal para utilizarla en las descripciones semánticas en OWL [81, 82, 154], que actualmente es una propuesta de trabajo del W3C [80]. Sin embargo, no hay razonador de lógica descriptiva capaz de tratar estas descripciones temporales por su alta complejidad computacional.

Consistencia A nivel de consistencia de objetivos, esta característica avanzada no se ha tenido en cuenta en la Web Tradicional, excepto WSOL [186, 188, 189] a nivel sintáctico y la propuesta de Ruiz-Cortés [166]. Ello es debido a que estas propuestas no tienen modelo operacional capaz de razonar sobre las descripciones de acuerdos para comprobar si contienen contradicciones. Respecto a la Web Semántica, cuyas propuestas sí están basadas en lógicas descriptivas o lógicas de primer orden, sí que abordan esta característica.

Quedan pendientes de estudio la consistencia a nivel de funciones de utilidad y probabilidades de ocurrencia de atributos no-controlables.

Conformidad No hay diferencias entre las propuestas tradicionales y semánticas. Prácticamente todas asumen implícitamente que los atributos son controlables. Sólo Ruiz-Cortés [166] asume en su tesis doctoral la no-controlabilidad de los atributos, aunque no hace un estudio de la conformidad cuando hay ambos tipos de atributos conjuntamente.

Valores de Negocio con Funciones de Utilidad Complejas La mayor parte de propuestas permiten expresar grados de importancia para expresar criterios de preferencias. Usualmente utilizan métodos pre-definidos difíciles de adaptar a nuevas necesidades. En cambio, sólo unas cuantas propuestas permiten describir las preferencias mediante funciones de utilidad. En la Web Tradicional tenemos los trabajos de Yu y Lin [209] y Canfora *et al.* [32], en la Web Semántica sólo el trabajo de WSAF [134–136]. Estas propuestas suelen estar restringidas a funciones monótonas crecientes o decrecientes que se utilizan en agregados de funciones de utilidad que asumen la independencia de los atributos.

Hay que avanzar en la posibilidad de permitir las funciones de utilidad con formas variadas y complejas, tales como exponenciales o sigmoides, no

necesariamente monótonas, que permitan agregados de funciones de utilidad de atributos no necesariamente independientes como mostramos en el capítulo §3, sin olvidarnos de su consciencia temporal, como mostramos en la figura §2.14. A lo que conocemos, sólo el trabajo de Ruiz-Cortés [166] alcanza a tener funciones de utilidad dependientes del tiempo, con formas variadas aunque asume la independencia de los atributos.

5.5. Resumen y Reseña Bibliográfica

En este capítulo hemos estudiado, en primer lugar, los principales problema que tienen las propuestas actuales en torno a tres ejes: abstracción, semántica formal y modelo operacional. Hemos llegado a la conclusión de que la mayor parte de sus limitaciones parten de la total ausencia de abstracción, ya que el modelo de abstracción coincide con el modelo operacional. Respecto al modelo operacional, hemos estudiado las diferentes técnicas que se han aplicado hasta el momento en el emparejamiento de servicios.

A continuación, hemos realizado un estudio más detallado respecto a las características del emparejamiento avanzado. Estas características avanzadas se han considerado como un marco de clasificación y comparación que se ha utilizado para nuestro estudio y se propone para futuras revisiones del estado del arte. Anteriormente, en [129] propusimos una versión preliminar de dicho marco, que ha sido utilizado por Tosic [187] para realizar un estudio del estado del arte. En [142] presentamos una comparación de propuestas respecto a la consciencia temporal. Estos trabajos han sido actualizados por el estudio que hemos presentado en este capítulo.

De este estudio podemos concluir que necesitamos un marco de trabajo para el emparejamiento de servicio, que debería (**1**) separar el modelo abstracto del modelo operacional, (**2**) definir de manera rigurosa el modelo abstracto, y (**3**) que el modelo operacional sea capaz de alcanzar todas las características de emparejamiento avanzado.

Capítulo 6

Marco de Trabajo HDM

He loves her. Love, it's a new starting!
On the other hand, our old ways were once new, weren't they?
On the other hand, they decided without parents, without the matchmaker.
On the other hand, did Adam and Eve have a matchmaker?
Oh, yes they did! And it seems these two have the same Matchmaker...

Haym Topol, Tevye
"Fiddler on the Roof", 1971

Debido a los problemas que hemos identificado en el capítulo anterior, proponemos el marco de trabajo HDM (Holistic Decision Maker). Está estructurado en varios niveles de abstracción. Nuestro principal objetivo es presentar este marco de trabajo y describir de manera rigurosa el modelo abstracto de emparejamiento mediante teoría de conjuntos, independiente del modelo operacional.

Este capítulo se estructura como sigue. En la sección §6.1 se hace una introducción de HDM. A continuación, en la sección §6.2 se define de manera rigurosa el concepto de oferta de acuerdo. Siguen las secciones §6.3, §6.4 y §6.5 donde presentamos la definición rigurosa de la consistencia, conformidad y optimalidad, respectivamente. Finalmente, en la sección §6.6 resumimos el capítulo.

6.1. Introducción

En los capítulos anteriores ya hemos expuesto que nuestro principal interés es proporcionar el soporte necesario para la automatización de las tareas de emparejamiento con un conjunto de características avanzadas. Esta necesidad se materializa con nuestro marco de trabajo para HDM, que proporciona un fundamento para el desarrollo de prototipos de emparejadores automáticos.

6.1.1. Niveles de Abstracción de HDM

La figura §6.1 muestra los niveles de abstracción de HDM. El objetivo del primer nivel es definir de manera rigurosa el modelo abstracto de emparejamiento; el objetivo del segundo nivel es definir el modelo operacional; finalmente, el objetivo del tercer nivel es el modelo de implementación. Describimos estos niveles con mayor detalle:

- *Modelo Abstracto*. Se describen las tareas del emparejamiento de una manera concisa y rigurosa. Hemos utilizado teoría de conjuntos para dar una semántica formal a las tareas de emparejamiento. Hay que resaltar que este modelo es abstracto, estable, libre de formalismos subyacentes para tener independencia del modelo operacional. Este modelo abstracto es de amplio alcance porque recoge todas las características de interés que hemos identificado.

- *Modelo Operacional*. En este nivel se define el soporte operacional del emparejamiento de servicios. Las definiciones rigurosas se transforman para interpretar las tareas de emparejamiento como problemas de satisfacción de restricciones. La programación con restricciones nos permite alcanzar la mayor parte de características avanzadas. Las transformaciones desde el modelo abstracto necesitan de un tratamiento específico de la consciencia temporal mediante un algoritmo de proyección temporal. Este nivel se describen en el capítulo §7.

- *Modelo de Implementación*. Este nivel proporciona la implementación del modelo operacional. Nuestra propuesta está basada en OPL-Studio de ILOG, que es el resolutor de restricciones que hemos integrado en nuestro prototipo. También hay que estudiar la adaptación de este resolutor a nuestras necesidades, como se describe en el capítulo §8.

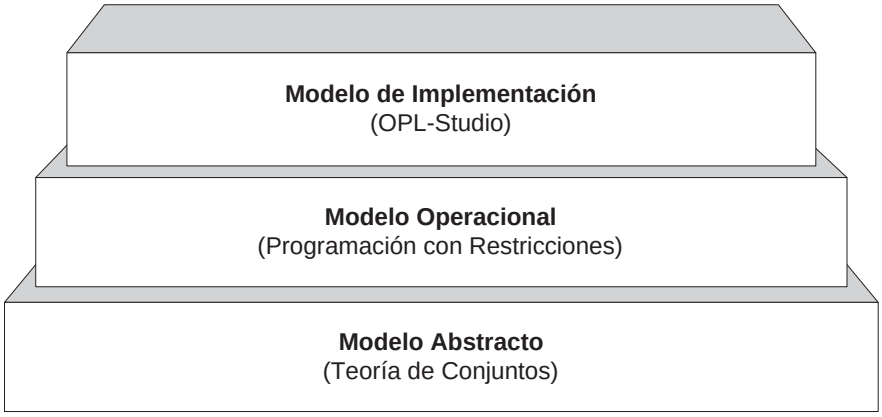


Figura 6.1: *Niveles del marco de trabajo HDM.*

6.2. Modelo Abstracto de HDM

Las ofertas de acuerdo, sean demandas u ofertas propiamente dicha, tienen una estructura abstracta que incluye el conjunto de atributos y sus dominios, los predicados que representan a los objetivos y determinan las regiones de acuerdo, los criterios de preferencias y las probabilidades de ocurrencia que caracterizan a los atributos no-controlables.

Definición 6.1 (Oferta de Acuerdo de Servicio) *Una oferta de acuerdo de servicio, denotada con* α*, es una tupla de la forma* (A, D, P, R, E) *donde:*

- A *representa el conjunto de atributos de calidad.*
- D *representa los dominios de los valores de calidad, es decir, los valores que toman los atributos. Dicho de otro modo, representa el espacio de acuerdos.*
- P *representa el predicado que determina el conjunto de puntos de acuerdos posibles, es decir, la región de acuerdos.*
- R *representa los criterios de preferencia sobre los puntos de acuerdo que se requieren.*
- E *representa la probabilidad de ocurrencia de los puntos de acuerdo que se garantizan.*

Detallamos ahora cada una de estas partes. La figura §6.2 ilustra estos conceptos.

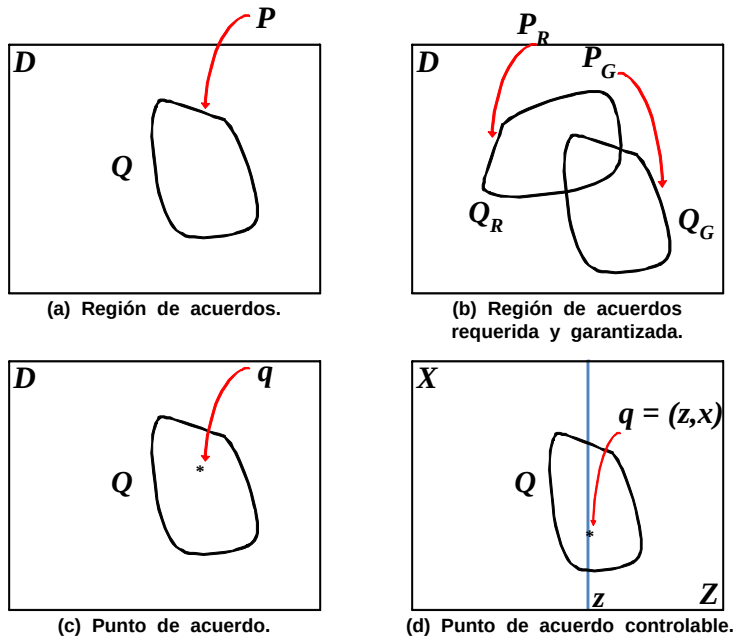


Figura 6.2: *Conceptos básicos de ofertas de acuerdo.*

6.2.1. Espacio de Acuerdos y Puntos de Acuerdo

A representa el conjunto de atributos de calidad, denotado como un conjunto no vacío $A = \{a_1 \ldots, a_n\}$ donde cada elemento a_i es un atributo de calidad.

D representa el espacio de acuerdos, es decir, el dominio de los puntos de acuerdo posibles de la oferta de acuerdo. Viene dado por el producto cartesiano $D = d_1 \times \ldots \times d_n$ donde el i-ésimo dominio d_i corresponde al i-ésimo atributo a_i de A. De esta manera, un punto de acuerdo viene dado por una tupla $q = (v_1, \ldots, v_n)$ que denota la asignación a cada atributo de calidad $a_i \in A$ de un valor en su dominio $v_i \in d_i$ tal que $i \in [1 \,..\, n]$.

Período de Vigencia Por convención, el atributo a_0, también representable como τ, denota al tiempo. Su dominio d_0 es el intervalo temporal correspondiente al período de vigencia de la oferta de acuerdo, que denotamos T_α.

Sea $q = (\tau_0, v_1, \ldots, v_n)$ un punto de acuerdo $q = (v_1, \ldots, v_n)$ vigente en el instante temporal $\tau = \tau_0$. En este contexto, si v representa los valores $v_1, \ldots, v_n$ entonces dicho punto de acuerdo también se denota $q = (\tau_0, v)$.

Controlabilidad Distinguimos entre atributos controlables y no-controlables, que denotamos como A_M y A_N, respectivamente. A su vez, D_M y D_N denotan los dominios correspondientes.

De esta manera, $q = (z, x)$ representan a un punto de acuerdo en el que se distinguen sus partes controlable z determinada por A_M y no-controlable x determinada por A_N. En este contexto, hablamos que z es un punto de acuerdo controlable, de manera que x son los valores que toman las variables no-controlables que están ligados al mismo.

Por otro lado, se asume que τ es un atributo controlable, aunque requiere un tratamiento específico, como mostramos a lo largo del presente capítulo.

6.2.2. Regiones de Acuerdo

La región de acuerdo puede interpretarse como el conjunto Q formado por aquellos puntos de acuerdo que satisfacen los términos de una oferta de acuerdo. La propiedad característica de este conjunto es un predicado P que representa los objetivos y períodos de vigencia de los términos de dicha oferta de acuerdo, que está definido sobre el dominio D:

$$Q = \{\, q \in D \mid P(q) \,\}$$

P_R y P_G denotan los predicados requeridos y garantizados, que corresponden a los objetivos de los términos requeridos y garantizados, respectivamente, de una oferta de acuerdo. Se cumple $P = P_R \wedge P_G$, de manera que un punto de acuerdo posible pertenece a Q si y sólo si satisface ambos predicados $Q = \{\, q \in D \mid P_R(q) \wedge P_G(q) \,\}$. A su vez, Q_R y Q_G denotan las regiones de puntos de acuerdo posibles requerida y garantizada, respectivamente.

6.2.3. Criterios de Preferencias

R denota una familia de relaciones de orden total dependientes del tiempo, que representan los criterios de preferencia incluidos en la oferta de acuerdo. Dados dos puntos de acuerdo $q_1, q_2 \in Q$, la expresión $q_1 \preceq_{R_{\tau_0}} q_2$ indica que el punto de acuerdo q_2 se prefiere al punto de acuerdo q_1 en el instante $\tau = \tau_0$. Suelen venir dadas por grados de similaridad o funciones de utilidad.

6.2.4. Probabilidades de Ocurrencia

E denota una familia de funciones de distribución dependientes del tiempo y de los atributos controlables. Representan las probabilidades de ocurrencia de los valores no-controlables en los puntos de acuerdos garantizados. $E(q_{\tau_0})$ denota la probabilidad de ocurrencia de un valor no-controlable x en el punto de acuerdo $q = (z, x)$ en el instante $\tau = \tau_0$. Suelen venir en forma de funciones de distribución o densidad, según los atributos sean discretos o continuos, respectivamente.

6.2.5. Ejemplo Ilustrativo

En la figura §6.3 se muestra un ejemplo de oferta de acuerdo según el esquema sintáctico de WS-Agreement y su representación abstracta. Nótese que los atributos y sus dominios tienen una correspondencia directa. No es el caso del resto de elementos. La conjunción de los términos que garantiza el proveedor se van a corresponder al predicado P_G de manera que éste viene dado como una expresión de implicación lógica donde las condiciones de vigencia (temporales y atributos controlables) son los antecedentes del predicado y los objetivos asociados son los consecuentes. Igualmente se hace con los términos que se requiere al cliente, que se van a corresponder con el predicado P_R. Los criterios de preferencias y probabilidades de ocurrencia también se construyen a partir de los valores de negocio de los términos.

6.3. Consistencia de una Oferta de Acuerdo

El emparejamiento de servicios supone que las ofertas de acuerdo, demandas u ofertas, deben ser consistentes, es decir, sus elementos constituyentes no deben contener contradicciones. La estudiamos a nivel de objetivos.

6.3.1. Operaciones Auxiliares

Nuestra noción de consistencia requiere la definición previa de los siguientes conceptos.

El filtro de una oferta de acuerdo permite obtener aquellos puntos de acuerdo que cumplen una determinada condición adicional sobre un subconjunto de los atributos.

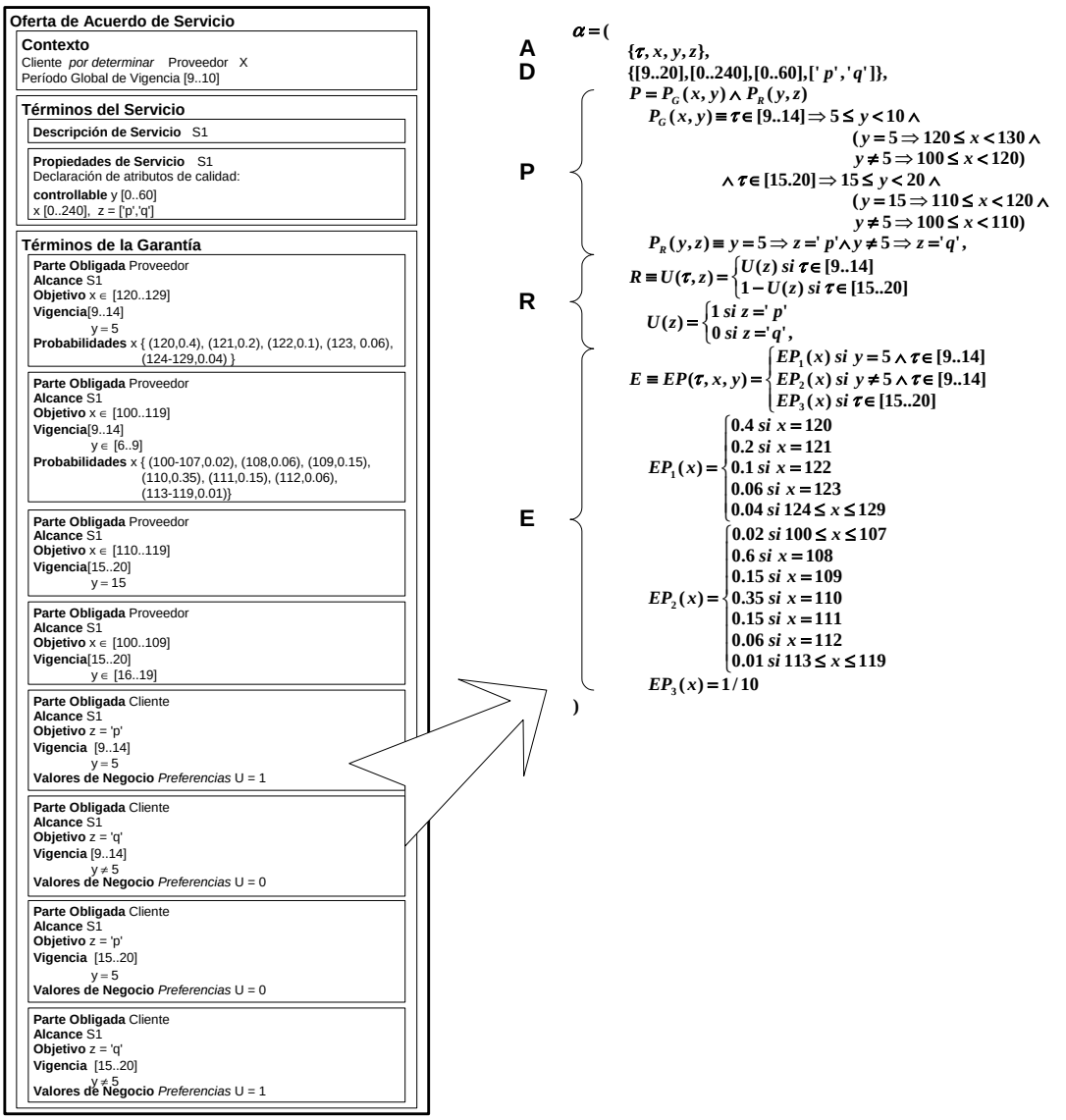


Figura 6.3: *Representación abstracta de una oferta de acuerdo.*

Definición 6.2 (Filtro) *Sea α una oferta de acuerdo dada por una tupla de la forma (A, D, P, R, E) y P' es un predicado definido sobre un subconjunto de atributos $A' \subseteq A$, entonces el filtro de α respecto a P', denotado como $\Phi(\alpha, P')$, resulta otra oferta de acuerdo cuya región de puntos de acuerdo posibles Q' sólo contiene aquellos puntos de acuerdo de Q que también satisfacen P':*

$$Q' = \{\, q \in Q \mid P'(q) \,\}$$

Abreviadamente, este filtro también lo denotamos como $Q^{\alpha}(P')$.

Por ejemplo, sea α_1 una oferta de acuerdo definida como sigue:

$$\alpha_1 = (\{x, y\}, \{[0\,..\,2], [0\,..\,2]\}, \{x < y\})$$

Su región de puntos de acuerdo posibles es $Q^{\alpha_1} = \{(0,1), (1,2)\}$. Si filtramos α_1 con el predicado $P'(x) \equiv x = 1$, entonces la región de puntos de acuerdo resultante es:

$$\Phi(\alpha_1, x = 1) = Q^{\alpha_1}(x = 1) = \{(1,2)\}$$

Filtro Temporal Un filtro temporal permite obtener la región de puntos de acuerdos correspondiente a una oferta de acuerdo que son válidos en un instante temporal concreto.

Abreviadamente, se denota $Q^{\alpha}(\tau = \tau_0)$.

Por ejemplo, sea α_2 una oferta de acuerdo vigente durante $[9\,..\,20]$ y definida como sigue:

$$\alpha_2 = (\{x, y\}, \{[0\,..\,2], [0\,..\,2]\}, \{\tau \in [9\,..\,14] \Rightarrow x < y \wedge \tau \in [15\,..\,20] \Rightarrow x = y\})$$

La región de puntos de acuerdo posibles en el instante temporal $\tau = 10$ del período de vigencia es:

$$\Phi(\alpha_2, \tau = 10) = Q^{\alpha_2}(\tau = 10) = \{(0,1), (1,2)\}$$

6.3.2. Noción de Consistencia

Una oferta de acuerdo consistente es aquélla en la que existe algún punto de acuerdo posible en cada instante temporal de su período de vigencia, es decir, no hay instantes temporales del período de vigencia donde no haya algún punto de acuerdo posible.

Definición 6.3 (Consistencia) *Sea α una oferta de acuerdo y T_α su período de vigencia, decimos que α es consistente si y sólo si en cada instante temporal de su período de vigencia su región de puntos de acuerdos posibles no está vacía:*

$$\text{consistente}(\alpha) \Leftrightarrow \forall \tau_0 \in T_\alpha \;\; Q(\tau = \tau_0) \neq \emptyset$$

6.3.3. Ejemplos Ilustrativos

Presentamos varios ejemplos para ilustrar la noción de consistencia, que corresponden a algunos de los casos de inconsistencia que presentamos en el capítulo §3.

Los intervalos temporales que conforman los períodos locales de vigencia pueden contener errores, no cubren completamente el período global de vigencia Sea la oferta de acuerdo α'_1 vigente durante $[9\,..\,20]$ definida como sigue:

$$\alpha'_1 = (\{x, y, z\}, \{[0..240], [0..60], [\text{'p'}, \text{'q'}]\}, P_R(x, y, z) \wedge P_G(z), -, -)$$

donde:

$$\begin{aligned} P_R^{\alpha'_1}(x, y, z) \equiv \tau \in [9\,..\,14] \Rightarrow (z = \text{'p'} \Rightarrow x \geq 120 \ \wedge \ y \leq 10) \\ \wedge \ (z = \text{'q'} \Rightarrow x \geq 100 \ \wedge \ y \leq 20) \\ \wedge \tau \in [16\,..\,20] \Rightarrow x \geq 100 \ \wedge \ y \leq 20 \\ P_G^{\alpha'_1}(z) \equiv z = \text{'p'} \ \vee \ z = \text{'q'} \end{aligned}$$

Nótese que no se define término alguna para el instante temporal $\tau = 15$, lo que significa que la región de puntos de acuerdo está vacía en dicho momento. Podemos concluir que la oferta de acuerdo no es consistente.

Inalcanzable en algún momento, debido a que los términos válidos durante un instante temporal son contradictorios Sea la oferta de acuerdo α''_1 vigente durante $[9\,..\,20]$ definida como sigue:

$$\alpha''_1 = (\{x, y, z\}, \{[0..240], [0..60], [\text{'p'}, \text{'q'}]\}, P_R(x, y, z) \wedge P_G(z), -, -)$$

donde:

$$\begin{aligned} P_R^{\alpha''_1}(x, y, z) \equiv \tau \in [9\,..\,15] \Rightarrow (z = \text{'p'} \Rightarrow x \geq 120 \ \wedge \ y \leq 10) \\ \wedge \ (z = \text{'q'} \Rightarrow x \geq 100 \ \wedge \ y \leq 20) \\ \wedge \tau \in [15\,..\,20] \Rightarrow 90 \leq x < 100 \ \wedge \ y \leq 20 \\ P_G^{\alpha''_1}(z) \equiv z = \text{'p'} \ \vee \ z = \text{'q'} \end{aligned}$$

Nótese que hay sendos términos válidos para el instante temporal $\tau = 15$. Por un lado, se requiere que $x \geq 100$ en cualquier caso; por otro lado, también se requiere $90 \leq x < 100$. Ambas condiciones son contradictorias entre sí, puesto que no hay valores del dominio que satisfaga ambas condiciones, lo

que significa que no hay punto de acuerdo posible vigente en dicho momento. Si la región de acuerdos está vacía en dicho instante temporal, significa que la oferta de acuerdo no es consistente.

6.4. Conformidad

La comprobación de la conformidad permite conocer si existe algún punto de acuerdo común entre sendas ofertas de acuerdo, oferta y demanda, comprobando que los valores de calidad garantizados por cada parte satisfacen los valores que requiere la otra. Hay múltiples factores a tener en cuenta, a saber: las ofertas de acuerdo son bilaterales, con objetivos simétricos y alta expresividad, hay atributos controlables y no-controlables y los objetivos tienen consciencia temporal.

Hay otros casos más sencillos, como cuando todos los atributos son controlables, o bien todos son no-controlables, cuyas definiciones son triviales y no consideramos en este capítulo.

6.4.1. Operaciones Auxiliares

Nuestra noción de conformidad requiere, aparte del filtro, la definición previa del siguiente concepto.

La proyección de atributos de una oferta de acuerdo permite obtener los valores que toman un subconjunto de atributos en la región de puntos de acuerdos posibles.

Definición 6.4 (Proyección) *Sea α una oferta de acuerdo dada por una tupla de la forma (A, D, P, R, E) y sendos subconjuntos de atributos A' y A'' que constituyen una partición de A, entonces la proyección de α sobre A', denotado como $\Pi(\alpha, A')$, resulta el conjunto de todos aquellos valores de los atributos de A' que forman parte de algún punto de acuerdo en Q:*

$$\Pi(\alpha, A') = \{\, q' = (v') \mid v' \in D', \exists v'' \in D'' \cdot q = (v', v'') \in Q \,\}$$

donde v'' denota unos valores correspondientes a los atributos de A'' que, añadidos a v', conforman un punto de acuerdo posible q de la oferta de acuerdo.

De manera equivalente, podemos denotar la proyección sobre las propias regiones de puntos de acuerdo de esta manera $\Pi(Q, A')$.

Ejemplo Sea α_1 una oferta de acuerdo definida sobre los atributos $\{u, v, x, y\}$. Todos ellos tienen el mismo dominio $[0\,..\,2]$. Se tiene que:

$$Q^{\alpha_1} = \{(1,2,0,0), (1,0,0,1), (1,0,0,2), (1,0,1,2), (1,1,1,2)\}$$

Algunos ejemplos de proyecciones en la oferta de acuerdo anterior:

$$\Pi(Q^{\alpha_1}, \{x, y\}) = \{(0,0), (0,1), (0,2), (1,2)\}$$
$$\Pi(Q^{\alpha_1}, \{u, v, y\}) = \{(1,2,0), (1,0,1), (1,0,2), (1,1,2)\}$$

6.4.2. Noción de Conformidad

La noción de conformidad tiene dos componentes: **(1)** la consciencia temporal y **(2)** la noción de conformidad propiamente dicha, que definimos por separado.

Consciencia Temporal Hay que tener en cuenta la cobertura temporal del período de vigencia de la demanda, es decir, la oferta debe ser conforme a la demanda en cada instante temporal de dicho intervalo temporal.

Definición 6.5 (Conformidad) *Sean δ una demanda y ω una oferta, decimos que ω es conforme a δ si y sólo si, en cada instante temporal de T_δ, período de vigencia de la demanda, existe algún punto de acuerdo controlable que satisface el predicado de conformidad:*

$$\text{conforme}(\omega, \delta) \Leftrightarrow \forall \tau_0 \in T_\delta \;\; \exists z \in D_M \cdot P_C^{\omega,\delta}(\tau_0, z)$$

donde z representa dicho punto de acuerdo controlable y $P_C^{\omega,\delta}$ es el predicado de conformidad entre ω y δ.

Predicado de Conformidad Asumimos que ambas ofertas de acuerdo incluyen términos garantizados basados en atributos tanto controlables como no-controlables.

Definición 6.6 (Predicado de Conformidad) *Sean δ una demanda y ω una oferta, ambas basadas en los mismos conjuntos de atributos A y dominios D, decimos que el punto de acuerdo controlable z es conforme en el instante temporal τ_0 si y sólo si **(1)** existe un valor no-controlable x que, ligado a z, satisfaga los requisitos de ambas ofertas de acuerdo, **(2)** no existe un valor no-controlable x' que, ligado a z, satisfaga los objetivos garantizados de*

la oferta sin satisfacer los objetivos requeridos de la demanda e, igualmente, ***(3)*** *no existe un valor no-controlable x'' que, ligado a z, satisfaga los objetivos garantizados de la demanda sin satisfacer los objetivos requeridos de la oferta:*

$$\begin{aligned} P_C^{\omega,\delta}(\tau_0, z) \Leftrightarrow \exists x \in D_N \cdot P_R^{\omega}(\tau_0, z, x) \wedge P_R^{\delta}(\tau_0, z, x) \\ \wedge (\forall x' \in D_N \cdot P_G^{\omega}(\tau_0, z, x') \Rightarrow P_R^{\delta}(\tau_0, z, x')) \\ \wedge (\forall x'' \in D_N \cdot P_G^{\delta}(\tau_0, z, x'') \Rightarrow P_R^{\omega}(\tau_0, z, x'')) \end{aligned}$$

donde P_R^{ω} y P_G^{ω} son los predicados correspondientes a objetivos requeridos y garantizados de la oferta, y P_R^{δ} y P_G^{δ} son los predicados correspondientes a objetivos requeridos y garantizados de la demanda, respectivamente.

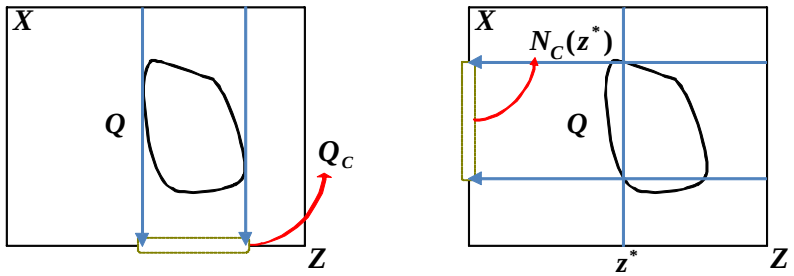
Regiones de Conformidad El conjunto de puntos de acuerdo *conformes* entre una demanda δ y una oferta ω incluye todos los puntos de acuerdo controlables que satisfacen el predicado de conformidad entre demanda y garantía en un instante temporal $\tau = \tau_0$:

$$Q_C^{\omega,\delta}(\tau_0) = \{\, z \in D_M \mid P_C^{\omega,\delta}(\tau_0, z) \,\}$$

La región no-controlable ligada a punto de acuerdo controlable que es conforme, en un instante temporal concreto, viene dado por:

$$N_C^{\omega,\delta}(\tau_0, z) = \Pi(\Phi(Q_C^{\omega,\delta}(\tau_0), A_M = z), A_N)$$

La figura §6.4(b) ilustra la aplicación de proyecciones y filtros para obtener las regiones de conformidad.



(a) Región de puntos de acuerdo que son conformes.

(b) Región no-controlable ligada a un punto de acuerdo controlable.

Figura 6.4: *Proyecciones y filtros para obtener regiones de conformidad.*

6.4.3. Ejemplos Ilustrativos

Presentamos varios ejemplos para ilustrar la noción de conformidad, que corresponden a algunos de los casos que presentamos en el capítulo §3.

