



UNIVERSIDAD CARLOS III DE MADRID

TESIS DOCTORAL

**DEFINICIÓN DE UN MODELO DE
REPRESENTACIÓN DEL CONOCIMIENTO
PARA PROCESOS DE ESTIMACIÓN DE
PRESUPUESTOS**

Autor:
José Luis López Cuadrado

Director/es:
Dr. Ángel García Crespo
Dra. Belén Ruiz Mezcua

DEPARTAMENTO DE INFORMÁTICA

Leganés, Diciembre de 2009

TESIS DOCTORAL

DEFINICIÓN DE UN MODELO DE REPRESENTACIÓN DEL CONOCIMIENTO PARA PROCESOS DE ESTIMACIÓN DE PRESUPUESTOS

Autor: José Luis López Cuadrado

Director/es: Dr. Ángel García Crespo y Dra. Belén Ruiz Mezcuca

Firma del Tribunal Calificador:

Firma

Presidente:

Vocal:

Vocal:

Vocal:

Secretario:

Calificación:

Leganés, de de

A María

Agradecimientos

Es difícil condensar en unas cuantas líneas la gratitud que siento hacia tantas personas que me han acompañado y me han ayudado a llegar hasta este punto.

En primer lugar agradecer a mis tutores, Ángel García Crespo y Belén Ruiz Mezcuca, la oportunidad que me dieron de trabajar a su lado y de alcanzar este punto. Gracias por todo.

No puedo dejar de mencionar a mi familia, a mis padres que me han dado todo, a mi hermana y cuñado, a mis suegros, mi cuñada y a toda mi familia por apoyarme en todo momento, especialmente en los momentos más bajos.

Gracias también a mis amigos. A Isra, compañero de fatigas, por estos años compartiendo trabajo, estancias, artículos, alegrías y penas. A Fernando, Ricardo, Juanmi, Isabel, Lourdes, Luis, María Jesús, Marilú, y los demás (seguro que alguno falta, no me lo tengáis en cuenta) por sus consejos y sus palabras de ánimo. Gracias a Luis, Bernabé y Sergio por estar ahí. Gracias también a Jesús, Diego, Maite, Vero, Pili, María, Eva y todos los del grupo por sus ánimos.

Pero lo más grande queda para el final. Estos últimos cuatro años el pilar fundamental sobre el que me he sostenido ha sido mi mujer. Gracias María por tu ayuda, tu comprensión y por transmitirme fuerza e ilusión para seguir adelante. En otra dedicatoria te decía *el mañana ha comenzado hoy*. Ese mañana llegó y comenzamos a andarlo juntos hace cuatro años, y hoy comienza otro mañana que nos llevará a otro punto que está aún por descubrir. Solo espero seguir encontrando mañanas junto a tí, pase lo que pase.

Con la emoción, las palabras se quedan cortas para mostrar la gratitud y el detalle hacia todos los que he mencionado y los que en este momento no me vienen a la memoria. Cada uno de los que ha estado junto a mí, quizás sin saberlo, ha puesto un granito de arena. Por tanto gracias a todos de corazón.

Summary

In order to satisfy the requirements for the european doctorate mention, the summary of this phd thesis has been written in English. This summary includes methodology, main contribution and conclusions. The chapter 9 “Conclusions” has also been written in English.

Price estimation is a critical process for the preparation of quotations in companies that have to compete with other companies by offering competitive prices in order to have demand of orders. For machining parts companies, the estimation process is very complex, as a big number of variants may appear. An expert in manufacturing is usually in charge of this process. However, the lack of qualifying experts in such factories implies extra work for the experts and they already have other critical tasks to which they must attend. The use of conventional software systems is usual as they automate the costs' estimation, and the knowledge based systems are one of the main alternatives for two reasons. On the one hand, this knowledge's application constitutes an open area of research that must be exploited. On the other hand, the research on systems based on knowledge raises the need of new models and methodologies that incorporate the knowledge's representation with the web for the creation of new expert systems based on the web.

In order to give an answer to the former two aspects that have been identified, this thesis presents a knowledge representation model for representing an expert's experience orientated to the quotation process. This representation model also covers the knowledge that can be obtained from other applications, allows the use of detailed estimation models that complement the expert's knowledge and incorporates the representation by using web technologies. The next contribution of this thesis is a solution engine based on the proposed model which is able to automate the represented processes in a web environment. The purpose of the engine is not restricted to the concrete domain of this thesis (it is generic), and it can be used for other domains in future research works. Finally a modeling tool based on the proposed model is presented. Such a tool allows the knowledge representation of an expert and at the same time it provides the validation by means of trace functionality.

Besides the objectives and hypotheses initially proposed, a formal definition of the model is presented. It makes possible the knowledge sharing between different experts in different places using a shared language.

The steps taken to conduct the research are the following:

- 1. State of the art review. In this phase, several price estimation techniques and tools within versatile manufacturing companies has been reviewed, paying special attention to the knowledge based techniques and tools. It has been found that the creation of these systems use more theoretic knowledge than knowledge provided by experience.*

-
2. *Problem analysis. In this phase research hypotheses have been put forward. Such hypotheses are based on the need of a knowledge representation model which is able to represent knowledge about the quotation process, the knowledge stored in external applications, and the information exchange with external agents when it is necessary to obtain additional information. Additionally a solution engine is required in order to automate the represented process and so that it is capable of obtaining similar results to those from an expert.*
 3. *Problem Solution. In this phase the knowledge representation model has been defined in order to solve the problems previously identified.*
 4. *Validation. In this phase a knowledge representation tool has been developed based on the proposed knowledge representation model. Also a solution engine has been developed in order to interpret and execute the processes modelled by means of the knowledge representation tool. Using the representation tool and the solution engine, a knowledge based system has been developed in order to validate the proposed hypotheses. The knowledge based system has been implemented in the context of a machining parts company.*
 5. *Analysis of the results obtained. In this phase the results of the validation phase were evaluated in order to determine the validity of the proposed model and tools.*
 6. *Documentation. During the overall research work the present thesis has been documented in order to generate the present document.*

After the elaboration, validation and analysis of the results, we have concluded that the proposed model can represent the knowledge of an expert in the domain of the quotation process for machining parts. Furthermore, the validation results show that the proposed model integrates the representation with the Web technologies. It allows, by means of the modelling tool and the solution engine, developing web based expert systems for the specific problem of the quotation process and, in future research works, for other domains.

Resumen

A fin de cumplir los requisitos para la mención europea del doctorado, el resumen fue escrito en inglés. A continuación se incluye el resumen en castellano.

La estimación de precios es un proceso crítico para la elaboración de presupuestos para en compañías que tienen que competir para obtener pedidos ofreciendo precios competitivos. En concreto, las compañías de mecanizado de piezas, dicho proceso es complejo, pues se pueden plantear un gran número de variantes. En este caso un experto en la fabricación está generalmente al cargo de esta tarea. Sin embargo, estos expertos tienen a cargo otras tareas, también importantes para la compañía, que deben atender. Es habitual la utilización de sistemas software que automaticen la estimación de costes, y los sistemas basados en conocimiento son una de las principales alternativas por dos razones fundamentales. Por un lado, la aplicación de este conocimiento constituye un área abierta de investigación que debe ser explotada. Por otro lado, la investigación acerca de los sistemas basados en conocimiento plantea la necesidad de nuevos modelos y metodologías que integren la representación del conocimiento con la web para la creación de nuevos sistemas expertos basados en web.

Para responder a los dos aspectos que se han identificado, la presente tesis plantea un modelo de representación del conocimiento para capturar la experiencia de un experto orientado a la estimación de presupuestos. Este modelo de representación cubre además el conocimiento que puede obtenerse de otras aplicaciones, permite la utilización de modelos de estimación detallados que complementen al conocimiento del experto e integra la representación con tecnologías web. La siguiente aportación de esta tesis doctoral es un motor de solución basado en el modelo propuesto capaz de automatizar los procesos representados, en un entorno web. Se trata de un motor de solución genérico que no es exclusivo para la implementación del proceso de representación de la fábrica en la que se ha llevado a cabo la validación. Finalmente se ha aportado una herramienta de modelado que permite la representación del conocimiento de un experto al tiempo que facilita el proceso de validación por medio de la funcionalidad de traza.

Además de los objetivos e hipótesis inicialmente planteados, la presente tesis aporta la definición formal del modelo de representación del conocimiento. Con ello se da la posibilidad de compartir la representación de distintos procesos entre expertos de diferentes lugares basándose en una definición común. La metodología seguida para la elaboración de esta tesis, ha consistido en:

- 1. Estudio del estado del arte. Se han analizado las distintas técnicas y herramientas empleadas en la estimación de precios dentro de las compañías de fabricación versátil,*

poniendo especial atención en las que se basan en conocimiento. En este sentido se ha encontrado que se emplean más conocimientos teóricos que conocimientos procedentes de la experiencia para la creación de estos sistemas.

- 2. Análisis del problema. En esta fase se ha realizado el planteamiento de las hipótesis de investigación, basadas en la necesidad de un modelo de representación capaz de representar el conocimiento sobre el proceso de estimación de presupuestos, el acceso al conocimiento que pueda contener otro sistema y el intercambio de información con agentes externos cuando sea necesario obtener información adicional. Adicionalmente es necesario un motor capaz de automatizar el proceso y que éste sea capaz de obtener resultados similares a los de un experto.*
- 3. Planteamiento de la solución. En esta fase se ha planteado el modelo de representación que se propone como solución a los problemas identificados.*
- 4. Validación. En esta fase se ha creado una herramienta de modelado basada en el modelo propuesto, así como un motor capaz de interpretar y ejecutar los procesos representados. A partir de ambas herramientas, se ha desarrollado un sistema basado en conocimiento y comprobación de las hipótesis de investigación. Para ello se ha modelado un proceso real en una fábrica de piezas mecanizadas.*
- 5. Análisis de los resultados obtenidos. Tras la validación se obtuvieron un conjunto de resultados, tanto a nivel de representación como a nivel de ejecución de proceso que fueron analizados para determinar la validez del modelo y las herramientas propuestas.*
- 6. Documentación. A lo largo de todo el proceso de elaboración de la tesis se ha generado la documentación que constituye la presente tesis doctoral.*

Las conclusiones tras la elaboración, validación y análisis de los resultados obtenidos, han mostrado que el modelo propuesto puede representar el conocimiento de un experto en el dominio de la fabricación de piezas mecanizadas. Además la validación realizada muestra como el modelo se integra con la Web permitiendo, gracias a la implementación de una herramienta de modelado y un motor de soluciones, la creación de sistemas basados en Web que afronten en concreto el problema del cálculo de precios, admitiendo problemas de otros dominios en futuros trabajos de investigación.

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Motivación	4
1.3. Descripción del Problema	5
1.4. Justificación	7
1.4.1. ¿Por qué es importante la estimación del precio?	7
1.4.2. ¿Por qué representar el conocimiento de un experto?	8
1.4.3. ¿Por qué un modelo nuevo de representación?	8
1.5. Objetivos	9
1.6. Metodología	10
1.7. Estructura de la Tesis Doctoral	12
2. Estado del Arte	15
2.1. Sistemas Basados en Conocimiento	15
2.1.1. Evolución Histórica	16
2.1.2. Características Generales de los Sistemas Basados en Conocimiento	19
2.2. Sistemas Expertos Basados en Web	21
2.3. El Proceso de Oferta y su Optimización	24
2.4. Modelos de Estimación de Costes	26
2.4.1. Técnicas de estimación cualitativas	27
2.4.2. Técnicas de estimación cuantitativas	28

2.4.3.	Discusión sobre los métodos de estimación	30
2.5.	Aplicaciones para Estimación de Precios y Costes	31
2.5.1.	Aplicaciones y modelos basados en tecnologías convencionales	31
2.5.2.	Aplicaciones y modelos basados en sistemas expertos	46
2.6.	Representación de Procesos Basada en Ontologías	63
2.6.1.	Mejora de los modelos clásicos de representación por medio de ontologías	65
2.6.2.	Representación de procesos basados en servicios web semánticos	66
2.6.3.	Otras aproximaciones	67
2.7.	Discusión	69
2.8.	Resumen	71
3.	Propuesta de Solución	73
3.1.	Hipótesis de Investigación	73
3.2.	Modelo Propuesto	74
3.3.	Resumen	76
4.	Modelo de Representación del Conocimiento	79
4.1.	Consideraciones Previas	79
4.2.	Modelo Formal	81
4.2.1.	Definición inicial del proceso de solución	81
4.2.2.	Hecho	82
4.2.3.	Acción	82
4.2.4.	Verificación	82
4.2.5.	Decisión	83
4.2.6.	Situación	84
4.2.7.	Contexto	85
4.2.8.	Resumen de definiciones	87
4.3.	Relación entre los Conceptos del Modelo Formal	88
4.4.	Extensión del Modelo: Hacia la Automatización	90
4.4.1.	Agentes que intervienen en un proceso	90
4.4.2.	Acciones que pueden realizarse en un proceso	92
4.5.	Modelo Conceptual	92
4.5.1.	Proceso	95
4.5.2.	Hecho	95
4.5.3.	Acción	96

4.5.4.	Situación	98
4.5.5.	Contexto	102
4.5.6.	Comportamiento	102
4.5.7.	Parámetro	107
4.5.8.	Decisiones	108
4.5.9.	Verificación	110
4.5.10.	Operador	111
4.5.11.	Condición	112
4.6.	Resumen	114
5.	Algoritmo de Solución	115
5.1.	Introducción	115
5.2.	Algoritmo de Solución: Control y Ejecución de un Proceso	116
5.2.1.	Activación de un proceso de solución	120
5.2.2.	Contexto y situación inicial	121
5.2.3.	Situaciones y acciones	122
5.2.4.	Recepción de nuevos hechos	125
5.2.5.	Validación de librerías	126
5.2.6.	Verificaciones	127
5.2.7.	Decisiones	128
5.2.8.	Resolución de conflictos y pila de decisiones	129
5.2.9.	Evaluación de condiciones	131
5.2.10.	Detención del proceso	133
5.2.11.	Selección de la siguiente situación	134
5.3.	Resumen: Semántica del Modelo Propuesto	134
5.4.	Propuesta de Arquitectura para un Motor de Solución	138
5.4.1.	Motor de Solución	140
5.4.2.	Interfaz de Comunicación	140
5.4.3.	Interfaz Software	142
5.4.4.	Base de Conocimiento	144
5.4.5.	Base de Hechos	145
5.5.	Detalles de Implementación	146
5.5.1.	Interfaz HTTP y control del proceso	146
5.5.2.	Extensiones de los conceptos de la ontología	148

5.5.3. Tags de control en las acciones externas	154
5.6. Resumen	155
6. Herramienta de Modelado	157
6.1. Descripción General	157
6.2. Organización General de la Herramienta de Modelado	158
6.3. Proceso de Solución	158
6.4. Representación de Contextos	160
6.5. Representación de Situaciones y Acciones	161
6.5.1. Situaciones interactivas	161
6.5.2. Situaciones automáticas o librerías	165
6.5.3. Situaciones virtuales	170
6.6. Representación de Verificaciones	171
6.7. Representación de Decisiones	173
6.8. Representación de Condiciones	177
6.9. Representación de Hechos	182
6.10. Representación de Operadores	185
6.11. Representación de Parámetros del Proceso	187
6.12. Gestión de Sesiones	189
6.13. Traza de Ejecución	191
6.14. Resumen	193
7. Representación del Proceso de Elaboración de Presupuestos	195
7.1. Introducción	195
7.2. Metodología	195
7.3. Descripción del Proceso de Estimación	199
7.4. Aplicación del Modelo	200
7.5. Resumen	233
8. Validación	235
8.1. Metodología de Validación	235
8.2. Validación del Modelo de Representación	237
8.3. Validación Empírica	239
8.3.1. Descripción de los casos de pruebas	241
8.3.2. Resultados obtenidos	242

Índice general

8.4. Resumen	248
9. Conclusions and Future Research	251
9.1. Conclusions	251
9.2. Future Research	254
9.3. Related Publications	255
10. Conclusiones y Futuras Líneas de Investigación	257
10.1. Conclusiones	257
10.2. Futuras Líneas de Investigación	260
10.3. Publicaciones Relacionadas	261
Bibliografía	263

Índice de figuras

1.1. Etapas de la investigación.	11
2.1. Relación entre sistemas basados en conocimiento y sistemas expertos.	16
2.2. Arquitectura general de un sistema experto.	19
4.1. Ejemplo del concepto decisión.	84
4.2. Conceptos del modelo de representación.	89
4.3. Modelo conceptual.	94
4.4. Relación entre proceso, contexto y comportamiento.	95
4.5. Distintos tipos de acción.	97
4.6. Clases de situación en función de las acciones.	98
4.7. Esquema de funcionamiento de una situación automática.	100
4.8. Esquema de funcionamiento de una situación interactiva.	100
4.9. Tipos de comportamiento en función de las situaciones.	103
4.10. Relaciones de los distintos tipos de comportamiento.	105
4.11. Relaciones del concepto decisión.	108
4.12. Relaciones del concepto condición.	113
5.1. Arquitectura de alto nivel.	118
5.2. Procesos de solución guiados.	119
5.3. Procesos de solución automatizados.	119
5.4. Algoritmo de control del motor de solución.	120