No-Conformidad debida a la cobertura incompleta del período global de vigencia de la demanda Sea la demanda δ_1 vigente durante $[9\,..\,20]$, en la que se garantiza la controlabilidad de z, definida como sigue:

$$\delta_1 = (\{x, y, z\}, \{[0..240], [0..60], [\text{'p'}, \text{'q'}]\}, P_R(x, y, z) \wedge P_G(z), -, -)$$

donde:

$$\begin{aligned} P_R^{\delta_1}(x, y, z) \equiv \tau \in [9\,..\,14] &\Rightarrow (z = \text{'p'} \Rightarrow x \geq 120 \ \wedge \ y \leq 10) \\ &\wedge \ (z = \text{'q'} \Rightarrow x \geq 100 \ \wedge \ y \leq 20) \\ \wedge\, \tau \in [15\,..\,20] &\Rightarrow x \geq 100 \ \wedge \ y \leq 20 \\ P_G^{\delta_1}(z) \equiv z = \text{'p'} \ \vee \ z &= \text{'q'} \end{aligned}$$

Por otro lado, sea la oferta ω_2' vigente durante $[9..14]$, en la que se garantiza la controlabilidad de y, definida como sigue:

$$\omega_2' = (\{x, y, z\}, \{[0..240], [0..60], [\text{'p'}, \text{'q'}]\}, P_R(y, z) \wedge P_G(x, y), -, -)$$

donde:

$$\begin{aligned} P_G^{\omega_2'}(x, y) \equiv \tau \in [9\,..\,14] &\Rightarrow 5 \leq y < 10 \\ &\wedge \ (y = 5 \Rightarrow 120 \leq x < 130 \\ &\wedge \ \ y \neq 5 \Rightarrow 100 \leq x < 120) \\ P_R^{\omega_2'}(y, z) \equiv y = 5 \Rightarrow z = \text{'p'} \ \wedge \ y \neq 5 &\Rightarrow z = \text{'q'} \end{aligned}$$

Nótese que los períodos de vigencia de ambas ofertas de acuerdo son diferentes. En concreto, la oferta propiamente dicha sólo está vigente en el intervalo temporal $[9..14]$ y, aunque la oferta es conforme durante el mismo, no cubre totalmente el período de vigencia de la demanda, que se extiende durante el intervalo temporal $[9\,..\,20]$, por lo que tenemos que concluir que la oferta no es conforme.

No-Conformidad debida a incompatibilidad entre requisitos y garantías Considerando la demanda del ejemplo anterior y la siguiente oferta ω_2'', en la que se garantiza la controlabilidad de y, que está vigente durante $[9\,..\,20]$, definida como sigue:

$$\omega_2'' = (\{x, y, z\}, \{[0..240], [0..60], [\text{'p'}, \text{'q'}]\}, P_R(y, z) \wedge P_G(x, y), -, -)$$

donde:

$$\begin{aligned} P_G^{\omega_2''}(x,y) \equiv \tau \in [9\,..\,14] \Rightarrow 5 \leq y < 10 \\ \wedge\ (y = 5 \Rightarrow 120 \leq x < 130 \\ \wedge\ \ y \neq 5 \Rightarrow 100 \leq x < 120) \\ \wedge\, \tau \in [15\,..\,20] \Rightarrow 15 \leq y < 25 \\ \wedge\ (y = 15 \Rightarrow 110 \leq x < 120 \\ \wedge\ \ y \neq 15 \Rightarrow 90 \leq x < 110) \\ P_R^{\omega_2''}(y,z) \equiv y = 5 \Rightarrow z = \text{'p'} \ \wedge\ y \neq 5 \Rightarrow z = \text{'q'} \end{aligned}$$

Hay que encontrar al menos un punto de acuerdo que satisfaga el predicado de conformidad en cada instante temporal de $T_{\delta_1} = [9\,..\,20]$. En nuestro ejemplo, podemos distinguir los instantes temporales en los siguientes intervalos $[9\,..\,14]$ y $[15\,..\,20]$, constatándose que la región de conformidad válida en el segundo de ellos está vacía:

$$\begin{aligned} \tau_0 \in [9\,..\,14] &\rightarrow Q_C^{\omega_2'',\delta_1}(\tau_0) \neq \emptyset \\ \tau_1 \in [15\,..\,20] &\rightarrow Q_C^{\omega_2'',\delta_1}(\tau_1) = \emptyset \end{aligned}$$

Durante el intervalo temporal $[9\,..\,14]$ cualquier par de valores x e y que cumple bien **(1)** las condiciones $120 \leq x < 130$ para $y = 5$ y $z =$ 'p', o bien **(2)** las condiciones $100 \leq x < 120$ para $y \neq 5$ y $z =$ 'q', satisface el predicado de conformidad. Sin embargo, durante el intervalo temporal $[15\,..\,20]$ los valores controlables de y que cumplen $y \neq 15$ no satisfacen el predicado de conformidad porque hay valores no-controlables de $x \in [90\,..\,99]$, ligados a y, que no satisfacen la condición $x \geq 100$.

Debido a la ausencia de puntos de acuerdo durante el intervalo temporal $[15\,..\,20]$, decimos que la oferta ω_2'' no cubre completamente el período de vigencia de la demanda δ_1, por lo que debemos concluir que esta oferta no es conforme.

6.5. Optimalidad

La optimación se presenta como un problema de toma de decisiones, desde la perspectiva del análisis multicriterio y la teoría de probabilidades. Además de todos los elementos a tener en cuenta para obtener las regiones de conformidad, aquí hemos de incorporar los siguientes factores de influencia, a saber: los criterios de preferencias aportados por ambas partes y las probabilidades de ocurrencia que caracterizan estadísticamente a los atributos no-controlables.

6.5.1. Utilidad de un Acuerdo

La utilidad del acuerdo entre una oferta y una demanda, que también referimos como *utilidad del emparejamiento*, nos permite obtener una valoración según los criterios de preferencias que tengamos a disposición.

6.5.1.1. Período de Vigencia de la Demanda

La utilidad del acuerdo se obtiene a partir de las utilidades de acuerdo calculadas en cada instante temporal del período de vigencia de la demanda.

Definición 6.7 (Utilidad de un Acuerdo – Período de Vigencia) *Sean ω y δ una oferta y demanda, respectivamente, entonces la utilidad del acuerdo entre ellos, denotada como $\mathcal{U}(\omega, \delta)$, viene dada por la utilidad agregada durante T_δ, período de vigencia de la demanda:*

$$\mathcal{U}(\omega, \delta) = \underset{\tau_0 \in \mathsf{T}_\delta}{\textbf{\textit{agr}}} \{ \mathcal{U}(\tau_0, \omega, \delta) \}$$

La estrategia para seleccionar el agregado más apropiado puede discurrir entre varias posibilidades:

- Una actitud *pesimista* o conservadora; el objetivo es encontrar la oferta que en algún momento proporcione el peor acuerdo.
- Una actitud *muy optimista*; el objetivo es encontrar la oferta que en algún momento proporcione el mejor acuerdo, pero arriesga ya que es posible que dicha oferta sea mucho peor en otros momentos.
- Una actitud *moderadamente optimista*; el objetivo es encontrar la oferta que proporciona acuerdo óptimo en el promedio, a lo largo del período de vigencia de la demanda.

6.5.1.2. Región de Conformidad

En un instante de tiempo concreto, la utilidad del acuerdo corresponde al punto de acuerdo controlable de la región de conformidad que es óptimo desde ambos puntos de vista, oferta y demanda.

Definición 6.8 (Utilidad de un Acuerdo – Región de Conformidad) *Sean ω y δ una oferta y demanda, respectivamente, entonces la utilidad del acuerdo*

entre ellos en un instante τ_0 del período de vigencia de la demanda, denotada como $\mathcal{U}(\tau_0, \omega, \delta)$, corresponde al punto de acuerdo controlable de la región de conformidad que, en dicho instante, maximiza el producto de sendas funciones de utilidad que representan los puntos de vista de la oferta y la demanda:

$$\mathcal{U}(\tau_0, \omega, \delta) = \underset{z \in Q_C^{\omega,\delta}(\tau_0)}{\textbf{\textit{máx}}} \{ U^{\delta/\omega}(\tau_0, z) \times U^{\omega/\delta}(\tau_0, z) \}$$

donde:

- $U^{\delta/\omega}(\tau_0, z)$ *es la utilidad del punto de acuerdo controlable* z *en* τ_0 *respecto a los criterios de preferencias de* δ *mientras se valoran las garantías dadas por* ω*.*
- $U^{\omega/\delta}(\tau_0, z)$ *es la utilidad del punto de acuerdo controlable* z *en* τ_0 *respecto a los criterios de preferencias de* ω *mientras se valoran las garantías dadas por* δ*.*

6.5.1.3. Utilidad de un Punto de Acuerdo Controlable

Para obtener la utilidad de un punto de acuerdo controlable, hay que tener en cuenta que las ofertas de acuerdo son bilaterales. Así pues, hay que comprobar tanto **(1)** las garantías de la oferta frente a los requisitos de la demanda, como **(2)** las garantías de la demanda frente a los requisitos de la oferta.

Definición 6.9 (Utilidad de un Punto de Acuerdo Controlable) *Sean ω y δ una oferta y demanda, respectivamente, z un punto de acuerdo controlable de $Q_C^{\omega,\delta}$ en el instante τ_0 durante T_δ, entonces hay sendas utilidades de dicho punto de acuerdo controlable, desde los puntos de vista de ω y δ, que denotamos $U^{\omega/\delta}(\tau_0, z)$ y $U^{\delta/\omega}(\tau_0, z)$, respectivamente, que vienen dadas por el agregado de valoraciones de utilidad de cada punto de acuerdo en la región garantizada que está ligada a z, desde ambos puntos de vista:*

$$U^{\omega/\delta}(\tau_0, z) = \underset{x \in N_G^{\delta}(\tau_0, z)}{\textbf{\textit{agr}}} \{ U^{\omega}(\tau_0, z, x), \mathcal{P}^{\delta}(\tau_0, z, x) \}$$
$$U^{\delta/\omega}(\tau_0, z) = \underset{x \in N_G^{\omega}(\tau_0, z)}{\textbf{\textit{agr}}} \{ U^{\delta}(\tau_0, z, x), \mathcal{P}^{\omega}(\tau_0, z, x) \}$$

donde:

- $U^{\delta}(\tau_0, z, x)$ *es la utilidad de un punto de acuerdo* $q = (z, x)$ *en el instante de tiempo* τ_0 *según la demanda, y* $U^{\omega}(\tau_0, z, x)$ *es la utilidad de un punto de acuerdo* $q = (z, x)$ *en el instante de tiempo* τ_0 *según la oferta.*

- $\mathcal{P}^{\delta}(\tau_0, z, x)$ *es la probabilidad de ocurrencia de un punto de acuerdo* $q = (z, x)$ *en el instante de tiempo* τ_0 *según la demanda, y* $\mathcal{P}^{\omega}(\tau_0, z, x)$ *es la probabilidad de ocurrencia de un punto de acuerdo* $q = (z, x)$ *en el instante de tiempo* τ_0 *según la oferta.*

La estrategia para seleccionar el agregado más apropiado puede discurrir entre varias posibilidades:

- Una actitud *pesimista* o conservadora; la referencia es el punto de acuerdo no-controlable con peor valoración de utilidad porque no hay garantías de disfrutar de valores con mejor valoración ni tampoco tenemos información sobre la probabilidad de ocurrencia:

$$\begin{aligned} \mathcal{U}^{\omega/\delta}(\tau_0, z) &= \underset{x \in N_G^{\delta}(\tau_0, z)}{\textbf{mín}} \{ \mathcal{U}^{\omega}(\tau_0, z, x) \} \\ \mathcal{U}^{\delta/\omega}(\tau_0, z) &= \underset{x \in N_G^{\omega}(\tau_0, z)}{\textbf{mín}} \{ \mathcal{U}^{\delta}(\tau_0, z, x) \} \end{aligned}$$

- Una actitud *optimista*; la referencia es el punto de acuerdo no-controlable con mejor valoración de utilidad ponderada por la probabilidad de que efectivamente vaya a ocurrir:

$$\begin{aligned} \mathcal{U}^{\omega/\delta}(\tau_0, z) &= \underset{x \in N_G^{\delta}(\tau_0, z)}{\textbf{máx}} \{ \mathcal{U}^{\omega}(\tau_0, z, x) \times \mathcal{P}^{\delta}(\tau_0, z, x) \} \\ \mathcal{U}^{\delta/\omega}(\tau_0, z) &= \underset{x \in N_G^{\omega}(\tau_0, z)}{\textbf{máx}} \{ \mathcal{U}^{\delta}(\tau_0, z, x) \times \mathcal{P}^{\omega}(\tau_0, z, x) \} \end{aligned}$$

- Una actitud *moderadamente optimista*; la referencia es el punto de acuerdo controlable con mejor promedio de valoraciones de utilidad de su región ligada de puntos de acuerdo no-controlables. Asumiendo que éstos pueden venir dados por cualquier dominio, la expresión se generaliza para obtener la utilidad esperada del punto de acuerdo controlable:

$$\begin{aligned} \mathcal{U}^{\omega/\delta}(\tau_0, z) &= \int_{x \in N_G^{\delta}(\tau_0, z)} \mathcal{U}^{\omega}(\tau_0, z, x) \mathcal{P}^{\delta}(\tau_0, z, x) dx \\ \mathcal{U}^{\delta/\omega}(\tau_0, z) &= \int_{x \in N_G^{\omega}(\tau_0, z)} \mathcal{U}^{\delta}(\tau_0, z, x) \mathcal{P}^{\omega}(\tau_0, z, x) dx \end{aligned}$$

6.5.2. Funciones de Utilidad de Puntos de Acuerdo

Las funciones de utilidad de puntos de acuerdo permiten reflejar los criterios de preferencias de una oferta de acuerdo. En el capítulo §3 repasamos los modelos clásicos de funciones de utilidad. Estos modelos están basados en

agregados de funciones de utilidad para un único atributo de calidad, ponderadas por pesos que denotan la importancia relativa del atributo. Por regla general, requieren la independencia de los atributos de calidad. Nótese que este requisito de independencia hace difícil tanto la definición de una función de utilidad que incluya atributos controlables y no-controlables ligados entre sí, como la definición de una función de utilidad con consciencia temporal, que requiere también una ligadura con el atributo temporal. Estas ligaduras entre diferentes atributos rompen la independencia entre los atributos, necesaria para mantener la linealidad de la función de utilidad. Todos ellos son factores que derivan hacia expresiones no-lineales.

Nuestro modelo propone una función de utilidad como un agregado de *funciones de utilidad parciales*. Estas funciones de utilidad parciales pueden estar definidas sobre varios atributos no necesariamente independientes, con consciencia temporal. En este caso, los pesos no denotan la importancia relativa de cada atributo, sino la importancia relativa de los propios criterios de preferencias que vienen dados por dichas funciones de utilidad parciales.

Definición 6.10 (Utilidad de un Punto de Acuerdo) *Sea α una oferta de acuerdo, la utilidad de un punto de acuerdo constituido por los valores v, denotada como $U(v)$, viene dada por el siguiente agregado ponderado:*

$$U(v) = \underset{i \in [1..q]}{\textbf{\textit{agr}}} \{ k_i, U_i(v_i) \}$$

donde U_i es la i-ésima función de utilidad parcial definida sobre el subconjunto de atributos $A_i = \{a_{i_1}, \ldots, a_{i_p}\} \subseteq A$ y k_i es su peso o importancia relativa. A menor utilidad, peor es la valoración de un punto de acuerdo.

Según la estrategia de toma de decisiones deseada, podemos aplicar cualquiera de los modelos, aditivo lineal, geométrico o valoración lógica, de las nociones de utilidad que mostramos en el capítulo §3.

Asignación de los Pesos Reseñamos brevemente este problema que puede ser complejo. Los pesos son propuestos por los clientes y proveedores, que suelen tener naturaleza semicualititativa. Podemos encontrar indicadores de preferencia tales como “alta prioridad” y “baja prioridad”, cada uno con un peso predeterminado y configurable, de manera que se normalizan para que cumplan las propiedades requeridas, es decir, deben tener un valor entre 0 y 1, sumando todos 1. El lector interesado puede consultar [166] para un estudio detallado.

6.5.3. Distribuciones de Probabilidad de Ocurrencia

En general, si los dominios de los atributos son discretos, las distribuciones de probabilidades de ocurrencias siguen funciones de distribuciones multinomiales. Si los dominios son continuos, entonces hablamos de funciones de densidad. En cualquier caso, para cada instante de tiempo, para cada valor controlable, la distribución multinomial (o densidad) asigna una probabilidad de ocurrencia a cada valor no-controlable ligados a ellos.

Hay que recordar que la principal propiedad de una función de distribución es que la suma de todas las probabilidades deba ser 1. En el caso de las funciones de densidad, la integral de la función en todo su dominio también debe ser 1. Este criterio debe cumplirse cuando se considera que todas las variables son independientes. Sin embargo, en nuestro contexto ya hemos comentado la dependencia respecto al eje temporal y los atributos controlables; en este caso, la propiedad anterior debe satisfacerse para la suma de probabilidades de los valores no-controlables, para cada intervalo temporal de interés y punto de acuerdo controlable, como mostramos a continuación.

6.5.4. Ejemplo Ilustrativo

En este ejemplo, determinamos la utilidad del acuerdo entre una demanda y una oferta, ilustrando los diferentes puntos de interés que hemos presentado hasta el momento. La oferta se corresponde a la oferta de acuerdo que hemos presentado en el ejemplo de la figura §6.3.

Demanda Sea la demanda δ_1 vigente durante $[9\,..\,20]$, en la que se garantiza la controlabilidad de z, definida como sigue:

$$\delta_1 = (\{x, y, z\}, \{[0..240], [0..60], [\text{'p'}, \text{'q'}]\}, P_R(x, y, z) \wedge P_G(z), -, -)$$

donde:

$$\begin{aligned} P_R^{\delta_1}(x, y, z) \equiv \tau \in [9\,..\,14] \Rightarrow (z = \text{'p'} \Rightarrow x \geq 120 \wedge y \leq 10) \\ \wedge\ (z = \text{'q'} \Rightarrow x \geq 100 \wedge y \leq 20) \\ \wedge\ \tau \in [15\,..\,20] \Rightarrow x \geq 100 \wedge y \leq 20 \\ P_G^{\delta_1}(z) \equiv z = \text{'p'} \vee z = \text{'q'} \end{aligned}$$

La demanda δ_1 define una relación de orden total R^{δ_1} por la que se prefieren **(1)** durante el intervalo temporal $[9\,..\,14]$ aquellas ofertas con $z = \text{'p'}$, con

valores mayores de x pero menores de y, mientras que **(2)** durante el intervalo $[15\,..\,20]$ aquellas ofertas con valores mayores de y pero menores de x.

Al no haber atributos no-controlables, la demanda no incluye probabilidades de ocurrencias.

Oferta Sea la oferta ω_2 vigente durante $[9\,..\,20]$, en la que se garantiza la controlabilidad de y, definida como sigue:

$$\omega_2 = (\{x, y, z\}, \{[0..240], [0..60], ['p', 'q']\}, P_R(x, y) \wedge P_G(y, z), -, -)$$

donde:

$$\begin{aligned} P_G^{\omega_2}(x, y) \equiv \tau \in [9\,..\,14] &\Rightarrow 5 \leq y < 10 \\ &\wedge\ (y = 5 \Rightarrow 120 \leq x < 130 \\ &\wedge\ \ y \neq 5 \Rightarrow 100 \leq x < 120) \\ \wedge\, \tau \in [15\,..\,20] &\Rightarrow 15 \leq y < 20 \\ &\wedge\ (y = 15 \Rightarrow 110 \leq x < 120 \\ &\wedge\ \ y \neq 15 \Rightarrow 100 \leq x < 110) \\ P_R^{\omega_2}(y, z) \equiv y = 5 \Rightarrow z = \text{'p'} \ \wedge\ y \neq 5 &\Rightarrow z = \text{'q'} \end{aligned}$$

La oferta ω_2 define una relación de orden total R^{ω_2} por la que se prefieren **(1)** aquellas demandas que garanticen $z =$ 'p' durante el intervalo temporal $[9\,..\,14]$ y **(2)** aquellas demandas con $z =$ 'q' durante el intervalo temporal $[15\,..\,20]$.

La oferta ω_2 define las probabilidades de ocurrencia E^{ω_2} del atributo no controlable x, que dependen del tiempo y del atributo controlable. Durante el intervalo temporal $[9\,..\,14]$ las probabilidades de ocurrencia siguen la siguiente función de distribución multinomial si $y = 5$:

120	**121**	**122**	**123**	**124-129**
0.40	0.20	0.10	0.06	0.04

o bien esta otra si $y \neq 5$:

100-107	**108**	**109**	**110**	**111**	**112**	**113-119**
0.02	0.06	0.15	0.35	0.15	0.06	0.01

Durante el intervalo temporal $[15\,..\,20]$ las probabilidades de ocurrencias siguen una distribución uniforme, es decir, todos los valores son equiprobables para cualquier valor controlable en cualquier instante de tiempo del período de vigencia de la demanda.

Regiones de Conformidad El conjunto de puntos de acuerdo conformes en el intervalo temporal de interés $\tau_0 \in [9\,..\,14]$ es:

$$Q_C^{\omega_2,\delta_1}(\tau_0) = \{\, y, z \mid z = \text{'p'} \Rightarrow y = 5 \;\wedge\; z = \text{'q'} \Rightarrow\; 6 \leq y < 10 \,\}$$

Mientras que en el intervalo temporal de interés $\tau_1 \in [15\,..\,20]$ tenemos:

$$Q_C^{\omega_2,\delta_1}(\tau_1) = \{\, y, z \mid z = \text{'q'} \;\wedge\; 15 \leq y < 20 \,\}$$

Ambos conjuntos de puntos de acuerdo conformes constituyen la región de conformidad entre la oferta y la demanda.

Función de Utilidad del Acuerdo La utilidad del acuerdo entre ω_2 y δ_1 viene dada por la siguiente expresión:

$$\mathcal{U}(\omega_2, \delta_1) = \underset{\tau_0 \,\in\, T_{\delta_1}}{\textbf{promedio}} \{ \underset{y,z \,\in\, Q_C^{\omega_2,\delta_1}(\tau_0)}{\textbf{máx}} U^{\delta_1/\omega_2}(\tau_0, y, z) \times U^{\omega_2/\delta_1}(\tau_0, y, z) \}$$

donde las funciones de utilidad de los puntos de acuerdos controlables son:

$$\begin{aligned} U^{\delta_1/\omega_2}(\tau_0, y, z) &= \underset{x \,\in\, N_G^{\omega_2}(\tau_0, y, z)}{\textbf{máx}} U^{\delta_1}(\tau_0, x, y, z) \times \mathcal{P}^{\omega_2}(\tau_0, x, y) \\ U^{\omega_2/\delta_1}(\tau_0, y, z) &= \underset{x \,\in\, N_G^{\delta_1}(\tau_0, y, z)}{\textbf{mín}} U^{\omega_2}(\tau_0, z) \quad = \quad U^{\omega_2}(\tau_0, z) \end{aligned}$$

Funciones de Utilidad de Puntos de Acuerdo A su vez, las funciones de utilidad de los puntos de acuerdo que corresponden a R^{δ_1} y R^{ω_2} son:

$$U^{\delta_1}(\tau, x, y, z) = \begin{cases} 0.7 * U^{\delta_1}(z) + 0.3 * U^{\delta_1}(x, y) & \text{si } \tau \in [9\,..\,14] \\ 1 - U^{\delta_1}(x, y) & \text{si } \tau \in [15\,..\,20] \end{cases}$$

$$U^{\omega_2}(\tau, z) = \begin{cases} U^{\omega_2}(z) & \text{si } \tau \in [9\,..\,14] \\ 1 - U^{\omega_2}(z) & \text{si } \tau \in [15\,..\,20] \end{cases}$$

donde las funciones de utilidad parciales se definen como sigue:

$$U^{\delta_1}(x, y) = \begin{cases} x/x + y & \text{si} \quad x \geq 90 \;\wedge\; y \leq 10 \\ 0 & \text{en caso contrario} \end{cases}$$

$$U^{\delta_1}(z) = U^{\omega_2}(z) = \begin{cases} 1 \text{ si } z = \text{'p'} \\ 0 \text{ si } z = \text{'q'} \end{cases}$$

Funciones de Distribución Las probabilidades de ocurrencia que corresponden a E^{ω_2} son:

$$\mathcal{P}^{\omega_2}(\tau, x, y) = \begin{cases} \mathcal{P}_1^{\omega_2}(x) \text{ si } y = 5 \wedge \tau \in [9\,..\,14] \\ \mathcal{P}_2^{\omega_2}(x) \text{ si } y \neq 5 \wedge \tau \in [9\,..\,14] \\ \mathcal{P}_3^{\omega_2}(x) \text{ si } \tau \in [15\,..\,20] \end{cases}$$

donde las funciones de distribución particulares son:

$$\mathcal{P}_1^{\omega_2}(x) = \begin{cases} 0.40 \text{ si } x = 120 \\ 0.20 \text{ si } x = 121 \\ 0.10 \text{ si } x = 122 \\ 0.06 \text{ si } x = 123 \\ 0.04 \text{ si } 124 \leq x \leq 129 \end{cases} \qquad \mathcal{P}_2^{\omega_2}(x) = \begin{cases} 0.02 \text{ si } 100 \leq x \leq 107 \\ 0.06 \text{ si } x = 108 \\ 0.15 \text{ si } x = 109 \\ 0.35 \text{ si } x = 110 \\ 0.15 \text{ si } x = 111 \\ 0.06 \text{ si } x = 112 \\ 0.01 \text{ si } 113 \leq x \leq 119 \end{cases}$$

$$\mathcal{P}_3^{\omega_2}(x) = 1/10$$

Tabla de Utilidades de los Puntos de Acuerdo Conformes A continuación, mostramos las tablas de resultados parciales para obtener las valoraciones de utilidad de los puntos de acuerdo conformes de $Q_C^{\omega_2,\delta_1}$:

- En el intervalo temporal $[9\,..\,14]$, si $z =$ 'p', las valoraciones de utilidad correspondientes a U^{δ_1} son:

y \ x	120	121	122	123	124	125	126	127	128	129
5	0.9880	0.9881	0.9882	0.9883	0.9884	0.9885	0.9885	0.9886	0.9887	0.9888

- En cambio, si $z =$ 'q' entonces:

y \ x	100	101	102	103	104	105	106	107	108	109
6	0.2830	0.2832	0.2833	0.2835	0.2836	0.2838	0.2839	0.2841	0.2842	0.2843
7	0.2804	0.2805	0.2807	0.2809	0.2811	0.2812	0.2814	0.2816	0.2817	0.2819
8	0.2778	0.2780	0.2782	0.2784	0.2786	0.2788	0.2789	0.2791	0.2793	0.2795
9	0.2752	0.2754	0.2757	0.2759	0.2761	0.2763	0.2765	0.2767	0.2769	0.2771

y \ x	110	111	112	113	114	115	116	117	118	119
6	0.2845	0.2846	0.2847	0.2849	0.2850	0.2851	0.2852	0.2854	0.2855	0.2856
7	0.2820	0.2822	0.2823	0.2825	0.2826	0.2828	0.2829	0.2830	0.2832	0.2833
8	0.2797	0.2798	0.2800	0.2802	0.2803	0.2805	0.2806	0.2808	0.2809	0.2811
9	0.2773	0.2775	0.2777	0.2779	0.2780	0.2782	0.2784	0.2786	0.2787	0.2789

- En el intervalo temporal $[15\,..\,20]$, todos los puntos de acuerdo tienen utilidad igual a 1 según δ_1.

Las valoraciones de utilidad anteriores se ven ponderadas por las probabilidades de ocurrencia que vienen dadas por $\mathcal{P}^{\omega_2}$:

- En el intervalo temporal $[9\,..\,14]$, si $z =$ 'p', entonces $\mathcal{U}^{\delta_1} \times \mathcal{P}^{\omega_2}$:

y \ x	**120**	**121**	**122**	**123**	**124**	**125**	**126**	**127**	**128**	**129**
5	0.3952	0.1976	0.0988	0.0593	0.0395	0.0395	0.0395	0.0395	0.0395	0.0395

- En cambio, si $z =$ 'q' entonces:

y \ x	**100**	**101**	**102**	**103**	**104**	**105**	**106**	**107**	**108**	**109**
6	0.0057	0.0057	0.0057	0.0057	0.0057	0.0057	0.0057	0.0057	0.0170	0.0426
7	0.0056	0.0056	0.0056	0.0056	0.0056	0.0056	0.0056	0.0056	0.0169	0.0423
8	0.0055	0.0056	0.0056	0.0056	0.0056	0.0056	0.0056	0.0056	0.0167	0.0419
9	0.0055	0.0055	0.0055	0.0055	0.0055	0.0055	0.0055	0.0055	0.0166	0.0416

y \ x	**110**	**111**	**112**	**113**	**114**	**115**	**116**	**117**	**118**	**119**
6	0.0996	0.0427	0.0171	0.0028	0.0028	0.0028	0.0028	0.0028	0.0028	0.0028
7	0.0987	0.0423	0.0169	0.0028	0.0028	0.0028	0.0028	0.0028	0.0028	0.0028
8	0.0979	0.0420	0.0168	0.0028	0.0028	0.0028	0.0028	0.0028	0.0028	0.0028
9	0.0970	0.0416	0.0167	0.0028	0.0028	0.0028	0.0028	0.0028	0.0028	0.0028

- En el intervalo temporal $[15\,..\,20]$, todos los puntos de acuerdo tienen 0.1 como utilidad ponderada.

Las valoraciones de utilidad respecto a $\mathcal{U}^{\omega_2}$ son:

- En el intervalo temporal $[9\,..\,14]$, la valoración de utilidad es 1 si $z =$ 'p' o bien 0 si $z =$ 'q'.
- En el intervalo temporal $[15\,..\,20]$, la valoración de utilidad es 0 si $z =$ 'p' o bien 1 si $z =$ 'q'.

Utilidad del Acuerdo Una vez que hemos obtenido todos los números necesarios, pasamos a calcular la utilidad del acuerdo entre ω_2 y δ_1:

- En el intervalo temporal $[9\,..\,14]$, el punto de acuerdo controlable que maximiza $\mathcal{U}^{\delta_1/\omega_2} \times \mathcal{U}^{\omega_2/\delta_1}$ es $y = 5$ y $z =$ 'p' con 0.3952 de valoración de utilidad.

- En el intervalo temporal $[15\,..\,20]$, los puntos de acuerdo controlables que maximizan $U^{\delta_1/\omega_2} \times U^{\omega_2/\delta_1}$ son aquellos que satisfacen $15 \leq y < 20$ y $z =$ 'q' con 0.1 de valoración de utilidad.

El promedio de utilidades durante el período de vigencia de δ_1 resulta $\frac{6\times 0.3952+6\times 0.1}{12} = 0.2476$ como valoración de utilidad global del acuerdo.

6.6. Resumen y Reseña Bibliográfica

En este capítulo hemos presentado el marco de trabajo HDM, nuestro tomador de decisiones holístico para el emparejamiento de servicios. Este marco de trabajo está basado en la separación de niveles de abstracción, de manera que nos ha permitido resolver los problemas que hemos comentado en el capítulo §5.

En este capítulo también hemos definido de manera rigurosa nuestro modelo de emparejamiento, que constituye el resultado de varios años de investigación que comenzaron con nuestros trabajos en [126, 130, 168]. En ellos, presentamos un modelo de emparejamiento preliminar, aunque ya incorporábamos características avanzadas tales como la expresividad para describir objetivos, la simetría de los objetivos y los criterios de preferencia. En estos trabajos, el modelo abstracto estaba basado en programación con restricciones, de manera que el modelo abstracto estaba muy influenciado por el modelo operacional. La consciencia temporal se incluyó como una extensión del modelo abstracto [131, 142] que después ha seguido una línea de investigación paralela para dotar de consciencia temporal la recomendación WS-Agreement [143, 144].

Durante el último año, el esfuerzo se ha centrado en obtener un modelo abstracto independiente del modelo operacional. Este modelo se ha definido de manera rigurosa mediante teoría de conjuntos, habiéndose conseguido que nuestro modelo de emparejamiento alcance todas las características avanzadas que presentamos en esta tesis doctoral.

Capítulo 7

Modelo Operacional de HDM

I'm going to make him an offer he can't refuse...

Marlon Brando, Vito Corleone
"The Godfather", 1972

En este capítulo presentamos el modelo operacional de HDM. Proporcionamos las transformaciones necesarias desde el modelo abstracto para interpretar las tareas de emparejamiento como problemas de satisfacción de restricciones. Asumimos la existencia de un lenguaje de restricciones genérico. Hay que añadir que este modelo operacional está apoyado por un algoritmo de proyección temporal para realizar un tratamiento específico de la consciencia temporal.

Este capítulo se estructura como sigue. En la sección §7.1 hacemos una introducción al modelo operacional y la necesidad del algoritmo de proyección temporal. A continuación, en la sección §7.2 se muestran los esquemas generales de transformación. En la sección §7.3 se muestra el tratamiento de la consciencia temporal y en la sección §7.4 la interpretación de las tareas de emparejamiento como problemas de satisfacción de restricciones. Finalmente, en la sección §7.5 se encuentran nuestran conclusiones.

7.1. Introducción

En este capítulo exponemos el modelo operacional de HDM. Este modelo tiene dos partes claramente diferenciadas, a saber, (**1**) el tratamiento específico de la consciencia temporal, y (**2**) la interpretación de las tareas de emparejamiento como problemas de satisfacción de restricciones (CSP), en el caso de la consistencia y conformidad; o bien como problemas de optimación (CSOP), en el caso de la optimalidad.

La programación con restricciones tiene unas características muy interesantes, que permiten afrontar nuestro modelo operacional:

- *Declarativid*ad. Sólo tenemos que definir el problema a resolver, sin necesidad de indicar cómo resolverlo, tarea de la que se encarga el resolutor de restricciones.

- *Expresividad*. Los resolutores de restricciones son capaces de procesar teóricamente cualquier expresión matemática como una función booleana.

Los CSP y CSOP correspondientes a las tareas del emparejamiento se obtienen transformando las ofertas de acuerdo, de manera que sus diferentes partes, tales como atributos, dominios, predicados, criterios de preferencia y probabilidades de ocurrencia, se transforman, respectivamente, en variables, dominios, restricciones y funciones-objetivos, según sea el caso. Para ello, asumimos la existencia de un lenguaje de restricciones genérico y un resolutor capaz de tratarlas.

Respecto al aspecto temporal, hemos definido un *algoritmo de proyección temporal* para apoyar a los resolutores de restricciones en las tareas de emparejamiento con consciencia temporal.

En el algoritmo de proyección temporal, consideramos la tupla que representa de manera abstracta a las ofertas de acuerdo (véase la figura §6.3). En esta representación, una oferta de acuerdo tiene un conjunto de atributos y dominios, donde el primero de ellos representa al tiempo y su dominio corresponde al período de vigencia de la oferta de acuerdo. A continuación los predicados, donde consideramos aquellos requeridos y garantizados de la oferta de acuerdo. Estos predicados siguen una estructura que reflejan el contenido de los términos, de manera que toman la forma de implicaciones lógicas donde los antecedentes son las condiciones de los períodos locales de vigencia y las condiciones respecto a los atributos controlables, y los consecuentes son las

condiciones de los objetivos que son vigentes en cada caso. El resto de partes no las consideramos hasta que tratemos específicamente la optimalidad.

En las ofertas de acuerdo puede haber solapamientos entre los períodos locales de vigencia, de manera que hablamos de una oferta de acuerdo ambigua porque puede haber diferentes objetivos válidos durante un mismo intervalo temporal. Esto es un problema para comprobar la consistencia, donde necesitamos conocer la región de acuerdos en cada instante temporal del período de vigencia. Basado en el algoritmo de proyección temporal, el aplanamiento trata de superar dicha ambigüedad, de manera que aplicándose sobre una oferta de acuerdo, obtenemos una secuencia de sub-ofertas. La figura §7.1 ilustra una traza sencilla de aplanamiento. Nótese que los predicados de cada sub-oferta sólo contienen los que son válidos durante su período global de vigencia, que se obtienen de manera que éstos no tengan solapamientos. Esta secuencia está ordenada temporalmente por los períodos globales de vigencia. Las sub-ofertas resultantes son fácilmente transformadas a una familia de CSP, donde hemos eliminado el aspecto temporal, de manera que la consistencia de la oferta de acuerdo se puede obtener a partir de los resultados de consistencia de cada sub-oferta.

Respecto a comprobar la conformidad, también necesitamos obtener los predicados requeridos y garantizados válidos en cada instante temporal del período global de vigencia de la demanda. Basado en el algoritmo de proyección temporal, el recorrido paralelo trata de obtener una secuencia de nuevos elementos, ordenada temporalmente, de manera que cada elemento contiene los predicados requeridos y garantizados de la demanda y la oferta que son válidos en un intervalo temporal. La secuencia resultante no es ambigua. La figura §7.2 ilustra una traza sencilla de recorrido paralelo. Igual que en el caso anterior, los elementos de esta secuencia son fácilmente transformables a una familia de CSP para comprobar la conformidad.

7.2. Esquema General de Transformaciones

Mostramos los esquemas generales de transformaciones que necesitamos para llevar a cabo las tareas de emparejamiento de servicios. Las figuras §7.3 y §7.4 muestran sendos esquemas para comprobar la consistencia y conformidad, respectivamente. La figura §7.5 muestra el esquema para obtener la utilidad de un acuerdo.

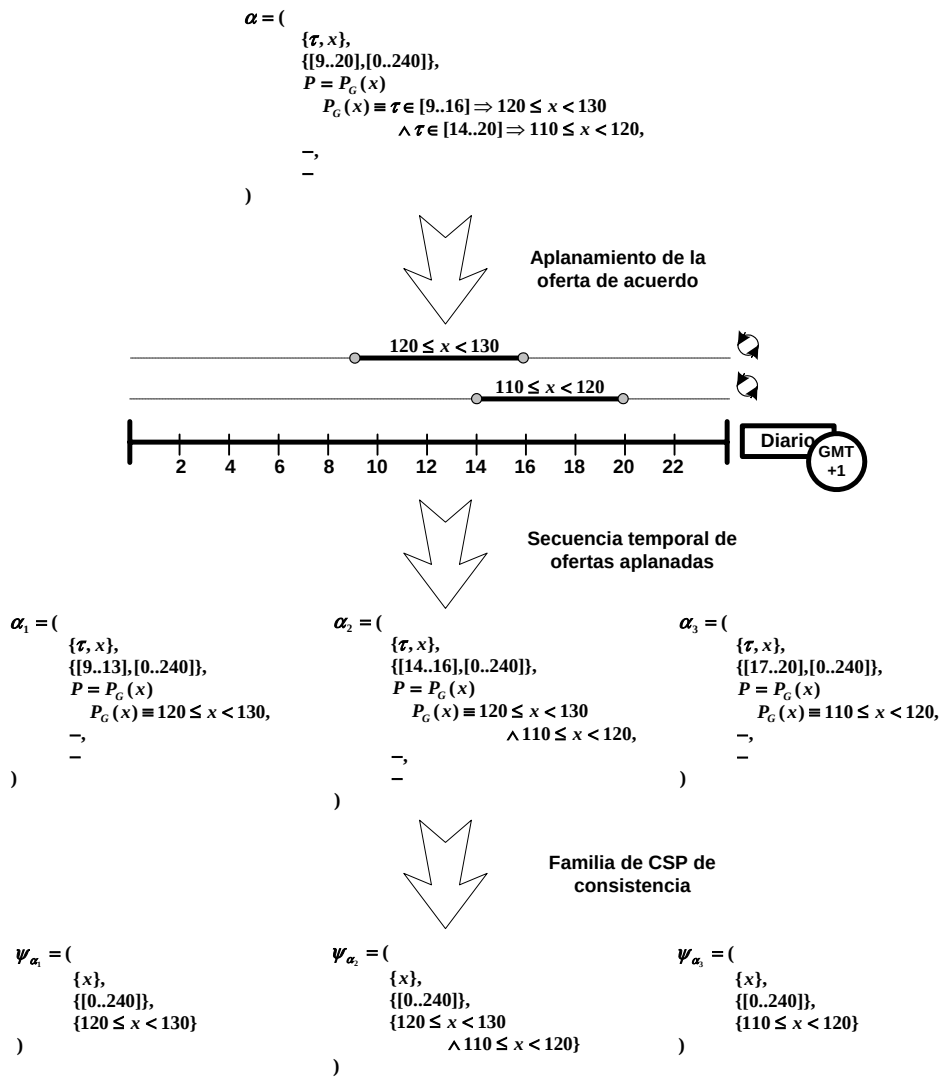


Figura 7.1: *Ejemplo de transformaciones para comprobar la consistencia.*

7.2.1. Consistencia

En la sección §6.3, hemos determinado que para comprobar la consistencia de una oferta de acuerdo, hemos de constatar su consistencia en cada instante temporal de su período de vigencia. Para ello, hemos de conocer qué predicados son válidos en cada momento. El aplanamiento nos permite conocer los *intervalos temporales de interés* de la oferta de acuerdo, que son los máximos intervalos temporales sin solapamientos durante los cuáles son válidos los mismos predicados.

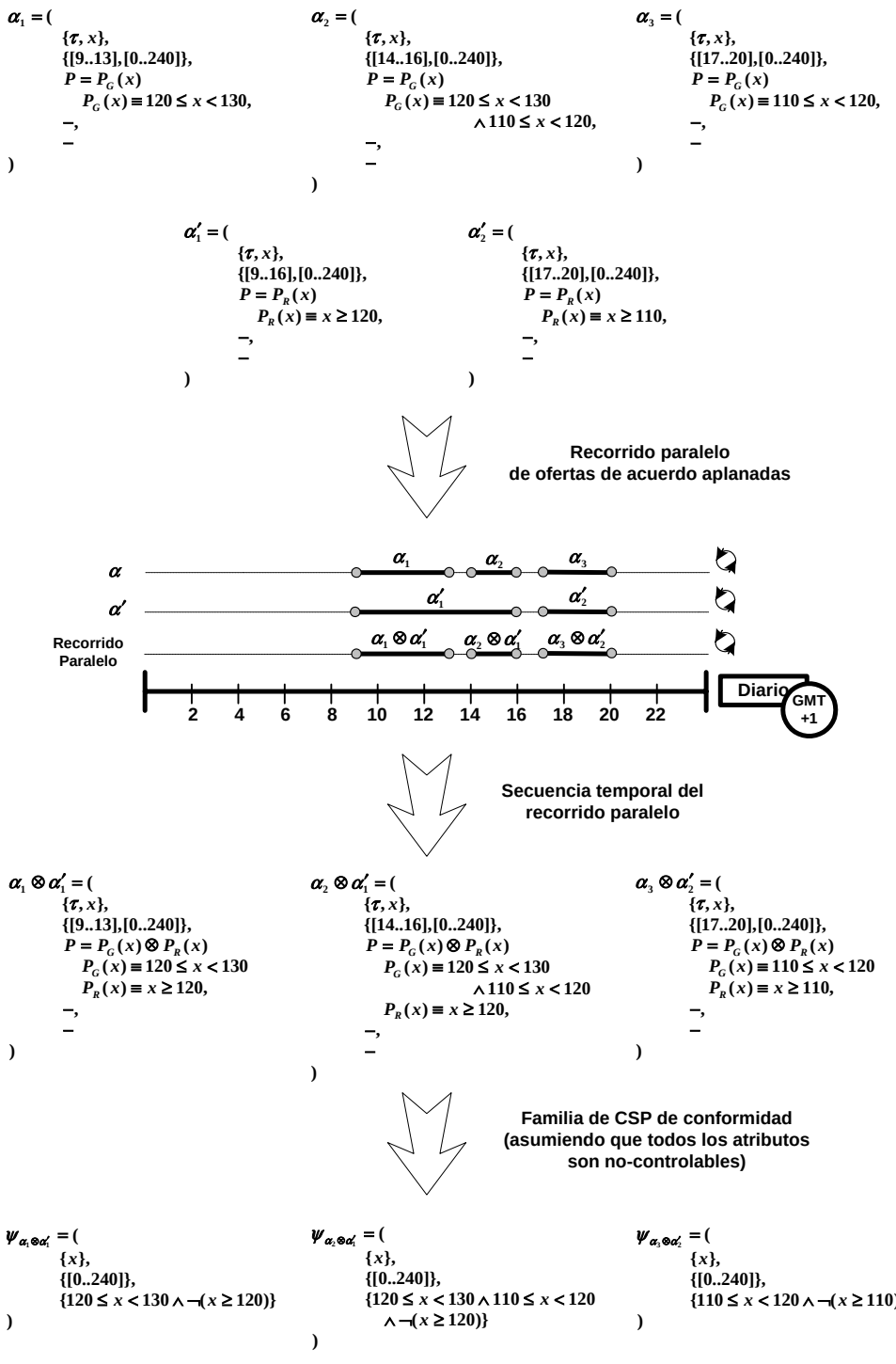


Figura 7.2: *Ejemplo de transformaciones para comprobar la conformidad.*

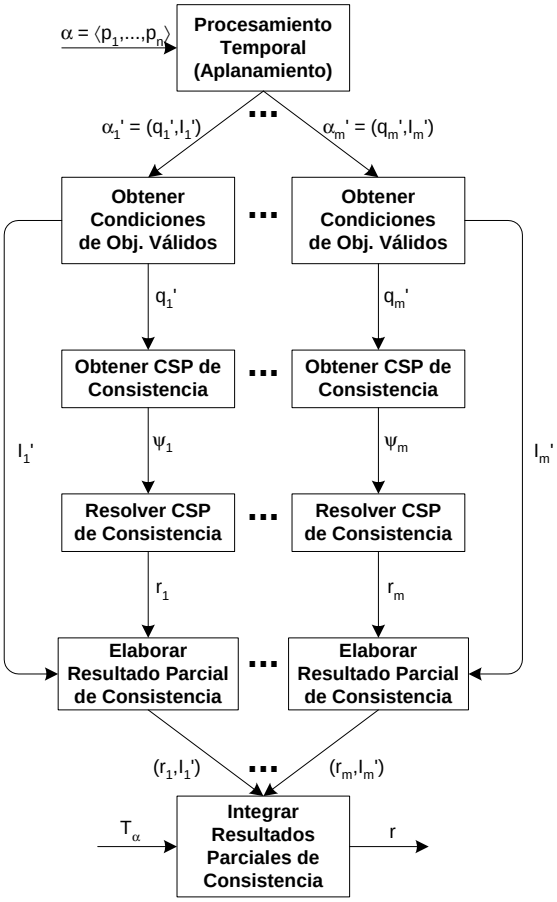


Figura 7.3: *Transformaciones para comprobar la consistencia.*

La figura §7.3 muestra el esquema general de transformaciones para comprobar la consistencia. Se comienza **(1)** llamando al algoritmo de aplanamiento de la oferta de acuerdo; a continuación, **(2)** para cada sub-oferta de la oferta aplanada, se extraen los predicados que se transforman al correspondiente *CSP de consistencia* y se comprueba su satisfactibilidad; finalmente **(3)** se integran todos los resultados parciales para comprobar que configuran una cobertura efectiva del período global de vigencia de la oferta de acuerdo, es decir, en todo instante temporal de dicho período de vigencia se ha comprobado la consistencia de los términos con validez en dicho momento.

Respecto al procesamiento temporal y la obtención del CSP de consistencia, las transformaciones básicas se explican en las secciones §7.3.1 y §7.4.1, respectivamente.

7.2.2. Conformidad

En la sección §6.4, hemos determinado que para comprobar la conformidad de una oferta a una demanda, hay que constatar la conformidad de dicha oferta en cada instante temporal del período de vigencia de dicha demanda. Para ello, hemos de conocer qué predicados son válidos en cada momento por ambas partes. El algoritmo de recorrido paralelo nos permite conocer los intervalos temporales de interés, que son los máximos intervalos temporales sin solapamientos derivados de la intersección temporal de ambas ofertas de acuerdo, ambas aplanadas, durante los cuales son válidos los mismos predicados en cada una de las partes.

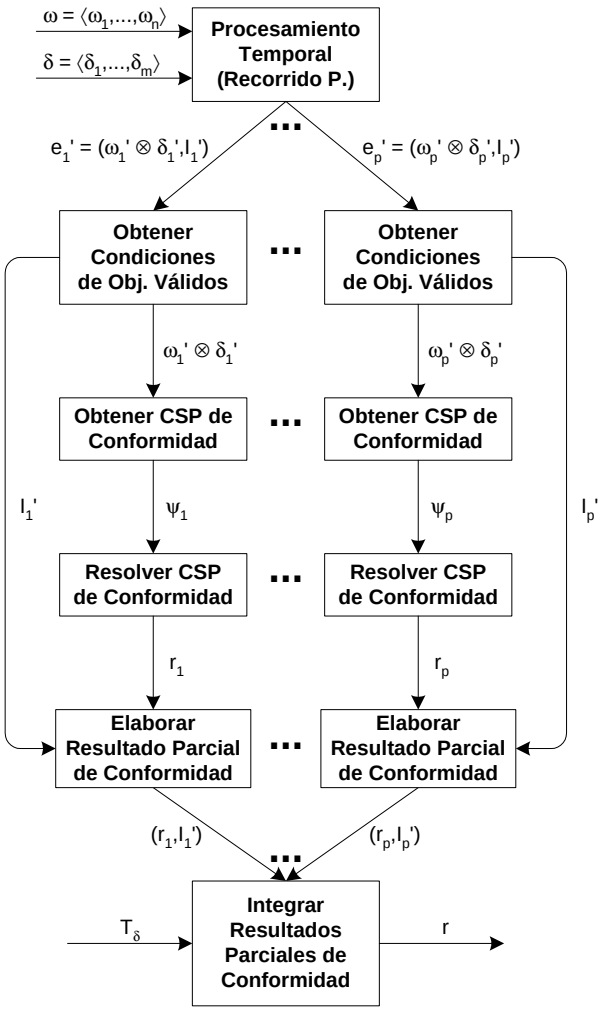


Figura 7.4: *Transformaciones para comprobar la conformidad.*

La figura §7.4 muestra el esquema general de transformaciones para comprobar la conformidad, similar al caso de la consistencia. Se comienza **(1)** llamando al algoritmo de proyección temporal para obtener la secuencia de recorrido paralelo; a continuación, **(2)** para cada elemento de dicha secuencia se extraen los predicados, requeridos y garantizados, procedentes de la oferta y la demanda, cuyas condiciones se transforman para obtener el correspondiente *CSP de conformidad* y se comprueba su satisfactibilidad; finalmente **(3)** se integran todos los resultados parciales para comprobar que configuran la cobertura efectiva del período de vigencia de la demanda, es decir, en todo instante temporal de dicho período de vigencia se ha comprobado la conformidad de los términos de sendas ofertas de acuerdo con validez en dicho momento.

Respecto al procesamiento temporal y la obtención del CSP de conformidad, las transformaciones básicas se explican en las secciones §7.3.2 y §7.4.2, respectivamente.

7.2.3. Optimalidad

En la sección §6.5 del modelo abstracto, hemos definido la optimalidad. Para obtener la utilidad del acuerdo entre una oferta y una demanda, en primer lugar hay que obtener los puntos de acuerdo óptimos en cada instante temporal del período de vigencia de la demanda. Para ello, utilizamos los intervalos temporales de interés que resultan del recorrido paralelo de ambas ofertas de acuerdo, que ya conocemos puesto que se ha debido comprobar previamente la conformidad.

La figura §7.5 muestra el esquema general de transformaciones para obtener la utilidad del acuerdo entre una oferta y una demanda. Se comienza: **(1)** obteniendo los elementos de la secuencia de recorrido paralelo; a continuación, **(2)** para cada elemento, se extraen los predicados garantizados, procedentes de la oferta y la demanda, que se transforman para obtener los correspondientes *CSOP de optimalidad*, que también incluyen las funciones de utilidad y distribución de probabilidades disponibles, y se obtienen la mejor solución junto a su valoración o utilidad, que se consideran punto de acuerdo óptimo en dicho intervalo temporal de interés; a continuación, **(3)** se reúnen los puntos de acuerdo obtenidos para constituir la secuencia temporal de puntos de acuerdo óptimos; finalmente, **(4)** se obtiene la utilidad del acuerdo mediante agregación de las valoraciones de utilidad de los puntos de acuerdo, que puede ponderarse por la duración de cada intervalo temporal de interés, o incluso por la utilidad de propio intervalo temporal si estuviese dicha función disponible.

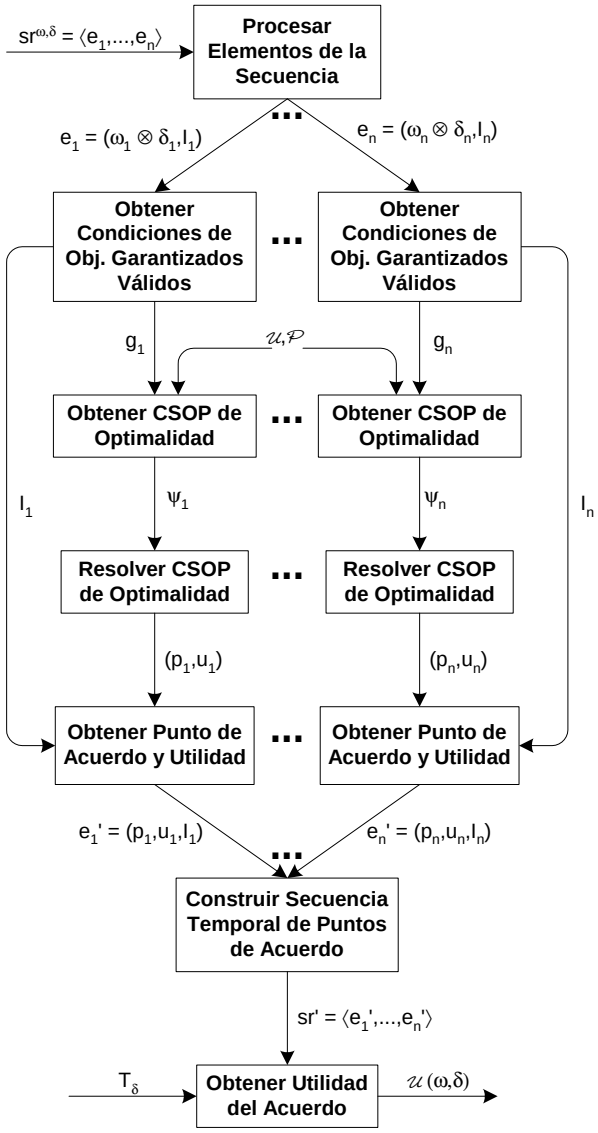


Figura 7.5: *Transformaciones para obtener la utilidad de un acuerdo.*

Respecto a la obtención del CSP de optimalidad para obtener la utilidad del punto de acuerdo óptimo en un intervalo temporal de interés concreto, las transformaciones básicas se explican en la sección §7.4.3.

7.3. Esquemas de Transformación Temporales

En esta sección, describimos las dos principales tareas de los esquemas de transformación temporales, a saber: el aplanamiento de una oferta de acuerdo y el recorrido paralelo. Ambas tareas utilizan el algoritmo de proyección temporal, que se define de manera rigurosa en el apéndice §B.

7.3.1. Aplanamiento de una Oferta de Calidad

El algoritmo de proyección temporal permite transformar una oferta de acuerdo en una *oferta de acuerdo aplanada*. Recuérdese que partimos de una oferta de acuerdo abstracta, de manera que vamos a considerar que su predicado contiene un conjunto de condiciones. Aquí consideramos que cada condición tiene un período local de vigencia y la condición propiamente dicha que corresponde al objetivo válido en dicho intervalo temporal. Así pues, la oferta de acuerdo aplanada consiste en una secuencia de sub-ofertas, de manera que cada sub-oferta incluye todas las condiciones del predicado válidas durante un mismo intervalo temporal. En cada sub-oferta dicha condición pasa a ser el predicado propiamente dicho y el intervalo temporal pasa a considerarse su período global de vigencia. La secuencia resultante debe satisfacer las propiedades de ordenación temporal y no-ambigüedad. En el algoritmo §7.1 se muestran los pasos del aplanamiento, que explicamos a continuación:

- En primer lugar, las condiciones del predicado de la oferta de acuerdo se ordenan temporalmente según su período local de vigencia para obtener una secuencia de condiciones. De esta secuencia, se extraen en cada paso las dos primeras.
- Para cada pareja de condiciones extraida, el algoritmo de proyección temporal determina la relación entre los intervalos temporales de sus períodos locales de vigencia. Dependiendo de dicha relación, se obtiene una secuencia de nuevas condiciones aplanada, que está ordenada y sin solapamientos, según explicamos en el apéndice §B.
- A continuación, en orden inverso para conservar el orden temporal, se procesa cada nuevo predicado obtenido por el algoritmo:
    - Si existe alguna otra condición en la secuencia original que también tenga validez durante su período de vigencia, entonces la nueva condición se queda pendiente y se añade a la cabecera de la secuencia original.

```
declaraciones
  condición: tupla[plv: it, cond: q]
    // plv período local de vigencia
    // it intervalo temporal
    // cond condición de objetivo
    // q condición expresada sobre atributos de calidad
  oferta: tupla[pgv: it, pred:conjunto{condición}]
    // pgv período global de vigencia
    // pred predicado
alg aplanamiento(of: oferta)
  res sof: secuencia(oferta) ⟵ [] // secuencia de sub-ofertas
comienzo
   sc, scp, scr: secuencia(condición)
   sc ⟵ obtener-y-ordenar-condiciones-temporalmente(of.pred)
  mientras sc <> [] hacer
    si sc.tamaño > 1  entonces
       scp ⟵ sc.extraer-condición-cabecera(2)
       scr ⟵ proyección-temporal(scp)
      para-cada c en scr en-orden-inverso
        si no hay-disyunción-temporal(c.plv,sc) entonces
          sc.añadir-en-cabecera(c)
          scr.eliminar(c)
        fin-si
      fin-para-cada
      para-cada c en scr
          subof: oferta ⟵
            crear oferta[pgv: c.plv, pred: conjunto{c}]
          sof.añadir-al-final(subof)
      fin-para-cada
    en-otro-caso
      si sc.tamaño = 1 entonces
        fcond: condición ⟵ sc.extraer-condición-cabecera(1)
        fsubof: oferta ⟵
            crear oferta[pgv: fcond.plv, pred: conjunto{fcond}]
        sof.añadir-al-final(fusbof)
      fin-si
    fin-si
  fin-mientras
fin
```

Algoritmo 7.1: *Algoritmo de aplanamiento de una oferta de acuerdo.*

  - En cambio, si no lo hay entonces se construye una nueva sub-oferta, de manera que el período local de vigencia de la condición pasa a ser período global de vigencia y ésta pasa a considerarse como predicado de la misma. Esta sub-oferta se añade al final de la secuencia de sub-ofertas que se constituye como resultado del aplanamiento.

- Así se procede hasta que no queden más parejas de condiciones en la secuencia original, en cuyo momento se añade al final una última sub-oferta en caso de que quedara pendiente una única condición.

La figura §7.6 muestra una traza ilustrativa:

1. Las dos primeras condiciones q_1 y q_2 son válidas durante [1..9] y [8..15], respectivamente. La proyección obtiene tres términos, de manera que se añade al resultado final una sub-oferta con período global de vigencia [1..7] y cuyo predicado contiene la condición q_1. Quedan pendientes las condiciones $q_1 \wedge q_2$ y q_2 durante [8..9] y [10..15], respectivamente.

2. A continuación, las siguientes condiciones $q_1 \wedge q_2$ y q_3 son válidas durante [8..9] y [8..21], respectivamente. La proyección obtiene dos condiciones, de manera que se añade al resultado final una sub-oferta con período global de vigencia [8..9] y cuyo predicado contiene la condición $q_1 \wedge q_2 \wedge q_3$. Queda pendiente la condición q_3 válida durante [10..21].

3. A continuación, las siguientes condiciones q_2 y q_3 son válidas durante [10..15] y [10..21], respectivamente. La proyección obtiene dos condiciones $q_2 \wedge q_3$ y q_3 válidas durante [10..15] y [16..21], respectivamente. Ambas condiciones quedan pendientes.

4. A continuación, las siguientes condiciones $q_2 \wedge q_3$ y q_4 válidas durante [10..15] y [10..17], respectivamente. La proyección obtiene dos condiciones, de manera que se añade al resultado final una sub-oferta con período global de vigencia [10..15] y cuyo predicado contiene la condición $q_2 \wedge q_3 \wedge q_4$. Queda pendiente la condición q_4 válida durante [16..17].

5. A continuación, las siguientes condiciones q_3 y q_4 son válidas durante [16..21] y [16..17], respectivamente. La proyección obtiene dos condiciones $q_3 \wedge q_4$ y q_3 válidas durante [16..17] y [18..21], respectivamente. Ambas condiciones quedan pendientes.

6. A continuación, las siguientes condiciones $q_3 \wedge q_4$ y q_5 son válidas durante [16..17] y [17..21], respectivamente. La proyección obtiene tres condiciones, de manera que se añaden al resultado final sendas sub-ofertas con períodos globales de vigencia [16..16] y [17..17] cuyos predicados

contienen las condiciones $q_3 \wedge q_4$ y $q_3 \wedge q_4 \wedge q_5$, respectivamente. Queda pendiente la condición q_5 válida durante [18 . . 21].

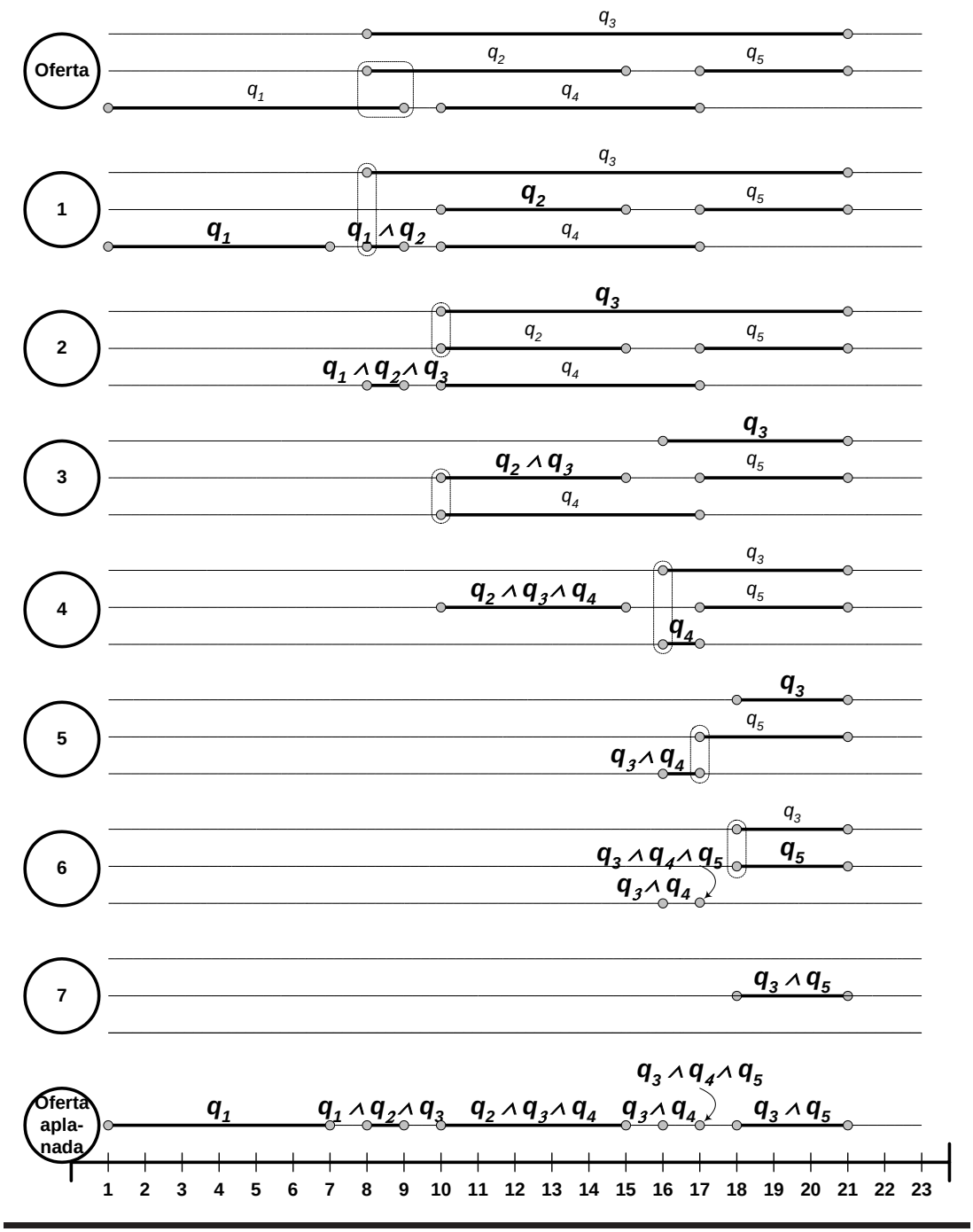


Figura 7.6: *Traza del aplanamiento de una oferta de acuerdo.*

7. Finalmente, las siguientes y últimas condiciones q_3 y q_5 son válidas durante $[18\,..\,21]$. La proyección temporal obtiene una única condición, de manera que se añade al resultado final una sub-oferta con período global de vigencia $[18\,..\,21]$ cuyo predicado contiene la condición $q_3 \wedge q_5$. Aquí termina el aplanamiento de esta oferta de acuerdo.

7.3.2. Recorrido Paralelo de Ofertas de Calidad

El algoritmo de proyección temporal, definido en el apéndice §B, también permite obtener la *secuencia de recorrido paralelo* de dos ofertas de acuerdo, ambas aplanadas. Los elementos de esta secuencia contienen los predicados de la sub-oferta y sub-demanda que tienen validez contemporáneamente durante un intervalo temporal de interés. Esta secuencia satisface las propiedades de ordenación temporal y no-ambigüedad.

La figura §7.7 muestra una traza ilustrativa. Sea una oferta (**Of**) y una demanda (**De**), ambas aplanadas. Se obtiene la secuencia de recorrido paralelo (**SR**) de la siguiente manera:

1. En primer lugar, los elementos de cabecera de ambas secuencias aplanadas tienen predicados cuyas condiciones son q_1 y q'_1, respectivamente, con validez durante el mismo intervalo temporal $[0\,..\,7]$. Por ser iguales, se añade a la secuencia de recorrido un nuevo elemento que incluye al predicado $q_1 \otimes q'_1$ con validez durante el intervalo temporal $[0\,..\,7]$[†1].