5.5. Comportamiento implementado para situaciones interactivas.	123
5.6. Comportamiento de situaciones automáticas.	124
5.7. Esquema de modelado como red de situaciones simple.	135
5.8. Esquema de modelado como red de contextos.	136
5.9. Arquitectura propuesta del motor de solución.	139
5.10. Esquema de la interfaz de comunicación.	142
5.11. Esquema de la interfaz software.	143
6.1. Pantalla principal de la herramienta de modelado.	159
6.2. Gestión de contextos con la herramienta de modelado.	160
6.3. Selección del contexto inicial del proceso.	161
6.4. Gestión de situaciones interactivas con la herramienta de modelado.	162
6.5. Vista de la ventana para crear o modificar una situación interactiva.	163
6.6. Vista de la ventana para añadir una situación interactiva existente a un contexto.	165
6.7. Visor de una acción externa en una situación interactiva.	166
6.8. Gestión de los hechos asociados a una situación.	166
6.9. Gestión de situaciones automáticas con la herramienta de modelado.	167
6.10. Vista de la ventana para crear o modificar una situación automática.	168
6.11. Vista de la ventana para añadir una situación automática existente a un contexto.	169
6.12. Vista de la ventana para gestionar las validaciones de una situación automática.	170
6.13. Pestaña para gestionar las verificaciones de una situación.	171
6.14. Ventana para crear una nueva verificación.	172
6.15. Ventana para modificar una verificación.	173
6.16. Ventana para añadir una verificación existente.	174
6.17. Vista de la pestaña para gestionar las decisiones para una situación.	174
6.18. Ventana para crear una nueva decisión para una situación.	175
6.19. Ventana para seleccionar el destino de una decisión.	175
6.20. Ventana para seleccionar una decisión existente para una situación.	176
6.21. Ventana para modificar una decisión en la situación y contexto seleccionados.	177
6.22. Zona de gestión de condiciones.	177
6.23. Ventana para crear una condición.	178
6.24. Selección de un objeto de conocimiento para un operando.	180
6.25. Selección de una condición existente.	182
6.26. Pestaña para la gestión de hechos.	183
6.27. Ventana para la creación de un hecho.	184

6.28. Pestaña para la gestión de valores iniciales.	184
6.29. Ventana para eliminar un conjunto de hechos.	185
6.30. Pestaña para la gestión de operadores.	186
6.31. Ventana para crear un nuevo operador.	186
6.32. Configuración general del proceso.	187
6.33. Pestaña para la gestión de sesiones del motor de solución.	190
6.34. Ventana para eliminar sesiones del motor de solución.	191
6.35. Selección del directorio a utilizar en la traza de ejecución.	192
6.36. Ventana de la traza de ejecución.	193
7.1. Modelado y automatización de la estimación.	197
7.2. Grafo común de decisión.	201
7.3. Datos generales para el presupuesto.	203
7.4. Grafo de situaciones para la estimación con tornos.	204
7.5. Selección del torno a utilizar.	205
7.6. Selección del material.	206
7.7. Forma y dimensiones de la pieza.	208
7.8. Situación interactiva: parámetros de operación para un torno.	213
7.9. Parámetros recomendados.	218
7.10. Selección de operaciones adicionales.	221
7.11. Parámetros de eliminación de pivotes.	222
7.12. Parámetros de avellanado.	224
7.13. Parámetros de chaflanado.	225
7.14. Situación para obtener los costes unitarios.	230
7.15. Resultados de la estimación.	233
8.1. Situaciones visitadas por prueba.	243
8.2. Número de situaciones visitadas por tipo de situación.	244
8.3. Resultados de la interacción realizada por cada usuario.	244

Índice de tablas

2.1. Sistemas de estimación convencionales.	32
2.2. Sistemas expertos para la estimación.	47
8.1. Resultados de estimación para tornos.	246
8.2. Detalle de estimación (en horas).	249

Acrónimos

- **ABC.** Activity Based Cost.
- **ANN.** Artificial Neural Network.
- **API.** Application Programming Interface.
- **ASP.** Active Server Pages.
- **BPEL.** Business Process Execution Language.
- **BPMN.** Business Process Modeling Notation.
- **CAD.** Computer Aided Design.
- **CAD.** Computer Aided Design.
- **CAM.** Computer Aided Manufacturing.
- **CAPP.** Computer Aided Process Planning.
- **CAPP.** Computer Aided Process Planning.
- **CBR.** Case Based Reasoning
- **CGI.** Common Gateway Interface.
- **CNC.** Computer Numerical Control.

- **CRM.** Customer Relationship Management.
- **CSS.** Cascade Style Sheet.
- **DL.** Description Logics.
- **DSS.** Decision Support Systems.
- **ERP.** Enterprise Resource Planning.
- **ETO.** Engineering To Order.
- **HTML.** Hypertext Markup Language.
- **HTTP.** Hypertext Transfer Protocol.
- **JSP.** Java Server Pages.
- **MTO.** Make to Order.
- **OWL.** Ontology Web Language.
- **PDA.** Personal Digital Assistant.
- **PDM.** Product Data Management.
- **PSM.** Problem Solving Methods.
- **RDF.** Resource Description Framework.
- **RFQ.** Request for Quotation.
- **SBC.** Sistema Basado en Conocimiento.
- **SCM.** Supply Chain Management.
- **SQL.** Structured Query Language.
- **SSEE.** Sistemas Expertos.
- **SWRL.** Semantic Web Rule Language.
- **UML.** Unified Modeling Language.
- **URL.** Universal Resource Locator.

Acrónimos

- **USB.** Universal Serial Bus.
- **XML.** eXtensible Markup Language.
- **XPATH.** XML Path Language.
- **XSL.** eXtensible Stylesheet Language.

El objetivo principal de la presente investigación es la definición de un modelo de representación del conocimiento basado en situaciones para la estimación de precios en la elaboración de un presupuesto de fabricación. Este capítulo realiza una breve introducción al problema de la representación del conocimiento, y al problema de la elaboración de presupuestos que será utilizado para la validación del modelo propuesto. Finalmente se describe la metodología seguida en el desarrollo de la tesis doctoral y se realiza un breve resumen de la estructura del resto del documento.

1.1. Contexto

El objetivo de la industria es convertir materias primas en productos elaborados a partir de unos recursos escasos, habitualmente con el objetivo de obtener un beneficio económico. Esta conversión implica un conjunto de procesos que comprenden desde la concepción inicial del producto hasta su entrega final al cliente. Para las pequeñas y medianas industrias que fabrican piezas, el proceso encargado de recibir las peticiones del cliente para ofrecerles un presupuesto es especialmente importante, pues puede determinar la ganancia o la pérdida de un pedido.

La inteligencia artificial es una rama de la informática cuyo objetivo es construir aplicaciones que se comporten de forma inteligente, emulando la inteligencia humana. Existe una doble perspectiva sobre inteligencia artificial (Gómez et al. 1997, Winston 1992): como ciencia, tratando de comprender la naturaleza de la inteligencia y reproducirla en dispositivos artificiales, y como ingeniería, tratando de construir máquinas y aplicaciones inteligentes que resuelvan problemas imitando la forma en que los resuelven los humanos. Dentro de la segunda perspectiva, cuando para resolver un problema se necesitan conocimientos específicos del dominio, entran en juego los sistemas basados en conocimiento. Un sistema basado en conocimiento es un programa inteligente que utiliza conocimiento y procedimientos de inferencia para resolver problemas que son lo suficientemente difíciles como para requerir experiencia humana significativa para resolverlo (Feigenbaum 1982). Con el avance de la informática, los sistemas inteligentes han sido aplica-

dos en la industria para resolver los problemas que tradicionalmente requerían un experto. Las técnicas de inteligencia artificial que se aplican habitualmente a los problemas de ingeniería en las fábricas son (Pham and Pham 1999):

- Sistemas expertos, probablemente la tecnología más madura en su aplicación industrial.
- Sistemas de lógica borrosa. Se basa en el razonamiento a partir de conocimiento poco preciso y con incertidumbre. La lógica borrosa tiene aplicaciones potenciales en áreas donde el conocimiento es impreciso, como control de procesos y control de máquinas.
- Aprendizaje inductivo: se basa en la extracción automática de conocimiento en forma de reglas if-then, usualmente a partir de un conjunto de ejemplos.
- Programación lógica inductiva: emplea lógica de predicados para representar ejemplos de entrenamiento y conocimiento del dominio.
- Redes de neuronas artificiales. Tienen un amplio espectro de aplicaciones: modelado, predicción, clasificación, reconocimiento de patrones, proceso de señal, optimización, etc.
- Algoritmos genéticos. Esta técnica resulta adecuada para resolver problemas de combinatoria compleja y optimización multi-parámetro.

Las distintas áreas específicas de fabricación, emplean técnicas distintas en función de los problemas a los que se enfrentan. Por ejemplo, en el área de la fabricación con herramientas de control numérico, Park and Kim (1998) realizan una revisión de técnicas de inteligencia artificial para el control de parámetros en procesos de mecanización.

Los sistemas basados en conocimiento han constituido una gran ayuda para la industria. Desde los años 50 hasta hoy, la informática ha ido abriéndose camino dentro de las compañías y se ha incrementado la utilización de los sistemas basados en conocimiento, obteniendo grandes beneficios. Además, la aparición de internet y su desarrollo ha aportado numerosas mejoras y posibilidades para las fábricas, tanto a nivel de negocio, con el comercio electrónico y la mejora de las comunicaciones, como a nivel de gestión interna, facilitando el acceso a la información desde cualquier lugar en cualquier momento y proporcionando avances en la automatización y mejora de procesos.

La informática ha contribuido a la mejora de los procesos implicados en la fabricación de productos, desarrollando aplicaciones que dan soporte a dichos procesos. La Asociación de Sistemas

de Automatización y Computación¹, perteneciente a la Sociedad de Ingenieros de Fabricación², define la Fabricación Integrada por Ordenador (CIM) como *la fusión total del proceso de fabricación a través del uso de sistemas integrados y la comunicación de datos, junto con una nueva filosofía de gestión que mejora la organización y la eficiencia del personal*. El concepto de fabricación integrada por ordenador aglutina gran cantidad de tecnologías y conceptos, de los que podemos destacar los siguientes:

- **CAD.** Computer Aided Design.
- **CAM.** Computer Aided Manufacturing.
- **ERP.** Enterprise Resource Planning.
- **CRM.** Customer Relationship Management.
- **SCM.** Supply Chain Management.
- **CNC.** Computer Numerical Control.
- **CAPP.** Computer Aided Process Planning.
- **PDM.** Product Data Management.

El concepto de CIM va más allá de las tecnologías que integra; se trata de una nueva forma de hacer negocios que conjuga la calidad total en la empresa, la mejora continua, la satisfacción del cliente, la utilización de una única base de datos con toda la información del producto para las decisiones de fabricación y producción en todos los departamentos, la eliminación de las barreras de comunicación entre los departamentos y la integración de todos los recursos de la empresa (Rehg 2005). Sin embargo la migración hacia un sistema de estas características puede tener un coste demasiado elevado para empresas pequeñas y medianas, pues cada empresa cuenta ya con aplicaciones que emplea en distintas fases del proceso de producción. La integración de estas aplicaciones, a menudo muy heterogéneas, supone un problema importante tanto a nivel de costes como a nivel de esfuerzo en el desarrollo.

Al problema de integración de aplicaciones antiguas, se unen la movilidad del personal y la globalización de los mercados, creando el problema de las “islas de experiencia”. Estas islas se pueden definir como equipos de trabajo geográficamente distribuidos, donde cada uno de

¹CASA: Computer and Automation Systems Association.

²SME: Society of Manufacturing Engineers

ellos cuenta con sus propias aplicaciones de soporte instaladas en sus ordenadores de trabajo, independientes de las de sus compañeros (Huang and Mak 2001b) (Huang and Mak 2001a). La definición de interfaces entre las aplicaciones de la empresa y la creación de repositorios comunes de información es, por tanto, una necesidad creciente en la actual época de globalización.

Esta investigación, que se enmarca en el contexto de la inteligencia artificial y los sistemas basados en conocimiento, propone la creación de un modelo que facilite la representación del conocimiento basado en situaciones para representar problemas de estimación de precios en pequeñas y medianas fábricas de piezas. El modelo plantea la combinación de la representación del conocimiento de expertos con el conocimiento que se encuentra en las aplicaciones existentes en la compañía, y orienta su operación a la web. Para validar el modelo se aplicará a la estimación de precios para la elaboración de presupuestos en una compañía pequeña de fabricación de piezas mecanizadas.

1.2. Motivación

Existe un gran consenso en afirmar que la elaboración de presupuestos de forma rápida y fiable es una tarea crítica para las compañías de fabricación versátil. Habitualmente la estimación de los tiempos de fabricación y entrega, así como del precio total del pedido es realizada por un ingeniero experto en base a las características del pedido y al proceso de fabricación necesario. Sin embargo, la falta de personal altamente cualificado en las fábricas supone una carga extra de trabajo para el experto, apartándole de otras tareas importantes que también requieren su atención. Por ello, es necesario aportar nuevos modelos que ayuden a la elaboración de presupuestos, descargando de trabajo a los ingenieros que podrán emplear su tiempo en otras tareas necesarias para la fábrica.

La elaboración de un presupuesto en cualquier ámbito, implica una serie de conocimientos que son necesarios para calcular el coste de todos y cada uno de los elementos que influyen en el precio final. Para el caso concreto de una empresa de fabricación de piezas, la elaboración de un presupuesto requiere conocimientos acerca del proceso de fabricación, los materiales, las máquinas, y las personas disponibles, por citar los más importantes. Un experto, en base a sus conocimientos del proceso de fabricación y del negocio, es el encargado de realizar todas las estimaciones. El problema surge cuando el experto no está disponible para realizar los presupuestos. ¿Puede otra persona hacerse cargo de dicha tarea? ¿La realizará con la misma eficiencia? ¿Cuál es el margen de error que existiría? Una alternativa es la utilización de un sistema basado en conocimiento.

La globalización de los negocios y el constante desarrollo de las comunicaciones ha convertido internet en una herramienta fundamental para la vida de las compañías. Éstas deben adaptarse al nuevo contexto, adaptando sus sistemas si quieren sobrevivir. Sin embargo, no solo las compañías deben adaptar su tecnología a las nuevas necesidades. También las herramientas y los métodos de trabajo deben adaptarse a los requisitos que supone internet. En el ámbito de los sistemas basados en conocimiento, los métodos utilizados anteriormente pueden no ser efectivos en el nuevo entorno Web (Duan et al. 2005), por lo que son necesarias nuevas propuestas para aprovechar todo el potencial que proporciona internet y la web.

1.3. Descripción del Problema

Las nuevas necesidades de las empresas requieren la utilización de internet tanto en las relaciones con sus clientes como a nivel interno, para integrar conocimiento y soluciones. Sin embargo las técnicas empleadas por los sistemas basados en conocimiento tradicionales, pueden no ser tan efectivas en entornos web (Duan et al. 2005) y no ofrecen un modelo integral de representación en este sentido. Los avances en los sistemas basados en conocimiento e internet han dado lugar a nuevas tecnologías que poco a poco se van incorporando para mejorar la eficiencia en el proceso de fabricación (Xie 2006). En la revisión del estado del arte se han encontrado diversos sistemas basados en conocimiento desarrollados a medida para la solución de distintos problemas. Cada problema ha sido resuelto planteando una aplicación, en función del tipo de problema a resolver y sus características, especialmente cuando se trata de sistemas en los que se utiliza una interfaz Web, o en los que es necesario una comunicación con otros sistemas antiguos. Estas soluciones se basan en el desarrollo de una interfaz software a medida sobre un sistema experto convencional, o en la aplicación de herramientas comerciales que permiten la interacción específica a través de la Web. Duan et al. (2005) ya plantean la necesidad de nuevos modelos y metodologías que adapten los sistemas basados en conocimiento a los entornos Web, pero, como se mostrará en el estudio del estado del arte, no se ha encontrado un modelo de representación genérico que dé una respuesta integral a estas nuevas necesidades.

Como se ha descrito, el proceso de elaboración de un presupuesto para atender la solicitud de un cliente, implica el conocimiento profundo del proceso de fabricación de cada pieza, así como la capacidad para determinar su coste total y el margen de beneficio. Esta tarea, por tanto, solo puede ser realizada por una persona con alta cualificación y experiencia. El problema reside en la falta de expertos para realizar estos presupuestos. Esto provoca que el personal cualificado emplee gran cantidad de tiempo realizando presupuestos en lugar de trabajar en otras actividades

importantes para la empresa.

La investigación de nuevos métodos de estimación considerando el punto de vista del proceso de estimación de un experto tiene un gran potencial (Niazi et al. 2006). Las múltiples relaciones entre los parámetros de fabricación de una pieza y su influencia en el precio, hace que se compliquen los cálculos (Jahna-Shahi et al. 2001), tanto a nivel de definición del modelo como a nivel de implementación del mismo. Tal y como afirman Niazi et al., la visión del experto facilitaría la resolución de este problema. La determinación de los factores subjetivos que influyen en la elaboración de un presupuesto, es un proceso en el que el conocimiento adquirido por una persona a través de la experiencia juega un factor fundamental.

La combinación de varias técnicas de estimación es la aproximación ideal para obtener resultados satisfactorios (Duverlie and Castelain 1999), (Layer et al. 2002) y (Lan and Ding 2007), por lo que la combinación de un sistema basado en conocimiento, junto con una técnica cualitativa que represente los cálculos que realiza habitualmente el experto de una fábrica puede dar óptimos resultados.

Como se puede observar en el capítulo 2, existen diversas líneas de investigación que tratan de resolver el problema de estimar el precio en la elaboración de presupuestos. Los sistemas convencionales aportan una solución puntual al problema, pero los sistemas basados en conocimiento tienen como ventaja no solo la automatización del proceso de determinación del precio, sino que además posibilitan que el conocimiento quede almacenado en un repositorio común. Este conocimiento, además de ser fácilmente actualizable, puede ser empleado para conservar el conocimiento en la compañía más allá de la permanencia de las personas y para formar nuevos expertos.

A pesar de que existen sistemas basados en conocimiento para la estimación a los que se accede a través de la web, en la revisión del estado del arte no se han encontrado sistemas aplicados a la elaboración de ofertas, que aprovechen además un modelo de representación específico que facilite su integración en la web y que permita aprovechar las ventajas descritas.