2. A continuación, los elementos de cabecera de ambas secuencias aplanadas tienen predicados cuyas condiciones son q_2 y q'_2, con validez durante los intervalos temporales $[8\,..\,8]$ y $[8\,..\,15]$, respectivamente. Dado que coinciden los instantes temporales en que ambos comienzan, entonces se añade a la secuencia de recorrido un nuevo elemento que incluye al predicado $q_2 \otimes q'_2$ con validez durante el intervalo temporal $[8\,..\,8]$, mientras que vuelve a colocarse en la cabecera de la secuencia aplanada correspondiente a la demanda un predicado pendiente cuya condición es q'_2 con validez durante el intervalo temporal $[9\,..\,15]$.

3. A continuación, los elementos de cabecera de ambas secuencias aplanadas tienen predicados cuyas condiciones son q_3 y q'_2, con validez durante los intervalos temporales $[9\,..\,14]$ y $[9\,..\,15]$, respectivamente. Dado que

[†1] La operación $q_1 \otimes q_2$ hace referencia a cualquier operador $\otimes$ que deseemos aplicar sobre los predicados cuyas condiciones son q_1 y q_2. En nuestro caso, dicho operador puede equivaler al predicado de conformidad o la expresión de optimalidad que hemos definido en el capítulo §6.

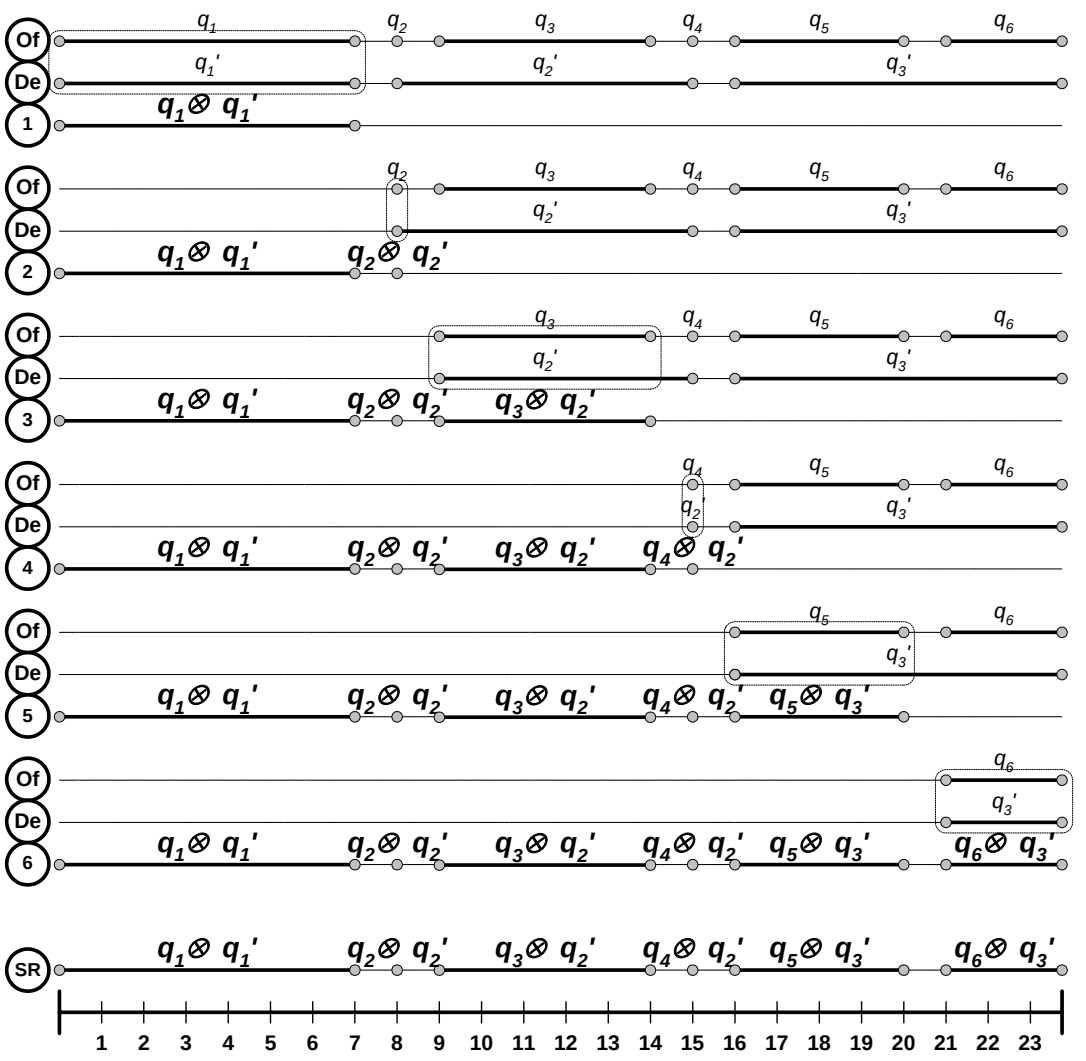


Figura 7.7: *Traza del recorrido paralelo de dos ofertas de acuerdo.*

nuevamente coinciden los instantes temporales en que ambos comienzan, entonces se añade a la secuencia de recorrido un nuevo elemento que incluye al predicado $q_3 \otimes q_2'$ con validez durante el intervalo temporal $[9..14]$, mientras que vuelve a colocarse en la cabecera de la secuencia aplanada correspondiente a la demanda un predicado cuya condición es q_2' con validez durante el intervalo temporal $[15..15]$.

4. A continuación, los elementos de cabecera de ambas secuencias aplanadas tienen predicados cuyas condiciones son q_4 y q_2', respectivamente, con validez durante el mismo intervalo temporal $[15..15]$. Por ser igua-

les, se añade a la secuencia de recorrido un nuevo elemento que incluye al predicado $q_4 \otimes q'_2$ con validez durante el intervalo temporal [15 . . 15].

5. A continuación, los elementos de cabecera de ambas secuencias aplanadas tienen predicados cuyas condiciones son q_5 y q'_3, con validez durante los intervalos temporales [16 . . 20] y [16 . . 24], respectivamente. Nuevamente coinciden los instantes temporales de comienzo, entonces se añade a la secuencia de recorrido un nuevo elemento que incluye el predicado $q_5 \otimes q'_3$ con validez durante el intervalo temporal [16 . . 20], mientras que vuelve a colocarse en la cabecera de la secuencia aplanada correspondiente a la demanda un predicado cuya condición es q'_3 con validez durante el intervalo temporal [21 . . 24].

6. Finalmente, quedan sendos elementos en ambas secuencias aplanadas que tienen predicados cuyas condiciones son q_6 y $q_5 \otimes q'_3$, respectivamente, con validez durante el mismo intervalo temporal [21 . . 24]. Por ser iguales, se añade a la secuencia de recorrido un último elemento que incluye al predicado $q_6 \otimes q'_3$ con validez durante el intervalo temporal [21 . . 24]. Aquí termina la traza del recorrido paralelo.

7.4. Esquemas de Tranformación a CSP/CSOP

En esta sección presentamos los esquemas básicos de transformación para obtener los CSP de consistencia y conformidad, y el CSOP de optimalidad. Las figuras §7.8 y §7.9 ilustran las transformaciones desde el modelo abstracto de una oferta de acuerdo a CSP y las transformaciones para interpretar las tareas de emparejamiento como CSP, respectivamente.

7.4.1. Consistencia

Dada una oferta de acuerdo de la que ya hemos despejado su componente temporal, el correspondiente CSP de consistencia está formado por (**1**) la declaración de variables y dominios correspondientes a los atributos de calidad, y (**2**) las restricciones correspondientes a las condiciones que contiene el predicado que determina la región de acuerdos.

En la figura §7.8 se ilustran diferentes transformaciones con diferentes ofertas de acuerdo. Si α representa una oferta de acuerdo, ψ_α denota el CSP de consistencia correspondiente. En cada una de las transformaciones la condición incluye algún tipo de operación diferente.

$\alpha_1 = ($ $\{x\},$ $\{[0..240]\},$ $P = P(x)$ $P(x) \equiv 120 \leq x < 130,$ $-,$ $-$ $)$	$\psi_{\alpha_1} = ($ $\{x\},$ $\{[0..240]\},$ $\{120 \leq x < 130\}$ $)$
$\alpha_2 = ($ $\{x, y\},$ $\{[0..240],[0..60]\},$ $P = P(x)$ $P(x) \equiv 120 \leq x < 130 \wedge 5 \leq y < 10,$ $-,$ $-$ $)$	$\psi_{\alpha_2} = ($ $\{x, y\},$ $\{[0..240],[0..60]\},$ $\{120 \leq x < 130, 5 \leq y < 10\}$ $)$
$\alpha_3 = ($ $\{x, y\},$ $\{[0..240],[0..60]\},$ $P = P(x)$ $P(x) \equiv 120 \leq x < 130 \wedge 5 \leq y < 10$ $\vee (110 \leq x < 130),$ $-,$ $-$ $)$	$\psi_{\alpha_3} = ($ $\{x, y\},$ $\{[0..240],[0..60]\},$ $\{(120 \leq x < 130 \wedge 5 \leq y < 10)$ $\vee (110 \leq x < 130)\}$ $)$
$\alpha_4 = ($ $\{x, y\},$ $\{[0..240],[0..60]\},$ $P = P(x)$ $P(x) \equiv 120 \leq x < 130 \wedge \neg(5 \leq y < 10),$ $-,$ $-$ $)$	$\psi_{\alpha_4} = ($ $\{x, y\},$ $\{[0..240],[0..60]\},$ $\{120 \leq x < 130, \neg(5 \leq y < 10)\}$ $)$
$\alpha_5 = ($ $\{x, z\},$ $\{[0..240],[' p', ' q']\},$ $P = P(x)$ $P(x) \equiv z = ' p' \Rightarrow 120 \leq x < 130$ $\wedge z = ' q' \Rightarrow 100 \leq x < 120,$ $-,$ $-$ $)$	$\psi_{\alpha_5} = ($ $\{x, z\},$ $\{[0..240],[' p', ' q']\},$ $\{z = ' p' \Rightarrow 120 \leq x < 130,$ $z = ' q' \Rightarrow 100 \leq x < 120\}$ $)$
$\alpha_6 = ($ $\{x, y, y'\},$ $\{[0..240],[0..60],[0..60]\},$ $P = P(x)$ $P(x) \equiv x \geq 120 \wedge y \geq 10$ $\wedge \forall y' \bullet 100x/(x + y') \geq 90,$ $-,$ $-$ $)$	$\psi_{\alpha_6} = ($ $\{x, y, y'\},$ $\{[0..240],[0..60],[0..60]\},$ $\{x \geq 120, y \geq 10,$ $\forall y' \bullet 100x/(x + y') \geq 90\}$ $)$

Figura 7.8: *Transformaciones desde modelo abstracto a CSP.*

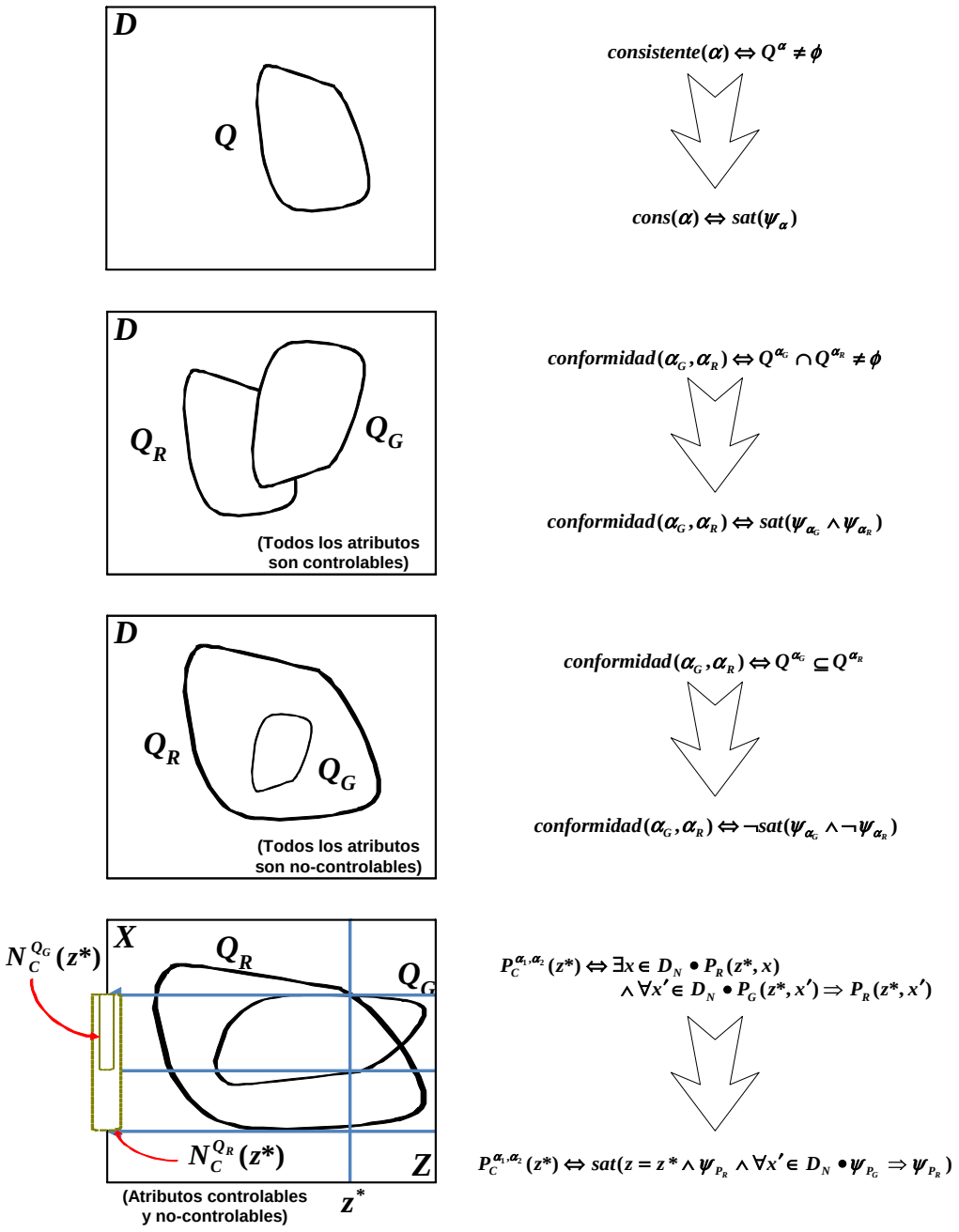


Figura 7.9: *Interpretación de tareas de emparejamiento como CSP.*

Nótese que hemos asumido un lenguaje de restricciones genérico que abarca las operaciones aritméticas, lógicas y relacionales habituales, más la cuantificación existencial y universal.

7.4.2. Conformidad

En la figura §7.9 se muestra cómo podemos interpretar la conformidad como un problema de satisfacción de restricciones. Abordamos los tres casos básicos de conformidad, a saber: la noción de conformidad cuando todos los atributos son controlables, o bien cuando todos los atributos son no-controlables, o bien cuando aparecen de ambos tipos.

Si consideramos que una oferta de acuerdo sólo contiene los predicados garantizados α_G y la otra sólo contiene los predicados requeridos α_R, cuyos CSP de consistencia son ψ_{α_G} y ψ_{α_R}, respectivamente. Entonces las transformaciones a los CSP de conformidad son como siguen.

En el primer caso, la intersección de las regiones de acuerdo de α_G y de α_R se corresponde a la satisfactibilidad de la disyunción de sus correspondientes CSP. Recuérdese que en este caso sólo hace falta que exista un punto de acuerdo que satisfaga garantías y requisitos.

En el segundo caso, la inclusión de la región de acuerdos garantizada de α_G en la región de acuerdos requerida de α_R se corresponde a la no-satisfactibilidad de la disyunción del CSP correspondiente a α_G frente a la negación del CSP correspondiente a α_R. Recuérdese que en este caso estamos comprobando que no existe punto de acuerdo que satisface a las garantías que no satisfaga también a los requisitos.

Por último, tenemos el complejo caso de que haya atributos controlables y no-controlables. En este caso debemos comprobar si la región ligada a un punto de acuerdo controlable z^* que satisface se incluye en la región ligada a dicho punto de acuerdo que satisface los requisitos. El CSP de conformidad se obtiene por una transformación directa utilizando sus correspondientes CSP. Sin embargo, hay que recordar que el cuantificador universal constituye un problema abierto de investigación en la programación con restricciones. En el capítulo §8 presentamos una solución simple, pero viable para nuestros propósitos.

7.4.3. Utilidad de un Acuerdo

Dados los términos que tienen validez en un intervalo temporal de interés, pertenecientes a sendas demanda y oferta que son conformes entre sí, el CSOP de optimalidad sigue la estructura del problema de optimación según la expresión definida en la sección §6.5:

$$\begin{array}{ll} \textbf{máx} & \mathcal{U}^{\delta}(z, x_1)\mathcal{P}^{\omega}(z, x_1) \times \mathcal{U}^{\omega}(z, x_2)\mathcal{P}^{\delta}(z, x_2) \\ \text{sujeto a} & \psi_{P_C^{\omega,\delta}(z)} \wedge \psi_{P_G^{\omega}(z,x_1)} \wedge \psi_{P_G^{\delta}(z,x_2)} \end{array}$$

donde $\mathcal{U}^{\omega}$ y $\mathcal{U}^{\delta}$ son las funciones de utilidad, y $\mathcal{P}^{\omega}$ y $\mathcal{P}^{\delta}$ son las distribuciones de probabilidad de ocurrencia, de oferta y demanda, respectivamente.

Hay que indicar que la expresividad de las funciones-objetivo que admiten los resolutores suelen estar restringidas. En el capítulo §8 presentamos alternativas para disponer de mayor expresividad.

Casos Particulares Si sólo hay atributos no-controlables, sin probabilidades de ocurrencias, entonces el CSOP de optimalidad adopta una visión pesimista:

$$\begin{array}{ll} \textbf{máx} & (1 - \mathcal{U}^{\delta}(z, x_1)) \times (1 - \mathcal{U}^{\omega}(z, x_2)) \\ \text{sujeto a} & \psi_{P_C^{\omega,\delta}(z)} \wedge \psi_{P_G^{\omega}(z,x_1)} \wedge \psi_{P_G^{\delta}(z,x_2)} \end{array}$$

En cambio, si sólo hay atributos controlables, entonces el CSOP de optimalidad adopta una visión optimista:

$$\begin{array}{ll} \textbf{máx} & \mathcal{U}^{\delta}(z, x_1) \times \mathcal{U}^{\omega}(z, x_2) \\ \text{sujeto a} & \psi_{P_C^{\omega,\delta}(z)} \wedge \psi_{P_G^{\omega}(z,x_1)} \wedge \psi_{P_G^{\delta}(z,x_2)} \end{array}$$

7.5. Resumen y Reseña Bibliográfica

En este capítulo hemos presentado modelo operacional de HDM. Principalmente, destacamos la interpretación de las tareas de emparejamiento como problemas de satisfacción de restricciones. La programación con restricciones proporciona la declaratividad y expresividad apropiadas para afrontar nuestro modelo operacional. Además, destacamos la relativa facilidad en los esquemas de transformación para obtener los CSP / CSOP de consistencia, conformidad y optimalidad, debido al hecho de que hemos asumido un lenguaje de CSP genérico.

De hecho, los principales problemas surgen durante la implementación y dependen en gran medida de las características del resolutor de restricciones utilizado, por ejemplo, la ausencia de cuantificación universal que necesitamos en nuestro predicado de conformidad. En el capítulo §8 comentamos nuestra experiencia con el resolutor OPL-Studio de ILOG.

Por otro lado, también hemos presentado el tratamiento de la consciencia temporal en el modelo operacional. El algoritmo de proyección temporal, que se define de manera rigurosa en el apéndice §B, es la base para el aplanamiento y el recorrido paralelo, pasos fundamentales en los esquemas de transformación necesarios para realizar las tareas de emparejamiento con consciencia temporal. Estos algoritmos tienen una complejidad computacional de orden $O(N \log N)$ y $O(N)$, respectivamente, siendo N el número de términos de las ofertas de acuerdo. Añadir que hemos asumido que el tiempo viene dado por una variable discreta, lo que es un enfoque práctico porque los períodos de vigencia suelen venir dados por intervalos temporales cuya granularidad es de horas y minutos en una jornada diaria.

Respecto a la línea de trabajo que hemos seguido, reseñar que en [129, 166, 168] propusimos el uso de programación con restricciones para dar soporte operacional al emparejamiento, del que podemos decir que somos pioneros. Estos trabajos fueron extendidos por [131] para añadir consciencia temporal. Posteriormente, en [142] estudiamos la necesidad de un tratamiento específico a la consciencia temporal mediante el algoritmo de proyección temporal, para despejar el aspecto temporal de la resolución de los problemas de restricciones.

Capítulo 8

Modelo de Implementación de HDM

It's alive! It's alive!

Colin Clive, Dr. Henry Frankestein
"Frankestein", 1931

En este capítulo presentamos el modelo de implementación de HDM. En la sección §8.1 introducimos OPL Studio de ILOG, que es el resolutor que hemos incorporado a nuestro prototipo. A continuación, en la sección §8.2 mostramos los esquemas de transformación para describir CSP como modelos OPL, que son aplicados en la sección §8.3 para obtener los modelos OPL correspondientes a los CSP de conformidad y optimalidad, introducidos en el capítulo §7. Posteriormente, en la sección §8.4 mostramos los resultados del experimento que hemos realizado para conocer la viabilidad de nuestra propuesta. Finalmente, en la sección §8.5 hacemos un resumen del capítulo.

8.1. OPL–Studio de ILOG

OPL–Studio (versión 3.6) de ILOG [95] es un entorno de desarrollo integrado para la resolución de restricciones, que incluye un componente COM para que pueda ser utilizado desde las aplicaciones. Los CSP y CSOP se escriben en el lenguaje OPL (*OPtimization Language*), que fue diseñado por Hentenryck [76, 78, 198]. En este contexto, los CSP y CSOP se denominan modelos OPL. OPL-Studio también incluye OPLScript, un lenguaje de *scripts* que permite definir aplicaciones que integran varios modelos OPL relacionados entre sí. Este lenguaje ofrece métodos diversos para invocar la resolución de un modelo y acceder a sus variables internas[†1].

En OPL podemos escribir restricciones lineales y no-lineales, que incluyen operadores relacionales y aritméticos y pueden combinarse mediante operadores lógicos. Estas expresiones pueden incluir variables con los siguientes tipos:

- Variables simples de tipo entero, real, rango o enumerado.
- Variables estructuradas, por ejemplo, registros.
- Arrays de variables indexados por tipos discretos finitos.

En OPLScript se disponen los mismos tipos que ofrece OPL. Además, se incorporan los arrays abiertos cuya característica principal es que permiten un número de elementos indeterminado a priori, lo que le convierten en estructuras de gran utilidad para coordinar las soluciones entre diferentes modelos OPL, como mostramos posteriormente.

8.1.1. Características de OPL

A pesar de la expresividad del lenguaje OPL, destacamos algunas características que deben ser tenidas en cuenta:

- No hay cuantificación universal de variables.
- Las expresiones y funciones-objetivo con tipos reales deben ser lineales. Esta característica obliga a utilizar variables con dominios enteros cuando se necesitan expresiones no-lineales.

[†1]El acceso a las variables internas desde un módulo de programación a otro es una práctica de diseño muy poco recomendada ya que aumenta el acoplamiento entre los módulos, por lo que esta técnica de comunicación entre módulos OPL debe utilizarse con cierta cautela [162].

- No se pueden definir variables de tipo conjunto, aunque podemos simular algún tipo de conjunto restringido.

- Sólo podemos optimizar a partir de una sola función-objetivo.

- Se permiten funciones-objetivo definidas por trozos, pero deben ser lineales y monótona crecientes. Las funciones que son decrecientes también están disponibles si definimos su complementaria y minimizamos, ya que máx $F \Longleftrightarrow$ mín $1 - F$.

8.1.2. Satisfacción de Restricciones con OPL

El lenguaje OPL ofrece diferentes constructores para especificar cómo se construye el árbol de búsqueda (sección §4.3) y las estrategias de exploración de dicho árbol, lo que permite que el resolutor pueda adaptarse para aprovechar las características de un modelo OPL y mejorar así la eficiencia.

A continuación, repasamos solamente los constructores que hemos utilizado en nuestro prototipo:

- El lenguaje OPL permite definir diferentes procedimientos de etiquetado estándar. Destacan los constructores `forall` y `tryall` para especificar nodos-*and* (selección de las variables) y nodos-*or* (tentativas nodeterministas para asignar valores a una variable dentro de su dominio) del árbol de búsqueda a explorar, respectivamente.

- El lenguaje OPL también permite definir las heurísticas para explorar el árbol de búsqueda. Se ofrecen varias estrategias predefinidas, que pueden adaptarse mediante parametrización. Entre ellas, sólo destacamos:

  - La búsqueda primero-en-profundidad entrelazada [139] es una estrategia que simula una exploración basada en sendas búsquedas en profundidad paralelas. El objetivo es evitar los fallos prematuros.

  - La búsqueda primero-el-mejor es una estrategia aplicable a problemas de optimación, que asume la existencia de una función objetivo que, pasada como parámetro, actúa como función de evaluación y representa la heurística que guía la exploración.

8.2. Descripción de CSP en OPL

En nuestro prototipo, asumimos el lenguaje OPL como código objeto (o intermedio) para especificar CSP de manera declarativa, ya que se trata, en nuestra opinión, de la propuesta más relevante al respecto. En el capítulo §7 mostramos que todo procesamiento de las ofertas de acuerdo va encaminado a obtener mediante transformación los CSP necesarios para llevar a cabo las tareas de emparejamiento. En este apartado, describimos cómo podemos describir CSP como modelos y *scripts* OPL que, a su vez, pueden ser resueltos bien directamente por el propio resolutor de OPL, o bien por otros resolutores que estimemos oportunos, tras el correspondiente procesamiento del modelo OPL para transformarlo al esquema necesario.

Hay que añadir que la utilización de OPL permite desligarnos de los detalles de los lenguajes de programación anfitriones de otras propuestas, que incluyen librerías para especificar y resolver CSP, por ejemplo, ILOG [94], CHOCO [108] o JaCoP [107].

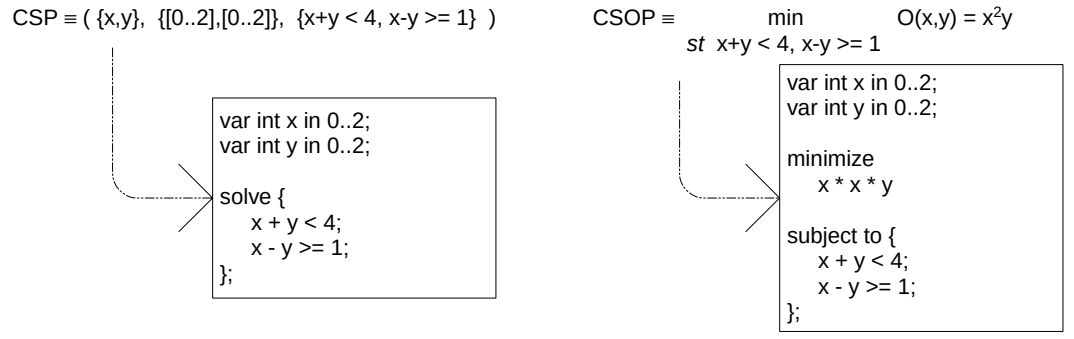
8.2.1. Esquemas Básicos

La figura §8.1 muestra los mapas para obtener el modelo OPL para especificar un CSP o un CSOP, respectivamente:

- En el caso de un CSP, el modelo OPL se compone de **(1)** una sección de declaraciones de variables y dominios, seguida de **(2)** una sección que incluye las restricciones a resolver.
- En el caso de un CSOP, el modelo OPL se compone de **(1)** una sección de declaraciones de variables y dominios, **(2)** una sección de optimación que incluye la función-objetivo a maximizar o minimizar, y **(3)** una sección que incluye las restricciones a las que está sujeta la función-objetivo.

8.2.2. Esquemas para Cuantificación Universal

En el capítulo §4 ya hemos introducido el problema de la cuantificación universal en la programación con restricciones. A continuación, mostramos los esquemas de transformación para resolver algorítmicamente un CSP con cuantificación universal mediante un *script* OPL que coordina la resolución de varios modelos OPL, que se ilustran en la figura §8.2.



(a) Mapa CSP a modelo OPL. **(b) Mapa CSOP a modelo OPL.**

Figura 8.1: *Ejemplos de mapas CSP o CSOP a modelos OPL.*

El *script* está compuesto de **(1)** una sección de declaraciones de los modelos OPL, que corresponden al CSP principal, el CSP cuantificado y el CSOP de optimación (opcional), seguida de **(2)** una sección de declaración de tipos y variables necesarios, que incluye un array abierto de candidatos, **(3)** una sección de procedimientos auxiliares y, finalmente, **(4)** el cuerpo del *script* propiamente dicho, que efectúa el procedimiento que sigue a continuación.

En primer lugar, se obtienen todas las soluciones del CSP principal. Para cada solución candidata se comprueba previamente si ya ha sido procesada[†2]. Esta búsqueda se hace de manera dicotómica con objeto de mejorar el rendimiento; para ello necesitamos que las soluciones candidatas vengan ordenadas de manera ascendente, lo que se consigue especificando el procedimiento de búsqueda estándar adecuado en la sección de búsqueda del CSP principal[†3]. A continuación, si el candidato aún no se ha procesado, entonces se incluye en el array abierto de candidatos, se inicializa el CSP cuantificado asignando las variables ligadas de la solución candidata, y se comprueba su satisfactibilidad. Si no lo es, entonces la solución candidata se considera que *es* una solución y se incluye en el array abierto de soluciones; en caso contrario se descarta. Por último, de manera opcional, el CSOP de optimación encuentra las mejores soluciones; sigue una estrategia de búsqueda en profundidad entrelazada que permite encontrar mayor número de soluciones óptimas, todas ellas ordenadas ascendemente por su utilidad.

[†2] Nótese que esto es posible porque una misma solución candidata puede estar asociada a varios valores ligados; así evitamos repetir procedimientos de resolución que tienen un alto coste computacional.

[†3] Nótese que la inserción no tiene que ser binaria, sino que las nuevas soluciones se van insertando directamente al final del array de candidatos, ya que se procesan de manera ordenada en el CSP principal.

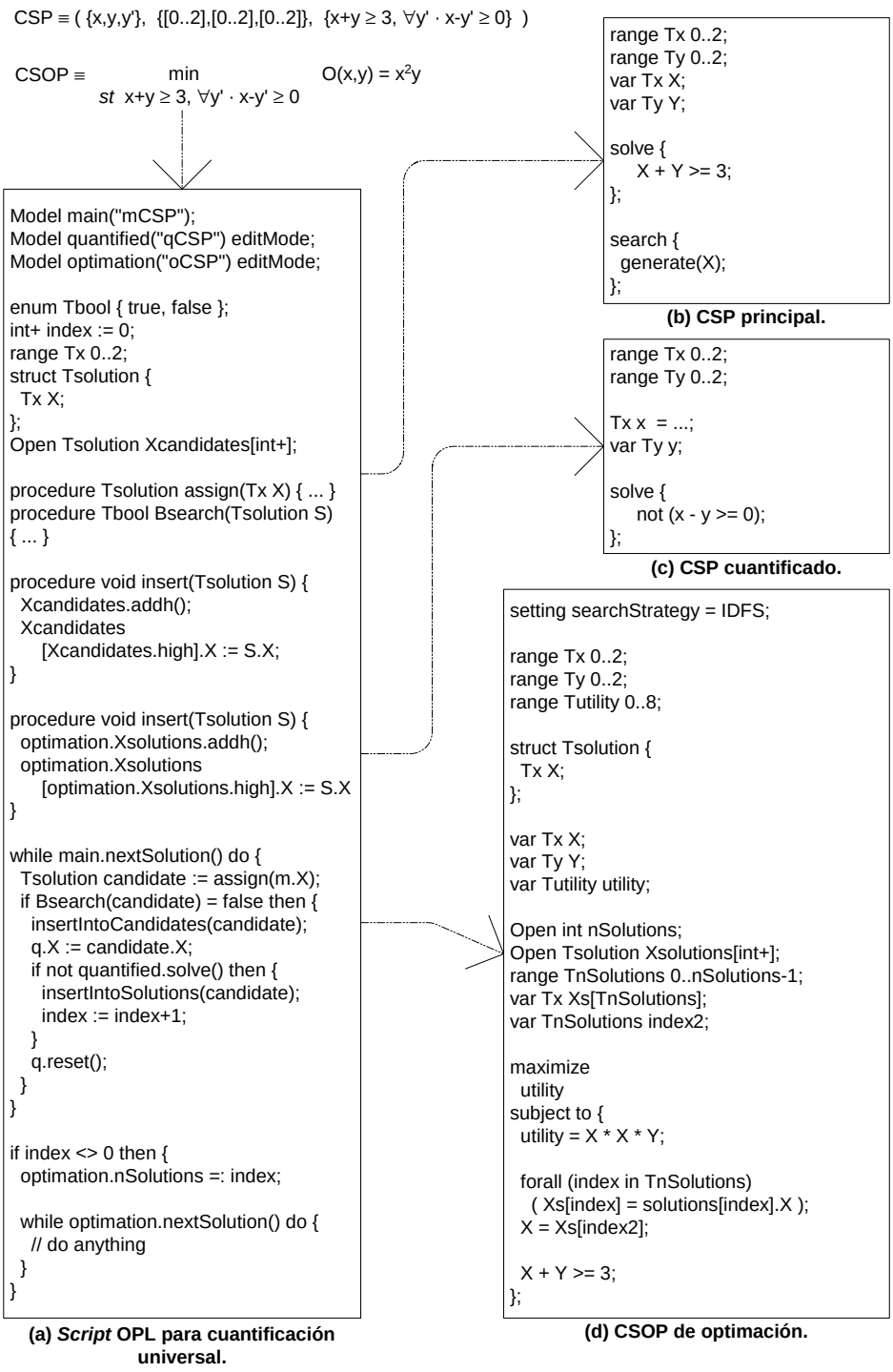


Figura 8.2: *Mapeando CSP cuantificado universalmente a OPL.*

En la figura §8.3 mostramos la traza para un pequeño ejemplo de resolución de un CSP con cuantificador universal. Nótese que podemos distinguir dos partes, que denominamos CSP principal y CSP cuantificado propiamente dicho. Para resolverlo, obtenemos el conjunto de soluciones del CSP principal $S = \{(1,2),(2,1),(2,2)\}$ y tomamos los valores de la proyección respecto a la variable x, que resulta el conjunto $\Pi_x(S) = \{1,2\}$. Entonces, para cada valor candidato de x se comprueba que todos los valores de la variable y' que tiene ligados satisfacen el CSP cuantificado. Nótese que este esquema está limitado a que la variable cuantificada sea discreta y finita, aunque tiene un orden lineal de complejidad computacional.