Por tanto, la presente tesis doctoral surge de la conjunción de dos problemas identificados:

- Los modelos que se emplean en la estimación de precios en las fábricas se centran en conocimientos teóricos frente a la posibilidad de aplicar la experiencia en la estimación (Niazi et al. 2006). Considerar la experiencia de un estimador favorece:
 - La simplificación de los modelos detallados, especialmente cuando se disponen de pocos datos acerca del producto a estimar.

- La experiencia representada queda almacenada y puede ser empleada para formar a otros expertos.
- Los nuevos entornos de comunicación, y más concretamente el desarrollo de la Web, hacen necesario el planteamiento de nuevos modelos de representación que aprovechen sus ventajas (Duan et al. 2005). Además, el conocimiento puede estar tanto en el experto como en otras aplicaciones. Por lo tanto son necesarios nuevos modelos adaptados a la web, y que tengan capacidad para representar el acceso al conocimiento de dichas aplicaciones.

En adelante, entenderemos por modelos detallados aquellos modelos de estimación que pueden ser expresados por medio de fórmulas matemáticas y que están basados en características detalladas del producto y/o del proceso de producción.

En base los problemas identificados, se plantea la creación de un modelo de representación del conocimiento para representar el proceso de decisión para estimar un precio. Este modelo de representación, al estar orientado a procesos de decisión, puede ser aplicado a otros dominios, pues no está restringido ni es específico para la elaboración de presupuestos en la fabricación de piezas. Para conseguir el acceso al conocimiento de las distintas aplicaciones que intervienen en el proceso, el modelo debe representar la forma de interactuar con las mismas.

1.4. Justificación

Como se ha indicado, la elaboración de un presupuesto en la fabricación de piezas requiere el conocimiento tanto del proceso de fabricación necesario como de los costes que conlleva el mismo. Por ello es necesario que uno o varios trabajadores de la fábrica posean los conocimientos suficientes se encarguen de llevarlo a cabo. Existen distintos modelos para el cálculo de precios y costes que se han empleado en la industria de fabricación. Sin embargo, como se mostrará en el apartado Estado del Arte, los modelos basados en la experiencia no han sido completamente explotados aún, constituyendo un área de investigación abierta (Niazi et al. 2006).

A continuación se plantean las cuestiones que fundamentan la presente de tesis doctoral.

1.4.1. ¿Por qué es importante la estimación del precio?

Un presupuesto fiable y realizado en un periodo corto de tiempo es fundamental en las empresas que fabrican bajo demanda. Un retraso en la elaboración del mismo puede suponer que el cliente se decida ante una oferta mejor por parte de una compañía competidora. Por otro lado

una respuesta rápida pero mal estimada puede suponer pérdidas económicas para la compañía, si se estima un precio por debajo de los costes de producción, o la pérdida del cliente, si el precio estimado es demasiado elevado.

Hvam et al. (2006), entre otros autores, señalan que, a pesar de la importancia para la fábrica del éxito en sus presupuestos, los ingenieros expertos dedicados a esta labor tienen otras muchas tareas importantes que desempeñar dentro de las fábricas. Esto provoca que, en algunos casos, la elaboración de presupuestos llegue a demorarse incluso semanas.

1.4.2. ¿Por qué representar el conocimiento de un experto?

Un problema se puede resolver siguiendo una serie de pasos hasta obtener la solución, empleando una determinada metodología (González and Dankel 1993). Esta aproximación puede llevar asociado un coste inaceptable de tiempo y dinero cuando es aplicada por una persona que no es experta en el dominio. En el caso de la estimación de precios, las relaciones entre las distintas características que puede tener la pieza y su coste son numerosas. La combinación de dichas características hace que el número de posibilidades se eleve exponencialmente, dificultando su representación por medio de un modelo matemático (Jahna-Shahi et al. 2001).

Dado que en el momento de elaborar el presupuesto es común no disponer de la información suficiente como para aplicar modelos detallados, es necesaria una aproximación diferente al problema de la estimación. Un experto en la fabricación, a lo largo de su trabajo anterior ha adquirido unos conocimientos que, si bien puede que no sean rigurosos, le sirven para reconocer las características principales del problema y formular una solución óptima. Un sistema basado en conocimiento permite la representación del conocimiento del experto y su aplicación a la estimación del precio para la elaboración del presupuesto de fabricación. Niazi et al. (2006) señalan que los sistemas basados en conocimiento actuales conceden más peso a los conocimientos teóricos que al conocimiento práctico derivado de la experiencia práctica de los trabajadores, por lo que la investigación en este aspecto tiene un potencial elevado.

1.4.3. ¿Por qué un modelo nuevo de representación?

Como se ha indicado en el apartado anterior, los modelos empleados actualmente en el dominio estudiado están basados principalmente en modelos teóricos (Niazi et al. 2006) y son necesarios nuevos modelos que recojan la experiencia práctica de los trabajadores cualificados. Dentro de la experiencia no solo será necesario considerar conocimientos en forma de reglas que representen heurísticas que el experto emplea en su trabajo diario. El conocimiento acerca de

otros sistemas que se emplean a diario en las fábricas es fundamental tanto en la estimación de precios como en la representación de procesos en general. El conocimiento puede estar en una hoja de cálculo, en una base de datos o en una aplicación de gestión de materiales de la fábrica. Además los últimos avances en la tecnología hacen que la interacción en los distintos procesos no sea necesariamente con personas, sino que se pueden plantear escenarios de interacción con otros sistemas software. Por ello las actuales necesidades de información de las fábricas requieren por un lado el acceso a información procedente de aplicaciones heterogéneas que dan soporte a los procesos industriales, y por otro el empleo de las tecnologías de comunicación para facilitar el acceso remoto a la información y el desarrollo de sistemas de comercio electrónico. Duan et al. (2005) señalan que las técnicas de representación del conocimiento tradicionales pueden no ser tan efectivas en los entornos actuales y son necesarios nuevos modelos y metodologías. Estos modelos de conocimiento sustentarán sistemas expertos integrados con las nuevas tecnologías de la comunicación.

Grove (2000) ya señala la necesidad de nuevas técnicas para aplicar los sistemas basados en conocimiento en el ámbito de internet en general y la web en particular. Adams (2001) señala que no todos los sistemas expertos pueden ser adecuados para los entornos web. Desde entonces numerosos sistemas basados en conocimiento han sido desarrollados, pero, como indican Duan et al. (2005) las tecnologías siguen siendo similares a las señaladas por Grove (2000) y siguen siendo necesarios nuevos modelos y metodologías.

El modelo de representación que sirve de base al sistema basado en conocimiento, debe permitir el acceso al conocimiento de otros sistemas de la fábrica y facilitar el acceso a la información, incorporándolo como un todo al conocimiento que se está representando. Además, dicho modelo debe poder ser implementable, es decir, que sobre dicho modelo se pueda definir un sistema capaz de interpretarlo y obtener resultados.

1.5. Objetivos

El objetivo y la principal aportación de la tesis doctoral, en base a los problemas identificados, es la definición de un modelo conceptual de representación del conocimiento basado en situaciones, capaz de representar problemas de toma de decisión, en particular de estimación de precios en la fabricación de piezas, en los que se requiere intercambio de información, ya sea con otros sistemas o con personas. Para ello se establecen los siguientes objetivos:

1. El modelo debe definir los elementos que permitan representar el proceso que sigue un experto para determinar un precio. Esta representación incluye:

- Las actividades que realiza el experto.
 - Las actividades que realizan otros agentes que puedan intervenir en el proceso. Con este objetivo se modela el intercambio de información con otros sistemas y con otras personas.
2. El modelo debe permitir representar el conocimiento relativo a otros modelos detallados de estimación que el experto emplee para la determinación del precio.
 3. Permitir que el modelo se integre con las tecnologías de comunicación actuales para permitir el intercambio de información. En concreto el modelo se orientará a la interacción a través de la web.
 4. Como continuación del objetivo anterior, el modelo debe permitir la automatización de la estimación de precios a partir de la definición de un proceso.
 - Permitiendo el intercambio de información con otras aplicaciones.
 - Permitiendo el intercambio de información con otros agentes implicados en el proceso.
 5. Como consecuencia de la automatización del proceso de estimación, se pretende liberar al experto de la tarea de elaboración de presupuestos. Un proceso automatizado servirá para que puedan hacerse cargo del proceso otras personas menos cualificadas, permitiendo al experto dedicarse a otras tareas en la fábrica.
 6. Validar el modelo propuesto, aplicándolo al dominio de la estimación de presupuestos de fabricación. Para ello se modelará el proceso de estimación de un experto en una fábrica de piezas mecanizadas.
 7. Validar la capacidad para automatizar el proceso modelado. Para ello se implementará un motor capaz de reproducir un proceso representado con el modelo propuesto y se automatizará la representación utilizada para validar el modelo, comparando los resultados con la estimación del experto.

1.6. Metodología

Para alcanzar los objetivos que se han definido en el apartado anterior, se han llevado a cabo los siguientes pasos (figura 1.1):

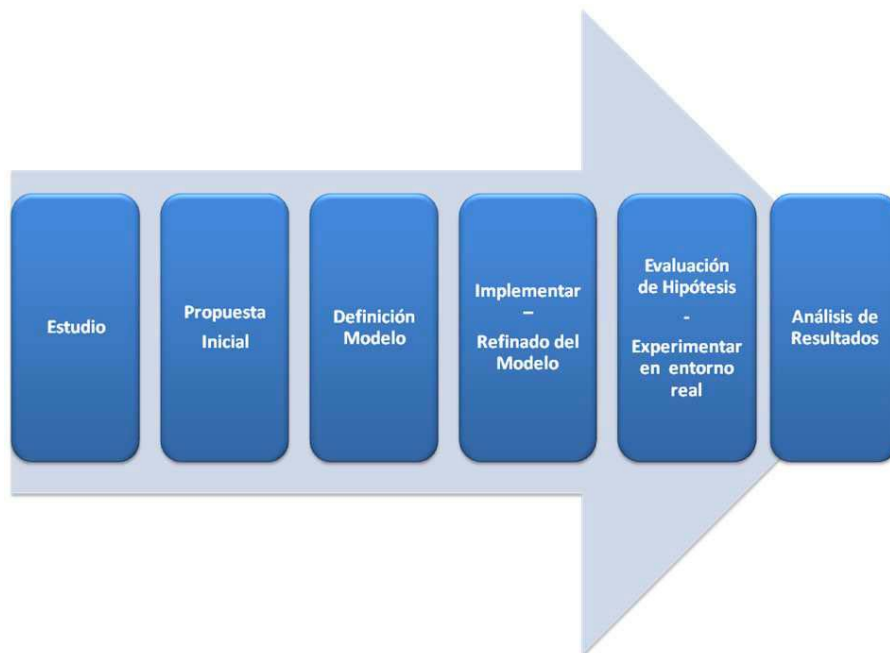


Figura 1.1: *Etapas de la investigación.*

1. Estudio del estado del arte. Para determinar la viabilidad del proyecto se ha comenzado con el estudio de los trabajos relacionados con el área en el que se desarrolla la presente tesis doctoral. Fruto de este estudio se han identificado los métodos y aplicaciones que se exponen en el capítulo 2. Este estudio se ha dividido en los siguientes pasos:
 - Estudio de los Sistemas Basados en Conocimiento basados en Web y los modelos de representación existentes.
 - Estudio de aspectos generales relativos a la industria y los sistemas expertos.
 - Estudio de los métodos de estimación de costes y precios que se emplean para la fabricación.
 - Estudio de aplicaciones para la estimación de presupuestos y costes de fabricación que no emplean técnicas de inteligencia artificial.
 - Estudio de sistemas basados en conocimiento para la estimación de presupuestos y costes de fabricación. Si bien los sistemas de cálculo de costes no elaboran presupuestos, la estimación de costes es una parte fundamental en su realización.
2. Análisis del problema de la estimación de precios y propuesta inicial. El desarrollo de la tesis surge a partir del problema detectado en una fábrica. En paralelo al estudio del estado

del arte se comenzó con la definición del problema y el estudio de distintas aproximaciones a la solución que pudieran emplearse. El estudio del problema se ha dividido en los siguientes pasos:

- Análisis del proceso de estimación que sigue el experto.
 - Identificación de la información que necesita el experto para estimar el precio y las fuentes para su obtención.
 - Identificación de los cálculos que realiza el experto en su estimación.
3. Definición del modelo de representación. Para representar el conocimiento se ha definido un modelo de representación que integra el sistema basado en conocimiento con la web. Para facilitar la aplicación dicho modelo se ha desarrollado una herramienta de modelado que facilita la gestión de los objetos representados, y un motor capaz de interpretarlo y ejecutarlo. Este desarrollo se ha realizado en paralelo a la definición del problema.
 4. Validación. En esta fase se ha comprobado que el modelo desarrollado puede representar el conocimiento relativo al problema, aplicándolo a la creación de un sistema basado en conocimiento para una fábrica, y que dicho sistema es capaz de realizar la estimación de precios de forma similar al experto de una fábrica.
 5. Análisis de los resultados. Tras la fase de validación se han estudiado los resultados obtenidos para determinar la validez de la propuesta.
 6. Documentación. La tarea de documentación se ha llevado a cabo a lo largo de todo el proceso de elaboración de la presente tesis doctoral. En las primeras fases, la tarea de documentación registra los resultados del estudio del estado del arte y del análisis del problema. Durante el desarrollo, se documentaron las características del modelo de representación creado y los aspectos relevantes del sistema basado en conocimiento. En la fase de validación, se han documentado los resultados de los procedimientos de validación seguidos, descritos en el capítulo 8. Finalmente se redactó la documentación definitiva, que constituye la presente memoria, a partir de la documentación generada a lo largo de las distintas fases.

1.7. Estructura de la Tesis Doctoral

En adelante, el presente documento se estructura de la siguiente forma:

-
- **Estado del arte.** En este capítulo se revisan los trabajos realizados hasta la fecha en el área de creación de presupuestos de fabricación y se repasan los aspectos relativos a la representación del conocimiento.
 - **Propuesta de Solución.** En este capítulo se define cuál es el problema que se pretende solucionar, y se describe una aproximación a la solución propuesta.
 - **Modelo de Representación del Conocimiento.** En este capítulo se describe de forma detallada el modelo de representación del conocimiento propuesto como solución.
 - **Algoritmo de Solución.** Este capítulo describe el algoritmo para automatizar un proceso modelado a partir del modelo de representación propuesto. Para la validación del modelo se describe una implementación de dicho algoritmo en un motor de solución.
 - **Herramienta de Modelado.** En este capítulo se describe una herramienta que soporta la representación del conocimiento con el modelo propuesto. A partir de dicha herramienta se pueden representar procesos de decisión que puedan ser automatizados por el motor de solución.
 - **Representación del Proceso de Elaboración de Presupuestos.** En este capítulo se describe la representación del proceso de elaboración de presupuestos que desarrolla una empresa de fabricación de piezas, empleando el modelo propuesto.
 - **Validación.** En este capítulo se aborda la validación del modelo propuesto y su aplicación al dominio de la estimación de precios para la elaboración de presupuestos.
 - **Conclusiones y Futuras Líneas de Investigación.** En este capítulo se plantean las conclusiones de la investigación que se ha llevado a cabo y se apuntan las líneas en las que se puede continuar el trabajo realizado.

Este capítulo realiza una revisión de los conceptos y la investigación realizada alrededor del problema de la elaboración de presupuestos y los sistemas basados en conocimiento. En primer lugar se realiza una introducción a los sistemas basados en conocimiento y a los sistemas expertos basados en Web. A continuación se realiza un repaso de diferentes propuestas de mejora del proceso de elaboración de presupuestos, así como los principales modelos económicos empleados en la estimación de costes y precios. Finalmente se revisan distintos sistemas, tanto convencionales como basados en conocimiento, fruto de líneas de investigación relacionadas con la estimación de precios para la elaboración de presupuestos en la fabricación de piezas. También se repasan los trabajos de investigaciones recientes sobre la representación del conocimiento y procesos de negocio en la red.

2.1. Sistemas Basados en Conocimiento

Los términos sistema experto, sistema basado en conocimiento o sistema experto basados en conocimiento son utilizados frecuentemente como sinónimos (Giarranto and Riley 1998). Un sistema basado en conocimiento se puede definir como un sistema informático que utiliza conocimiento sobre un dominio concreto para resolver un problema (González and Dankel 1993). Los sistemas expertos son aquellos sistemas basados en conocimiento que utilizan el conocimiento de un experto en un dominio determinado (Giarranto and Riley 1998). Sin embargo, el término sistema experto se aplica frecuentemente a cualquier sistema que utiliza tecnología de sistemas expertos (lenguajes específicos, programas o incluso hardware diseñado para ayudar al desarrollo y ejecución de sistemas expertos) (Giarranto and Riley 1998). Los sistemas expertos son probablemente la tecnología de Inteligencia Artificial más madura en su aplicación industrial (Pham and Pham 1999). La figura 2.1 muestra gráficamente la relación entre los SBC, SSEE y la Inteligencia Artificial.

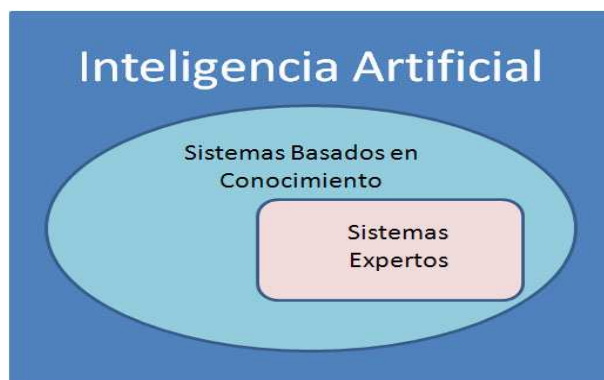


Figura 2.1: Relación entre sistemas basados en conocimiento y sistemas expertos (Gómez et al. 1997).

2.1.1. Evolución Histórica

La descripción de la evolución de la inteligencia artificial y los sistemas basados en conocimiento se basa en los estudios de Giarranto and Riley (1998), González and Dankel (1993) y artículos relacionados que se citan a lo largo del texto.