Éste debe transformarse a un esquema con cuantificador existencial, como se comenta en el apartado §8.2.3:

$$\forall y' \cdot x - y' \geq 0 \equiv \neg\exists y' \cdot \neg(x - y' \geq 0)$$

En nuestro ejemplo, para $x = 1$ hay un valor $y' = 2$ que satisface la restricción transformada, por lo que $(1,2)$ se descarta como solución del CSP[†4]. Para $x = 2$, ningún valor la satisface, de manera que $(2,1)$ y $(2,2)$ se confirman como soluciones.

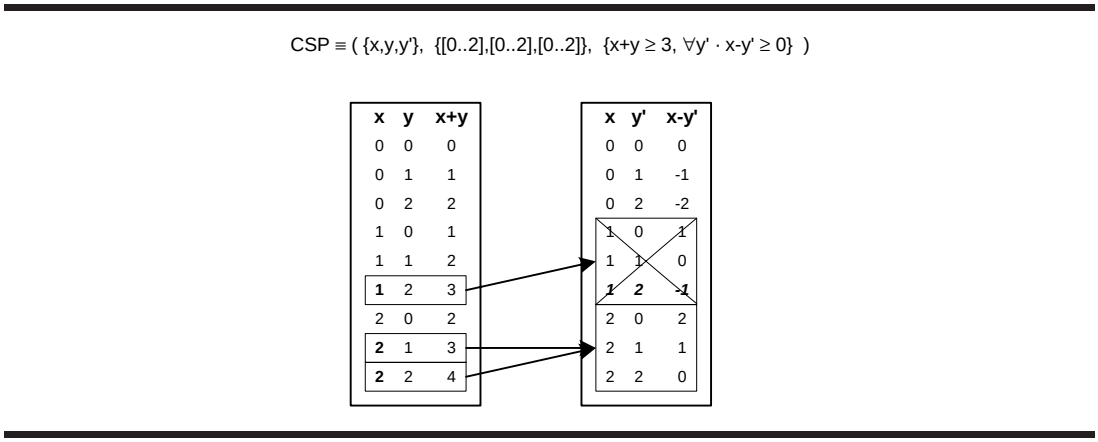


Figura 8.3: *Traza de resolución de un CSP cuantificado.*

A pesar de que nuestra solución al problema está limitada a variables cuantificadas con dominios discretos y finitos, es viable puesto que, por regla general, los atributos controlables suelen tener dominios enumerados para indicar diferentes opciones de una oferta de servicio, por lo que se ajusta perfectamente a nuestras necesidades.

[†4]De hecho, las soluciones del CSP cuantificado reflejan qué valores de y' hacen que la solución principal sea descartada.

8.2.3. Esquemas para Conformidad

En el contexto del predicado de conformidad, hay que comprobar la inclusión de sendos espacios de soluciones (sección §6.4). Hasta lo que sabemos, la inclusión de espacios no está soportada directamente por ningún resolutor, pero sí podemos comprobarla de manera indirecta aplicando el operador de *implicación de restricciones* definido por Marriott y Stuckey [125].

Definición 8.1 (Implicación de CSP) *Sea ψ_1 un CSP que viene dado por una tupla de la forma (V, D, C_1) y ψ_2 otro CSP que vienen dado por (V, D, C_2), el espacio de soluciones de ψ_1 es un subconjunto del espacio de soluciones de ψ_2 si y sólo si el CSP definido como $(V, D, C_1 \longrightarrow C_2)$ es satisfactible, o su equivalente CSP $(V, D, C_1 \wedge \neg C_2)$ no lo es.*

Hay que añadir que la inclusión de la negación es una fuente de ineficiencia, como mostramos en la sección §8.4.

8.2.4. Otras Consideraciones

Finalmente, explicamos brevemente nuestra solución a algunas características del lenguaje OPL, que hemos mencionado anteriormente:

Tipo Conjunto Para transformar un atributo de tipo conjunto hay que definir un tipo enumerado que contiene a priori los posibles elementos del conjunto, de manera que la variable de tipo array indica la pertenencia o no-pertenencia del elemento. Todas las operaciones de conjunto y enumerado habituales pueden expresarse, aunque de una manera compleja y poco intuitiva. Por ejemplo, si definimos en OPL el enumerado `Tz` para declarar que p, q, r y s son los elementos pre-determinados del conjunto, y la variable z como un array lógico indexado por `Tz`:

enum Tz {p, q, r, s};
var int z[Tz] **in** 0 . . 1;

entonces la restricción de igualdad $z = \{p, s\}$ debe marcar a 1 la posición de los elementos p y s, y dejar a 0 las correspondientes a q y r:

$$z[p] = 1 \;\&\; z[q] = 0 \;\&\; z[r] = 0 \;\&\; z[s] = 1;$$

Restricciones No–Lineales con Variables Reales Transformamos las variables a enteras y escalamos *lo necesario* las variables y constantes implicadas hasta obtener una determinada precisión.

Por ejemplo, si x e y son variables reales ligadas con la restricción:

$$x/y \geq 0.95 \wedge x \leq y$$

entonces si sólo re-definimos las variables como enteras, la restricción no tiene solución puesto que la división entera siempre resulta 0. Hay que despejar y escalar para obtener la precisión mínima de manera que la restricción tenga solución:

$$100 * x \geq 95 * y \wedge x \leq y$$

Funciones–Objetivo Ante la expresividad para definir las funciones-objetivo en OPL, se hace preferible maximizar o minimizar el valor de una variable auxiliar a la que asignamos el valor objetivo mediante restricciones, representado *ad hoc* la función objetivo. Además, para describir las funciones-objetivo se puede abarcar toda la expresividad que ofrece el lenguaje OPL para describir las restricciones.

8.3. Tareas de Emparejamiento en OPL

En las figuras §8.4 y §8.5 se muestran esquemáticamente los modelos OPL que resultan con la transformación de los CSP de conformidad y optimalidad, respectivamente.

En el caso del CSP de conformidad, se aplica la transformación específica de los CSP cuantificados. En este esquema, el modelo OPL principal contiene las restricciones correspondientes a los predicados de objetivos requeridos de ambas ofertas de acuerdo. Este modelo OPL es invocado por un *script* OPL, de manera que para cada punto de acuerdo requerido que se obtiene por el modelo OPL principal, el *script* comprueba si es un punto de acuerdo conforme utilizando sendos modelos OPL correspondientes a la parte cuantificada del predicado de conformidad.

En el caso del CSOP de optimalidad, hay que maximizar una variable auxiliar que se corresponde a la utilidad combinada del emparejamiento de ambas ofertas de acuerdo. Esta variable toma el valor del producto de otras tantas variables auxiliares que se corresponden a las valoraciones de utilidad de demanda frente a la oferta y oferta frente a la demanda. A su vez, cada una de estas variables auxiliares toman sus valores a partir de restricciones que se incluyen al CSP, que se corresponden a las funciones de utilidad y distribuciones de probabilidad aplicadas. A estas restricciones se añaden los puntos de acuerdo conformes, que proceden del *script* OPL, junto a las restricciones correspondientes a los predicados de los objetivos garantizados de ambas ofertas de acuerdo.

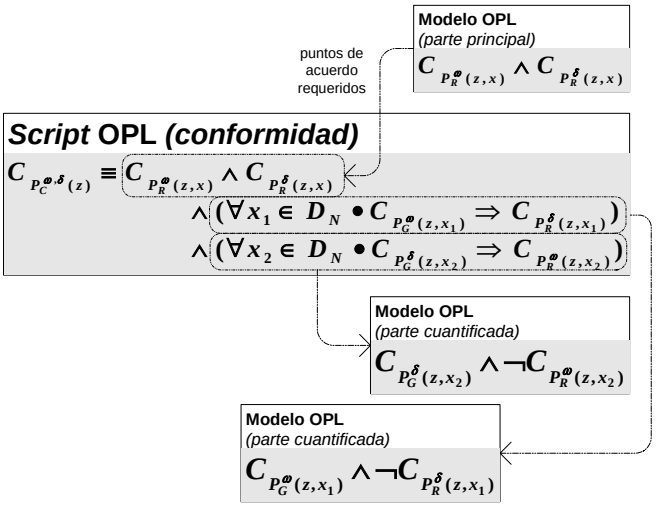


Figura 8.4: *Esquema de modelos y scripts de OPL para la conformidad.*

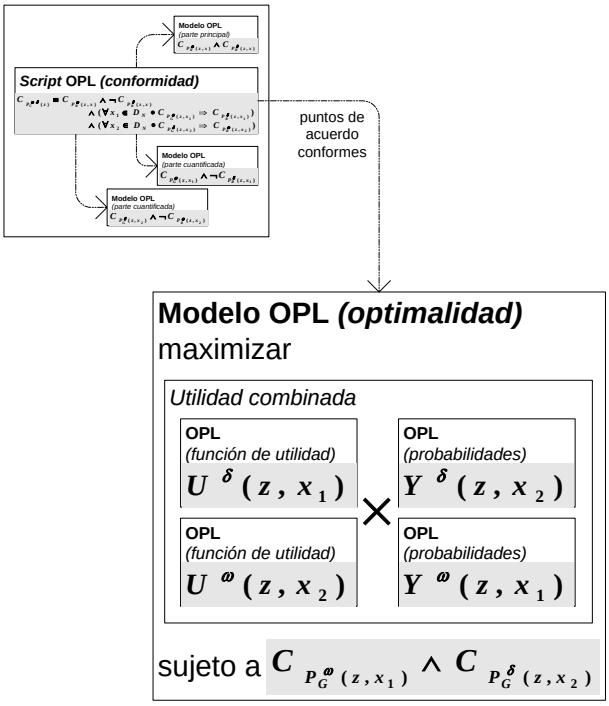


Figura 8.5: *Esquema de modelos y scripts de OPL para la optimalidad.*

8.4. Resultados Experimentales

Para evaluar la implementación de nuestra prototipo y estudiar la viabilidad de nuestra propuesta, hemos realizado varias pruebas experimentales. Éstas también fueron dedicadas a justificar de manera empírica algunas de las decisiones de diseño tomadas respecto al intermediario de calidad, como hemos presentado en el capítulo §8.

Nuestro prototipo se ha implementado para la plataforma Microsoft .NET; nuestras pruebas se han ejecutado sobre una máquina con Windows 2000 Professional, equipado con un microprocesador 1.8 Ghz AMD Athlon y 512 MB de memoria RAM, conectado a una red LAN Ethernet de 100 Mbits. Durante las pruebas, la máquina no ha sido desconectada de la red porque el resolutor OPL necesita comunicarse con un servidor de licencias, así que la ejecución de estas pruebas ha tenido lugar en períodos de mínimo uso de la red. Los servicios de Windows se han reducido al mínimo para evitar tantas interferencias externas como fuera posible, aunque sin detener el recolector de basura de la plataforma.

Para reducir el efecto de los posibles *outliers* provocados por interferencias ocasionales del sistema operativo o la red, se han medido los tiempos medios de 30 ejecuciones, descartando los tiempos máximo y mínimo en cada ejecución del experimento.

En general, hemos medido la latencia del resolutor, o tiempo transcurrido, para los casos correspondientes a comprobar la conformidad y selección óptima. Nótese que la consistencia puede considerarse como un caso particular de la conformidad, desde el punto de vista del resolutor, ya que ambas tareas se basan en comprobar la satisfactibilidad de un CSP / CSOP. Entre las variables del experimento que hemos tenido en cuenta destacan:

- Aridad de las restricciones que conforman las expresiones correspondientes a los objetivos garantizados y requeridos de los términos de las ofertas de acuerdo.

- Número de variables, correspondientes a atributos de calidad.

- Número de modelos OPL, correspondientes a las tareas de emparejamiento sobre ofertas de acuerdo.

8.4.1. Descripción General del Experimento

Las pruebas que hemos realizado son:

- La comprobación de la conformidad con ofertas de acuerdo cuyos términos tienen objetivos asimétricos. En esta prueba, comparamos la utilización de un resolutor de restricciones frente a una solución *ad hoc* que comprueba la conformidad mediante programación directa.
- La comprobación de la conformidad con ofertas de acuerdo cuyos términos tienen objetivos simétricos no-puntuales. Distinguimos entre comprobar la intersección no-vacía de los espacios de soluciones frente a comprobar la inclusión de los espacios de soluciones de los correspondientes CSP de conformidad.
- La reedición de los experimentos de la plataforma AgFlow [212, 213] para medir el rendimiento de la selección óptima, En esta ocasión, comparamos la programación con restricciones frente a técnicas de programación lineal para obtener la solución óptima.

Hay que recordar que el prototipo que hemos desarrollado es un emparejador de única-vía que no incluye la controlabilidad de los atributos de calidad ni la consciencia temporal.

8.4.2. Conformidad

8.4.2.1. Ofertas de Acuerdos con Objetivos Asimétricos

Intuitivamente, si los términos garantizados sólo incluyen objetivos basados en pares atributos / valor, entonces la utilización de un resolutor de restricciones para comprobar la conformidad parece demasiado generosa. En este primer experimento, hemos estudiado el rendimiento de los resolutores de restricciones frente a soluciones programadas *ad hoc*.

Datos Experimentales Los datos experimentales han consistido en un conjunto de ofertas y demandas, de manera que a cada oferta se le corresponde una demanda. Cada oferta incluye términos que contienen asignaciones a n atributos, de tipo entero byte sin signo. Cada demanda incluye términos con restricciones simples sobre dichos atributos. Los valores de n van desde desde 10 a 100 cada 10, y desde 100 a 2000 cada 100. Cada ejecución del experimento

ha comprobado si una oferta es conforme a su correspondiente demanda, y se ha probado tanto la solución programada *ad hoc* como el resolutor OPL.

```
class TestingSatisfiability
{
        public TestingSatisfiability()
        {}

        public bool isSatisfiable()
        {
                bool bResult;

                // Oferta
                byte x1 = 5, x2 = 5;

                // Demanda
                if ( x1 > 2 && x2 < 10 )
                        bResult = true;
                else
                        bResult = false;

                return bResult;
        }
}
```

(a) Solución programada *ad hoc*.

```
var int x1 in 0..255;
var int x2 in 0..255;

solve {
        // Oferta
        ( x1 = 5 & x2 = 5 )
        &
        // Demanda
        ( x1 > 2  &  x2 < 10);
};
```

(b) Modelo OPL (intersección).

```
var int x1 in 0..255;
var int x2 in 0..255;

solve {
        // Oferta
        ( x1 = 5 & x2 = 5 )
        & not
        // Demanda
        ( x1 > 2  &  x2 < 10);
};
```

(c) Modelo OPL (subconjuntos).

Figura 8.6: *Datos experimentales para conformidad / objetivos asimétricos.*

La figura §8.6 muestra el programa *ad hoc* (a) y los modelos OPL (b y c) de la ejecución para $n = 2$. Cada ejecución ha sido como sigue:

- Para probar la solución programada *ad hoc*, un pequeño programa ha incluido el método `IsSatisfiable()` para comprobar si la oferta, que viene dada por términos garantizados basados en asignaciones a las variables que representan los atributos, es conforme a la demanda, cuyos términos vienen dados por expresiones condicionales sobre tales variables. Este programa ha sido creado, compilado y ejecutado dinámicamente para obtener el resultado de conformidad, aunque sólo se han medido los tiempos de compilación y ejecución.

- Para probar el resolutor de restricciones OPL, se han creado sendos modelos OPL. El primero de ellos, que se muestra en la figura §8.6(b), comprueba si los espacios de soluciones correspondientes a la demanda y oferta tienen una intersección no-vacía; mientras que el segundo, que se muestra en la figura §8.6(c), comprueba si el espacio de soluciones correspondiente a la oferta es un subconjunto del espacio de soluciones correspondiente a la demanda. Estos modelos también se han creado dinámicamente y pasados al resolutor OPL, que los ha compilado para obtener la solución, de manera que sólo se han medido los tiempos de carga y resolución del modelo.

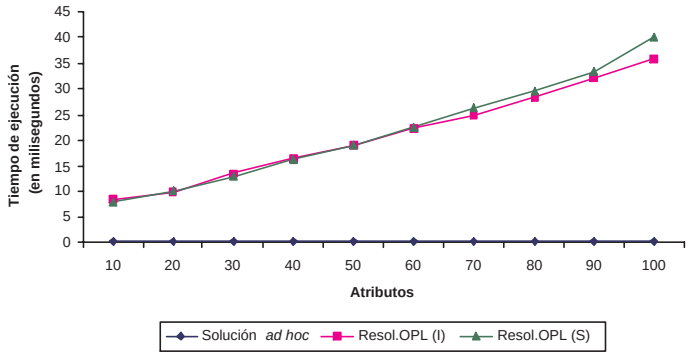


Figura 8.7: *Resultados del estudio de conformidad / objetivos asimétricos.*

Hay que destacar que la inclusión puede tener un gran coste computacional, debido a las negaciones inherentes en el predicado de conformidad. Por ello, un objetivo complementario ha sido verificar si es mejor comprobar previamente la intersección no-vacía, de manera que en caso de que sea negativa ya sabemos que no hay solución, evitando el costoso chequeo de comprobar la inclusión de los espacios de soluciones.

Estudio Empírico Los resultados experimentales se muestran en la figura §8.7, que demuestran con claridad que es preferible evitar el resolutor en el caso de que las ofertas vengan determinadas con términos garantizados basados en pares atributo / valor. La latencia de la solución programada *ad hoc* es imperceptible y se comporta de manera lineal. El resolutor OPL también tiene comportamiento lineal mientras que $n \leq 500$, entonces el resolutor OPL tiene un comportamiento polinomial de segundo grado.

La figura §8.7 también muestra que la latencia de la comprobación de subconjuntos no es mucho mayor que la comprobación de la intersección, por lo que pensamos que no merece la pena comprobar previamente la intersección en un contexto de objetivos asimétricos.

Hay que tener en cuenta que la asignación directa de valores a las variables debida a los términos garantizados es asumida por los procedimientos de etiquetado estándar durante la construcción del árbol de búsqueda, de manera que la resolución es muy eficiente en estos casos.

```
var int x0 in 0..255;
var int x1 in 0..255;
var int x2 in 0..255;

solve {
 // Demanda
 ( x0 > 4 & x1 > 2 & x2 > 10)
 // Oferta
 &  ( x0 > 4 & x1 > 2 & x2 > 10)
};
```

(a) Expresiones binarias (sat).

```
var int x0 in 0..255;
var int x1 in 0..255;
var int x2 in 0..255;

solve {
 // Demanda
 ( x0 > 4 & x1 > 2 & x2 > 10)
 // Oferta
 &  ( x0 > 4 & x1 > 2 & x2 < 10)
};
```

(b) Expresiones binarias (no-sat).

```
var int x0 in 0..255;
var int x1 in 0..255;
var int x2 in 0..255;

solve {
 // Demanda
 ( x0 + x1 + x2 > 10)
 // Oferta
 &  ( x0 + x1 + x2 > 10)
};
```

(c) Expresiones ternarias (sat).

```
var int x0 in 0..255;
var int x1 in 0..255;
var int x2 in 0..255;

solve {
 // Demanda
 ( x0 + x1 + x2 > 10)
 // Oferta
 &  ( x0 + x1 + x2 < 10)
};
```

(d) Expresiones ternarias (no-sat).

```
var int x0 in 0..255;
var int x1 in 0..255;
var int x2 in 0..255;

var int aux1 in 0..255;

solve {
 // Demanda
 ( x0 + x1 = aux1 & aux1 + x2 > 10)
 // Oferta
 &  ( x0 + x1 = aux1 & aux1 + x2 > 10)
};
```

(e) Expresiones binarizadas (sat).

```
var int x0 in 0..255;
var int x1 in 0..255;
var int x2 in 0..255;

var int aux1 in 0..255;

solve {
 // Demanda
 ( x0 + x1 = aux1 & aux1 + x2 > 10)
 // Oferta
 &  ( x0 + x1 = aux1 & aux1 + x2 < 10)
};
```

(f) Expresiones binarizadas (no-sat).

Figura 8.8: *Datos experimentales para conformidad / objetivos simétricos.*

8.4.2.2. Ofertas de Acuerdo con Objetivos Simétricos

En el apartado anterior, ya hemos comparado las posibilidades de basar la conformidad en la intersección de espacios de soluciones, frente a la inclusión que, a priori, es una operación más costosa computacionalmente. En el contexto de ofertas de acuerdo con objetivos asimétricos, ya hemos comprobado que no hay grandes diferencias.

En este apartado, presentamos un segundo experimento para medir la latencia del resolutor OPL para comprobar la conformidad en el contexto de ofertas de acuerdo con objetivos simétricos no-puntuales, según todas las consideraciones anteriores.

Datos Experimentales Los datos experimentales han consistido de ofertas y sus correspondientes demandas. Cada caso del experimento está basado en n atributos, con n desde 2 a 10. Hemos limitado el rango de n debido al esperado comportamiento exponencial. Cada caso se ha ejecutado para atributos basados en dominios de rango pequeño (enteros cortos sin signo) y medio (enteros sin signo). Hemos ejecutado tres pruebas, para comprobar diferentes patrones de demandas y ofertas:

- Modelos OPL de demandas y ofertas con restricciones binarias. Se comprueba tanto la satisfactibilidad como la no-satisfactibilidad, tanto para el caso de la intersección como el de la inclusión.
- Modelos OPL de demandas y ofertas cuyas restricciones incluyen una operación aritmética con aridad k, insertada en una comparación binaria. La aridad de las operaciones viene dada por el número de sus operandos. Cada caso se restringe de manera que satisfaciera que $k = n$.
- Modelos OPL de demandas y ofertas cuyas restricciones incluyen, como antes, una operación aritmética con aridad k, tal que $k = n$, insertada en una comparación binaria, aunque esta vez las restricciones se han transformado a su equivalente binaria[†5].

La figura §8.8 muestra los modelos OPL correspondientes a la comprobación de la intersección para las tres pruebas (a, b y c, respectivamente). los modelos OPL basados en la comprobación de la inclusión se construyen de manera similar, pero no se muestran en la figura.

Estudio Empírico Los resultados experimentales se muestran en la figura §8.9, de manera que las figuras en el lado izquierdo corresponden a la comprobación de la intersección, mientras que las figuras en el lado derecho a la comprobación de la inclusión. Estos experimentos demuestran de forma clara que la aridad de las restricciones implicadas afecta a la latencia del resolutor de restricciones.

Las figuras §8.9(a) y §8.9(b) muestran la latencia cuando demandas y ofertas sólo contienen comparaciones simples, de manera que el resolutor OPL se comporta casi linealmente.

Sin embargo, las figuras §8.9(c) y §8.9(d) muestran la latencia cuando las demandas y ofertas contienen operaciones aritméticas con aridad creciente,

[†5]La binarización es un proceso por el que cualquier restricción con aridad k se transforma a un conjunto equivalente de restricciones binarias. Esta es una cuestión importante porque la mayoría de los resolutores de restricciones están optimizados para resolverlas [177].

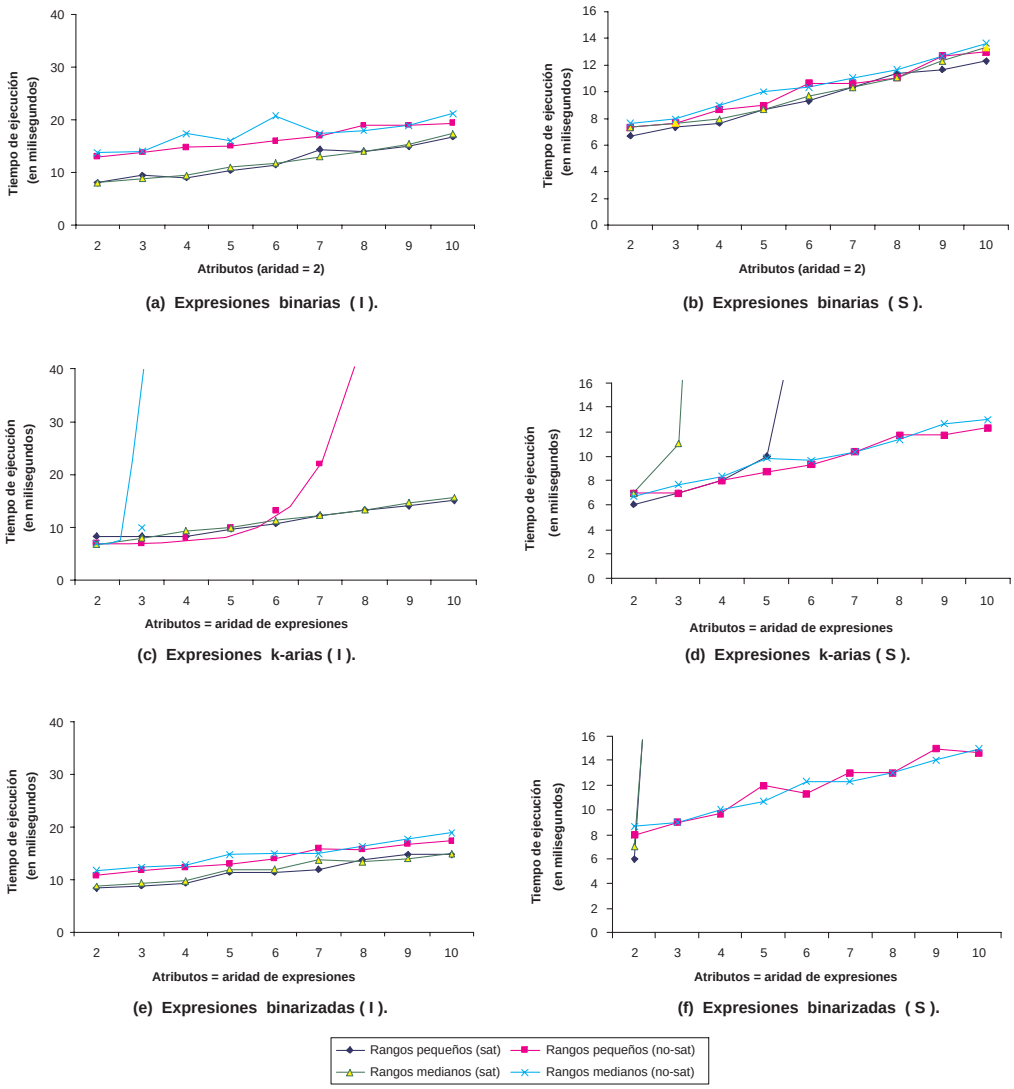


Figura 8.9: *Resultados del estudio de conformidad / objetivos simétricos.*

entonces la latencia es mayor en el caso de la no-satisfactibilidad (comprobación de la intersección, a la izquierda) o la satisfactibilidad (comprobación de la inclusión, a la derecha) porque estos casos no tienen solución y el resolutor OPL lleva a cabo un proceso de *backtracking* hasta que comprueba el dominio completo del espacio de soluciones. Si los rangos de variables son mayores (además), entonces los tiempos de latencia son dramáticamente superiores.

Nótese que éste es el peor caso, donde por regla general los resolutores tienen un comportamiento exponencial. Por otro lado, el resolutor OPL se com-

porta linealmente en los casos complementarios (donde sí hay alguna solución).

Afortunadamente, la mayoría de los resolutores están mejorados cuando gestionan restricciones binarias. Por ejemplo, tiene un comportamiento lineal, como se muestra en la figura §8.9(e) en el caso de la comprobación de la intersección. Sin embargo, la comprobación de la inclusión tiene un comportamiento exponencial, como se muestra en la figura §8.9(f), en el caso de la satisfactibilidad.

Esta prueba ha mostrado que, si las restricciones se binarizan entonces es mejor comprobar la intersección previamente. En este caso, si el resultado es vacío, entonces la oferta es no-conforme. De otra manera, hay que llevar a cabo la comprobación de la inclusión para la conformidad.

8.4.3. Selección Óptima

Este experimento ha reeditado las pruebas publicadas sobre AgFlow [212, 213]. En concreto, hemos centrado nuestras pruebas a la optimación local, es decir, la selección de la mejor oferta para una demanda, pero utilizando programación con restricciones. De nuevo, nuestro objetivo ha sido verificar si es mejor utilizar el resolutor de restricciones, o si se prefiere invocar directamente una solución programada *ad hoc*, en el contexto de ofertas de acuerdos con objetivos asimétricos.

Datos Experimentales Los datos del experimento han consistido en N_D demandas, de manera que cada demanda podría seleccionar entre un conjunto de N_F ofertas. El número de atributos se ha fijado a 5, que es el número de atributos considerados en AgFlow. Los valores de N_D van desde 10 a 80 cada 10, mientras que los valores de N_F van desde 10 hasta 40 cada 10. Las ofertas y demandas han sido creadas de manera aleatoria; cada oferta asigna un valor a cada atributo y cada demanda asigna un peso a cada atributo.

La figura §8.10 muestra el programa *ad hoc* (a) y el modelo OPL (b) que corresponden a sendas demanda y oferta de la instancia $N_D = 80$ y $N_F = 40$ del experimento. Nótese que estos casos han sido generados sin restricciones, de manera que todos ellos son conformes a sus correspondientes demandas. Cada caso se ha probado de esta manera:

- Para probar la solución programada *ad hoc*, un pequeño programa incluye el método `GetOptimum()` para obtener la oferta óptima, según el

```
class TestingSelection
{
        const int NumOffers = 40, NumParams = 5;

        public TestingSelection() {}
        private int getBestScore(double [][] normalOffers) { ... }
        private double [][] normalize(int [][] offers) { ... }

        public string getBestOffer()
        {
                // Conjunto de nombres de ofertas candidatas
                string [] nameOffers = new string []
                 { "Offer_80_40_1_1.cs", ..., "Offer_80_40_1_40.cs" };

                // Conjunto de ofertas candidatas
                int [][] offers = new int [NumOffers][];
                offers[0] = new int [NumParams] {20,40,90,90,2};
                ...
                offers[39] = new int [NumParams] {20,30,90,90,2};

                return nameOffers[getBestScore(normalize(offers))];
        }

        private double getScore(double [] normalOffer)
        {
                double score = 0;
                double [] weigths = new double [] { 10, 0, 40, 30, 20 };

                for (int i = 0; i < NumParams; i++)
                        score = score + weigths[i] * normalOffer[i];

                return score;
        }
}
```

(a) Solución programada *ad hoc*.

```
var int P in 0..1000;
var int D in 0..1000;
var int A in 0..100;
var int S in 0..100;
var int R in 0..100;

enum offers {offer_80_40_1_1,...,offer_80_40_1_40};
var offers selected_offer;

maximize

    0.10 * (100 - piecewise{0.1->1000;0} P)
  + 0 * (100 - piecewise{0.1->1000;0} D)
  + 0.40 * piecewise{1->100;0} A
  + 0.30 * piecewise{1->100;0} S
  + 0.20 * piecewise{1->100;0} R

subject to {
  (
    selected_offer = offer_80_40_1_1
    &
    ( P = 20  & D = 40 & A = 90 & S = 90 & R = 2 )
  )
  \/
  ...
  \/
  (
    selected_offer = offer_80_40_1_40
    &
    ( P = 20 & D = 30 & A = 90 & S = 90 & R = 2 )
  );
};
```

(b) Modelo OPL.

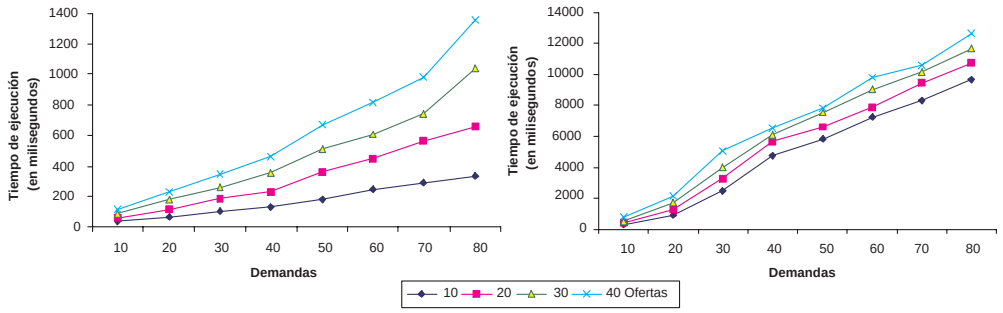
Figura 8.10: *Datos experimentales para la selección óptima.*

mismo esquema que utiliza AgFlow. Este programa ha sido creado, compilado y ejecutado dinámicamente para obtener el resultado, pero sólo se han medido los tiempos de compilación y ejecución.