El nacimiento de la Inteligencia Artificial se sitúa en la década de los 50. Los primeros trabajos consistían en juegos de ajedrez y damas, capaces en algunos casos de vencer a jugadores humanos y de aprender por medio de técnicas de aprendizaje automático.

En el año 1957 Rosenblatt inventa el Perceptrón y comienza el desarrollo del General Problem Solver (GPS) por parte de Newell, Shaw y Simon. GPS (Newell and Simon 1972) es una técnica de resolución de problemas de propósito general, cuyo objetivo era resolver un problema de la misma forma en que lo haría un humano. GPS presenta la característica de separar el conocimiento de los métodos de resolución del problema, por cuanto el sistema es de propósito general, el método de resolución es siempre el mismo, determinando los datos el problema a resolver y el conocimiento necesario para resolverlo.

En 1958 John McCarthy desarrolla el lenguaje LISP en el MIT, que aún hoy es utilizado. Más tarde, en el año 1965, comienza el primer trabajo de desarrollo en sistemas expertos: DENDRAL (Buchanan and Feigenbaum 1978). El sistema reconocía estructuras moleculares de compuestos desconocidos, de la misma forma que lo hacía un químico humano. En ese mismo año, Lofti Zadeh (Zadeh 1965) publica un artículo presentando la teoría de conjuntos borrosos, que será la base para los sistemas de lógica borrosa.

En el año 1973 surge MCYN (Shortliffe 1976), un sistema experto de diagnóstico médico que es el primero en separar explícitamente la base de conocimiento del motor de inferencia. Gracias a esta capacidad surge EMYCIN (Empty MYCIN), el primer shell de sistemas expertos.

Hasta este momento, los sistemas expertos se creaban a medida, y no podían ser reutilizados sus componentes.

En 1976 comienza el desarrollo de PROSPECTOR (Duda et al. 1978), un sistema experto para prospección minera. Este sistema descubrió un depósito mineral valorado en 100 millones de dólares. Por otro lado XCON (McDermott and Bachant 1984), desarrollado también en la década de los 70, fue el primer sistema experto comercial. Este sistema utilizó para la configuración de ordenadores, y sirvió para que Digital Equipment Company (DEC), ahorrara una cantidad estimada en millones de dólares.

Desde finales de los 70, convergen los tres conceptos básicos subyacentes en los sistemas expertos actuales: la utilización de reglas para modelar la forma en que los humanos resuelven problemas, la separación entre el conocimiento y el motor de inferencia, y la utilización del conocimiento como clave para la experiencia (Giarranto and Riley 1998).

El éxito de estos primeros sistemas, junto con los beneficios que las compañías obtuvieron a partir de su aplicación, hicieron que las empresas se interesaran en el desarrollo de sistemas expertos. Se incrementó la inversión, y se desarrollaron cientos de sistemas en diversos dominios. Hasta este momento gran parte de los sistemas expertos que se desarrollaban estaban basados en reglas.

A mediados de los 80 la NASA desarrolla CLIPS, que aplica el algoritmo de pattern matching RETE (Forgy 1982). Durante la segunda mitad de la década de los 80 la tendencia cambia hacia sistemas orientados a objetos.

En los últimos años, las aplicaciones industriales de los sistemas expertos han ido creciendo, gracias a la madurez de la tecnología y la gran expansión de la informática. Se han realizado estudios acerca de los factores de éxito de los sistemas expertos aplicados a la fabricación (Guimaraes et al. 1995)(Metaxiotis, Askounis and Psarras 2002), en la ingeniería (Nurminen et al. 2003), y, en distintas áreas de negocio, analizando futuras líneas de investigación (Metaxiotis and Psarras 2003). Rao and Miller (2004) identifican nuevos puntos de aplicación para los sistemas expertos en el análisis de productividad. La tendencia actual apunta a que los sistemas expertos se integrarán en aplicaciones o paquetes de software más general (Metaxiotis, Askounis and Psarras 2002). Un estudio completo acerca de las metodologías y aplicaciones de sistemas expertos entre los años 1995 y 2004 puede encontrarse en el estudio de Liao (2005).

Entre estos sistemas expertos aplicados a la industria, Metaxiotis, Psarras and Askounis (2002) presentan un sistema experto basado en reglas para realizar la programación de una planta industrial, utilizando la aplicación NEXPERT OBJECT (TM) de Neuron Data. Otras aplicaciones de los sistemas expertos en la industria están relacionadas con el asesoramiento para la toma

de decisiones. Por ejemplo, Balachandra (2000) emplea un sistema experto basado en reglas para identificar las variables de contexto ideales para el proyecto de desarrollo de un nuevo producto. Cuando las ha identificado propone los factores de éxito asociados al proyecto y la prioridad de cada uno de ellos. También la implementación de servicios de calidad para el cliente son un área de aplicación importante, como indican Eppinette et al. (1997).

La lógica borrosa es otra técnica que se ha empleado con éxito en el desarrollo de sistemas expertos para la industria. Por ejemplo, Jiang and Hsu (2003) emplean un sistema de lógica borrosa para evaluar la viabilidad de fabricar un producto determinado a partir de datos en las primeras fases de diseño. Collan and Liu (2003) combinan la lógica borrosa con un sistema de agentes para crear un sistema de soporte a la decisión para la asignación de presupuesto en las inversiones en bienes capital. Dentro del área de CIM, Kumar and Midha (2004) aplican la lógica borrosa para crear un sistema experto que ayude a determinar los componentes que debe tener un sistema de Product Data Management en una determinada empresa, en base a sus características y necesidades específicas.

A nivel de fabricación de piezas mecanizadas también existen diferentes ejemplos de sistemas expertos a distintos niveles. Nikolopoulos and Assimakopoulos (2003) presentan un sistema experto para realizar predicciones en el entorno industrial y de negocios, integrando un sistema de soporte a la decisión con un sistema experto. Tang et al. (2001) muestran un sistema basado en reglas de producción para optimizar el diseño para sistemas de estampado. Arezoo et al. (2000) describen un sistema experto creado en PROLOG, para la selección de herramientas y parámetros de corte en la fabricación de piezas a partir de sus características. El artículo de Er and Dias (2000) muestra un sistema experto basado en reglas creado con la herramienta Crystal para decidir el proceso de producción de piezas. Abdalla and Knight (1994) aplican un sistema experto basado en reglas para determinar las operaciones de mecanización necesarias para fabricar una pieza a partir de las características requeridas. El sistema, a partir de dichas operaciones, calcula el coste de fabricación. Siong et al. (1992) utilizan un sistema experto creado con LISP para la selección de los módulos de fijación para la mecanización de una pieza, determinando las operaciones necesarias y los parámetros de operación a partir de las características de la pieza, y calculando el precio de la fijación y de la pieza mecanizada. Hunter et al. (2007) presentan un modelo de representación para el proceso de inspección de las características geométricas de las piezas fabricadas, creando un sistema basado en conocimiento alrededor para automatizar la planificación y ejecución de actividades de inspección.

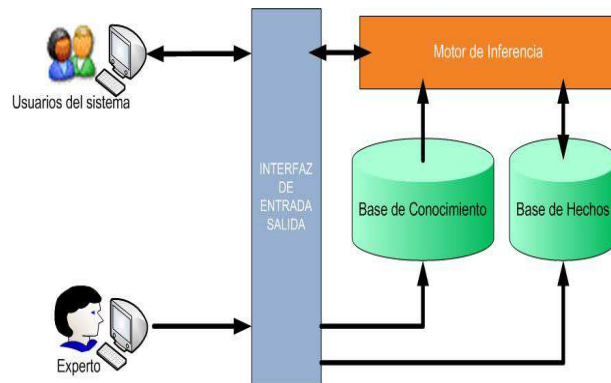


Figura 2.2: Arquitectura general de un sistema experto (Gómez et al. 1997).

2.1.2. Características Generales de los Sistemas Basados en Conocimiento

Generalmente los sistemas basados en conocimiento se diseñan de forma diferente a los sistemas convencionales, debido a que, en general, los sistemas basados en conocimiento tratan de solucionar problemas que no tienen una solución algorítmica (Giarranto and Riley 1998). La figura 2.2 muestra la arquitectura básica de un sistema basado en conocimiento, que principalmente consta de (Gómez et al. 1997):

- Una interfaz de entrada/salida, para que el usuario pueda enviar datos y recibir las respuestas del sistema, y el experto pueda introducir sus conocimientos.
- Una base de hechos en la que se almacena la información que se recibe del entorno.
- Una base de conocimiento, en la que se encuentran las reglas que un experto en el dominio del problema utiliza en la toma de decisiones.
- Un motor de inferencia, para elegir la decisión a tomar partiendo de las reglas y los conocimientos adquiridos.
- Un módulo de explicación de las decisiones que ha tomado.

Las técnicas de representación del conocimiento empleadas con más frecuencia en el desarrollo de sistemas expertos son:

- Lógica de predicados: se basa en proposiciones que expresan relaciones entre objetos así como cualidades y atributos de dichos objetos. Estas relaciones entre atributos se denominan predicados.

- Reglas: son una forma natural de expresar el conocimiento (González and Dankel 1993). Tienen una estructura del tipo If <condición> then <acción>, donde el lado izquierdo de la regla se denomina antecedente o premisa, y el lado derecho se denomina consecuente o conclusión. El consecuente de una regla que se cumple puede generar un nuevo conjunto de hechos o añadir nuevos hechos a los que ya existían. La inferencia en estos sistemas se realiza encadenando reglas. Cuando el razonamiento se realiza desde hechos iniciales para llegar a unas conclusiones, se denomina encadenamiento hacia adelante (forward chaining). Si, por el contrario, el razonamiento parte de las conclusiones para determinar los hechos que las producen, se denomina encadenamiento hacia atrás (backward chaining). Estos sistemas también son llamados sistemas de producción.
- Redes semánticas: representan objetos, conceptos o situaciones en forma de nodos y relaciones entre ellos (Giarranto and Riley 1998).
- Frames o marcos: representan el conocimiento relacionado con un problema determinado. Contienen un conjunto de pares atributo-valor que representan una situación, un objeto o un proceso del mundo real (González and Dankel 1993). Los marcos organizan los conocimientos del dominio en árboles, también llamados jerarquías, o en grafos, ambos constituidos por especialización de conceptos generales en conceptos más específicos, siendo las técnicas de inferencia utilizadas: equiparación, para clasificar entidades en una jerarquía; herencia simple y herencia múltiple para compartir propiedades distribuidas en la jerarquía de conceptos; y valores activos y métodos para representar la conducta del sistema (Gómez et al. 1997).
- Objetos: son una colección de información que representa una entidad perteneciente al mundo real y una descripción de cómo esta información es manipulada (González and Dankel 1993).
- Lógica Borrosa: se basa en representar el conocimiento utilizando variables lingüísticas de vaga definición, similares a las que utiliza un experto en su trabajo diario, expresándolas en forma de conjuntos y funciones de pertenencia.

Otros tipos de razonamiento empleados en la creación de sistemas basados en conocimiento son las redes de neuronas artificiales y el razonamiento basado en casos. Sin embargo algunos autores ven el razonamiento basado en casos como un tipo específico de reglas (Chan 2005).

2.2. **Sistemas Expertos Basados en Web**

En los últimos años Internet se ha convertido en un área de negocio. Junto con la evolución de la informática y las comunicaciones, la ingeniería del conocimiento ha avanzado en el desarrollo de los sistemas basados en conocimiento y sistemas expertos, aportando sistemas de gestión del conocimiento cada vez más efectivos. El objetivo de estos sistemas es ayudar a la creación, transferencia y aplicación del conocimiento en las organizaciones (Alavi and Leidner 2001). Cada vez se identifica más el conocimiento como un activo muy importante de las organizaciones. El conocimiento aporta importantes ventajas competitivas para el que lo posee, y es necesario gestionarlo adecuadamente.

Se pueden distinguir tres clases de sistemas expertos en función de su conexión a internet:

- **Sistemas Expertos stand-alone.** Se trata de sistemas expertos que se ejecutan en un ordenador y no se interconecta con ningún otro.
- **Sistemas Expertos basados en Internet.** A diferencia de los anteriores, el sistema basado en Internet accede y envía información a otros equipos a través de Internet para desarrollar su función.
- **Sistemas Expertos basados en Web.** Dentro de los sistemas expertos basados en Internet, los sistemas basados en Web son aquellos que interactúan con el usuario a través de la Web.

A mediados de los 90 comienzan a descubrirse las nuevas posibilidades que aporta la integración de las tecnologías web con los sistemas expertos y otras técnicas de inteligencia artificial (Parkinson et al. 1994).

Los sistemas expertos en esta evolución se han aplicado a múltiples áreas, proliferando cada vez más los sistemas expertos a los que se accede a través de internet gracias al espectacular auge de esta tecnología en los últimos años. Internet puede ser tanto una fuente de datos para ofrecer soluciones, como un medio para acceder a la ayuda del sistema desde cualquier lugar (Lin et al. 2002). Un ejemplo de sistema experto basado en web es el desarrollado por Slater (1999) a partir de la herramienta RuleWorks, que fue empleado para asesorar acerca de los componentes más adecuados en la fabricación de ordenadores en la empresa Digital (que posteriormente se convirtió en Compaq). Otros ejemplos de sistemas expertos basados en web son aplicaciones a la planificación estratégica de Márketing (Li 2005), o de apoyo a la toma de decisiones, calculando la complejidad de distintos aspectos del proceso productivo de la empresa, a partir de

sus características concretas (Efstathiou et al. 2002). En otro ámbito, Thomson and Willoughby (2004) proponen crear un sistema experto que ayude a decidir la utilización de un determinado herbicida.

Grove (2000) realizó un estudio en el que aporta las principales razones por las que es beneficiosa la utilización de sistemas expertos basados en web, entre las que se incluyen la facilidad de acceso remoto al sistema, las posibilidades de interfaz de usuario que proporcionan los navegadores y la independencia de la plataforma para su ejecución.

Algunos de estos sistemas basados en web han sido desarrollados ad-hoc, utilizando tecnologías como PERL, ASP o CGI, combinadas con HTML para generar la interfaz de usuario. Por debajo de estas tecnologías web, se encuentran lenguajes específicos como Prolog, CLISP o JESS para representar el conocimiento y las reglas de decisión. También existen herramientas comerciales para generar sistemas expertos basados en web, como EXSYS de CORVID, Expertise2Go o ILOG.

Tanto los sistemas creados ad-hoc, como las aplicaciones comerciales, siguen fundamentalmente dos modelos de comunicación: la comunicación se realiza a través de un applet, o bien se utiliza una interfaz de tipo CGI, ASP o similar.

Kim et al. (2005) clasifican los sistemas basados en web en función de la ubicación del motor de inferencia:

- Sistemas ubicados en el servidor. El sistema experto corre en el servidor, y el usuario accede a través del navegador. Kim et al. (2005) clasifican estos sistemas a su vez en:
 - Programas CGI. El servidor web realiza una llamada al programa pasándole los parámetros según el estándar CGI.
 - Scripts en el lado del servidor. El motor de inferencia se ha desarrollado en entornos como JSP, PHP o ASP.
 - Embebidos en el servidor. El sistema experto es un módulo que se integra como parte del servidor mediante una API.
- Sistemas en el cliente. El sistema experto se descarga y se ejecuta en el cliente. A su vez, Kim et al. (2005) clasifica estos sistemas en:
 - Visores externos. El sistema experto es desarrollado como un programa independiente, y se invoca a través del navegador Web.
 - Applet de java. El sistema está implementado en un applet que se ejecuta en el navegador web del cliente.

Tomando como base los problemas de rendimiento que pueden tener los sistemas basados en web, ya sea en el cliente o en el servidor, Kim et al. (2005) opinan que en este tipo de sistemas es más eficiente una aproximación basada en HTML, proponiendo un mecanismo de inferencia basado en los hiperenlaces de los documentos HTML.

Sin embargo los sistemas expertos basados en web no siempre son una solución óptima para todos los problemas. En ocasiones no es factible utilizar la web como medio de acceso al sistema experto, bien por motivos de seguridad de las comunicaciones y de los datos, o bien por tiempo de respuesta, por la falta de consideraciones específicas de diseño (Adams 2001) o por falta de conectividad.

Los problemas de seguridad de las comunicaciones y de los datos en determinados casos pueden ser resueltos con técnicas de codificación de datos. Por otro lado, los aspectos de seguridad en el acceso pueden paliarse con la utilización de certificados e incluso, para sistemas especialmente sensibles, técnicas de seguridad biométrica. En cuanto a los problemas de disponibilidad de la red de comunicaciones y velocidad de transmisión, a pesar de la constante evolución de los sistemas de comunicación, suponen un factor de incertidumbre y un riesgo que debe ser tomado en consideración a la hora de implementar un sistema para un dominio concreto.

Sin embargo, el factor acerca de los criterios de diseño para el sistema experto basado en web no es un problema menor. Por tanto no es suficiente con que el sistema sea accesible a través de internet, sino que la representación del conocimiento y el motor de inferencia deben orientarse también a la web para obtener un resultado óptimo.

El creciente desarrollo de aplicaciones web que incorporan sistemas expertos ha motivado que los ingenieros web comiencen a tratar la problemática específica de la integración de estas tecnologías (Dokas 2005). Duan et al. (2005) identifican una serie de oportunidades que quedan pendientes para aprovechar las ventajas de los sistemas expertos basados en web, como el desarrollo de nuevas metodologías, herramientas y técnicas de representación del conocimiento, pues las tradicionales pueden no ser adecuadas en un entorno web.

En el ámbito de la industria, la integración de tecnologías de inteligencia artificial y de conocimiento con la tecnología de internet aplicada a los problemas de fabricación, lleva a un nuevo paradigma que permite alcanzar ciclos de desarrollo muy cortos y permite aprovechar oportunidades repentinas que surgen en el mercado (Xie 2006). En la línea de integración de gestión de conocimiento y gestión de procesos de negocio Jung et al. (2007) proponen una arquitectura cuyo objetivo es combinar las ventajas de ambos paradigmas: sin embargo esta aproximación se centra en aspectos relativos a la definición, utilización y evolución del conocimiento pero no en la creación de sistemas basados en dicho conocimiento .

La utilización de cuestionarios se adapta perfectamente a la aplicación de tecnologías web asociadas a sistemas expertos. La interacción entre el usuario y el sistema experto implica un diálogo web que debe ser controlado (Yu 2004). Frente a aplicaciones que encadenan las preguntas de forma estática, Yu (2004) propone un módulo para generar un diálogo, que se manifieste en una elección inteligente de la siguiente cuestión a realizar, y en conjuntos de posibles respuestas que reflejen el conocimiento que posee el sistema. Este es un caso claro de aprovechamiento de internet tanto para llegar a los clientes, como para ofrecerles soluciones personalizadas, gracias a un sistema basado en conocimiento.

2.3. El Proceso de Oferta y su Optimización

Se han realizado numerosas clasificaciones sobre las compañías que se dedican a la fabricación de productos en función de las características del proceso de fabricación y de los tipos de productos que tratan. Por un lado, existen compañías que fabrican productos estandarizados que lanzan al mercado, o que mantienen en stock para hacer posteriormente frente a los pedidos que reciben. Por otro lado, otras compañías fabrican bajo demanda los encargos que reciben: las compañías de fabricación versátil son aquellas negocian con el cliente y pujan con otras compañías por cada orden que reciben, personalizando el producto para cada una de ellas (Kingsman et al. 1996). Dentro de las compañías de fabricación versátil se puede distinguir entre compañías Engineering-to-Order (ETO), en las que el pedido del cliente requiere también la elaboración de un diseño, y las empresas Make-to-Order (MTO), que fabrican el producto a partir del diseño que proporciona el cliente (Kingsman and de Souza 1997). Estos conceptos han sido ampliamente estudiados y varios autores han propuesto completas taxonomías que amplían, definen y clasifican estos términos, como es el caso de Amaro et al. (1999).

Las compañías de fabricación versátil suelen ser pequeñas y medianas empresas que se dedican a fabricar bajo demanda los pedidos recibidos. Pero para conseguir el pedido, es necesario negociar previamente con el cliente las condiciones de cantidad, tiempo de entrega y precio. Este proceso es denominado proceso de petición de oferta, en inglés Request for Quotation (RFQ).

El proceso de RFQ es vital en las compañías de fabricación versátil. Dejando a un lado el tiempo de entrega del producto, una estimación de precio por debajo del coste de producción puede provocar pérdidas económicas para la compañía, en algunos casos importantes, y retrasos en la entrega del producto. En el otro extremo, una estimación de precio muy por encima del precio de coste, puede conllevar que el cliente realice el pedido a una empresa competidora que le ofrezca un precio mejor. Otro factor determinante para ganar o perder un pedido, es el tiempo

de respuesta al cliente desde que éste solicita el presupuesto. Una respuesta demasiado lenta puede suponer que el cliente decida encargar el pedido a otra compañía que respondió mucho antes.

Un estudio de 12 compañías de fabricación versátil concluyó que el proceso de estimación de precio seguía un conjunto de pasos comunes (Kingsman et al. 1996) y (Kingsman and de Souza 1997). Para determinar el precio, es necesario estimar el coste de distintas áreas que intervendrán posteriormente en la fabricación del producto. Normalmente la estimación de coste es realizada por un ingeniero o un experto en cada área de fabricación. Este personal experto no suele estar dedicado exclusivamente a la estimación, sino que tiene asignadas otras tareas relacionadas con el diseño o la producción. Por ello habitualmente necesitan cierto tiempo para realizar la tarea de estimación. Si a esto añadimos el hecho de que el número de expertos cualificados para la estimación del coste en las compañías suele ser reducido, la estimación puede llegar a demorarse incluso varias semanas.

Dada la importancia del proceso de oferta, distintas líneas de investigación tratan de optimizarlo, unas de forma integral, y otras mejorando partes concretas del proceso.

Desde el punto de vista de la negociación entre el cliente y los distintos suministradores, la optimización se puede enfocar desde una doble vertiente: la automatización del proceso a través de internet, y el soporte a la decisión. Así por ejemplo se han desarrollado sistemas de workflow basados en servicios web y agentes para la colaboración entre empresas, que han sido aplicados en procesos de recepción de pedidos y negociación de precios entre un cliente y varios proveedores (Wang et al. 2006). Siguiendo la línea de mejorar la negociación de precio, se están desarrollando e-marketplaces donde se ponen en contacto empresas y suministradores (Beil and Wein 2003). Aprovechando que estos e-marketplaces permiten automatizar parcialmente el proceso de RFQ, se han propuesto mecanismos de pujas entre una fábrica y sus suministradores para seleccionar la mejor oferta entre varios suministradores. Calosso et al. (2003) estudian el proceso de negociación de ofertas en el entorno de empresas MTO, proponiendo tres modelos de programación lineal para ser utilizados por las distintas partes implicadas en la evaluación de ofertas y elaboración de pujas.

El término *proceso de especificación* (Hvam and Have 1998) engloba las tareas de ventas, diseño e ingeniería para un producto. Este proceso incluye la gestión de pedidos, pujas, configuración de producto, etc. En ocasiones, para reducir los tiempos de este proceso de especificación y ayudar a mejorar la competitividad de la empresa, es necesaria la reingeniería de procesos. Hvam et al. (2004) propone un método para el análisis y reestructuración de procesos de especificación del producto, integrando las nuevas tecnologías de información. El proceso de reingeniería pue-

de incluir la introducción de nuevos elementos software que den soporte al nuevo proceso, como pueden ser los sistemas expertos, para conseguir reducir el tiempo de elaboración de una oferta. En la línea de mejorar el proceso de ofertas, Kennedy and Shao (1989) han aplicado sistemas expertos representando las reglas sobre cómo mejorar el proceso RFQ en pequeñas fábricas. También Ahmadi and Tirupati (1996) aplican sistemas de soporte a la decisión para evaluar pedidos de un cliente y decidir si aceptarlos o no en base a la programación de actividades de la fábrica.

La especial naturaleza de las empresas ETO restringen la aplicación de metodologías dentro de la cadena de suministros, lo que afecta también al proceso de elaboración de ofertas. Las mejoras en el soporte software de la fábrica permiten obtener respuestas más rápidas, realizar mejores planificaciones y facilitar la innovación en los productos (Hicks et al. 2000).

Uno de los problemas principales con los que se encuentra una compañía de fabricación versátil a la hora de tratar las peticiones de oferta son las estimaciones de coste, que posteriormente servirán para determinar el precio de venta (Kingsman et al. 1996). Es en este área donde se centran diversas líneas de investigación que se tratan a continuación.

2.4. Modelos de Estimación de Costes

Para la determinación del precio a ofrecer al cliente, la compañía debe en primer lugar estimar el coste de fabricación del producto. Cuanto más rápida y fiable sea la estimación del precio, mayores posibilidades de éxito tendrá la oferta. Sin embargo, la estimación del coste no es una tarea trivial. Aunque en ocasiones el cliente proporciona planos y diseños con mayor o menor nivel de detalle, en una fase tan temprana del proceso de producción, la información sobre el producto a fabricar puede ser escasa e imprecisa.

Habitualmente las compañías de fabricación versátil no estiman los costes tal y como sugieren los libros de texto, sino que dichos métodos son el punto de partida para la estimación real (Kingsman and de Souza 1997). En este sentido, es habitual que las estimaciones se lleven a cabo por parte de expertos en la fabricación del producto para el que se solicita la oferta, y que parte de dichas estimaciones estén basadas en reglas no escritas y en la experiencia previa del estimador.

Las siguientes subsecciones describen una clasificación de las distintas técnicas y modelos de estimación empleadas en el desarrollo de aplicaciones de cálculo de costes de fabricación tomando como base el estudio de Niazi et al. (2006), completado con los puntos de vista de otros autores.

2.4.1. Técnicas de estimación cualitativas

Las técnicas de estimación cualitativas son aquellas que están basadas principalmente en el análisis del nuevo producto en comparación con productos previamente desarrollados. Las técnicas de estimación cualitativas se pueden clasificar en métodos intuitivos y métodos analógicos (Niazi et al. 2006).

Métodos intuitivos

Están basadas en la experiencia anterior. El conocimiento puede ser almacenado en forma de reglas, árboles de decisión, juicios, etc. y ser utilizado posteriormente para realizar la estimación. Estas técnicas se pueden clasificar a su vez en (Niazi et al. 2006):

- Metodología basada en casos. Los sistemas basados en casos recogen un número elevado de productos anteriores realizados en la compañía, almacenando información relativa a sus características y a sus costes. Cuando se quiere estimar el coste de un nuevo producto, estos sistemas buscan un caso anterior que sea similar al nuevo para dar una estimación. Una vez encontrado un caso similar al que se desea tratar, se aplica una función para adaptar el caso anterior a las nuevas circunstancias que se desean estimar. Para Duverlie and Castelain (1999), estos sistemas basados en casos proporcionan una solución rápida, y, al considerar solo una parte de los parámetros del producto, pueden ofrecer distintas alternativas para los parámetros que no se conocen en base a los casos anteriores. Ésto suele ser útil cuando no existe un conocimiento formalizado en el área de trabajo. En la fase de adaptación se pueden emplear otros métodos (como sistemas de lógica borrosa) y, aunque en ocasiones sea impreciso, siempre proporcionará un caso aproximado que podrá tomar como referencia el usuario. Sin embargo entre las desventajas de esta metodología, se encuentra la dificultad de dar criterios generales para indexar los casos, y la necesidad de un determinado número de casos de base, casos de indexación, una medida de similitud y una función de adaptación (Duverlie and Castelain 1999).
- Sistemas de soporte a la decisión. El propósito principal de estos sistemas es asistir a los estimadores para la toma de decisiones óptimas a distintos niveles del proceso de estimación, mediante la utilización del conocimiento de expertos en el dominio (Niazi et al. 2006). En estos sistemas se incluyen:
 - Sistemas Basados en Reglas.
 - Lógica Borrosa.

- Sistemas expertos.

Métodos analógicos

Se basan en la similitud del nuevo producto con casos anteriores. Los métodos analógicos basan su justificación en tres áreas (Duverlie and Castelain 1999): la psicología cognitiva, los sistemas basados en reglas y el razonamiento analógico. Dentro de estos métodos se encuentran:

- Análisis de regresión.
- Redes de Neuronas Artificiales. Algunos ejemplos de utilización de redes de neuronas para el cálculo de costes se puede ver en Stockton and Wang (2004). También se han empleado redes de neuronas para la estimación de costes de mantenimiento (Park and Seo 2004), o del ciclo de vida completo (Seo et al. 2002) a partir de la información del diseño conceptual.
- Duverlie and Castelain (1999) clasifican también el razonamiento basado en casos dentro de los métodos analógicos.

2.4.2. Técnicas de estimación cuantitativas

Se basan en el análisis del diseño detallado del producto, sus características y los procesos necesarios para fabricarlo. Dentro de las técnicas cuantitativas se encuentran los métodos paramétricos y los métodos analíticos (Niazi et al. 2006).

Métodos paramétricos

Se derivan de la aplicación de métodos estadísticos para definir el coste como una función de distintas variables del producto. Para poder aplicar los métodos paramétricos es necesario conocer el valor de los parámetros en los que se basa el modelo. Dichos parámetros no tienen por qué describir completamente el producto, sino que puede tratarse de un conjunto limitado que describan sus características principales. Estos métodos funcionan como una caja negra, por lo que no requieren amplios conocimientos del producto para emplearlo. Además estos métodos son rápidos en la obtención de resultados. Por otro lado su principal desventaja es que los parámetros no tomados en cuenta a la hora de definir las fórmulas pueden ser relevantes en determinados casos (Duverlie and Castelain 1999). Como parte de los métodos paramétricos, se encuentran los también los llamados “métodos estadísticos”, que incluyen los basados en datos

históricos y exámenes empíricos para obtener información causal sobre la relación entre características del producto y costes (Layer et al. 2002). Dentro de esta clasificación Layer et al. (2002) incluyen también los métodos basados en análisis de regresión, técnicas de optimización y redes de neuronas artificiales.

Métodos analíticos

Los métodos analíticos se basan en la descomposición del producto en unidades elementales, operaciones y actividades que representan los diferentes recursos utilizados en el proceso de fabricación. El coste se calcula como una suma de todos los componentes identificados. Otros autores se han referido a estos métodos como “basados en ingeniería” (Ruffo et al. 2006), “método detallado” (Seo et al. 2002) o “basado en la descomposición” del trabajo necesario para la fabricación en tareas elementales (Duverlie and Castelain 1999). En este apartado también se incluye el método “generativo-analítico” (Layer et al. 2002), que identifica los procesos relevantes para la creación del producto, y derivan los costes agregando cada uno de estos procesos. Estas técnicas se descomponen en (Niazi et al. 2006):

- Basados en operación. Calculan el coste a partir de los tiempos de fabricación, tiempos no operativos y tiempos de preparación de operaciones. Esta aproximación se utiliza en fases avanzadas del diseño, debido a la naturaleza información necesaria para su empleo.
- Basados en descomposición. En esta aproximación, el coste se calcula sumando todos los costes incurridos durante el ciclo de producción, incluyendo materiales y costes indirectos.
- Basados en tolerancias. El coste se calcula a partir de márgenes de tolerancia sobre distintos aspectos del diseño del producto.
- Basados en características del producto. Se basa en la identificación de las diferentes características del producto que están relacionadas con el coste del mismo. Estas características pueden estar relacionadas con elementos de diseño y/o elementos de operación.
- Método ABC (Activity-Based Costing) (Cooper and Kaplan 1988). Se centra en el cálculo de los costes incurridos al realizar las actividades necesarias para la fabricación del producto.

2.4.3. Discusión sobre los métodos de estimación

A pesar de que las diferentes clasificaciones están basadas en conceptos similares, la terminología empleada dista de ser homogénea. Así Lan and Ding (2007) y Chougule and Ravi (2006) clasifican los métodos de estimación en intuitivos, paramétricos, analógicos y analíticos, mientras que Shehab y Abdalla ofrecen hasta tres clasificaciones diferentes (Shehab and Abdalla 2001), (Shehab and Abdalla 2002a) y (Shehab and Abdalla 2002b).

Un aspecto que contemplado por Niazi et al. (2006) es el de los métodos tradicionales de estimación. Dichos métodos tradicionales, se pueden clasificar en (Zhang et al. 1996):

- Métodos de descomposición detallada.
- Métodos de descomposición simplificada (diseñados para la estimación en las fases iniciales de diseño).
- Métodos basados en el grupo tecnológico (basados en la similitud).
- Métodos basados en regresión y el método ABC.

Otros métodos no recogidos en la clasificación de Niazi, se basan en calcular las estimaciones de precio evaluando el beneficio y el nivel de riesgos asociado al proyecto (Muntslag 1994), en lugar de utilizar los costes de producción.

De los métodos citados, los paramétricos proporcionan estimaciones rápidas y se pueden basar en un número limitado de parámetros. Esto puede hacerlos ideales para las fases iniciales del desarrollo, y por ende a la estimación de coste para realizar una oferta (Veeramani and Joshi 1997) y (Chougule and Ravi 2006). Por otro lado entre los métodos más fiables destacan los métodos analíticos, pero dichos métodos no suelen ser adecuados en las fases iniciales de desarrollo debido a la detallada información que requieren.

La utilización de un único método de estimación no suele ser suficiente para una estimación fiable en las fases iniciales. Los autores señalan que es necesario combinar varias aproximaciones para obtener un resultado óptimo (Duverlie and Castelain 1999), (Layer et al. 2002) y (Lan and Ding 2007). Estudios comparativos entre técnicas analógicas y paramétricas concluyen que, aun obteniendo mejores resultados con la aproximación analógica, es razonable la combinación de ambos métodos, por ejemplo buscando un caso anterior que sea similar al problema a resolver y adaptarlo al caso objetivo con fórmulas paramétricas (Duverlie and Castelain 1999).

2.5. Aplicaciones para Estimación de Precios y Costes

Para aplicar los diferentes métodos de estimación, se han desarrollado distintas aplicaciones informáticas. Los requisitos fundamentales que debe cumplir un sistema de estimación de precios consisten en proporcionar alta fiabilidad y rapidez en la estimación (Chen et al. 2003).

Uno de los problemas que se han encontrado en este estudio es la escasez de sistemas dedicados a la estimación durante el proceso de oferta. Por ello se han incluido modelos y sistemas de estimación de precios y costes que están orientados a las fases iniciales del proceso de fabricación, lo que hace posible que sean empleados en la elaboración de ofertas. A título ilustrativo se han incluido en el estudio sistemas de estimación que no se encuentran dentro del dominio de fabricación de piezas mecanizadas, pero que son relevantes para la presente investigación por sus características.

Las siguientes subsecciones describen los principales sistemas y modelos encontrados, distinguiendo entre los sistemas basados en tecnologías convencionales y los sistemas basados en técnicas de inteligencia artificial.

2.5.1. Aplicaciones y modelos basados en tecnologías convencionales

La tabla 2.1 muestra los principales sistemas para la estimación de costes y precios de fabricación de piezas mecanizadas, así como las principales metodologías que se han propuesto para tal fin, que no están basadas en técnicas de Inteligencia Artificial.

En la tabla 2.1, la columna web indica si el sistema está basado en web o no. La columna objeto describe el objetivo final del sistema o modelo descrito, mientras que el área indica la rama de la industria a la que se orienta el sistema. La columna descripción realiza una breve descripción del sistema. Finalmente la columna método indica cuál de los métodos de estimación indicados anteriormente se desarrolla.

Tabla 2.1: Sistemas de estimación convencionales.