- Para probar el resolutor de restricciones, el modelo OPL también ha sido creado dinámicamente y pasado al resolutor, que lo compila y obtiene una solución. El tiempo medido sólo ha incluido la carga y resolución del modelo.

Estudio Empírico Los resultados experimentales se muestran en la figura §8.11. De nuevo, las medidas obtenidas demuestran que es preferible utilizar una solución programada *ad hoc* que un resolutor de restricciones en el contexto de ofertas de acuerdo con objetivos asimétricos.

El resolutor OPL tiene un comportamiento cuadrático abierto, debido a las técnicas de programación lineal entera que aplica, aunque con mayores valores absolutos de latencia.



(a) Selección óptima con solución programada *ad hoc*. **(b) Selección óptima con resolutor OPL.**

Figura 8.11: *Resultados del estudio de selección óptima.*

8.5. Resumen

En este capítulo hemos presentado algunos aspectos del modelo de implementación de HDM. Hemos revisado las características de ILOG OPL–Studio, que es el resolutor de restricciones que hemos utilizado, y las implicaciones que su uso ha tenido respecto a la implementación de las tareas de emparejamiento. En concreto, hemos constatado el problema para tratar la cuantificación universal y la solución que hemos proporcionado mediante *scripts* de OPL. Sin embargo, esta solución está restringida a variables discretas y finitas. Por otro lado, hay que recordar que esta limitación ya era obligada porque OPL restringe el uso de variables reales a expresiones lineales.

Por otro lado, los estudios empíricos han permitido confirmar las decisiones de diseño que hemos tomado respecto a HDM. Estos experimentos se realizaron sobre nuestro prototipo, del que en [168] presentamos una versión preliminar de su arquitectura. Posteriormente fue presentado en [127, 128, 132]. El lector interesado puede examinar una prueba de conceptos en la siguiente página web:

`http://www.tdg-seville.info/topics/procurement.html`

La tabla §8.1 resume los rangos de tratabilidad que obtuvimos. En ella, N representa el número de atributos y k la aridad de las restricciones. Nótese que la programación con restricciones es excesivamente costosa en el caso de la conformidad con objetivos simétricos basada en la inclusión, que se corresponde a la definición de conformidad mas compleja de las que hemos presentado. Sin embargo, estos experimentos han sido preliminares y hay que completarlos con el objetivo de ampliar el abanico a diferentes grados de ari-

	N	k
Objetivos asimétricos	500	–
Obj. simétricos y conformidad basada en intersección:		
binarias	–	2
no binarias	6 – 7	6
binarizadas	–	2
Obj. simétricos y conformidad basada en inclusión:		
binarias	–	2
no binarias	4 – 5	4
binarizadas	2	2

Tabla 8.1: *Rangos de tratabilidad.*

dad para un mismo número de atributos y estudiar el efecto del tratamiento de la controlabilidad y la consciencia temporal en el rendimiento de las tareas de emparejamiento.

Estos experimentos pendientes deben complementar los estudios analíticos que hemos realizado sobre el comportamiento computacional de los algoritmos propuestos para la controlabilidad y consciencia temporal. En el capítulo §8 mostramos que el tratamiento de las variables controlables tiene orden lineal, aunque restringido a variables discretas y finitas. Igualmente, en el apéndice §B mostramos que el comportamiento del algoritmo de proyección temporal también tiene orden lineal, de manera que el aplanamiento resulta con orden $O(N \log N)$ y el recorrido paralelo $O(N)$. Estos últimos se han presentado en el capítulo §7.

Parte IV

Epílogo

Capítulo 9

Conclusiones y Trabajo Futuro

I'll be back!

Arnold Schwarzenegger, The Terminator
"The Terminator", 1984

9.1. Conclusiones

Como conclusión de este trabajo de tesis doctoral podemos afirmar que:

"*Es posible extender los actuales modelos de ofertas de acuerdo para tener en cuenta escenarios de emparejamiento automático más complejos. Dicha extensión puede llevarse a cabo sobre un marco de trabajo descrito con rigor, estructurado en varios niveles y que dispone de una implementación eficiente.*"

Este marco de trabajo lo hemos denominado HDM, acrónimo de *Holistic Decision Maker*, pues por una parte su funcionalidad esencial se corresponde con la de un tomador de decisiones, y por otra ha sido necesario utilizar una perspectiva holística para poder integrar adecuadamente las características necesarias para soportar los escenarios de emparejamiento avanzados. De no haber tomado esta perspectiva holística se habrían quedado sin descubrir algunas de las características añadidas, concretamente las que han surgido de la interrelación entre características más básicas. Por ejemplo, el uso conjunto de atributos controlables y no controlables y la inclusión de consciencia temporal en los valores de negocio.

9.2. Discusión sobre Decisiones Tomadas

A continuación, reseñamos brevemente algunas de las principales decisiones que hemos tomado durante el desarrollo de la tesis y que, en cierto modo, determinan algunas de las limitaciones y extensiones de nuestra propuesta:

- *¿Es adecuada la técnica de especificación empleada para describir el nivel abstracto de HDM?* La teoría de conjuntos y la lógica de predicados proporciona una notación sencilla de utilizar y comprender, por lo que en principio parece acertada su elección. No obstante, el no usar una notación formal presenta las limitaciones habituales de las notaciones informales.

  Extensión Utilizar una técnica de especificación formal como Z [96]. Al ser un lenguaje fuertemente tipado, se obtienen beneficios tales como que podemos comprobar la consistencia del modelo abstracto de manera automática. Merece al pena señalar que WSDL en su última versión ha sido especificado en Z.

- *Dado que determinar la satisfactibilidad de un CSP es un problema NP-duro*[†1]*, ¿es razonable plantear una implementación con resolutores de restricciones?* Existen dos razones. En primer lugar, la programación con restricciones ha sido calificada como "Santo Grial" por Freuder [59] ya que permite establecer los problemas de una manera declarativa, utilizando restricciones muy expresivas, que resolvemos mediante resolutores sin necesidad de especificar cómo resolverlos. De por sí, esta ventaja puede ser suficiente como para asumir un alto coste computacional. No obstante, y en segundo lugar, el rango de tratabilidad (coste computacional) de HDM, obtenido en los experimentos sobre nuestro prototipo, ha demostrado ser suficiente.

  Extensión Hay que repetir estos experimentos en nuevos prototipos que incluyan las características más avanzadas y determinar la complejidad computacional resultante.

- *¿Por qué no utilizar en el modelo operacional la programación lineal, más eficiente, tal y como hacen Zeng et al. [213] y otros autores?* Ya Smith[177] argumentaba que la representación de un problema mediante CP es frecuentemente más cercana al problema original porque las variables de los CSP se corresponden directamente a las entidades del problema y las restricciones se pueden expresar directamente con una

[†1] La resolución de un CSP con restricciones binarias es un problema NP-completo [121].

formulación más simple, mientras que en IP ó MIP el modelado debe adaptarse a una representación matemática más estricta[†2].

- *¿Por qué no se ha interpretado un acuerdo con consciencia temporal como un CSPT (Temporal CSP)?* En general, los TCSP establecen restricciones sobre variables que representan eventos que ocurren en ciertos intervalos o instantes temporales. *Todas* las variables representan eventos. Las restricciones establecen relaciones temporales entre estos eventos, según fueron definidas originalmente por Allen [7, 8]. Hay extensiones de los resolutores para resolver estos tipos de problemas [147, 174], a los que podemos añadir la existencia de lógicas descriptivas temporales [13] y lógicas temporales [120].

  Conclusión La estructura prefijada de los actuales TCSP no es adecuada a nuestras necesidades, puesto que en nuestro caso no todas las variables corresponden a eventos temporales. Por ello, hemos optado por desarrollar un algoritmo de proyección temporal, que tiene un coste computacional lineal.

  Extensión En cualquier caso, hay que volver a estudiar las posibilidades de los TCSP cuando haya propuestas que admitan que haya variables que no representen eventos temporales. De esta manera, el modelo operacional quedaría expresado únicamente en términos de CSP.

9.3. Extensiones y Trabajo Futuro

La tabla §9.1 resume las características avanzadas que tiene HDM, distinguiendo el nivel de abstracción donde se han abordado. En esta tabla **MA** significa que dicha característica se ha tenido en cuenta en el modelo abstracto; **MO** significa que dicha característica es tenida en cuenta por el modelo operacional; y **MI** significa que dicha característica se ha tenido en cuenta en el modelo de implementación.

A la vista de esta tabla, el trabajo pendiente de realizar es la incorporación de algunas de las características avanzadas a nuestro prototipo, aunque hay que indicar que los algoritmos y esquemas de transformación necesarios para su implementación se han definido y se incluyen en esta memoria de tesis doctoral.

[†2] Hentenryck [76] por un lado, y Smith *et al.* [178] por otro, ofrecen interesantes comparaciones entre CP e IP desde un punto de vista práctico, incluyendo ejemplos concretos y estudios experimentales.

Característica Avanzada	MA	MO	MI
Expresividad de Objetivos	√	√	√
Simetría de Objetivos	√	√	√
Atributos Controlables y No-Controlables	√	√	√
Uso Conjunto de Atributos Contr. y No-Contr.	√	√	~
Caracterización Estadística de los Atrib. No-Contr.	√	√	~
V.N. con Funciones de Utilidad Complejas	√	√	~
Consciencia Temporal	√	√	~

Tabla 9.1: *Estado de desarrollo de las características avanzadas en HDM.*

Los resultados de esta memoria de tesis no pueden ser sino la línea de comienzo de nuevas e interesantes líneas de investigación, que permiten ampliar el abanico de posibilidades del emparejamiento de servicios. Estas líneas de trabajo futuro se exponen a continuación.

9.3.1. Marco de Trabajo

El modelo abstracto de HDM se ha definido de manera independiente a las notaciones actuales. Aunque esa pueda ser una gran virtud porque proporciona independencia, también puede ser un gran defecto puesto que no hemos tenido en cuenta ninguna propuesta de recomendación como WS-Agreement (aunque se ha tomado como referencia, no se ha incluido en HDM), OASIS u otra similar. Para integrar una recomendación en HDM, hay que añadir un nivel más de abstracción que nos proteja de las variaciones que pudieran haber. Este nivel se denomina *modelo característico*. Por ejemplo, si asumimos WS-Agreement como modelo característico, la figura §9.1 muestra los niveles de abstracción del futuro marco de trabajo de HDM. En este caso particular, hemos de extender WS-Agreement para alcanzar todas las características avanzadas, de manera que hemos de estudiar cómo incorporarlas a la recomendación. En este sentido, ya hemos estudiado la consciencia temporal en el contexto de WS-Agreement [144].

Por otro lado, las propuestas utilizan mayoritariamente una única técnica en el modelo operacional. No obstante, no siempre es posible que la técnica escogida sea capaz de alcanzar o cubrir completamente el modelo abstracto. Así pues, se plantea la necesidad de un *soporte multi-paradigma* para permitir la integración de diferentes modelos operacionales, cada uno dedicado a un aspecto para que se pueda ampliar el alcance global de la propuesta.

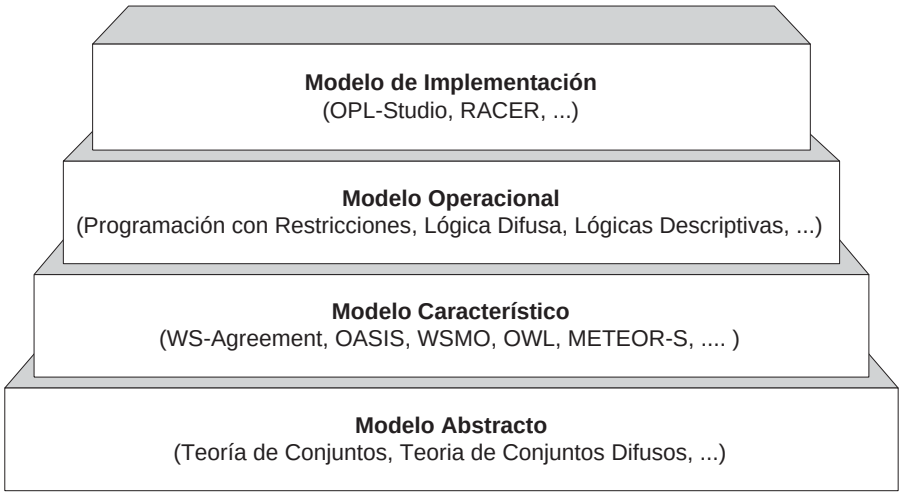


Figura 9.1: *Niveles de abstracción de HDM en propuestas futuras.*

Por ejemplo, podemos extender el modelo de abstracción para tener en cuenta el emparejamiento funcional. El modelo operacional más adecuado pensamos que debe ser un emparejador semántico basado en lógicas descriptivas. Para integrar los resultados del razonamiento semántico y la programación con restricciones (el soporte que proponemos para el emparejamiento extra-funcional), necesitamos un soporte multi-paradigma que puede ser proporcionado por razonadores híbridos tales como se proponen en [15, 73].

9.3.2. Nuevas Características Avanzadas

Hemos identificado nuevas características para el modelo de emparejamiento avanzado. Algunas de ellas se tratan de modificaciones de las existentes para incorporar aspectos novedosos que no hemos tenido en cuenta en esta tesis doctoral:

- *Emparejamiento con Grados de Conformidad*. En nuestro modelo abstracto, la noción de conformidad es rígida, es decir, la respuesta es “sí” o “no” hay conformidad entre dos ofertas de acuerdo. Esta noción puede resultar demasiado rígida y podemos necesitar flexibilizarla cuando, por ejemplo, no encontremos ofertas de acuerdo que sean conformes. Independientemente del modelo abstracto que pudiéramos tener [58, 113], el modelo operacional puede estar basado en:

- Figuradamente, los criterios de preferencias infieren unas curvas o líneas de iso-calidad sobre las regiones de acuerdos, que mostramos en la figura §9.2. La comprobación de la conformidad puede interpretarse como un *problema de satisfacción de múltiples objetivos* [85] que puede resolverse mediante *programación por metas* [35, 179]. En este contexto, se utilizan las líneas de iso-calidad para establecer unos valores umbrales o niveles mínimos de aspiración.
- Los grados de conformidad también pueden abordarse con diferentes propuestas avanzadas de programación con restricciones, que enumeramos: *restricciones ponderadas* [61, 175], *restricciones probabilísticas* [55], *restricciones posibilísticas* [172] basadas en lógica de posibilidades [211], *restricciones difusas* [48] basadas en la teoría de conjuntos difusos [49], *restricciones parciales* [61], *restricciones con valoración* [173], *restricciones jerarquizadas* [31] y *restricciones blandas* [29] basadas en estructuras de semi-anillo. Todas ellas suponen nuevas fronteras a explorar para nuestro modelo operacional.

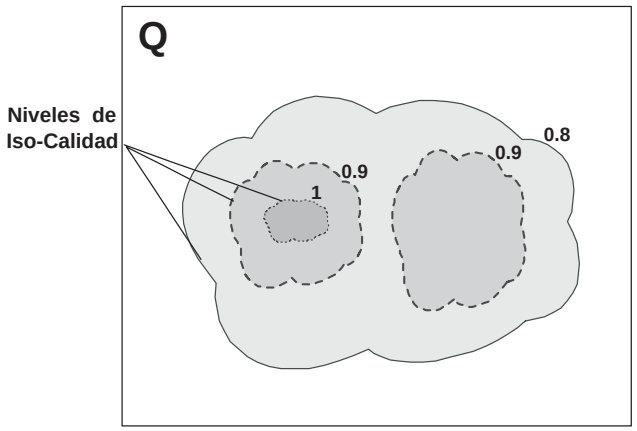


Figura 9.2: *Interpretación geométrica con niveles de iso-calidad.*

- *Detección y Reparación Automática de Inconsistencias.* En nuestro modelo abstracto hemos tenido en cuenta numerosos casos de inconsistencia. Sin embargo, no hemos propuesto ninguna técnica para detectar exactamente qué terminos es el causante de la inconsistencia, ni tampoco cómo se puede corregir la inconsistencia. En este caso, la diagnosis de sistemas parece el paradigma para abordar la detección de inconsistencias. En este sentido, nuestro grupo de investigación ya ha dado los primeros pasos para el análisis de inconsistencias en modelos de características [191].

- *Controlabilidad de los Objetivos*. En nuestro modelo abstracto, hemos considerado que los atributos de calidad pueden ser controlables o no-controlables. Aunque esta característica es un paso adelante en el estado del arte, puede extenderse la noción de controlabilidad hacia los objetivos, para tener en cuenta que sean los propios objetivos los que sean controlables, pues puede darse el caso de que un atributo sea controlable en un objetivo, y no-controlable en otro. Hay que estudiar y adaptar el modelo abstracto del problema para tener en cuenta la controlabilidad los objetivos, así como estudiar la técnica en el modelo operacional más adecuada para realizarla.

9.3.3. Integración con FAST

En cuanto a la integración de HDM en plataformas basadas en arquitecturas orientadas a servicios, cuestión importante para su puesta en producción por la industria, estamos trabajando su integración en el contexto que proporciona FAST (*Framework for Automating Services Trading*) [56]. FAST[†3] tiene como objetivo proporcionar un marco de trabajo flexible para crear sistemas de intermediación de servicios. Uno de sus componentes es un tomador de decisiones, papel que puede asumir HDM.

Resumimos algunas de las características avanzadas que el contexto de FAST proporciona o puede ayudar a proporcionar:

- *Composición de Servicios*. HDM tiene una visión exclusivamente local de la optimalidad. Esta visión es insuficiente cuando hay que afrontar el problema de encontrar la mejor combinación de ofertas para múltiples demandas que optimizaran criterios globales o estratégicos tales como minimizar el número de proveedores.

- *Interoperabilidad Semántica*. HDM asume la conformidad entre dos ofertas de acuerdo basadas en el mismo conjunto de atributos. Sin embargo, hay problemas si las definiciones de los atributos son diferentes, en cuyo caso se pueden estudiar equivalencias o correspondencias, como se propone en [166]. Estas situaciones sólo son una pequeña parte del problema de la *interoperabilidad semántica* en el tratamiento de información procedente de fuentes heterogéneas, basadas en diferentes ontologías pero relativas al mismo dominio, para encontrar una ontología de consenso [180].

[†3] http://www.factories.us.es

- *Estudios Empíricos.* Hemos realizado estudios empíricos para obtener los rangos de tratabilidad para resolver los CSP y CSOP necesarios para llevar a cabo las tareas de emparejamiento. Sin embargo, estos experimentos sólo han abordado algunas de las características avanzadas, a saber, expresividad y simetría de los objetivos. Una vez realizada la integración de HDM en FAST, hay que repetir los experimentos para ratificar los resultados de los estudios analíticos que hemos realizado sobre los algoritmos utilizados para la controlabilidad y consciencia temporal. Por otro lado, también se necesita el diseño de *benchmarks* que, de manera automática, sea capaz de ejecutar experimentos sobre un prototipo de emparejador y nos permita clasificarlo según las características que satisface.

9.3.4. Semantizar HDM

El campo de nuestra propuesta es la Web Tradicional. Sin embargo, no hay que negar que los servicios web semánticos constituyen una línea de investigación que ha tenido un gran grado de desarrollo en los últimos años. De hecho, en el capítulo §5 ya hemos incorporado en nuestro estudio comparativo las propuestas de la Web Semántica y hemos mostrado que suelen alcanzar a la mayor parte de las características avanzadas.

Una futura extensión de nuestra propuesta es semantizar HDM, esto es, asumir en el futuro nivel característico propuestas de la Web Semántica tales como WSMO, OWL-S o METEOR-S. Ello implica la utilización de modelos operacionales basados en, por ejemplo, lógicas descriptivas. Ya que estas técnicas no son capaces de alcanzar una gran expresividad, necesitamos resolutores de restricciones para superar esta limitación. Así pues, necesitamos un soporte multi-paradigma para incorporar los resultados de los razonadores de lógicas descriptivas o motores de inferencia lógicas con los resultados de los resolutores de restricciones. Los primeros pasos para estudiar este problema ya lo hemos dado y los primeros resultados se han presentado en [66].

Al final de mi tesis doctoral, me gustaría concluir con una cita algo más extensa del séptimo arte, espléndido reflejo de lo que podría ser el objetivo transcendental de todo trabajo de investigación:

[...] We all want to help one another. Human beings are like that.
[...] The way of life can be free and beautiful, but we have lost the way.
[...] Machinery that gives abundance has left us in want.
Our knowledge has made us cynical;
our cleverness, hard and unkind.
We think too much and feel too little.
More than machinery, we need humanity.
More than cleverness, we need kindness and gentleness.
Without these qualities, life will be violent and all will be lost.
The airplane and the radio have brought us closer together.
The very nature of these inventions cries out for the goodness in men;
cries out for universal brotherhood;
for the unity of us all.
Even now my voice is reaching millions throughout the world,
millions of despairing men, women, and little children,
victims of a system that makes men torture and imprison innocent people.
[...] It is written that the Kingdom of God is within man,
not one man nor a group of men,
but in all men! In you!
You, the people, have the power,
the power to create machines,
the power to create happiness!
You, the people, have the power to make this life free and beautiful,
to make this life a wonderful adventure.
Then in the name of democracy, let us use that power.
Let us all unite. Let us fight for a new world,
a decent world that will give men a chance to work,
that will give youth a future and old age a security.
[...] Let us fight to free the world!
To do away with national barriers!
To do away with greed, with hate and intolerance!
Let us fight for a world of reason, a world where science
and progress will lead to the happiness of us all.

Charles Chaplin, The Jewish Barber
"The Great Dictator", 1940

Parte V

Apéndices

Apéndice A

Drammatis Formulae

En este apéndice se presentan unas tablas-resumen de las notaciones que se han utilizado a lo largo de la presente memoria de tesis. En primer lugar, la sección §A.1 presenta las notaciones correspondientes a la programación con restricciones. Posteriormente, la sección §A.2 presenta las notaciones del emparejamiento de servicios.

A.1. Programación con Restricciones

Notación y Concepto	
V	Conjunto de variables $V = \{v_1, \ldots, v_n\}$
V_1, V_2	Subconjuntos de variables, conforman una partición de V
D	Dominio del espacio de soluciones $D = d_1 \times \ldots \times d_n$
$d_1, \ldots, d_n$	Dominios de las variables, tal que d_i es el dominio de v_i
$\nu = (\nu_1, \ldots, \nu_n)$	Asignación en el dominio D
$\nu \in D$	Asignación que pertenece al espacio de soluciones
ν_1, ν_2	Asignaciones parciales respecto a subconjuntos V_1 y V_2
C	Restricción
$C_1, \ldots, C_m$	Conjunción de restricciones $C_1 \wedge \ldots \wedge C_m$
$C(\nu)$	La asignación ν satisface la restricción C
$\neg C(\nu)$	La asignación ν no satisface la restricción C
ψ	CSP, problema de satisfacción de restricciones
(V, D, C)	Tupla de un CSP
$\mathrm{sol}(\psi)$ $\mathrm{sol}(C)$	Espacio de soluciones del CSP, o su restricción
$\mathrm{sat}(\psi)$ $\mathrm{sat}(C)$	Satisfabilidad del CSP, o su restricción
s	Solución de un CSP (asignación que satisface la restricción)
s_1, s_2	Soluciones parciales respecto subconjuntos V_1 y V_2
$\Pi_{V_1}(\psi)$ $\Pi(\psi, V_1)$	Proyección del CSP respecto al conjunto de variables V_1
$\Pi_{V_1}(C)$ $\Pi(C, V_1)$	Proyección de la restricción respecto al conjunto de variables V_1
F	Función-objetivo del problema de optimación
$\mathrm{sol}(\psi, F)$	Conjunto de soluciones (óptimas) del CSOP
$\mathrm{sol}(C, F)$	Conjunto de soluciones (óptimas) de la restricción
$\mathrm{val}(\psi, F)$	Valoración de las soluciones (óptimas) del CSOP
$\mathrm{val}(C, F)$	Valoración de las soluciones (óptimas) de la restricción

A.2. Modelo del Emparejamiento de Servicios

A.2.1. Definición de Ofertas de Acuerdo

Notación y Concepto	
(A, D, P, R, E)	Tupla de una oferta de acuerdo
A	Conjunto de atributos de calidad $A = \{a_1, \ldots, a_n\}$
A_1, A_2	Subconjuntos de atributos que conforman una partición de A
D	Dominio de los puntos de acuerdo posibles $D = d_1 \times \ldots \times d_n$
$d_1, \ldots, d_n$	Dominios de los atributos, tal que d_i es el dominio de a_i
$v \quad v_1 \quad v_2$	Asignaciones de valores a los atributos en A, A_1 o A_2 en los dominios D, D_1 o D_2
$q = (v_1, \ldots, v_n)$	Punto de acuerdo, dado por asignaciones a los valores en el dominio D
$A_M \quad A_N$	Subconjuntos de atributos controlables y no-controlables
$D_M \quad D_N$	Dominios de los subconjuntos A_M y A_N
$q = (z, x)$	Punto de acuerdo: valores controlables y valores no-controlables
T	Período de vigencia de la oferta de acuerdo
τ	Atributo del tiempo
$\tau_0 \quad \tau_1$	Instantes temporales de un período de vigencia
$q = (\tau_0, v_1, \ldots, v_n)$ $q = (\tau_0, v)$ q_{τ_0} $q(\tau_0)$	Un punto de acuerdo posible en el instante τ_0
Q	Región de puntos de acuerdo posibles, o región de acuerdos
P	Predicado que define el conjunto Q
$P_R \quad P_G$	Predicados de requisitos y garantías
R	Criterios de preferencias de la oferta de acuerdo
$q_1 \preceq_{R_{\tau_0}} q_2$ $q_1(\tau_0) \preceq_R q_2(\tau_0)$ $(\tau_0, v_1) \preceq_R (\tau_0, v_2)$	Relación de orden entre puntos de acuerdo posibles en τ_0
E	Probabilidades de ocurrencia de atributos no-controlables en objetivos garantizados
$E(q_{\tau_0}) \quad E_{q_{\tau_0}} \quad E_{q(\tau_0)}$	Probabilidad de un punto de acuerdo en el instante τ_0

A.2.2. Operaciones sobre Ofertas de Acuerdo

Notación y Concepto	
α	Oferta de acuerdo
T_α	Período de vigencia de la oferta de acuerdo
$Q^\alpha \quad Q(\alpha)$ $M^\alpha \quad M(\alpha)$ $N^\alpha(z) \quad N(\alpha, z)$	Región de puntos de acuerdo posibles Región de puntos de acuerdo controlables Región no-controlable ligada a un punto de acuerdo controlable z
$Q^\alpha(\tau_0) \quad Q(\alpha, \tau_0)$ $M^\alpha(\tau_0) \quad M(\alpha, \tau_0)$ $N^\alpha(\tau_0, z) \quad N(\alpha, \tau_0, z)$	Región de puntos de acuerdo posibles en τ_0 Región de puntos de acuerdo controlables en τ_0 Región no-controlable ligada a un punto de acuerdo controlable z en τ_0
$\Phi(\alpha, P') \quad \Phi(Q, P')$ $\alpha(P') \quad Q(P')$	Filtro de la oferta de acuerdo, o su región de puntos de acuerdo posibles, por el predicado P' Representaciones alternativas
$\Phi(\alpha, A' = q) \quad \Phi(Q, A' = q)$ $\alpha(A' = q) \quad Q(A' = q)$	Filtro dado por un punto de acuerdo definido sobre un subconjunto de atributos en A' Representaciones alternativas
$\Phi(\alpha, \tau = \tau_0) \quad \Phi(Q, \tau = \tau_0)$	Filtro temporal de la oferta de acuerdo (obtener los puntos de acuerdo posibles que están vigentes en el instante τ_0)
$\Pi(\alpha, A') \quad \Pi(Q, A')$	Proyección de la oferta de acuerdo, o su conjunto de valores, por el subconjunto de atributos A'
$\alpha(\tau) \quad Q(\tau)$	Proyección temporal de la oferta de acuerdo

A.2.3. Tareas del Emparejamiento de Servicios

Notación y Concepto	
$\alpha \quad \omega \quad \delta$	Ofertas de acuerdo: ofertas y demandas
$T_\alpha \quad T_\omega \quad T_\delta$	Períodos de vigencia de ofertas de acuerdo
$\text{consistente}(\alpha)$	Consistencia de una oferta de acuerdo
$\text{conforme}(\omega, \delta)$	Conformidad de la oferta a la demanda
$P_C^{\omega,\delta}(\tau_0, z)$	Predicado de conformidad para punto de acuerdo controlable z en el instante τ_0
$Q_C^{\omega,\delta}(\tau_0)$	Región de puntos de acuerdo controlables y conformes en el instante τ_0
$N_C^{\omega,\delta}(\tau_0, z)$	Región no-controlable ligada a un punto de acuerdo z controlable y conforme en el instante τ_0
$\mathcal{U}(\omega, \delta)$	Utilidad de un acuerdo (de emparejamiento) entre ω y δ
$\mathcal{U}(\tau_0, \omega, \delta)$	Utilidad de un acuerdo (de emparejamiento) entre ω y δ en el instante τ_0
$U^{\delta/\omega}(\tau_0, z)$	Utilidad de un punto de acuerdo controlable z en τ_0 respecto los criterios de preferencia de δ mientras valoran garantías dadas por ω
$U^{\omega/\delta}(\tau_0, z)$	Utilidad de un punto de acuerdo controlable z en τ_0 respecto los criterios de preferencia de ω mientras valoran garantías dadas por δ
$U(\tau_0, z, x)$	Utilidad de un punto de acuerdo $q = (z, x)$ en el instante τ_0
$\mathcal{P}(\tau_0, z, x)$	Probabilidad de ocurrencias de un punto de acuerdo $q = (z, x)$ en el instante τ_0

Apéndice B

Algoritmo de Proyección Temporal

En este apéndice presentamos la definición rigurosa de nuestro algoritmo de proyección temporal, utilizado tanto para realizar el aplanamiento de ofertas de acuerdo, como para obtener la secuencia de recorrido paralelo de ofertas de acuerdo.

Este apéndice se estructura como sigue. En la sección §B.1 hacemos una introducción a la proyección temporal. A continuación, en la sección §B.2 presentamos la definición rigurosa del algoritmo. Posteriormente, en las secciones §B.3 y §B.4 demostramos las propiedades del algoritmo y su complejidad computacional, respectivamente.

B.1. Introducción

Una secuencia temporal está formada por ítemes ordenados temporalmente, de manera que cada ítem contiene un elemento asociado a un intervalo temporal. Si los intervalos temporales no tienen solapamientos, decimos que es una secuencia no-ambigua. El algoritmo de proyección temporal toma dos secuencias para obtener una nueva, donde cada uno de sus ítemes contiene los elementos de las secuencias entrantes que están vigentes durante un mismo intervalo temporal, de manera que la secuencia sigue siendo no-ambigua.

Nuestro algoritmo de proyección temporal está inspirado en el operador temporal de unión de Böhlen *et al.* [26], que extendemos para tener en cuenta todas las relaciones entre intervalos temporales, según fueron definidas por Allen [7, 8]. Hasta lo que conocemos, no hay propuesta similar que aborde el problema desde nuestro punto de vista.

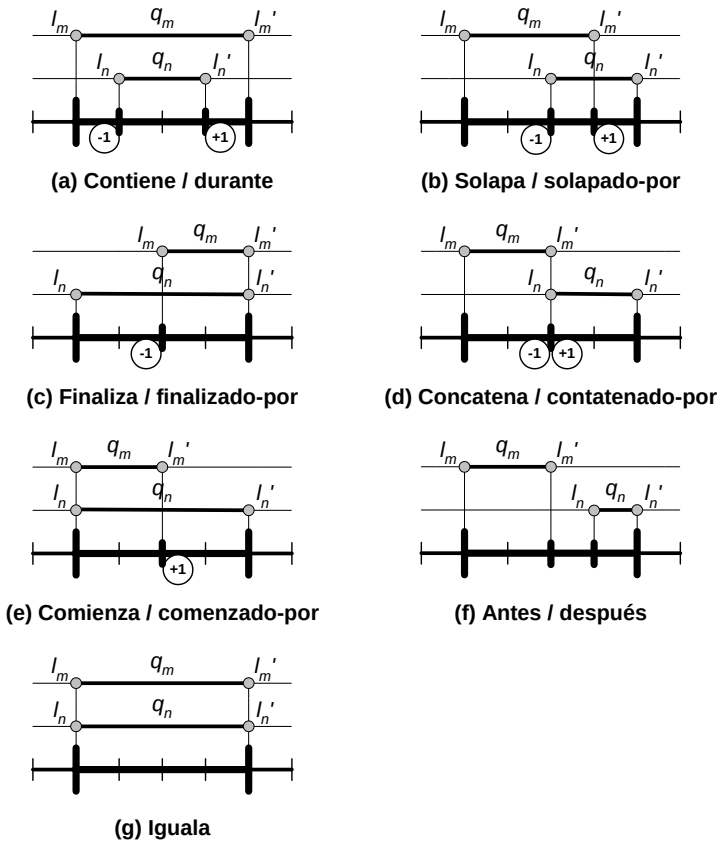


Figura B.1: *Relaciones primitivas entre intervalos temporales de Allen.*

B.2. Definición del Algoritmo

La semántica operacional del algoritmo de proyección temporal esta inspirada en el enfoque estructural basado en sistemas de transiciones propuesto por Plotkin [161].