Artículo	Web	Objeto	Área	Descripción	Método
Lan and Ding (2007)	Sí	Estimación de precios para prototipado rápido.	Fabricación de piezas mediante estereolitografía.	Presenta un sistema web basado en la combinación de dos modelos de estimación. Uno paramétrico para la estimación inicial de precios y otro analítico para una estimación detallada.	Paramétrico y Analítico
Tu et al. (2007)	Sí	Estimación de costes y control posterior. Sistema de gestión de la producción.	Hojas de metal	Describe un modelo de estimación basado en índices para productos conocidos, que puede ser empleado también nuevos productos.	Analógico
Masmoudi et al. (2007)	No	Sistema de cálculo de costes.	Piezas que para su fabricación requieren soldadura. El mecanizado es una parte de la preparación de este proceso.	Propone una herramienta de soporte a la decisión para determinar los costes de ensamblado mediante soldadura, asociando variables técnicas, económicas y de soldadura de la pieza.	Analógico y Analítico
Continúa en la página siguiente					

Tabla 2.1 Sistemas de estimación convencionales. Continuación.

Artículo	Web	Objeto	Área	Descripción	Método
Ruffo et al. (2006)	No	Estimación de costes.	Sinterizado Láser	Presenta un modelo experimental de estimación de costes para fabricación de piezas por medio de sinterizado láser.	Paramétrico y Analítico
Jiao and Helander (2006)	Sí	Estimación de costes dentro de un sistema general.	Inyección en Molde	Sistema web colaborativo en el que los clientes pueden diseñar la pieza que necesitan, que incluye un módulo de costes.	Analítico
Continúa en la página siguiente					

Tabla 2.1 Sistemas de estimación convencionales. Continuación.

Artículo	Web	Objeto	Área	Descripción	Método
Silva et al. (2006)	Sí	Soporte a la Decisión	Fabricación de moldes para inyección de plástico.	Describe un sistema DSS llamado MAPP basado en web que planifica y controla la producción en una empresa de fabricación de moldes. Presenta un módulo para ayudar a determinar el precio mostrando datos anteriores para pedidos similares. Evita procesos complejos de estimación porque utiliza un conjunto reducido de parámetros dado que no se dispone del diseño detallado. Integra documentación y comunicación.	Ninguno. Da soporte con información de datos históricos.
Continúa en la página siguiente					

Tabla 2.1 Sistemas de estimación convencionales. Continuación.

Artículo	Web	Objeto	Área	Descripción	Método
Venkatadri et al. (2006)	No	Modelo para la estimación de precios y tiempos de entrega en RFQ.	General.	Presenta un modelo de soporte a la decisión que representa la cadena de suministro. A partir de la cadena representada se determina el precio y tiempo de entrega óptimos, asumiendo que el precio puede ser diferente en función del tiempo. Además genera el plan de la cadena de suministro.	Paramétrico.
Brinke et al. (2004)	No	Arquitectura para la estimación de costes.	Es genérico.	Presenta una arquitectura genérica para la determinación de costes, planteada para ser incluida en un sistema de control de costes de funcionalidad más amplia.	Configurable
Ben-Arieh and Li (2003)	Si	Negociación, en base a los costes de fabricación.	Mecanización	Propone una arquitectura y un sistema de agentes para la coordinación del proceso de oferta entre varias empresas.	El módulo es configurable, en el ejemplo propone ABC.
Continúa en la página siguiente					

Tabla 2.1 Sistemas de estimación convencionales. Continuación.

Artículo	Web	Objeto	Área	Descripción	Método
Beil and Wein (2003)	Si	Mecanismo para optimización del proceso RFQ electrónico.	Es genérico.	Propone un mecanismo de pujas para RFQ electrónico. Los suministradores ofrecen un precio con unas características (calidad, tiempo etc.) en cada ronda. Con la información de las rondas, aproxima la función de coste de cada suministrador, y selecciona la mejor oferta.	No aplica.
Chen et al. (2003)	No	Estimación de precios para RFQ.	Es genérico. Proporciona un ejemplo de mecanización.	Utiliza diseño axiomático para determinar los módulos, requisitos y cálculos necesarios para crear un sistema de ofertas asistido por ordenador. Implementa un sistema para piezas mecanizadas como ejemplo.	Intuitivo y Analítico
Continúa en la página siguiente					

Tabla 2.1 Sistemas de estimación convencionales. Continuación.

Artículo	Web	Objeto	Área	Descripción	Método
Koonce et al. (2003)	No	Estimación de costes.	Es genérico	Describe un módulo de estimación de coste para un entorno inteligente de producción (FIPER). Puede utilizar varias técnicas de estimación. Se basa en la descomposición del producto y la aplicación de relaciones entre sus características y ecuaciones de coste asociadas.	Paramétrico y Analítico
Jung (2002)	No.	Sistema de cálculo de costes.	Piezas mecanizadas.	Propone un sistema de estimación de costes basado en características para piezas mecanizadas. El modelo matemático de estimación se basa en 4 características principales de las piezas con las que trabaja. En el futuro plantea la integración del sistema en la Web.	Analítico.

Continúa en la página siguiente

Tabla 2.1 Sistemas de estimación convencionales. Continuación.

Artículo	Web	Objeto	Área	Descripción	Método
Lan (2002)	No	Estimación de Costes.	Mecanización máquinas de control numérico.	Desarrolla un modelo de cálculo de tiempo óptimo de mecanización, minimizando de coste de fabricación con máquinas . Programado en MATLAB, aplica simulación numérica (multiplicadores de Lagrange) para obtener el coste en función del tiempo de entrega.	Paramétrico
Layer et al. (2002)	No	Recopilación de modelos de coste.	Varios.	Presenta una recopilación de trabajos sobre estimación de coste realizados en Alemania hasta el año 2002. Propone una clasificación de métodos de costes.	Varios
Locascio (2000)	No	Modelo de coste para diseño de nuevos productos.	Ensamblado de componentes electrónicos.	Describe un modelo de costes basado en los tiempos de producción y de ensamblado para una planta de ensamblado de componentes electrónicos de Motorola.	Analítico, basado en ABC
Continúa en la página siguiente					

Tabla 2.1 Sistemas de estimación convencionales. Continuación.

Artículo	Web	Objeto	Área	Descripción	Método
Maropoulos et al. (2000) y Maropoulos and Baker (2000)	No	Cálculo de costes en función de las herramientas a utilizar.	Mecanización	Presenta un sistema que selecciona las operaciones necesarias para mecanizar una pieza en función de sus características. A partir de las operaciones selecciona las máquinas que pueden emplearse, y en base a las características calcula tiempo de mecanizado, y, por extensión, el coste.	Analítico
Ben-Arieh (2000)	No	Cálculo de costes de fabricación de piezas.	Fabricación de piezas mecanizadas	Describe un sistema para calcular el coste de mecanización de piezas, seleccionando a partir de sus características el proceso de mecanización necesario. En función de la tasa de eliminación de material calcula el tiempo de mecanización y a partir del tiempo obtiene el coste.	Analítico.
Continúa en la página siguiente					

Tabla 2.1 Sistemas de estimación convencionales. Continuación.

Artículo	Web	Objeto	Área	Descripción	Método
Duverlie and Castelain (1999)	No	Estimación de costes.	General	Compara el modelo de estimación paramétrico con el modelo analógico utilizando CBR. Aporta ventajas e inconvenientes. De su experimentación concluye que CBR obtiene mejores resultados, pero que la mejor opción es combinar ambos modelos	Paramétrico y Analógico.
Ou-Yang and Lin (1997)	No	Estimación de costes.	Mecanización	Presenta un sistema de estimación de costes a partir de las características de una pieza representada a través de una herramienta CAD. El sistema determina el proceso de mecanización necesario y, a partir del mismo, su coste.	Analítico
Veeramani and Joshi (1997)	No	Estimación de precios para RFQ.	Mecanización	Describe metodologías y modelos para la estimación de precios, y propone una arquitectura de sistema.	Analítico
Continúa en la página siguiente					

Tabla 2.1 Sistemas de estimación convencionales. Continuación.

Artículo	Web	Objeto	Área	Descripción	Método
Aderoba (1997)	No	Estimación de costes.	Job-shops	Propone un modelo para cálculo de costes y tiempos de producción en un job-shop, que puede ser utilizado para evaluar pedidos.	Analítico. Basado en ABC
Taiber (1996)	No	Modelo de costes para un CAPP.	Mecanización	Presenta un sistema que genera un proceso óptimo de fabricación basado en las características de la pieza a mecanizar. Propone un algoritmo para generar alternativas basado en combinatoria. Valora el empleo de algoritmos genéticos para generarlas indicando que es difícil validar los procesos obtenidos. Para cada alternativa de fabricación generada, determina su coste y selecciona la mejor opción.	Analítico
Continúa en la página siguiente					

Tabla 2.1 Sistemas de estimación convencionales. Continuación.

Artículo	Web	Objeto	Área	Descripción	Método
Muntslag (1994)	No	Método de estimación de precios para elaborar presupuestos	Empresas ETO.	Propone un modelo para estimar precios en la elaboración de presupuestos que, en lugar de tomar en consideración los modelos convencionales basados en procesos y características, emplea análisis de riesgos y beneficios para determinar el precio a ofrecer al cliente.	Basado en riesgos y beneficios
Downs and Trappey (1992)	No	Metodología para elaborar un programa de estimación de costes	Genérico.	Presenta una metodología para el diseño de software de estimación de costes. Propone un conjunto de bloques secuenciales que representan los materiales, las operaciones y los costes de cada una de ellas. El proceso se representa combinando los bloques de operación y de material necesarios para obtener el producto. En función de dichos bloques se estima el coste.	Analítico

Continúa en la página siguiente

Tabla 2.1 Sistemas de estimación convencionales. Continuación.

Artículo	Web	Objeto	Área	Descripción	Método
Doney (1971)	No	Estimación de precios para RFQ.	Mecanización en job-shop	Presenta un sistema para determinación de precios en base a módulos programados en C.	Analítico.

Un rápido análisis de la tabla nos permite ver que desde los años 70 ya se plantea que la informática puede ayudar a resolver uno de los problemas más importantes de las fábricas, como es la determinación del precio de fabricación en una fase previa al diseño. Este es el caso de Doney (1971), que ya plantea una aplicación modular para la determinación del precio de piezas mecanizadas para una fábrica. Con el paso del tiempo, a medida que nos acercamos al momento actual, aparecen más sistemas que se integran en la Web, fruto de las nuevas necesidades de comunicación que surgen en las compañías con la globalización de los mercados y movilidad del personal.

Podemos distinguir dos bloques fundamentales dentro del apartado de los sistemas que no emplean técnicas de inteligencia artificial, que en este análisis se han denominado “sistemas convencionales”: por un lado los estudios metodológicos y por otro los sistemas de estimación que los implementan.

Entre los estudios metodológicos dos de los autores más citados son Boothroyd y Dewhurst, tanto por sus modelos de estimación como por sus trabajos en racionalización de diseños (Dewhurst and Boothroyd 1988) y optimización de los costes de desarrollo de los proyectos (Boothroyd and Reynolds 1989).

El proceso de estimación es lento y requiere la intervención de numerosas áreas de la compañía, por lo que se han desarrollado métodos específicos para la estimación de costes y precios de fabricación. Sin embargo, aunque existen diversos modelos y metodologías de estimación de costes, no siempre están orientados específicamente a la estimación para realizar una oferta (Veeramani and Joshi 1997). Muntslag (1994) señala que en el momento de realizar una oferta se desconocen demasiados factores que pueden afectar al precio final, por lo que propone una aproximación al precio tomando en cuenta factores más generales asociados a los riesgos que conlleva el pedido y los beneficios que éste puede reportar a la compañía. Veeramani and Joshi (1997)

realiza una revisión de metodologías para responder rápidamente a peticiones de presupuesto, señalando que es imperativo el desarrollo de metodologías de soporte a la decisión y tecnologías para implantar el comercio electrónico y favorezcan dichas respuestas rápidas. Aderoba (1997) propone un modelo de estimación analítico, basado en el Modelo ABC, cuyo principal objetivo es la estimación del coste de fabricación de nuevos productos, pero que puede ser aplicado a la evaluación de pedidos.

A nivel de descripción de modelos y metodologías, también es relevante la ya citada aportación de Duverlie and Castelain (1999) comparando la estimación entre un modelo paramétrico y una metodología analógica basada en casos. En su trabajo Duverlie and Castelain (1999) aplican ambos modelos a la fabricación de pistones y comparan los resultados de ambos modelos. La conclusión es que los mayores beneficios se obtendrían combinando ambos, aplicando el método paramétrico para la evaluación y adaptación de casos anteriores. También Locascio (2000) presenta un modelo analítico basado en el método ABC, que aplica en una fábrica de ensamblado de Motorola. Lan (2002) presenta un modelo para la estimación de coste de piezas mecanizadas, creado a partir de los multiplicadores de Lagrange que posteriormente implementa en Matlab para su evaluación. Más actuales son el modelo de Venkatadri et al. (2006), un modelo genérico de programación lineal con algunas restricciones orientado a la fase de RFQ, y de Ruffo et al. (2006), modelo híbrido entre paramétrico y basado en ingeniería, con algunas aproximaciones estadísticas, orientado a estimación de costes de fabricación mediante sinterizado láser. Sin embargo, estos modelos no se reflejan en un sistema web ni recogen experiencia práctica.

También a nivel metodológico, pero esta vez en el área del desarrollo de software, Downs and Trappey (1992) analiza desde el punto de vista de la Ingeniería del Software el diseño de software de estimación de costes. En el artículo se propone una aproximación a la estimación basada en un conjunto de bloques almacenados en una base de datos, que pueden ser de dos tipos diferentes: de operación o de material. Cada bloque dispone de un conjunto de ecuaciones para calcular los costes asociados la característica que representan. Según esta metodología, una pieza se representa como el conjunto de bloques que representan las operaciones y los materiales que son necesarios para fabricarla. Una vez construida la secuencia de bloques necesaria para producir la pieza, el sistema sería capaz de calcular el coste a partir de las ecuaciones asociadas cada uno de ellos. La ampliación de este tipo de sistemas consistiría en añadir a la base de datos los nuevos bloques que sean necesarios para los nuevos productos. La estructura de módulos software y los requisitos de un sistema de estimación de precios es específicamente tratado por Chen et al. (2003) a partir de la técnica de diseño axiomático. Ambas aproximaciones se basan en sistemas que no contemplan la aproximación desde el punto de vista de los sistemas basados

en conocimiento.

En cuanto a los sistemas desarrollados, después de la ya citada aplicación de Doney (1971), los esfuerzos han continuado en distintas áreas y con distintos objetivos. En algunos casos el cálculo de costes no se aplica directamente a la estimación de un precio, sino que se aplica como criterio de selección de procesos de fabricación o alternativas de diseño. Por ejemplo Taiber (1996) aplica un modelo analítico para la selección de alternativas de fabricación en un sistema CAPP. Ou-Yang and Lin (1997) desarrolla un sistema de cálculo de costes para la evaluación de procesos de fabricación, basado en un modelo analítico que toma los parámetros de funcionamiento de ficheros CAD con los diseños de las piezas y bases de datos con los parámetros de fabricación y coste. Otros sistemas como el de Ben-Arieh (2000) calculan el coste combinando una aproximación variante (basada en datos anteriores) para determinar los parámetros de fabricación, calcular el tiempo de fabricación y estimar el coste final de la pieza a partir de un modelo detallado. En línea similar se encuentran los sistemas de Maropoulos et al. (2000) y Maropoulos and Baker (2000). Por otro lado Jung (2002) desarrolla un sistema de estimación de costes de piezas mecanizadas aplicando un modelo analítico basado en sus características y las operaciones necesarias para su mecanización.

El desarrollo de módulos configurables para integrar en sistemas más amplios ha sido tratado por Koonce et al. (2003) y Brinke et al. (2004), mientras que en los últimos años, la tendencia de los sistemas se orienta hacia el soporte a la decisión, como presentan Venkatadri et al. (2006), Silva et al. (2006) y Masmoudi et al. (2007). Estos últimos crean un sistema híbrido entre los métodos analítico y paramétrico, junto con un configurador de productos, empleado para calcular el coste de operaciones de soldadura en la fabricación de piezas (Masmoudi et al. 2007). Este sistema se implementa sobre una base de datos ACCESS y no es accesible a través de la web.

Dada la importancia de la elaboración de presupuestos para compañías de prototipado rápido, Lan and Ding (2007) presentan un sistema web de estimación, capaz de proporcionar el precio de una pieza a partir de un fichero con la definición de la misma, basado en dos modelos diferentes: uno general y otro detallado. El modelo general es paramétrico y se basa en un conjunto de características muy sencillas que permiten calcular el peso de la pieza, para, a partir del mismo, obtener una estimación del coste de fabricación y el precio de venta. El modelo detallado aplica un método analítico, basado en el tiempo de fabricación de la pieza. Este sistema aprovecha las ventajas de los modelos detallados de estimación, pero no tiene detrás un sistema basado en conocimiento, que permita gestionar y evolucionar dichos modelos.

Otros sistemas interesantes para la estimación de costes son los configuradores de productos, pues facilitan la definición de las características del producto para las decisiones y cálculos

posteriores. Un ejemplo de este tipo de sistemas es el propuesto por Jiao and Helander (2006).

Aparte de los modelos de cálculos de precios y costes y los sistemas que los implementan, otros estudios se ocupan también del problema de la estimación de precios pero desde el punto de vista de mejorar y optimizar el proceso. Así Beil and Wein (2003) proponen un mecanismo electrónico de pujas entre distintos proveedores por un pedido. En función de las pujas recibidas y las condiciones de la oferta, el sistema aproxima la función de coste de cada suministrador para decidir cuál es la mejor oferta. Por otro lado (Ben-Arieh and Li 2003) presentan un sistema de agentes para la coordinación del proceso de oferta entre varias empresas. Ambos sistemas siguen una dirección que toma fuerza en los últimos años: la automatización de procesos, en este caso del proceso de oferta, de forma que todas las transacciones se realicen de forma electrónica y automática, tomando decisiones inteligentes y minimizando la intervención humana.

Otra de las líneas en boga en los últimos años, especialmente en los niveles directivos, es el de los sistemas de soporte a la decisión. Silva et al. (2006) es un ejemplo de la utilización de estos sistemas para apoyar la determinación del precio proporcionando al estimador información relevante acerca de la decisión que debe tomar.

Como se puede observar en la tabla resumen, en los últimos años aparecen más sistemas integrados en la web, y muchos de ellos son sistemas de ámbito más general que incluyen módulos de estimación de costes.

2.5.2. Aplicaciones y modelos basados en sistemas expertos

La tabla 2.2 muestra los principales sistemas expertos orientados al cálculo de costes y la estimación de precios que se han desarrollado en los últimos años.

Tabla 2.2: Sistemas expertos para la estimación.

Artículo	Web	Objeto	Área	Descripción	Método
Ko et al. (2007)	Si	Evaluación de costes de fabricación.	Fabricación de piezas por inyección en moldes.	Presenta un sistema basado en conocimiento para la evaluación de los costes de fabricación por inyección en moldes. El sistema está embebido en un pequeño dispositivo USB provisto de un servidor Web a través del cual interactúa el usuario. Además proporciona una interfaz para interactuar con herramientas CAD del mercado.	Paramétrico
Bouaziz et al. (2006)	No	Estimación de costes procesos, coste y tiempos de fabricación.	Fabricación de piezas en moldes.	Presenta un sistema que determina el proceso de fabricación de una pieza por analogía con otras piezas fabricadas anteriormente. A partir del proceso de fabricación y las características de la pieza determina el coste.	Analógico y Analítico

Pasa a la página siguiente

Tabla 2.2: Sistemas expertos para la estimación. Continuación

Artículo	Web	Objeto	Área	Descripción	Método
Chougule and Ravi (2006)	Sí.	Estimación de costes en diseño.	Fabricación de piezas a partir de metal fundido.	Describe un sistema de estimación de costes paramétrico. Emplea CBR para la determinación del proceso de fabricación de la pieza. El sistema se integra en una plataforma inteligente de ingeniería.	Analógico y Paramétrico
Hvam et al. (2006)	Sí.	Mejora de proceso.	Construcción de plantas de cemento.	Ofrece una visión de cómo se puede crear un sistema basado en conocimiento para la estimación en la fase RFQ a partir de un configurador de productos.	Configurador.
Niazi et al. (2006)	No	Revisión de técnicas de estimación.	Varios.	Presenta una visión general de métodos de estimación, una clasificación técnica, y como una revisión del estado del arte.	Varios.

Pasa a la página siguiente

Tabla 2.2: Sistemas expertos para la estimación. Continuación

Artículo	Web	Objeto	Área	Descripción	Método
Shehab and Abdalla (2006)	No	Estimación de costes en la fase de diseño.	Ensamblado de piezas.	Describe un sistema experto que decide la técnica de ensamblado mejor para un producto dado. A partir de sus características, determina tiempo de ensamblado y coste. Además ofrece alternativas de rediseño.	Analítico
Xie (2006)	Sí	Diseño rápido de productos.	Varias. Caso de estudio Inyección en molde.	Propone un DSS que integra varias herramientas y bases de datos con la Web e Internet. Representa en EXPRESS y calcula costes por analogía (si existen casos anteriores) o analítico.	Analógico y Analítico
Chan (2005)	No.	Estimación de precios para elaborar presupuestos.	Chapado de relojes.	Presenta un sistema que combina CBR, lógica borrosa y reglas para estimar el precio del chapado de relojes. Si hay caso anterior lo propone, y si no busca utilizando reglas y lógica borrosas los casos que mejor se ajusten.	Intuitivo
Pasa a la página siguiente					

Tabla 2.2: Sistemas expertos para la estimación. Continuación

Artículo	Web	Objeto	Área	Descripción	Método
Hvam et al. (2004)	No	Reingeniería del proceso RFQ.	Cementera.	Estudia la forma de llevar a cabo la reingeniería del proceso de ofertas, a partir de un sistema de información basado en conocimiento.	Configurador.
Park and Seo (2004)	No	Estimación de costes de mantenimiento.	Genérico.	Sistema de estimación de costes de mantenimiento para productos en las primeras fases del diseño conceptual. Basado en redes de neuronas que calculan el coste a partir de un conjunto de atributos del producto.	Intuitivos (ANN)
Chan (2003) y Chan and Lewis (2000)	No	Estimación de coste, materiales y proceso de fabricación para piezas.	Mecanización	Presenta un sistema experto desarrollado en CLISP, que representa el conocimiento sobre materiales y procesos para mecanizar piezas. El sistema determina las operaciones necesarias en función de las características de la pieza, calculando su coste, junto con parámetros de operación.	Analítico
Pasa a la página siguiente					

Tabla 2.2: Sistemas expertos para la estimación. Continuación

Artículo	Web	Objeto	Área	Descripción	Método
Maropoulos et al. (2003)	N/E	Planificación inteligente de procesos	Mecanización y ensamblado	Propone un sistema de apoyo al diseño que optimiza los costes en la fase inicial de diseño. A partir de un conjunto reducido de características, genera un plan de producción, y varios alternativos, valorando su coste y su tiempo de producción.	Paramétrico.
Tang et al. (2003)	No.	Cálculo de costes a partir del diseño.	Troquelado de láminas de metal.	Describe un sistema basado en conocimiento que en la fase de diseño hace una estimación de costes. El conocimiento lo representa por medio de reglas de producción, relacionando las características de la pieza a fabricar. Calcula el coste a partir de las características de la pieza y el proceso para fabricarla.	Analítico.

Pasa a la página siguiente

Tabla 2.2: Sistemas expertos para la estimación. Continuación

Artículo	Web	Objeto	Área	Descripción	Método
Vidal et al. (2003)	Si	Determinación del coste para una orden en fábrica de muebles.	Fabricación de muebles de madera.	Describe un sistema de gestión de Workflow que modela el proceso de elaboración de presupuestos para la elaboración de muebles de madera con el fin de optimizarlo y automatizarlo.	No descrito
Wang et al. (2003)	No	Determinación de coste de fabricación en la fase de diseño.	Fabricación de piezas por inyección en moldes.	Propone un sistema de estimación de costes basado en casos, que aproxima la similitud por medio de redes de neuronas. Si no hay casos anteriores, lo calcula de modo tradicional según dos modelos reconocidos y lo introduce como un nuevo caso en la librería de casos.	Intuitivo

Pasa a la página siguiente

Tabla 2.2: Sistemas expertos para la estimación. Continuación

Artículo	Web	Objeto	Área	Descripción	Método
Seo et al. (2002)	No	Estimación de costes a los largo del ciclo de vida.	Varios.	El artículo compara modelo de regresión con modelo de redes de neuronas backpropagation para la estimación de costes asociados al ciclo de vida de un producto, en la fase de diseño conceptual. Concluye que las redes de neuronas obtienen mejores resultados que la regresión.	Analógico.
Pasa a la página siguiente					

Tabla 2.2: Sistemas expertos para la estimación. Continuación

Artículo	Web	Objeto	Área	Descripción	Método
Sharma and Gao (2002)	Si (interfaz ActiveX)	Evaluación de viabilidad de diseños y estimación de operaciones a realizar	Fabricación. No se restringe a un área concreta.	Presenta FBCDS (Feature Based Conceptual Design System), una herramienta de diseño preliminar a partir de modelos incompletos o abstractos. Representa las características en forma de árbol. Está integrado con herramientas que generan plan de proceso y utiliza CLISP para evaluar los modelos que aún no están completamente definidos, valorando tiempo y coste. El sistema devuelve una base de datos con un plan de proceso.	Analítico
Shehab and Abdalla (2002b) y Shehab and Abdalla (2002a)	No	Diseño	Mecanizado e inyección en moldes.	Describe un modelo de costes para piezas mecanizadas implementando un sistema basado en conocimiento que genera procesos de fabricación y determina su coste.	Analítico, basado en Ou-Yang and Lin (1997)

Pasa a la página siguiente

Tabla 2.2: Sistemas expertos para la estimación. Continuación

Artículo	Web	Objeto	Área	Descripción	Método
Jahna-Shahi et al. (2001)	N/E.	Estimación de costes.	Proceso de láminas de metal	Describe un modelo de estimación de costes basado en lógica borrosa. Se centra en la representación borrosa de las variables que no corresponden al proceso de fabricación propiamente dicho, pero que afectan al tiempo, como pueden ser las condiciones del entorno de operación.	Intuitivo.
Arezoo et al. (2000)	No	Selección de herramientas y parámetros de operación.	Mecanización	Presenta un sistema experto basado en lógica de predicados, que selecciona las herramientas de corte y sus parámetros, buscando coste o tiempo óptimo según indique el usuario.	Analógico.
Pasa a la página siguiente					

Tabla 2.2: Sistemas expertos para la estimación. Continuación

Artículo	Web	Objeto	Área	Descripción	Método
Wei and Eg-belu (2000)	No.	Cálculo de costes.	Fabricación de piezas mecanizadas	El sistema genera y optimiza el proceso de fabricación necesario para una pieza determinada, y propone un sistema para la estimación del coste de fabricación a partir del mismo. Incluye los procesos básicos de mecanizado. La selección de la máquina más adecuada se realiza a partir de reglas en un árbol de decisión.	Paramétrico.
Bidanda et al. (1998)	No	Análisis de procesos y cálculo de costes.	Fabricación de piezas a partir de metal fundido.	Muestra un sistema experto basado en reglas para el análisis del proceso y determinación de costes de fabricación para la elaboración de presupuestos. Emplea análisis de regresión.	Analógico.
Pasa a la página siguiente					

Tabla 2.2: Sistemas expertos para la estimación. Continuación

Artículo	Web	Objeto	Área	Descripción	Método
Mohamed and Celik (1998)	No	Diseño preliminar de edificios.	Construcción de edificio.	Describe un sistema experto para la estimación de costes en la fases iniciales de diseño sin información detallada. Cuando dispone de diseño detallado tiene un modo de estimación más completo.	Analógico
Kingsman and de Souza (1997) y Kingsman et al. (1996)	N/E.	Estimación de costes y decisión de precios.	Empresas MTO.	Presenta un sistema experto para la estimación de costes y el cálculo de precios a partir de heurísticas. Kingsman et al. (1996) indica los problemas clásicos de respuesta en compañías make-to-order, y propone como solución la utilización de sistema experto.	Analógico

Pasa a la página siguiente

Tabla 2.2: Sistemas expertos para la estimación. Continuación

Artículo	Web	Objeto	Área	Descripción	Método
Qiqin et al. (1996)	No	Definición de procesos y cálculo de presupuestos.	Líneas de producción compuestas de Máquinas-Herramienta modulares.	Describe un subsistema que combina CAD, Sistemas Expertos y análisis de sistemas, entre otras técnicas, para planificar una línea de producción y estimar el presupuesto de fabricación. El sistema forma parte de un sistema más amplio de diseño.	Analítico
Cunningham and Smart (1993)	No	Estimación de costes para decidir proveedores	Elaboración de piezas para coches.	Analiza el sistema empleado por la compañía Ford para estimar los costes de las piezas y poder así valorar las ofertas de los proveedores. Genera planes de fabricación y calcula el coste de fabricación de los distintos proveedores para decidir cuál es la mejor oferta.	Analítico

Pasa a la página siguiente

Tabla 2.2: Sistemas expertos para la estimación. Continuación

Artículo	Web	Objeto	Área	Descripción	Método
Kadidal and Bidanda (1993)	No	Estimación de precios a partir del coste de fabricación.	Piezas a partir de metal fundido	Presenta un sistema para la estimación de costes de fabricación de piezas a partir de metal fundido. Ayuda a análisis de viabilidad, selección de parámetros básicos para fabricarla pieza, selección de material, estimación de pesos y determinación de coste.	Paramétrico
Kennedy and Shao (1989)	No	Evaluar el proceso de gestión de pedidos de las compañías y propone mejoras para el mismo.	Pequeñas compañías de fabricación.	Describe un sistema experto para asesorar sobre la mejora del proceso de gestión de pedidos en pequeñas empresas de fabricación	No aplica

Como se observa en la tabla 2.2 los primeros sistemas expertos en el dominio aparecen a comienzos de los años 90, mientras que la tendencia a la orientación a la web comienza a marcarse a partir del año 2000. Pese a que autores como Xie et al. (2003) remarcan las ventajas para el proceso de fabricación de integrar en la web de los sistemas inteligentes y las oportunidades de aplicar el conocimiento práctico de los expertos de las fábricas, se han encontrado pocos sistemas expertos basados en web y menos aún orientados a la elaboración de presupuestos (Niazi et al. 2006).

A principios de los 90, Kadidal and Bidanda (1993) y, posteriormente, Bidanda et al. (1998) desarrollan un sistema experto para calcular el coste de elaboración de piezas a partir de metal fundido, empleando objetos y reglas para representar el conocimiento. El sistema realiza un análisis de viabilidad de la fabricación y determina los parámetros básicos de los procesos que son necesarios para calcular el tiempo de fabricación a través de un análisis de regresión. Para estimar el coste, además de los datos del proceso, el sistema experto selecciona el material a utilizar y calcula la cantidad necesaria del mismo.

Para evaluar y calcular el coste de fabricación de piezas de la compañía Ford en Europa, se desarrolló un sistema experto capaz de estimar el coste de las mismas a partir de sus características (Cunningham and Smart 1993). El sistema constaba de dos módulos: un kernel con información sobre cómo calcular el coste del plan, y un módulo de modelos de proceso con el conocimiento sobre las operaciones posibles, coste de las mismas y formulación de planes de fabricación. Para la representación de las piezas, la selección de procesos y la estimación del coste, el sistema experto combina diversas técnicas como heurísticas, lógica borrosa, orientación a objetos, generación y prueba de alternativas, simulación, y búsqueda primero en profundidad limitada.

Qiqin et al. (1996) integran un sistema experto con herramientas CAD, de las que extraen la información acerca de la pieza a fabricar para tomar decisiones, aplicando técnicas de análisis de sistemas. A partir de las características de la pieza a mecanizar, este sistema experto determina las operaciones necesarias para su elaboración, y las máquinas que pueden llevarlas a cabo. El sistema dispone de un módulo de evaluación para ayudar al experto a tomar decisiones.

Los principales problemas para el tratamiento de pedidos de los clientes en las empresas de fabricación versátil son (Kingsman et al. 1996): estimación de tiempo, coste y carga de trabajo, y realización del trabajo en el tiempo previsto. A partir del citado estudio Kingsman et al. (1996) proponen un sistema experto de soporte a la decisión para la elaboración de ofertas, en el que un módulo se encarga de analizar propuestas realizadas en el pasado para determinar la probabilidad de aceptación de la que se está elaborando, mientras que otro módulo se ocupa de generar fechas de entrega posibles teniendo en cuenta la carga de la factoría. A partir de los resultados de los módulos anteriores, un tercer módulo se encarga de generar un conjunto de propuestas alternativas. El usuario puede analizar cada una de ellas, y utilizar reglas que le ayuden a tomar una decisión. Estas reglas se describen con mayor detalle en el artículo de Kingsman and de Souza (1997). El módulo de estimación guarda para futuros análisis las estimaciones de tiempo y coste para cada producto, junto con un grado de confianza en la estimación, la similitud con trabajos anteriores y la experiencia que se obtuvo finalmente con el producto. El módulo de reglas

presenta un conjunto de preguntas al estimador y, en función de las respuestas, da una serie de recomendaciones (por ejemplo, reducir el margen de beneficio para bajar el precio y tener mayor probabilidad de éxito). El sistema de inferencia utiliza el razonamiento forward chaining, junto con un sistema de pesos para, en caso de tener varias reglas activadas, ejecutar la que más peso tenga.

Existen características del producto que influyen en el tiempo y el coste de las actividades de fabricación, que hacen difícil la construcción de un modelo matemático detallado de estimación. Esto es debido a las múltiples interrelaciones que existen entre los diferentes parámetros que se utilizan para realizar estimación. Por ello Jahna-Shahi et al. (2001) investiga la aplicación técnicas de lógica borrosa para la estimación de coste de fabricación.

Shehab and Abdalla (2002b) desarrollan un sistema experto para cálculo de costes de producción capaz de asesorar en conflictos que puedan surgir entre diseño y fabricación durante el desarrollo del producto. Junto con la estimación de costes recomienda, a partir de las reglas contenidas en su base de conocimiento, los procesos de mecanización apropiados, los parámetros a aplicar y la secuencia de operaciones adecuada. El sistema calcula todos los costes asociados al material, basándose el modelo propuesto por Ou-Yang and Lin (1997).

Sharma and Gao (2002) emplea un sistema experto como herramienta de soporte al diseño conceptual de un producto a partir en las características del mismo, incluyendo el cálculo de costes de fabricación. Utilizando reglas de producción, se ha desarrollado otro sistema experto integrado en la web para el diseño y fabricación de piezas a partir de hojas metálicas, que proporciona una estimación de coste a partir del proceso de producción que determine (Xie, Tu, Aitchison, Dunlop and Zhou 2001), (Xie, Tu, Liu and Zhou 2001) y (Xie et al. 2003).

Maropoulos et al. (2003) presenta un sistema de soporte a la toma de decisiones durante la fase de diseño, que incluye facilidades para determinar el proceso de fabricación de la pieza, calculando el coste del mismo. El sistema de Maropoulos et al. (2003), está basado en las características de la pieza a fabricar. El modelo que describe, trata emplear la menor cantidad posible de información y permitir la estimación en las fases iniciales del diseño. A partir de las características del producto, este sistema genera varios procesos de fabricación que se aproximan al óptimo, calculando para cada uno de ellos su coste, calidad y tiempo de entrega, a fin de seleccionar el más adecuado. El cálculo de coste se basa en las actividades necesarias para fabricar la pieza.