B.2.1. Estructuras de Datos

Sea una secuencia temporal de n ítemes $D = \langle c_1, \ldots, c_n \rangle$, de manera que el i-ésimo ítem c_i viene dado por una tupla (q_i, I_i) donde q_i representa un elemento asociado al i-ésimo intervalo temporal $I_i = [l_i \,..\, l'_i]$ con $l_i \leq l'_i$. Destacamos dos propiedades de las secuencias temporales:

- Una secuencia temporal D se considera que es no-ambigua si y sólo si no hay solapamientos entre los intervalos temporales de sus ítemes:

$$\begin{aligned} \text{no}-\text{ambigüedad}(D) \Longleftrightarrow \\ \forall i, j \in [1 \,..\, n] \cdot\, i \neq j \Rightarrow I_i \textbf{ antes } I_j \vee I_j \textbf{ antes } I_i \\ \wedge\, i = j \Rightarrow I_i \textbf{ igual a } I_j \end{aligned}$$

- Una secuencia temporal D está ordenada temporalmente si y sólo si los intervalos temporales de sus ítemes están ordenadas de manera ascendente en el tiempo:

$$\text{ordenado}(D) \Longleftrightarrow \forall i \in [1 \,..\, n-1] \cdot I_i \textbf{ antes } I_{i+1}$$

Los predicados correspondientes a las relaciones entre intervalos temporales definidos por Allen [7, 8] se ilustran en la figura §B.1.

B.2.2. Transiciones y Configuraciones

Sea el sistema de transiciones dado por la tupla $(\Gamma, T, \longrightarrow_{PT})$ donde Γ representa al conjunto de configuraciones, T al subconjunto de configuraciones terminales y $\longrightarrow_{PT}$ al conjunto de transiciones entre las configuraciones. Éstas vienen dadas por tuplas de la forma $\gamma = (M, N, R)$ donde M, N y R son secuencias temporales.

Definimos el algoritmo de proyección temporal como una relación de equivalencia, de manera que R es la secuencia temporal que resulta de "proyectar

temporalmente" las secuencias temporales M y N si y sólo si existe una única traza finita de configuraciones en el sistema de transiciones que, partiendo de una configuración inicial de la forma $\gamma_I = (M, N, \langle\rangle)$, llegamos a una configuración terminal $\gamma_F = (\langle\rangle, \langle\rangle, R)$, es decir:

$$\text{proyTemporal}(M, N) = R \quad \equiv \quad (M, N, \langle\rangle) \longrightarrow^*_{PT} (\langle\rangle, \langle\rangle, R)$$

Entrada Ambas secuencias temporales M y N son no-ambiguas y están ordenadas temporalmente:

$$\text{no}-\text{ambigüedad}(M) \wedge \text{no}-\text{ambigüedad}(N) \wedge \\ \text{ordenado}(M) \wedge \text{ordenado}(N)$$

Salida La secuencia temporal R resultante de la proyección temporal también es no-ambigua y está ordenada temporalmente:

$$\text{no}-\text{ambigüedad}(R) \wedge \text{ordenado}(R)$$

B.2.3. Transiciones debidas a los Intervalos Temporales

El conjunto de transiciones viene dado por las relaciones entre intervalos temporales, cuyas primitivas han sido definidas por Allen [7, 8], que mostramos en la figura §B.1. En principio, sólo se necesitan definir siete reglas de transición, puesto que el resto de reglas se corresponden a las denominadas relaciones inversas:

$$I_m \textbf{ contiene } I_n \equiv I_n \textbf{ durante } I_m$$
$$I_m \textbf{ finaliza } I_n \equiv I_n \textbf{ finalizado-por } I_m$$
$$I_m \textbf{ comienza } I_n \equiv I_n \textbf{ comenzado-por } I_m$$
$$I_m \textbf{ solapa } I_n \equiv I_n \textbf{ solapado-por } I_m$$
$$I_m \textbf{ concatena } I_n \equiv I_n \textbf{ concatenado-por } I_m$$
$$I_m \textbf{ antes } I_n \equiv I_n \textbf{ después } I_m$$

Sean $M = \langle m|M'\rangle$ y $N = \langle n|N'\rangle$ dos secuencias temporales, donde m y n representan a sus primeros ítemes, y M' y N' al resto de sus ítemes, respectivamente. Los elementos asociados a los ítemes m y n se denotan como q_m y q_n, y los intervalos temporales asociados como $I_m = [l_m \,..\, l'_m]$ y $I_n = [l_n \,..\, l'_n]$, respectivamente.

En cada paso se comprueba la relación entre los intervalos temporales de los primeros ítemes m y n, definiéndose las siguientes reglas de transición:

1. Si los intervalos temporales I_m y I_n **son iguales** entonces tenemos la situación mostrada en la figura §B.1(g). En este caso, se añade a la secuencia temporal R un nuevo ítem cuyo intervalo temporal es I_m y su elemento asociado incluye q_m y q_n:

$$\frac{I_m \ \textbf{iguala} \ I_n}{(M, N, R) \longrightarrow_{PT} (M', N', R')}$$

   donde $R' = R \oplus \langle q_m \otimes q_n, [l_m \,..\, l'_m] \rangle$.

   Nótese que el operador $\oplus$ representa la operación de añadir un nuevo ítem a la secuencia temporal, y el operador $\otimes$ representa una operación genérica que se aplica para agrupar sendos elementos asociados.

2. Si el intervalo temporal I_m **contiene** al intervalo temporal I_n entonces tenemos la situación mostrada en la figura §B.1(a). En este caso, se añade a la secuencia temporal R sendos ítemes de manera que **(1)** el intervalo temporal del primer ítem es $[l_m \,..\, l_n - 1]$[†1] y su elemento asociado incluye q_m, y **(2)** el intervalo temporal del segundo ítem es $[l_n \,..\, l'_n]$ y su elemento asociado incluye q_m y q_n:

$$\frac{I_m \ \textbf{contiene} \ I_n}{(M, N, R) \longrightarrow_{PT} (\langle q_m, [l'_n + 1 \,..\, l'_m] \rangle \oplus M', N', R' \rangle}$$

   donde $R' = R \oplus \langle q_m, [l_m \,..\, l_n - 1] \rangle \oplus \langle q_m \otimes q_n, [l_n \,..\, l'_n] \rangle$.

   Nótese que aún queda un remanente correspondiente al primer ítem de la secuencia M, que habrá de tenerse en cuenta en la siguiente transición anteponiendo a M un ítem cuyo intervalo temporal es $[l'_n + 1 \,..\, l'_m]$ y su elemento asociado incluye q_m.

3. Si el intervalo temporal I_m tiene lugar **durante** el intervalo temporal I_n entonces tenemos la situación mostrada en la figura §B.1(a). Esta situación es equivalente a decir que el intervalo temporal I_n *contiene* al intervalo temporal I_m, por lo que podemos plantear una regla de transición al efecto:

$$\frac{I_m \ \textbf{durante} \ I_n}{(M, N, R) \longrightarrow_{PT} (N, M, R)}$$

[†1] Asumiendo que el tiempo tiene un dominio discreto, las expresiones -1 o $+1$ aplicadas a un instante temporal denotan los instantes temporales inmediatamente anterior o siguiente, respectivamente.

4. Si el intervalo temporal I_m **está contenido en y finaliza con** el intervalo temporal I_n entonces tenemos la situación mostrada en la figura §B.1(c). En este caso, se añade a la secuencia temporal R sendos ítemes de manera que **(1)** el intervalo temporal del primer ítem es $[l_n \,..\, l_m - 1]$ y su elemento asociado incluye q_n, y **(2)** el intervalo temporal del segundo ítem es $[l_m \,..\, l'_m]$ y su elemento asociado incluye q_m y q_n:

$$\frac{I_m \ \textbf{finaliza} \ I_n}{(M, N, R) \longrightarrow_{PT} (M', N', R')}$$

donde $R' = R \oplus \langle q_n, [l_n \,..\, l_m - 1]\rangle \oplus \langle q_m \otimes q_n, [l_m \,..\, l'_m]\rangle$.

5. Si el intervalo temporal I_m **contiene y está finalizado por** el intervalo temporal I_n entonces tenemos la situación mostrada en la figura §B.1(c). Esta situación es equivalente a decir que el intervalo temporal I_n *está contenido en y finaliza con* el intervalo temporal I_m, por lo que podemos plantear una regla de transición al efecto:

$$\frac{I_m \ \textbf{finalizado-por} \ I_n}{(M, N, R) \longrightarrow_{PT} (N, M, R)}$$

6. Si el intervalo temporal I_m **está contenido en y comienza con** el intervalo temporal I_n entonces tenemos la situación mostrada en la figura §B.1(e). En este caso, se añade a la secuencia temporal R un nuevo ítem cuyo intervalo temporal es $[l_m \,..\, l'_m]$ y su elemento asociado incluye q_m y q_n:

$$\frac{I_m \ \textbf{comienza} \ I_n}{(M, N, R) \longrightarrow_{PT} (M', \langle q_n, [l'_m + 1 \,..\, l'_n]\rangle \oplus N', R')}$$

donde $R' = R \oplus \langle q_m \otimes q_n, [l_m \,..\, l'_m]\rangle$.

Nótese que aún queda un remanente correspondiente al primer ítem de la secuencia temporal N, que habrá de tenerse en cuenta en la siguiente transición anteponiendo a N un ítem cuyo intervalo temporal es $[l'_m + 1 \,..\, l'_n]$ y elemento asociado incluye q_n.

7. Si el intervalo temporal I_m **contiene y es comenzado por** el intervalo temporal I_n entonces tenemos la situación mostrada en la figura §B.1(e). Esta situación es equivalente a decir que el intervalo temporal I_n *está contenido en y comienza con* el intervalo temporal I_m, por lo que podemos plantear una regla de transición al efecto:

$$\frac{I_m \ \textbf{comenzado-por} \ I_n}{(M, N, R) \longrightarrow_{PT} (N, M, R)}$$

8. Si el intervalo temporal I_m **es anterior y solapa con** el intervalo temporal I_n entonces tenemos la situación mostrada en la figura §B.1(b). En este caso, se añade a la secuencia temporal R sendos ítemes de manera que **(1)** el intervalo temporal del primer ítem es $[l_m \,..\, l_n - 1]$ y su elemento asociado incluye q_m, y **(2)** el intervalo temporal del segundo ítem es $[l_n \,..\, l'_m]$ y elemento asociado incluye q_m y q_n:

$$\frac{I_m \ \textbf{solapa} \ I_n}{(M, N, R) \longrightarrow_{PT} (M', \langle q_n, [l'_m + 1 \,..\, l'_n]\rangle \oplus N', R')}$$

donde $R' = R \oplus \langle q_m, [l_m \,..\, l_n - 1]\rangle \oplus \langle q_m \otimes q_n, [l_n \,..\, l'_m]\rangle$.

Nótese que aún queda un remanente correspondiente al primer ítem de la secuencia temporal N, que habrá de tenerse en cuenta en la siguiente transición anteponiendo a N un término cuyo intervalo temporal es $[l'_m + 1 \,..\, l'_n]$ y elemento asociado incluye q_n.

9. Si el intervalo temporal I_m **no es anterior y está solapado por** el intervalo temporal I_n entonces tenemos la situación mostrada en la figura §B.1(b). Esta situación es equivalente a decir que el intervalo temporal I_n *es anterior y solapa con* el intervalo temporal I_m, por lo que podemos plantear una regla de transición al efecto:

$$\frac{I_m \ \textbf{solapado-por} \ I_n}{(M, N, R) \longrightarrow_{PT} (N, M, R)}$$

10. Si el intervalo temporal I_m se **concatena con** el intervalo temporal I_n entonces tenemos la situación mostrada en la figura §B.1(d). En este caso, se añade a la secuencia temporal R sendos ítemes de manera que **(1)** el intervalo temporal del primer ítem es $[l_m \,..\, l'_m - 1]$ y su elemento asociado incluye q_m, y **(2)** el intervalo temporal del segundo ítem es $[l'_m \,..\, l'_m]$ y su elemento asociado incluye q_m y q_n:

$$\frac{I_m \ \textbf{concatena} \ I_n}{(M, N, R) \longrightarrow_{PT} (M', \langle [l'_m + 1 \,..\, l'_n], q_n\rangle \oplus N', R')}$$

donde $R' = R \oplus \langle q_m, [l_m \,..\, l'_m - 1]\rangle \oplus \langle q_m \otimes q_n, [l'_m \,..\, l'_m]\rangle$.

Nótese que aún queda un remanente correspondiente al primer ítem de la secuencia temporal N, que habrá de tenerse en cuenta en la siguiente transición anteponiendo a N un ítem cuyo intervalo temporal es $[l'_m + 1 \,..\, l'_n]$ y su elemento asociado incluye q_n.

11. Si el intervalo temporal I_m **está concatenado por** el intervalo temporal I_n entonces tenemos la situación mostrada en la figura §B.1(d). Esta situación se puede expresar justamente al revés, por lo que podemos plantear una regla de transición al efecto:

$$\frac{I_m \ \textbf{concatenado-por} \ I_n}{(M, N, R) \longrightarrow_{PT} (N, M, R)}$$

12. Si el intervalo temporal I_m **es anterior** al intervalo temporal I_n entonces tenemos la situación mostrada en la figura §B.1(f). En este caso, se añade a la secuencia temporal R el ítem m:

$$\frac{I_m \ \textbf{antes} \ I_n}{(M, N, R) \longrightarrow_{PT} (M', N, R')}$$

donde $R' = R \oplus m$.

13. Si el intervalo temporal I_m **es posterior** al intervalo temporal I_n entonces tenemos la situación mostrada en la figura §B.1(f). Esta situación es equivalente a decir que el intervalo temporal I_n *es anterior* al intervalo temporal I_m, por lo que podemos plantear una regla de transición al efecto:

$$\frac{I_m \ \textbf{después} \ I_n}{(M, N, R) \longrightarrow_{PT} (N, M, R)}$$

B.2.4. Transiciones hacia Estados Terminales

Sean M y N sendas secuencias temporales; definimos un par de transiciones hacia el estado terminal, que corresponden a los casos donde alguna de ellas está vacía:

1. Si la secuencia temporal M está vacía, entonces se añaden los ítemes restantes de la secuencia temporal N a R:

$$\frac{}{(\langle\rangle, N, R) \longrightarrow_{PT} (\langle\rangle, \langle\rangle, R \oplus N)}$$

2. Si la secuencia temporal N está vacía, entonces se añaden los ítemes restantes a la secuencia temporal M a R:

$$\frac{}{(M, \langle\rangle, R) \longrightarrow_{PT} (\langle\rangle, \langle\rangle, R \oplus M)}$$

B.3. Propiedades del Algoritmo

La definición rigurosa facilita la demostración de que el algoritmo es determinista, correcto y que tiene una complejidad computacional de orden lineal.

Hemos definido una regla de transición para cada tipo de relación entre intervalos temporales. El conjunto de relaciones entre intervalos temporales es completo, es decir, incluye a todas las posibles relaciones entre intervalos [7, 8]. Por ello, podemos decir que *siempre* es posible aplicar una regla de transición, según el tipo de relación que se trate en cada caso.

B.3.1. Determinista

Hay que demostrar la propiedad de que *para todo par de secuencias temporales, la secuencia temporal resultante es única*:

$$\begin{aligned}\forall M, N\ \exists R, R' \ \cdot\ & \text{proyTemporal}(M, N) = R \\ & \wedge\ \text{proyTemporal}(M, N) = R' \Longrightarrow R = R'\end{aligned}$$

Demostración Por *reductio ad absurdum*, vamos a asumir que existe un par de secuencias temporales M y N para los cuáles hay sendas secuencias temporales R y R' que son resultantes:

$$\begin{aligned}(M, N, \langle\rangle) &\longrightarrow^*_{PT} (\langle\rangle, \langle\rangle, R) \\ (M, N, \langle\rangle) &\longrightarrow^*_{PT} (\langle\rangle, \langle\rangle, R')\end{aligned}$$

Esto significa que existe alguna configuración γ_k a partir de la cuál se diferencian los caminos, uno hacia R y otro hacia R':

$$(M, N, \langle\rangle) \longrightarrow_{PT} \gamma_1 \longrightarrow_{PT} \ldots \longrightarrow_{PT} \gamma_k \begin{cases} \nearrow\ \gamma_{k+1} \longrightarrow_{PT} \gamma_n \longrightarrow_{PT} (\langle\rangle, \langle\rangle, R) \\ \searrow\ \gamma'_{k+1} \longrightarrow_{PT} \gamma'_n \longrightarrow_{PT} (\langle\rangle, \langle\rangle, R') \end{cases}$$

Pero ello contradice el hecho de que siempre haya una única regla a aplicar en cada transición, por lo que podemos concluir que siempre es posible llegar a una única secuencia temporal R en todos los casos y, por tanto, el algoritmo de proyección temporal es determinista. ■

B.3.2. Corrección

Hay que demostrar la propiedad de que *para todo par de secuencias temporales, ambas no-ambiguas y ordenadas temporalmente, la única secuencia temporal resultante también es no-ambigua y ordenada temporalmente*:

$$\forall M, N\ \exists R \cdot \text{proyTemporal}(M, N) = R \land \\ \text{no}-\text{ambigüedad}(M) \land \text{no}-\text{ambigüedad}(N) \land \\ \text{ordenado}(M) \land \text{ordenado}(N) \implies \text{no}-\text{ambigüedad}(R) \land \text{ordenado}(R)$$

Demostración Por *reductio ad absurdum*, vamos a asumir que existe un par de secuencias temporales M y N que satisfacen las propiedades requeridas, para las cuáles la secuencia de recorrido resultante R no las satisface. Hay sendas causas posibles, R es ambigua o no está ordenada temporalmente:

- Para que R sea ambigua, tiene que haber solapamientos entre los intervalos temporales de sus ítemes. Sin embargo, esta circunstancia no es posible porque las reglas de transición aplicables lo evitan por construcción: si hay que dividir un intervalo temporal, no se permite que el instante divisorio pertenezca a sendos intervalos temporales resultantes.
- Para que R no esté ordenada temporalmente, significa que existe un par de intervalos temporales consecutivos que no lo están. Sin embargo, esta circunstancia no es posible porque, por construcción, siempre iteramos ambas secuencias temporales en orden temporal ascendente y así establecemos todos los intervalos temporales de la secuencia temporal resultante.

Por tanto, en ambos casos llegamos a una contradicción, por lo que podemos concluir que dadas sendas secuencias temporales no-ambiguas y ordenadas temporalmente, la secuencia temporal resultante también es no-ambigua y está ordenada temporalmente, por lo que podemos concluir que el algoritmo de proyección temporal es correcto. ■

B.4. Complejidad del Algoritmo

En primer lugar determinamos la cota inferior y superior del número de pasos para obtener el resultado de la proyección. Hay que demostrar la propiedad de que *para todo par de secuencias temporales, la secuencia temporal resultante se obtiene en un número finito de pasos*:

$$(M, N, \langle\rangle) \longrightarrow_{PT} \gamma_1 \longrightarrow_{PT} \gamma_2 \longrightarrow_{PT} \ldots \longrightarrow_{PT} \gamma_n \longrightarrow_{PT} (\langle\rangle, \langle\rangle, R)$$
$$n < \infty$$

Demostración Sea M y N sendas secuencias temporales, de manera que $|M|$ y $|N|$ denotan el número de ítemes que las constituyen. Sea n el número de pasos para obtener la secuencia temporal resultante. Para determinar n, estudiamos los siguientes casos límite:

- En el mejor de los casos, efectuamos tantos pasos como número de ítemes tenga la menor secuencia temporal, más un último paso adicional que añade a la secuencia temporal resultante aquellos ítemes de la secuencia temporal mayor que aún no han sido procesados.
- En el peor de los casos, puede ocurrir que a cada paso se aplique alguna de las reglas en las que queda remanente de un ítem por procesar. Estos casos corresponden a las relaciones temporales **contiene**, **comienza**, **solapa** y **concatena**; y sus inversas **durante**, **comenzado-por**, **solapado-por**, **concatenado-por**. De esta manera, puede doblarse el número de pasos que finalmente tiene el recorrido.

De esta manera, podemos establecer que el número de pasos para obtener R está limitado por una cota inferior y otra superior:

$$\text{mín}\{|M|, |N|\} + 1 \leq n \leq 2 \times \text{máx}\{|M|, |N|\} + 1$$

Por lo que podemos concluir que el algoritmo de proyección temporal siempre termina en un número finito de pasos. ■

Debido al recorrido de ambas secuencia, se puede concluir que el algoritmo de proyección temporal tiene una complejidad computacional de orden lineal $O(N)$, donde N es el número máximo de ítemes de ambas secuencias temporales.

Bibliografía

[1] R. Afandi, J. Zhang y C. Gunter. AMPol-Q: Adaptive Middleware Policy to Support QoS. En *4th Intl. Conf. on Service-Oriented Computing*, vol. 4294 de *LNCS*, págs. 165–178, Chicago, IL, Diciembre 2006. Springer Verlag

[2] R. Aggarwal, K. Verma, J. Miller y W. Milnor. Constraint Driven Web Service Composition in METEOR-S. En *1st IEEE Intl. Conf. on Services Computing*, págs. 23–30, Shanghai, China, Septiembre 2004. IEEE Computer Society

[3] M. Aiello, G. Frankova y D. Malfatti. What's an Agreement? A Formal Analysis and an Extension of WS-AGREEMENT. Informe Técnico DIT-05-039, University of Trento, Abril 2005

[4] M. Aiello, G. Frankova y D. Malfatti. What's in an Agreement? An Analysis and an Extension of WS-Agreement. En *3rd Intl. Conf. on Service-Oriented Computing*, vol. 3826 de *LNCS*, págs. 424–436, Amsterdam, The Netherlands, Diciembre 2005. Springer Verlag

[5] R. Al-Ali, O. Rana, D. Walker, S. Jha y S. Sohail. G-QoSM: Grid Service Discovery using QoS Properties. *Journal of Computing and Informatics*, 21(4):363–382, 2002

[6] A. S. Ali, O. Rana, R. Al-Ali y D. Walker. UDDIe: An Extended Registry for Web Services. En *IEEE Intl. Workshop on Service Oriented Computing: Models, Architectures and Applications at SAINT Conference*. IEEE Press, Enero 2003

[7] J. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, Noviembre 1983

[8] J. Allen. Time and Time Again: The Many Ways to Represent Time. *International Journal of Intelligent Systems*, 6(4):341–355, 1991

[9] M. Andrea-Rodríguez y M. Egenhofer. Determining Semantic Similarity among Entity Classes from Different Ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):442–456, Febrero 2003

[10] A. Andrieux, K. Czakowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke y M. Xu. Web Services Agreement Specification (WS-Agreement) Version 1.1 draft 20, Septiembre 2006

[11] K. Apt. *Principles of Constraint Programming*. Cambridge University Press, 2003

[12] K. Apt. Arrays, Bounded Quantification and Iteration in Logic and Constraint Logic Programming. *Science of Computer Programming*, 26 (1–3):133–148, 1996

[13] A. Artale y E. Franconi. A Survey of Temporal Extensions of Description Logics. *Annals of Mathematics and Artificial Intelligence*, 30(1–4):171–210, 2000

[14] F. Baader, D. Calvanese, D. McGuinness, D. Nardi y P. Patel-Schneider. *The Description Logic Handbook*. Cambridge University Press, 2002

[15] F. Baader y P. Hanschke. A Schema for Integrating Concrete Domains into Concept Languages. En *12th Intl. Joint Conference on Artificial Intelligence*, págs. 452–457, Sydney, Australia, Agosto 1991. Morgan Kaufmann

[16] F. Baader, I. Horrocks y U. Sattler. Description Logics as Ontology Languages for the Semantic Web. En *Mechanizing Mathematical Reasoning: Essays in Honor of Jörg Siekmann on the Occasion of His 60th Birthday*, vol. 2605 de *LNAI*, págs. 228–248. Springer Verlag, 2005

[17] F. Baader, C. Lutz, M. Miličić, U. Sattler y F. Wolter. A Description Logic Based Approach to Reasoning about Web Services. En *WWW Workshop on Web Service Semantics*, Chiba City, Japan, Mayo 2005

[18] S. Bajaj, D. Box, D. Chappell, F. Curbera, G. Daniels, P. Hallam-Baker, M. Hondo, C. Kaler, D. Langworthy, A. Nadalin, N. Nagaratnam, H. Prafullchandra, C. von Riegen, D. Roth, J. Schlimmer, C. Sharp, J. Shewchuck, A. Vedamuthu, . Yalçinalp y D. Orchad. Web Services Policy 1.2 – Framework (WS-POLICY), Abril 2006

[19] L. Baresi, S. Guinea y P. Plebani. WS-Policy for Service Monitoring. En *6th Intl. VLDB Workshop Technologies for E-Services*, vol. 3811 de *LNCS*, págs. 72–83, Trondheim, Norway, Septiembre 2005. Springer Verlag

[20] B. Benatallah, M. Hacid, A. Leger, C. Rey y F. Toumani. On Automating Web Services Discovery. *The International Journal on Very Large Data Bases*, 14(1):84–96, Marzo 2005

[21] B. Benatallah, M. Hacid, C. Rey y F. Toumani. Request Rewriting-based Web Service Discovery. En *2nd Intl. Web Semantic Conference*, vol. 2870 de *LNCS*, págs. 242–257, Sanibel Island, FL, Octubre 2003. Springer Verlag

[22] F. Benhamou, F. Goualard, E. Languénou y M. Christie. Universally Quantified Constraint Solving. Informe Técnico 00.5, Institut de Recherche en Informatique de Nantes, France, 2000

[23] T. Berners-Lee, J. Hendler y O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, Mayo 2001

[24] A. Bernstein, E. Kaufmann, C. Bürki y M. Klein. How Similar Is It? Towards Personalized Similarity Measures in Ontologies. En *7th Internationale Tagung Wirtshaftsinformatik 2005: eEconomy, eGovernment, eSociety*, págs. 1347–1366, Bamberg, Germany, Febrero 2005

[25] A. Bernstein y M. Klein. Towards High-Precision Service Retrieval. En *1st Intl. Semantic Web Conference*, vol. 2342 de *LNCS*, págs. 84–101, Sardinia, Italy, Junio 2002. Springer Verlag

[26] M. Böhlen, R. Busatto y C. Jensen. Point versus Interval-based Temporal Data Models. En *14t IEEE Intl. Conf. on Data Engineering*, págs. 192–200, Orlando, FL, Febrero 1998. IEEE Computer Society Press

[27] D. Bianchini, V. de Antonellis, B. Pernici y P. Plebani. Ontology-based Methodology for e-Service Discovery. *Elsevier Information Systems*, 31 (4):361–380, Junio 2006

[28] A. Bilgin y M. Singh. A DAML-Based Repository for QoS-Aware Semantic Web Service Selection. En *2nd Intl. Conf. on Web Services*, págs. 368–375, San Diego, CA, Julio 2004. IEEE Computer Society Press

[29] S. Bistarelli, U. Montanari y F. Rossi. Constraint Solving over Semirings. En *14th Intl. Joint Conference on Artificial Intelligence*, págs. 624–630, Montreal, Canada, 1995. Morgan Kaufmann

[30] L. Bordeaux y E. Monfroy. Beyonf NP: Arc-Consistency for Quantified Constraints. En *8th Int. Conf. on Principles and Practices of Constraint Programming*, vol. 2470 de *LNCS*, págs. 371–386, Ithaca, NY, 2002. Springer Verlag

[31] A. Borning, R. Duisberg, B. Freeman-Benson, K. Kramer y M. Wolf. Constraint Hierarchies. En *2nd Intl. Conf. on Object-Oriented Programming Systems, Languages, and Applications*, págs. 48–60, Orlando, FL, 1987. ACM Press

[32] G. Canfora, M. di Penta, R. Esposito y M. Villani. An Approach for QoS-aware Service Composition based on Genetic Algorithms. En *6th Genetic and Evolutionary Computation Conference*, págs. 1069–1075, Washington, DC, Junio 2005. ACM Press

[33] J. Cardoso y A. Sheth. Semantic E-Workflow Composition. *Journal of Intelligent Information Systems*, 21(3):191–225, Noviembre 2003

[34] F. Casati y M. Shan. Dynamic and Adaptive Composition of E-Services. *Elsevier Information Systems*, 26(3):143–163, Mayo 2001

[35] A. Charnes, W. Cooper y R. Ferguson. Optimal Estimation of Executive Compensation by Linear Programming. *Management Science*, 1(2):138–151, Enero 1955

[36] Y. Chen, Z. Li, Q. Jin y C. Wang. Study on QoS Driven Web Services Composition. En *8th Asia-Pacific Web Conference*, vol. 3841 de *LNCS*, págs. 702–707, Harbin, China, Enero 2006. Springer Verlag

[37] T. O. Coalition. OWL-S: Semantic Markup Language for Web Services, Noviembre 2004

[38] G. Collins. Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition. En *2nd GI Conference on Automata Theory and Formal Languages*, vol. 33 de *LNCS*, págs. 134–183, Kaiserslauten, Germany, 1975. Springer Verlag

[39] S. Colucci, T. di Noia, E. di Sciascio, F. Donini y M. Mongiello. Concept Abduction and Contraction for Semantic-based Discovery of Matches and Negotiation Spaces in an E-Marketplace. *International Journal of Electronic Commerce*, 4(4):345–361, Winter 2005

[40] H. Comon y P. Lescanne. Equational Problems and Disunification. *Journal of Symbolic Computation*, 7(3–4):371–425, 1989

[41] G. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 1963

[42] J. de Bruijn, C. Feier, U. Keller, R. Lara, A. Polleres y L. Predoiu. WSML Reasoning Survey, Noviembre 2005

[43] J. de Bruijn, H. Lausen, R. Krummenacher, A. Polleres, L. Predoiu, M. Kifer y D. Fensel. The Web Service Modeling Language WSML, Octubre 2005

[44] R. de Landtsheer. Solving CSP Including a Universal Quantification. En *2nd Intl. Conf. on Multiparadigm Programming in Mozart/Oz*, vol. 3389 de *LNCS*, págs. 200–210, Charleroi, Belgium, 2005. Springer Verlag

[45] T. di Noia, E. di Sciascio, F. Donini y M. Mongiello. A System for Principled Matchmaking in an Electronic Marketplace. *International Journal of Electronic Commerce*, 8(4):9–37, Summer 2004

[46] G. Dobson, R. Lock y I. Sommerville. QoSOnt: An Ontology for QoS in Service-Centric Systems. En *31st EUROMICRO Conference on Software Engineering and Advanced Applications*, págs. 80–87, Porto, Portugal, Agosto/Septiembre 2005. IEEE Computer Society Press

[47] G. Dobson y A. Sánchez-Macián. Towards Unified QoS/SLA Ontologies. En *3rd IEEE Intl. ICWS/SCC Workshop on Semantic and Dynamic Web Processes*, págs. 169–174, Chicago, IL, Septiembre 2006. IEEE Computer Society Press

[48] D. Dubois, H. Fargier y H. Prade. Propagation and Satisfaction of Flexible Constraints. En *Fuzzy Sets, Neural Networks and Soft Computing*, págs. 166–187. Van Nostrand Reinhold, New York, 1994

[49] D. Dubois y H. Prade. *Readings in Fuzzy Sets for Intelligent Systems*. Morgan Kaufmann, 1993

[50] J. Dujmovic. A Method for Evaluation and Selection of Complex Hardware and Software Systems. En *22nd Intl. Conf. for the Resource Management and Performance Evaluation of Enterprise Computing Systems*, págs. 368–378, 1996

[51] M. Dumas, B. Benatallah, N. Russell y M. Spork. A Configurable Matchmaking Framework for Electronic Marketplaces. *Electronic Commerce Research and Applications*, 3(1):95–106, Spring 2004

[52] B. M. (editor). Policy Core Information Model (PCIM) Extensions, Enero 2003

[53] E. K. (editor). Quality Model for Web Services (WSQM-2.0), Septiembre 2005

[54] B. Farbey y A. Finkelstein. Software Acquisition: a Business Strategy Analysis. En *5th Intl. IEEE Symposium on Requirements Engineering*,

págs. 76–83, Toronto, Canada, Agosto 2001. IEEE Computer Society Press

[55] H. Fargier y J. Lang. Uncertainty in Constraint Satisfaction Problems: A Probabilistic Approach. En *2nd European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, vol. 747 de *LNCS*, págs. 97–104, Granada, Spain, 1993. Springer Verlag

[56] P. Fernández-Montes, M. R.-A. de Reyna y R. Corchuelo-Gil. Towards an Automated Trading Process. *UPGRADE*, 7(5):29–33, Octubre 2006

[57] A. Finkelstein, G. Spanoudakis y M. Ryan. Software Package Requirements and Procurement. En *8th Intl. IEEE Workshop on Software Specification and Design*, págs. 141–145, Schloss Velen, Germany, Marzo 1996. IEEE Press

[58] X. Franch y J. Carvallo. Using Quality Models in Software Package Selection. *IEEE Software*, 20(1):34–41, 2003

[59] E. Freuder. In Persuit of the Holy Grail. *Constraints*, 2(1):57–61, Abril 1997

[60] E. Freuder y M. Wallace. Science and Substance: A Challenge to Software Engineers. *Constraints IEEE Intelligent Systems*, 2000

[61] E. Freuder y R. Wallace. Partial Constraint Satisfaction. *Artificial Intelligence*, 58(1–3):21–70, 1992

[62] T. Frühwirth y S. Abdennadher. *Essentials of Constraint Programming*. Springer, 2003

[63] S. Frølund y J. Koistinen. QML: A Language for Quality of Service Specification. Informe Técnico HPL-98-10, Hewlett–Packard, 1998

[64] S. Frølund y J. Koistinen. Quality-of-Service Specification in Distributed Object Systems. *Distributed Systems Engineering Journal*, 5(4), 1998

[65] A. Gao, D. Yang, S. Tang y M. Zhang. QoS-Driven Web Service Composition with Inter Service Conflicts. En *Proceedings of the 8th Asia-Pacific Web Conference*, vol. 3841 de *LNCS*, págs. 121–132, Harbin, China, Enero 2006. Springer Verlag

[66] J. García, D. Ruiz, A. Ruiz-Cortés, O. Martín-Díaz y M. Resinas. An Hybrid, QoS-Aware Discovery of Semantic Web Services using Constraint Programming. En *5th Intl. Conf. on Service-Oriented Computing*, vol. 4749 de *LNCS*, págs. 69–80, Vienna, Austria, Septiembre 2007. Springer Verlag

[67] R. Garfinkel y G. Nemhauser. *Integer Programming*. John Wiley & Sons, Nueva York, NY, 1972

[68] I. Gent, P. Nightingale y A. Rowley. Encoding Quantified CSPs as Quantified Boolean Formulae. Informe Técnico APES-79-2004, APES Research Group, 2004

[69] K. Golden y J. Frank. Universal Quantification in a Constraint-Based Planner. En *6th Intl. Conf. on AI Planning & Scheduling*, Toulouse, France, 2002

[70] J. González-Castillo, D. Trastour y C. Bartolini. Description Logics for Matchmaking of Services. Informe Técnico HPL-2001-265, Hewlett-Packard, Octubre 2001

[71] D. Gouscos, M. Kalikakis y P. Georgiadis. An Approach to Modeling Web Service QoS and Provision Price. En *1st IEEE/WISE Web Services Quality Workshop*, págs. 121–130, Rome, Italy, 2003. IEEE Computer Society Press

[72] P. Grefen, K. Ludwig y S. Angelov. A Three-Level Framework for Process and Data Management of Complex E-services. *Intl. Journal of Cooperative Information Systems*, 12(4):455–485, Diciembre 2003

[73] V. Haarslev y R. Möller. Practical Reasoning in RACER with a Concrete Domain for Linear Inequations. En *2002 Intl. Workshop on Description Logics*, vol. 53 de *Workshop Proceedings*, págs. 91–98, Toulouse, France, Abril 2002. CEUR.

[74] V. Haarslev y R. Moller. Description of the RACER System and its Applications. En *2001 Intl. Description Logics Workshop*, vol. 49 de *Workshop Proceedings*, Stanford, CA, Agosto 2001. CEUR.

[75] V. Haarslev y R. Moller. RACER: A Core Inference Engine for the Semantic Web. En *2th Intl. ISWC Workshop on Evaluation of Ontology-based Tools*, vol. 87 de *Workshop Proceedings*, Sanibel Island, FL, Octubre 2003. CEUR.

[76] P. Hentenryck. Constraint and Integer Programming in OPL. *Informs Journal on Computing*, 14(4):345–372, 2002

[77] P. Hentenryck y V. Saraswat. Strategic Directions in Constraint Programming. *ACM Computing Surveys*, 28(4), Diciembre 1996

[78] P. V. Hentenryck. *The OPL Optimization Programming Language*. The MIT Press, Cambridge, MA, 1999

[79] M. Hepp. Semantic Web and Semantic Web Services: Father and Son or Iindivisible Twins? *IEEE Internet Computing*, 10(2):85–88, Marzo / Abril 2006

[80] J. Hobbs y F. Pang. Time Ontology in OWL, Septiembre 2006

[81] J. Hobbs y J. Pustejovsky. Annotating and Reasoning about Time and Events. En *AAAI Spring Symposium on Logical Formalization of Commonsense Reasoning*, Stanford, CA, Marzo 2003

[82] J. Hobbs y J. Pustejovsky. An Ontology of Time for the Semantic Web. *ACM Transactions on Asian Language Processing, Special Issue on Temporal Information Processing*, 3(1):66–85, Marzo 2004

[83] K. Hoffman, S. Michaylov y P. Stuckey. LP-based Combinatorial Problem Solving. *Annals of Operations Research*, 4:145–194, 1985

[84] Y. Hoffner, S. Field, P. Grefen y H. Ludwig. Contract-driven Creation and Operation of Virtual Enterprises. *Computer Networks*, 37(2):111–136, Octubre 2001

[85] C. Holt, F. Modigliani y H. Simon. A Linear Decision Rule for Production and Employment Scheduling. *Management Science*, 2(1):1–30, Octubre 1955

[86] I. Horrocks. Using an Expressive Description Logic: FaCT or Fiction? En *6th Intl. Conf. on Principles of Knowledge Representation and Reasoning*, págs. 636–647, Trento, Italy, Junio 1998. Morgan Kaufmann

[87] I. Horrocks. FaCT and iFaCT. En *1999 Description Logics Workshop*, vol. 22 de *Workshop Proceedings*. CEUR, 1999.

[88] I. Horrocks, P. Patel-Schneider y F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The Making of a Web Ontology Language. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(1):7–26, Diciembre 2003

[89] I. Horrocks y U. Sattler. A Tableaux Decision Procedure for $\mathcal{SHOIQ}$. En *19th Intl. Joint Conference on Artificial Intelligence*, págs. 448–453, Edinburgh, Scotland, Julio / Agosto 2005. Professional Book Center

[90] J. Hu, C. Guo, H. Wang y P. Zou. Quality Driven Web Services Selection. En *1st IEEE Intl. Conf. on e-Business Engineering*, págs. 681–688, Beijing, China, Octubre 2005. IEEE Computer Society Press

[91] C. Hwang y K. Yoon. *Multiple Attribut Decision Making : Methods and Applications*. Springer-Verlag, 1981

[92] IBM. WebSphere Mathmaking Environment, 2002.

[93] ILOG. ILOG CPLEX 6.0: Related Documentation, 1998.

[94] ILOG. ILOG Solver 5.1: Related Documentation, 1998.

[95] ILOG. ILOG OPL Studio 3.6.1: Related Documentation, 2002.

[96] ISO. Information Technology – Z Formal Specification Notation – Syntax, Type System and Semantics, 2002

[97] L. Kagal, T. Finin y A. Joshi. Declarative Policies for Describing Web Service Capabilities and Constraints. En *W3C Workshop on Constraints and Capabilities for Web Services*, Redwood Shores, CA, Octubre 2004

[98] T. Kawamura, J.-A. de Blasio, T. Hasegawa, M. Paolucci y K. Sycara. Preliminary Report of Public Experiment of Semantic Service Matchmaker with UDDI Business Registry. En *1st Intl. Conf. on Service-Oriented Computing*, vol. 2910 de *LNCS*, págs. 208–224, Trento, Italy, 2003. Springer Verlag

[99] A. Keller y H. Ludwig. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. Informe Técnico RC22456 (W0205-171), IBM, Mayo 2002

[100] U. Keller, R. Lara, A. Polleres, I. Toma, M. Kifer y D. Fensel. WSMO Web Service Discovery, Noviembre 2004

[101] M. Kifer, R. Lara, A. Polleres, C. Zhao, U. Keller, H. Lausen y D. Fensel. A Logical Framework for Web Service Discovery. En *Workshop on Semantic Web Services*, vol. 119 de *Workshop Proceedings*, Hiroshima, Japan, Noviembre 2004. CEUR.

[102] M. Klein y A. Bernstein. Searching for Services on the Semantic Web using Process Ontologies. En *The Emerging Semantic Web: Selected Papers from the 1st Intl. Semantic Web Working Symposium*, vol. 75 de *Frontiers in Artificial Intelligence*. IOS Press, 2002

[103] M. Klein y A. Bernstein. Towards High-Precision Service Retrieval. *IEEE Internet Computing*, 8(1):30–36, Enero / Febrero 2004

[104] M. Klusch, B. Fries, M. Khalid y K. Sycara. OWLS-MX: Hybrid OWL-S Service Matchmaking. En *1st Intl. AAAI Fall Symposium on Agents and the Semantic Web*, Arlington, VA, Noviembre 2005

[105] J. Koistinen y A. Seetharaman. Worth–based Multi–Category Quality–of–Service Negotiation in Distributed Object Infrastructures. En *2nd Intl. Enterprise Distributed Object Computing Workshop*, La Jolla, USA, 1998

[106] V. Kolovski, B. Parsia, Y. Katz y J. Hendler. Representing Web Service Policies in OWL-DL. En *4th Intl. Semantic Web Conference*, vol. 3729 de *LNCS*, págs. 461–475, Galway, Ireland, Noviembre 2004. Springer

[107] K. Kuchcinski. Constraint-Driven Scheduling and Resource Assignment. *ACM Transactions on Design Automation of Electronic Systems*, 8(3):355–383, Julio 2003

[108] F. Laburthe y N. Jussien. CHOCO Constraint Programming System, 2004.

[109] J. Larrosa y P. Meseguer. Algoritmos para Satisfacción de Restricciones. *Revista Iberoamericana de Inteligencia Artificial*, 20:31–42, 2003

[110] E. Lawler y D. Wood. Branch-and-Bound Methods: A Survey. *Operations Research*, 14:699–719, 1966

[111] K. Lee, J. Jeon, W. Lee, S. Jeong y S. Park. QoS for Web Services: Requirements and Possible Approaches, 2003

[112] L. Li y I. Horrocks. A Software Framework for Mathmaking Based on Semantic Web Technology. En *12th Intl. World Wide Web Conference*, págs. 331–339, Budapest, Hungary, Mayo 2003. ACM Press

[113] X. Liu. A Quantitative Approach for Assessing the Priorities of Software Quality Requirements. *Journal of Systems and Software*, 42(8):105–113, Agosto 1998

[114] Y. Liu, A. Ngu y L. Zeng. QoS Computation and Policing in Dynamics Web Service Selection. En *13th Intl. World Wide Web Conference*, págs. 66–73, Nueva York, NY, Mayo 2004. ACM Press

[115] H. Ludwig. Electronic Contracts. En *Technology Supporting Business Solutions*. Nova Science Publishers, Inc., 2003

[116] H. Ludwig, A. Dan y R. Kearny. Cremona: An Architecture and Library for Creation and Monitoring of WS-Agreements. En *2nd Intl. Conf. on Service-Oriented Computing*, págs. 65–74, New York, NY, Noviembre 2004. ACM Press

[117] H. Ludwig, H. Gimpel, A. Dan y R. Kearny. Template-Based Automated Service Provisioning – Supporting the Agreement-Driven Service Life-Cycle. En *3rd Intl. Conf. on Service-Oriented Computing*, vol. 3826 de *LNCS*, págs. 283–296, Amsterdam, The Netherlands, Diciembre 2005. Springer Verlag

[118] H. Ludwig, A. Keller, A. Dan y R. King. A Service Level Agreement Language for Dynamic Electronic Services. Informe Técnico RC22316 (W0201-112), IBM, Enero 2002

[119] H. Ludwig, A. Keller, A. Dan, R. King y R. Franck. Web Service Level Agreement (WSLA) Language Specification. Informe Técnico wsla-2003/01/28, IBM, Enero 2003

[120] J. Ma y B. Knight. Reifid Temporal Logics: An Overview. *Artificial Intelligence Review*, 15:189–217, 2001

[121] A. Mackworth. Consistency in Network of Relations. *Artificial Intelligence*, 8:99–118, 1977

[122] N. Mamoulis y K. Stergiou. Algorithms for Quantified Constraint Satisfaction Problems. Informe Técnico APES-78-2004, APES Research Group, 2004

[123] A. Mani y A. Nagarajan. Understanding Quality of Service for Web Services, 2002.

[124] C. Marchetti, B. Pernici y P. Plebani. A Quality Model for E-service Based Multi-Channel Adaptive Information Systems. En *1st Intl. WISE Web Services Quality Workshop*, págs. 165–172, Rome, Italy, Diciembre 2003. IEEE Press

[125] K. Marriott y P. Stuckey. *Programming with Constraints: An Introduction*. The MIT Press, 1998

[126] O. Martín-Díaz, D. Benavides, J. Peña y M. Toro. Un Tratamiento Sensible a la Calidad para la Adquisición de Servicios Web. En *VIII Jornadas de Ingeniería del Software y Bases de Datos*, págs. 209–220, Alicante, España, 2003

[127] O. Martín-Díaz, A. Ruiz-Cortés, D. Benavides, A. Durán y M. Toro. A Quality-aware Approach to Web Services Procurement. En *4th Intl. VLDB Workshop Technologies for E-Services*, vol. 2819 de *LNCS*, págs. 42–53, Berlin, Germany, Diciembre 2003. Springer Verlag

[128] O. Martín-Díaz, A. Ruiz-Cortés, R. Corchuelo y A. Durán. A Management and Execution Environment for Multi-Organisational Web-based Systems. En *ZOCO: Métodos y Herramientas para el Comercio Electrónico*, págs. 79–88, San Lorenzo del Escorial, Spain, 2002

[129] O. Martín-Díaz, A. Ruiz-Cortés, R. Corchuelo y M. Toro. A Framework for Classifying and Comparing Web Services Procurement Platforms. En *1st Intl. WISE Web Services Quality Workshop*, págs. 156–164, Rome, Italy, 2003. IEEE Press

[130] O. Martín-Díaz, A. Ruiz-Cortés, A. Durán, D. Benavides y M. Toro. Automating the Procurement of Web Services. En *1st Intl. Conf. on Service-Oriented Computing*, vol. 2910 de *LNCS*, págs. 91–103, Trento, Italy, Diciembre 2003. Springer Verlag

[131] O. Martín-Díaz, A. Ruiz-Cortés, A. Durán y C. Müller. An Approach to Temporal-Aware Procurement of Web Services. En *3rd Intl. Conf. on Service-Oriented Computing*, vol. 3826 de *LNCS*, págs. 170–184, Amsterdam, The Netherlands, Diciembre 2005. Springer Verlag

[132] O. Martín-Díaz, A. Ruiz-Cortés y M. Toro. Un Entorno de Administración y Ejecución de Sistemas Multiorganizacionales basados en la Web. En *Workshop on Web Engineering (WebE'02)*, págs. 27–40, San Lorenzo del Escorial, Spain, 2002

[133] A. Maurino, S. Modafferi y B. Pernici. Reflective Architectures for Adaptive Information Systems. En *1st Intl. Conf. on Service-Oriented Computing*, vol. 2910 de *LNCS*, págs. 115–131, Trento, Italy, Diciembre 2003. Springer Verlag

[134] E. Maximilien y M. Singh. A Framework and Ontology for Dynamic Web Services Selection. *IEEE Internet Computing*, 8(5):84–93, Septiembre / Octubre 2004

[135] E. Maximilien y M. Singh. Towards Autonomic Web Services Trust and Selection. En *2nd Intl. Conf. on Service Oriented Computing*, págs. 212–221, New York, NY, 2004. ACM Press

[136] E. Maximilien y M. Singh. Agent-based Trust Model Involving Multiple Qualities. En *4th Intl. Joint Conference on Autonomous Agents and Multiagent Systems*, págs. 519–526, Utrecht, The Netherlands, 2005. ACM Press

[137] D. McGuinness y F. van Harmelen. OWL: Web Ontology Language Overview, Febrero 2004

[138] S. McIlraith, T. Son y H. Zeng. Semantic Web Service. *IEEE Intelligent Systems*, 16(2):46–53, Marzo / Abril 2001

[139] P. Meseguer. Interleaved Depth-First Search. En *15th Intl. Joint Conference on Artificial Intelligence*, págs. 1382–1387, Nagoya, Japan, 1997. Morgan Kaufmann

[140] METEOR-S Team Members. METEOR-S: Semantic Web Services and Processes, 2003

[141] J. Miller. *Professional Decision-Making: A Procedure for Evaluating Complex Alternatives*. Praeger Publishers, 1970

[142] C. Müller, O. Martín-Díaz, A. Ruiz-Cortés y J. García. Consistencia y Conformidad en un Contexto Temporal. En *ZOCO: Métodos y Herramientas para el Desarrollo de Aplicaciones*, págs. 15–24, Sitges, Spain, Octubre 2006

[143] C. Müller, O. Martín-Díaz, M. Resinas, P. Fernández y A. Ruiz-Cortés. A WS–Agreement Extension for Specifying Temporal Properties in SLA's. En *III Jornadas Científico-Técnicas en Servicios Web y SOA*, Zaragoza, España, Septiembre 2007

[144] C. Müller, O. Martín-Díaz, A. Ruiz-Cortés, M. Resinas y P. Fernández. Improving Temporal–Awareness of WS–Agreement. En *5th Intl. Conf. on Service-Oriented Computing*, vol. 4749 de *LNCS*, págs. 193–206, Vienna, Austria, Septiembre 2007. Springer Verlag

[145] B. Moore, E. Ellesson, J. Strassner y A. Westerinen. Policy Core Information Model – Version 1 Specification, Febrero 2001

[146] R. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1966

[147] M. Mouhoub. Reasoning with Numeric and Symbolic Time Information. *Artificial Intelligence Review*, 21:25–56, 2004

[148] N. Mukhi, P. Plebani, I. Silva-Lepe y T. Mikalsen. Supporting Policy-driven Behaviours in Web Services: Experiences and Issues. En *2nd Intl. Conf. on Service Oriented Computing*, págs. 322–328, New York, NY, 2004. ACM Press

[149] N. Oldham, K. Verma, A. Sheth y F. Hakimpour. Semantic WS-Agreement Partner Selection. En *15th Intl. World Wide Web Conference*, págs. 697–706, Edinburgh, Scotland, Mayo 2006. ACM Press

[150] OMG. UML Profile for Modelling Quality of Service and Fault Tolerance Characteristics and Mechanisms, Junio 2004

[151] J. O'Sullivan y D. Edmond. When and Where is a Service? Investigating Temporal and Locative Service Properties. En *IEEE Intl. Workshop on Service Oriented Computing: Models, Architectures and Applications at SAINT Conference*, págs. 90–94. IEEE Computer Society Press, Enero 2003

[152] J. O'Sullivan, D. Edmond y A. ter Hofstede. What's in a Service? Towards Accurate Description of Non-Functional Service Properties. *Distributed and Parallel Databases*, 12(2–3):117–133, Septiembre / Octubre 2002

[153] J. O'Sullivan, D. Edmond y A. ter Hofstede. Formal Descriptions of Non-Functional Service Properties. Informe Técnico FIT-TR-2005-01, Business Process Management Group, Centre for Information Technology Innovation, Queensland University of Technology, 2005

[154] F. Pang y J. Hobbs. Time in OWL-S. En *Proc. of the AAAI Spring Symposium on Semantic Web Services*, págs. 29–36, Stanford, CA, 2004

[155] M. Paolucci, T. Kawamura, T. Payne y K. Sycara. Importing the Semantic Web in UDDI. En *1st Intl. Workshop on Web Services, E-Business and the Semantic Web*, vol. 2512 de *LNCS*, págs. 225–236, Toronto, Canada, Mayo 2002. Springer Verlag

[156] M. Paolucci, T. Kawamura, T. Payne y K. Sycara. Semantic Matching of Web Services Capabilities. En *1st Intl. Semantic Web Conf.*, vol. 2342 de *LNCS*, págs. 333–347, Sardinia, Italy, Junio 2002. Springer Verlag

[157] I. Papaioannou, D. Tsesmetzis, I. Roussaki y M. Anagnostou. A QoS Ontology Language for Web-Services. En *20th IEEE Intl. Conf. on Advanced Information Networking and Applications*, págs. 101–106, Vienna, Austria, Abril 2006. IEEE Computer Society Press

[158] M. Papazoglou, P. Traverso, S. Dustdar, F. Leymann y B. Krämer. Service-Oriented Computing: A Research Roadmap. En *Dagstuhl Seminar on Service Oriented Computing*, 2006

[159] P. Pau y J. Schicho. Quantifier Elimination for Trigonometric polynomials by Cylindrical Trigonometric Decomposition. Informe Técnico, Research Institute for Symbolic Computation, 1995

[160] G. Piccinelli y L. Mokrushin. Dynamic Service Aggregation in Electronic Marketplaces. Informe Técnico HPL-2001-31, Hewlett-Packard, Febrero 2001

[161] G. D. Plotkin. A Structural Approach to Operational Semantics. Informe Técnico, Aarhus University, Denmark, 1981

[162] R. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill Professional, ed. 6th, 2004

[163] C. Priest. A Conceptual Architecture for Semantic Web Services. En *3rd Intl. Semantic Web Conference*, vol. 3298 de *LNCS*, págs. 395–409, Hiroshima, Japan, Noviembre 2004. Springer Verlag

[164] S. Ran. A Model for Web Services Discovery with QoS. *ACM SIGecom Exchanges*, 4(1):1–10, Spring 2003

[165] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler y D. Fensel. Web Service Modeling Ontology. *Applied Ontology*, 1(1):77–106, 2005

[166] A. Ruiz-Cortés. *Una Aproximación Semicualitativa al Tratamiento Automático de Requisitos de Calidad*. Tesis Doctoral, E.T.S. de Ingeniería Informática. Dpto. de Lenguajes y Sistemas Informáticos. Universidad de Sevilla, 2002

[167] A. Ruiz-Cortés, R. Corchuelo, A. Duran y M. Toro. Automated Support for Quality Requirements in Web-services-based Systems. En *8th IEEE Workshop on Future Trends of Distributed Computing Systems*, Bologna, Italy, 2001. IEEE Press

[168] A. Ruiz-Cortés, O. Martín-Díaz, A. Durán y M. Toro. Improving the Automatic Procurement of Web Services using Constraint Programming. *International Journal of Cooperative Information Systems*, 14(4):439–467, Diciembre 2005

[169] A. Sahai, A. Durante y V. Machiraju. Towards Automated SLA Management for Web Services. Informe Técnico HPL-2001-310 (R.1), HP Software Technology Laboratory, Palo Alto, CA, Julio 2002

[170] A. Sahai, V. Machiraju, M. Sayal, A. van Morseel y F. Casati. Automated SLA Monitoring for Web Services. En *13th IFIP/IEEE Intl. Workshop on Distributed Systems, Operations and Management*, vol. 2506 de *LNCS*, págs. 28–41, Montreal, Canada, Octubre 2002. Springer Verlag

[171] A. Sahai, J. Ouyang, V. Machiraju y K. Wurster. Specifying and Guaranteeing Quality of Service for Web Services through Real Time Measurement and Adaptive Control. Informe Técnico HPL-2001-134, HP Laboratories, Palo Alto, CA, 2001

[172] T. Schiex. Possibilistic Constraint Satisfaction Problems or "How to Handle Soft Constraints? En *8th Intl. Conf. on Uncertainty in Artificial Intelligence*, págs. 268–275, Stanford, CA, 1992

[173] T. Schiex, H. Fargier y G. Verfaillie. Valued Constraint Satisfaction Problems: Hard and Easy Problems. En *14th Intl. Joint Conference on Artificial Intelligence*, págs. 631–639, Montreal, Canada, 1995. Morgan Kaufmann

[174] E. Schwalb y L. Vila. Temporal Constraints: A Survey. *Constraints*, 3 (2–3):129–149, Junio 1998

[175] L. G. Shapiro y R. Haralick. Structural Descriptions and Inexact Matching. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 3: 504–519, 1981

[176] Y. Shi, G. Huang, L. Zhang y B. Shi. A Method to Select the Optimum Web Services. En *8th Asia-Pacific Web Conference*, vol. 3841 de *LNCS*, págs. 868–873, Harbin, China, Enero 2006. Springer Verlag

[177] B. Smith. A Tutorial on Constraint Programming. Informe Técnico 95.14, School of Computing, University of Leeds, 1995

[178] B. Smith, S. Brailsford, P. Hubbard y H. Williams. The Progressive Party Problem: Integer Linear Programming and Constraint Programming Compared. En *1st Intl. Conf. on Principles and Practice of Constraint Programming*. ACM Press, 1995

[179] W. Stadje. On The Relationship of Goal Programming and Utility Functions. *Business and Economics*, 23(1):61–69, Febrero 1979

[180] L. Stephens y M. Huhns. Concensus Ontology: Reconciling the Semantics of Web Pages and Agents. *IEEE Internet Computing*, 5(5):92–95, 2001

[181] K. Sycara, M. Klusch, S. Widoff y J. Lu. Dynamic Service Matchmaking among Agents in Open Information Environments. *SIGMOD Record*, 28(1):47–53, 1999

[182] K. Sycara, M. Paolucci, A. Ankolekar y N. Srinivasan. Automated Discovery, Interaction and Composition of Semantic Web Services. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 1(1):27–46, Diciembre 2003

[183] K. Sycara, M. Paolucci, J. Soudry y N. Srinivasan. Dynamic Discovery and Coordination of Agent-based Semantic Web Services. *IEEE Internet Computing*, 8(3):66–73, Mayo / Junio 2004

[184] H. Taha. *Operations Research: An Introduction*. Prentice Hall, 1997

[185] M. Tian, A. Gramm, T. Naumowicz, H. Ritter y J. Schiller. A Concept for QoS Integration in Web Services. En *1st IEEE/WISE Web Services Quality Workshop*, págs. 149–155, Rome, Italy, 2003. IEEE Computer Society Press

[186] V. Tosic. *Service Offerings for XML Web Services and Their Management Application*. Tesis Doctoral, Carleton University, Canada, Agosto 2004

[187] V. Tosic. WSOL versus Related Work. Informe Técnico SCE-04-07, Carleton University, Junio 2004

[188] V. Tosic y B. Pagurek. On Comprehensive Contractual Descriptions of Web Services. Informe Técnico 635, The University of Western Ontario, London, Canada, Octubre 2004

[189] V. Tosic, B. Pagurek, K. Patel, B. Esfandiari y W. Ma. Management Applications of the Web Service Offerings Language. *Information Systems*, 30(7):564–586, Noviembre 2005

[190] D. Trastour, C. Bartolini y J. González-Castillo. A Semantic Web Approach to Service Description for Matchmaking of Services. Informe Técnico HPL-2001-183, Hewlett-Packard, Julio 2001

[191] P. Trinidad, D. Benavides, A. Durán, A. Ruiz-Cortés y M. Toro. Automated Error Analysis for the Agilization of Feature Modeling. *Journal of Systems and Software*, 2007

[192] K. Trivedi. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. John Wiley & Sons, 2001

[193] E. Tsang. *Foundations of Constraint Satisfaction*. Academic Press, 1995

[194] D. Tsarkov y I. Horrocks. FaCT++ Description Logic Reasoner: System Description. En *3rd Intl. Joint Conference on Automated Reasoning*, vol. 4130 de *LNCS*, págs. 292–297, Seattle, WA, Agosto 2006. Springer

[195] D. Tsesmetzis, I. Roussaki, I. Papaioannou y M. Anagnostou. QoS Awareness Support in Web-Service Semantics. En *Advanced Intl. Conference on Telecommunications*, Guadeloupe, French Caribbean, Febrero 2006. IEEE Computer Society Press

[196] A. Tversky. Features of Similarity. *Psychological Review*, 84(4):327–352, Julio 1977

[197] A. Uszok, J. Bradshaw, R. Jeffers, M. Johnson, A. Tate, J. Dalton y S. Aitken. Policy and Contract Management for Semantic Web Services. En *1st AAAI Spring Symposium on Semantic Web Services*, Stanford University, CA, Marzo 2004

[198] P. van Hentenryck, L. Perron y J. Puget. Search and Strategies in OPL. *ACM Transactions on Computational Logic*, 1(2):285–320, 2000

[199] D. Veit, C. Weinhardt y J. Müller. Multi-Dimensional Matchmaking for Electronic Marketplaces. *Applied Artificial Intelligence*, 16(9–10):853–869, Octubre 2002

[200] L. Vu, K. Aberer y M. Hauswirth. Probabilistic Estimation of Service Quality Conformance and User Behaviours. Informe Técnico LSIR-REPORT-2006-013, Distributed Information Systems Laboratory, Ecole Polytechnique Fédérale de Lausanne, 2006

[201] L. Vu, M. Hauswirth, F. Porto y K. Aberer. A Search Engine for QoS-enabled Discovery of Semantic Web Services. *Special Issue of the International Journal of Business Process Integration and Management*, 1(4): 244–255, 2006

[202] L. Vu, F. Porto, M. Hauswirth, S. Gerlach, O. Tajmouati y K. Aberer. QoS-enabled Semantic Web Service Discovery: A User's Perspective Approach. Informe Técnico LSIR-REPORT-2006-012, Distributed Information Systems Laboratory, Ecole Polytechnique Fédérale de Lausanne, 2006

[203] X. Wang, T. Vitvar, M. Kerrigan y I. Toma. A QoS-aware Selection Model for Semantic Web Services. En *4th Intl. Conf. on Service-Oriented Computing*, vol. 4294 de *LNCS*, págs. 390–401, Chicago, IL, Diciembre 2006. Springer Verlag

[204] A. Ward, T. Lozano-Pérez y W. Seering. Extending the Constraint Propagation of Intervals. En *11th Intl. Joint Conference on Artificial Intelligence*, págs. 1453–1458, Detroit, Michigan, 1989

[205] A. Ward, T. Lozano-Pérez y W. Seering. Algebraic Solutions to Interval Linear Equations and their Applications. En *IMACS/GAMM Intl. Symposium on Numerical Methods and Error Bounds*, págs. 224–233, Oldenburg, Germany, 1995

[206] G. Weikum. The Web in 2010: Challenges and Opportunities for Database Research. En *Dagstuhl Seminars 10th Anniversary: Informatics – 10 Years Back. 10 Years Ahead*, vol. 2000 de *LNCS*, págs. 1–23, Schloss Dagstuhl, Germany, Septiembre 2001. Springer Verlag

[207] E. Wohlstadter, S. Tai, T. Mikalsen, I. Rouvellou y P. Davanbu. GlueQoS: Middleware to Sweeten Quality-of-Service Policy Interations. En *26th Intl. Conf. on Software Engineering*, págs. 189–199, Edinburg, Scotland, UK, Mayo 2004. IEEE Computer Society Press

[208] T. Yu y K. Lin. Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints. En *3rd Intl. Conf. on Service-Oriented Computing*, vol. 3826 de *LNCS*, págs. 130–143, Amsterdam, The Netherlands, Diciembre 2005. Springer Verlag

[209] T. Yu y K. Lin. Service Selection Algorithms for Web Services with End-to-End QoS constraints. *Information Systems and E-Business Management*, 3(2):103–326, Julio 2005

[210] M. Yu-jie, C. Jian, Z. Shen-shen y Z. Jian-hong. Interactive Web Service Choice-Making Based on Extended QoS Model. En *5th Intl. Conf. on Computer and Information Technology*, págs. 1130–1134, Shanghai, China, Septiembre 2005. IEEE Computer Society Press

[211] L. Zadeh. Fuzzy Sets. *Information and Control*, 8:338–353, 1965

[212] L. Zeng, B. Benatallah, M. Dumas, Kalagnanam y Q. Sheng. Quality Driven Web Services Composition. En *12th Intl. World Wide Web Conference*, págs. 411–421, Budapest, Hungary, 2003. ACM Press

[213] L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam y H. Chang. QoS-Aware Middleware for Web Services Composition. *IEEE Transactions on Software Engineering*, 30(5):311–327, Mayo 2004

[214] C. Zhou, L. Chia y B. Lee. DAML-QoS Ontology for Web Services. En *2nd Intl. Conf. on Web Services*, págs. 472–479, San Diego, CA, Julio 2004. IEEE Computer Society Press

[215] C. Zhou, L. Chia y B. Lee. Semantics in Service Discovery and QoS Measurement. *IEEE IT Professional, Technology Solutions for the Enterprise*, 7(2):29–34, Marzo / Abril 2005

[216] C. Zhou, L. Chia y B. Lee. Service Discovery and Measurement based on DAML-QoS Ontology. En *14th Intl. World Wide Web Conference*, págs. 1070–1071, Chiba, Japan, Mayo 2005. ACM Press

[217] J. Zinky, D. Bakken y R. Schantz. Architectural Support for Quality of Service for CORBA Objects. *Theory and Practice of Object Systems*, 3 (1):55–73, Abril 1997

This document was typeset on 2007/10/18 using RG-BOOK α2.11 for LaTeX$_{2\epsilon}$.
Should you want to use this document class, please send mail to
contact@tdg-seville.info.