Tang et al. (2003) presentan un sistema experto que cubre todo el proceso de diseño, incorporando reglas de producción para relacionar las características del producto que se está diseñando con el coste que conlleva. Este tipo de sistemas son utilizados principalmente en escenarios de

estudio de alternativas, impacto de cambios y seguimiento de costes.

Chan (2003) describe en detalle un sistema experto para determinar la viabilidad de fabricación de un producto y una estimación de su coste, no restringido a un dominio de fabricación concreto. El sistema, desarrollado con CLISP y C, recibe como entrada la geometría de la pieza a fabricar y las especificaciones de diseño, y calcula el coste siguiendo el modelo descrito por Chan and Lewis (2000). Para validar el sistema, Chan compara los resultados de la estimación para un conjunto de piezas, con los resultados obtenidos por otras compañías a las que solicita información sobre las mismas piezas, concluyendo que los resultados similares. El autor indica que las desviaciones entre la estimación del sistema y los resultados obtenidos por las compañías con las que ha comparado, pueden deberse a las herramientas especiales de cada fábrica, ya que éstas pueden afectar al coste y la distribución de los gastos generales, pues cada compañía puede tener una política diferente de asignación de costes.

Wang et al. (2003) aplica razonamiento basado en casos para la estimación de costes de fabricación de piezas por inyección de plástico en molde. Para calcular la función de similitud entre casos utilizan una red de neuronas. Cuando el sistema encuentra casos similares, adapta el coste del caso anterior aplicando una función, mientras que si no hay casos similares, calcula el coste empleando los métodos específicos conocidos. Cuando el sistema estima el coste de una pieza para la que no dispone de referencias anteriores, el usuario puede corregir el resultado obtenido (en caso de no estar de acuerdo con el mismo) antes de incorporarlo en la librería de casos.

Combinando razonamiento basado en casos, reglas y lógica borrosa, Chan (2005) desarrolla un sistema para la estimación de precios en el proceso de chapado de relojes. En este área, la dificultad para estimar el peso del material, lleva a tremendos errores en las ofertas de precios, pues no hay tiempo de hacer pruebas. Para la estimación Chan emplea razonamiento basado en casos. Cuando no hay casos previos, los módulos de reglas y de lógica borrosa se utilizan para seleccionar casos aproximados que se puedan utilizar. El sistema obtiene resultados con una media de error del 14,72 %, inferior al 20,05 % que tiene una persona para sus casos de estudio.

Chougule and Ravi (2006) crean un sistema experto que estima los costes de piezas fabricadas a partir de metal fundido en las fases iniciales de diseño. El sistema obtiene la información acerca de la geometría de la pieza a partir de herramientas de diseño tridimensional. Para la estimación del coste combina un modelo paramétrico con un modelo analógico, aplicando razonamiento basado en casos. En función de la geometría de la pieza y del proceso de fabricación necesario para la obtención de la misma, el sistema calcula coste de fabricación. El modelo se ha implementado en una plataforma llamada webICE (web-based Intelligent Collaborative

Engineering), desarrollada en el laboratorio de los autores.

El sistema de Shehab and Abdalla (2006) utiliza una representación híbrida de conocimiento, combinando reglas de producción, frames y objetos, para la estimación de costes durante la fase de diseño del producto. A partir de las características del producto decide la técnica de ensamblado más económica, calculando sus costes asociados. El sistema propone modificaciones sobre el diseño para mejorar el coste. El sistema desarrollado por Xie (2006) para diseño y desarrollo rápido de productos, incluye también un módulo de cálculo de costes que lo complementa para realizar estimaciones.

Fuera del ámbito de la fabricación de piezas, también se han aplicado sistemas expertos para la estimación de costes en el área de la construcción, a partir de diseños y especificaciones poco detalladas, tal y como muestran los ejemplos de (Mohamed and Celik 1998) y (Mohamed and Celik 2002).

Por otro lado, combinando la idea de sistemas basados en web con los sistemas embebidos en pequeños dispositivos, Ko et al. (2007) presenta un sistema basado en conocimiento para evaluar costes de fabricación de piezas fabricadas mediante inyección en molde. El sistema está embebido en un pequeño dispositivo USB, dotado de un sistema operativo basado en Linux que dispone de un servidor Web. El usuario a través de un navegador Web puede acceder al sistema, permitiendo además la interacción con otros sistemas CAD convencionales. Para representar el conocimiento emplea marcos almacenados en formato XML.

La integración de la web con sistemas inteligentes, junto con la reingeniería del proceso de ofertas, ha dado muy buenos resultados en determinadas áreas, como la construcción de plantas de cemento (Hvam et al. 2006). Otra aproximación a la mejora del proceso de oferta a través de sistemas expertos es la de Kennedy and Shao (1989), con un sistema experto que asesora sobre las mejoras que puede acometer una empresa en su proceso de recepción de pedidos, en base al conocimiento de empresas y consultoras de prestigio.

Finalmente, al igual que en el caso de los sistemas convencionales, también se han creado sistemas expertos capaces de estimar costes y precios a partir de configuradores de productos tal y como señala Helo (2006). Sin embargo éstos se basan en la selección de características y componentes que implican un conocimiento detallado del producto.

2.6. Representación de Procesos Basada en Ontologías

Una ontología es una especificación formal explícita de una conceptualización compartida (Studer et al. 1998). Studer et al. (1998) explica que la conceptualización se refiere a que es un

modelo abstracto que identifica los principales conceptos de un dominio, que es explícita porque los conceptos y las restricciones se definen explícitamente, que es formal en el sentido de que debe ser interpretada por máquinas y que es compartida pues captura conocimiento aceptado por un conjunto de personas. En el caso de esta tesis, el dominio se refiere a la representación basada en situaciones que se ha definido en el modelo formal. Los conceptos y sus relaciones, pues, serán los descritos en dicha sección, y quedarán explícitamente definidos de acuerdo al lenguaje seleccionado.

Gomez-Perez et al. (2004) identifica tres tipos de lenguajes para la definición de ontologías:

- Basados en Inteligencia Artificial, como Ontolingua, LOOM, OCML o FLogic.
- Lenguajes basados en etiquetas, como RDF, DAML+OIL o OWL.
- Otras aproximaciones no basadas en inteligencia artificial, como UML, diagramas entidad-relación o scripts SQL.

Fensel (2000) además identifica los siguientes tipos de ontologías:

- Ontologías de dominio son aquellas que capturan conocimiento válido para un dominio particular (como electrónica, medicina, mecánica o dominio digital).
- Ontologías de metadatos que proveen un vocabulario para describir el contenido de fuentes de información online. Un ejemplo de este tipo de ontologías es Dublin Core ¹.
- Ontologías genéricas o de sentido común, que tratan de capturar conocimiento general sobre el mundo, proporcionando nociones básicas y conceptos para objetos como tiempo, espacio, estados, eventos, etc (Fridman-Noy and Hafner 1997)
- Ontologías de representación. Esas ontologías proveen entidades representacionales sin establecer que debería ser representado. Una conocida ontología representacional es la Ontología de Frames (Gruber 1993), que define conceptos como frames, slots y restricciones de slots, permitiendo la expresión de conocimiento en entornos orientados a objetos.
- Otros tipos de ontología denominadas ontologías de método y tarea (Fensel et al. 1997) (Studer et al. 1996). Las ontologías de tareas proveen términos específicos para tareas particulares, mientras que las ontologías de métodos proveen términos específicos para métodos de resolución de problemas

¹<http://dublincore.org>

Por una parte, las ontologías establecen un vocabulario compartido entre los expertos de un dominio, permitiendo compartir información entre personas y empresas en base a una terminología común. Por otra parte, gracias a los nuevos lenguajes ontológicos basados en XML, como el *Ontology Web Language*² (OWL) las ontologías han facilitado no solo la compartición de información entre sistemas informáticos, sino que también han contribuido a la creación de sistemas inteligentes basados en las conceptualizaciones que representan.

Las ontologías han sido ampliamente utilizadas en el campo de la automatización de procesos así como en la representación del conocimiento. En el campo de los procesos industriales, las tecnologías semánticas y los Servicios Web proporcionan el marco ideal para la implementación de los nuevos métodos de fabricación y la representación del conocimiento acerca de los mismos.

Parte del esfuerzo investigador en la representación semántica de procesos de fabricación se centra en los Servicios Web Semánticos. Las tendencias actuales en este sentido se pueden clasificar en:

- Mejora de los modelos clásicos de representación por medio de ontologías.
- Representación de procesos basados en servicios web semánticos.
- Descripción semántica de productos y servicios y desarrollo de sistemas de soporte a la decisión para la mejora de procesos.

2.6.1. Mejora de los modelos clásicos de representación por medio de ontologías

En esta sección se repasan algunas aportaciones recientes a dirigidas a la mejora de los modelos clásicos de representación de procesos.

Poernomo and Umarov (2008) enriquece la definición de redes de Petri empleando ontologías para generar especificaciones de procesos en lenguaje B. Sin embargo, el trabajo de Poernomo and Umarov está orientado hacia la representación formal del proceso y su validación, sin considerar conocimiento adicional. Andonoff et al. (2005) utiliza redes de Petri con objetos (PNO) y servicios web, proporcionando un conjunto de reglas para convertir las especificaciones PNO en especificaciones OWL-S de forma automática. Este estudio se centra nuevamente la representación del proceso por medio de servicios web semánticos sin incluir conocimiento experto relacionado con el proceso.

²<http://www.w3.org/2004/OWL/>

La notación BPMN (Business Process Management Notation) es una notación ampliamente utilizada para representar desde una perspectiva de gráfica los flujos de trabajo y procesos de negocio. Sin embargo, para la aproximación que proponemos en esta tesis es necesario una aproximación más orientada al conocimiento.

2.6.2. Representación de procesos basados en servicios web semánticos

En esta subsección se repasan algunas aproximaciones recientes al modelado de procesos basado en servicios web.

Dong et al. (2008) propone una arquitectura para utilizar recursos de fabricación basados en servicios web, utilizando tecnologías semánticas. Esta aproximación está basada en la integración de diferentes servicios y recursos web pero no incluye una representación explícita de conocimiento acerca del proceso. Montagut and Molva (2005) describe otra arquitectura basada en el lenguaje BPEL para definir flujos de trabajo distribuidos basados en servicios web. Posteriormente desarrollaron un nuevo protocolo para el soporte de transacciones complejas en dichos procesos distribuidos (Montagut et al. 2008). Sin embargo, la aproximación de Montagut et al. se basa en la existencia de servicios web disponibles para la ejecución de las distintas tareas.

Desde el punto de vista del usuario, entendiéndolo como usuario al experto que está representando el proceso, Dietze et al. (2008) propone un metamodelo para la definición de procesos por medio de situaciones. Esta aproximación combina la descripción de situaciones con los objetivos que se deben alcanzar en cada una de ellas. La forma de alcanzar estos objetivos se realiza por medio de servicios web. En la aproximación de Dietze et al. se orienta a la descripción de los objetivos y los servicios web que pueden alcanzarlos. Sin embargo la obtención de dichos objetivos se realiza solamente a través de servicios web y se orienta hacia la composición y resolución automática.

También existen estudios referidos de la representación de las diferentes tareas ejecutadas en un proceso. Tran and Tsuji (2007) propone el lenguaje OWL-T, una extensión del lenguaje OWL para representar plantillas de tareas que representan demandas de negocio. Sin embargo, este lenguaje se orienta a la composición automática de servicios. Ye et al. (2006) presenta un sistema de gestión de pedidos basado en conocimiento y en el flujo de trabajo necesario. Este sistema se basa en la utilización de varias ontologías, incluyendo métodos de resolución de problemas (PSMs) y reglas en formato SWRL (Semantic Web Rule Language) para permitir la ejecución de actividades. Sin embargo la aproximación de Ye et al. no tiene un enfoque genérico.

Relacionado con la automatización de procesos, recientemente se han realizado diversos estudios sobre la composición y orquestación de servicios web para mejorar el nivel de automa-

tización de procesos industriales, como por ejemplo Vitvar et al. (2008) o Liu and Nie (2007). Arroyo et al. (2007) describe la aplicación de un framework basado en la creación de coreografías basadas en servicios web semánticos. Shen et al. (2007) propone una aproximación basada en ontologías para la resolución de conflictos en la composición de servicios. Koumoutsos and Thramboulidis (2008) se centra en la automatización de la negociación entre sistemas a partir de un framework basado en servicios web semánticos y ontologías de negociación. Sin embargo estas aproximaciones se centran en la ejecución por medio de servicios web por encima de la representación del conocimiento. Esto implica que para resolver el problema por medio de estas técnicas es necesaria la existencia de un servicio web adecuado y que sea posible acceder al mismo.

Por tanto un problema común en las aproximaciones mostradas es que en su mayoría requieren la existencia de servicios web para la ejecución de una tarea. Pedrinaci et al. (2008) destaca que la intervención humana en los procesos implica un problema de escalabilidad. Pedrinaci et al. (2008) presenta una parte del proyecto SUPER, que ha obtenido resultados prometedores en la escalabilidad de gestión de procesos de negocio explotando la utilización de servicios web semánticos. Sin embargo, a pesar de los problemas de escalabilidad, no todas las tareas pueden ser resueltas de forma automática por medio de servicios web y es necesario contemplar esa posibilidad.

2.6.3. Otras aproximaciones

En la línea de automatización de procesos, la descripción semántica de productos y servicios permite compartir definiciones de productos entre compañías y localizar y componer automáticamente servicios web. Zhao and Liu (2008a) y Zhao and Liu (2008b) presenta un modelo de información de producto basado en EXPRESS. Este modelo está representado en OWL y SWRL. El framework propuesto por Zhao and Liu está basado en el razonador Jena, reglas Jess y Protege³. Sin embargo esta aproximación no representa el proceso y su automatización, sino que se centra en el modelado del producto. Alexakos et al. (2005) introduce una arquitectura basada en ontologías para la integración de sistemas de distintas compañías por medio de servicios web. Este trabajo se caracteriza por la descripción semántica de las compañías y de los servicios que ofrece cada una de ellas, pero una vez más se basa en la definición de servicios web.

Otro aspecto importante para los procesos industriales es la gestión de los roles que representan cada interviniente en un determinado proceso. Filipowska et al. (2008) presenta dos ontologías para describir procesos de negocio y los roles organizativos implicados en los mismos

³<http://protege.stanford.edu>

(Business Roles Ontology y Business Functions Ontology). Sin embargo esta aproximación se centra en el control de acceso a la información dentro del proceso.

Por otra parte, desde el punto de vista del soporte a las personas implicadas en un determinado proceso, los sistemas de soporte a la decisión basados en tecnologías semánticas han sido aplicadas para la optimización de procesos industriales. Noor et al. (2006) presenta un framework genérico de soporte a la decisión para el proceso de formalización de pedidos. El principal objetivo de esta aproximación es el soporte para las decisiones complejas y poco estructuradas que se producen en algunos procesos de negocio como la formalización de un pedido. Sin embargo el framework propuesto por los autores no cubre el control del proceso y la ejecución de tareas. Chen and Sairamesh (2006) presenta una arquitectura para el modelado de procesos de negocio, evaluándolos y generando recomendaciones para su transformación y mejora. Esta aproximación, centrada en la industria de la automoción, tampoco cubre la ejecución del proceso ni lo complementa con conocimiento. Chen (2008) desarrolla una ontología para controlar el acceso por parte de los miembros de una empresa virtual al conocimiento compartido. La ontología de Chen representa reglas de acceso a datos de forma genérica, pero no representa procesos de negocio propiamente dichos. Kamsu Fogueum et al. (2008) presenta una ontología para la descripción de procesos industriales. Basada en la descripción ontológica del proceso, se pueden definir nuevas experiencias y lecciones aprendidas para reutilizarlas posteriormente con el objetivo de mejorar. Sin embargo, esta aproximación se centra en la representación de las lecciones aprendidas pero no contempla la ejecución de tareas. Pandit and Zhu (2007) desarrolló una ontología sobre equipamiento eléctrico, utilizada para seleccionar la mejor alternativa de diseño para el desarrollo completo del proceso de ingeniería. Sin embargo esta aproximación cubre representación y compartición del conocimiento y evalúa alternativas de diseño, pero la ejecución y control del proceso queda fuera de su alcance.

Pero una vez que el conocimiento ha sido semánticamente representado y estructurado, es necesario que las compañías obtengan un beneficio de esta estructura. Por esta razón un aspecto importante es el razonamiento y la inferencia basadas en ontologías. Razonadores genéricos, como Pellet, pueden ser utilizados con subconjuntos de OWL. Otra opción es representar la definición la ontología por medio de reglas y objetos de lenguajes específicos como Jess, Prolog o CLISP, que pueden proporcionar un rendimiento mejor que los razonadores genéricos (Dellas et al. 2007). También se han desarrollado mecanismos de inferencia ad-hoc para tratar problemas de dominios específicos (Meditkos and Bassiliades 2008), (Verstichel et al. 2008), (Wu et al. 2008), (Pan 2007) o (Zhou et al. 2006). Sin embargo estos razonadores son capaces de inferir nuevo conocimiento a partir de las ontologías pero no conectan la definición de la ontología con

los requisitos para solucionar el problema (Brewster and O'Hara 2004).

2.7. Discusión

La generación de presupuestos referidos a la fabricación de piezas requiere conocimiento acerca al proceso de fabricación necesario para la obtención de la pieza y sus costes. Por esta razón es necesario que uno o varios empleados de la fábrica posean suficiente conocimiento para estimar dichos presupuestos.

Un presupuesto fiable generado en un corto periodo de tiempo es fundamental en compañías que fabrican bajo demanda. Un retraso en la elaboración del presupuesto puede causar que el cliente elija una oferta mejor de un competidor. Por otro lado, como se ha indicado, una estimación rápida pero incorrecta, puede acarrear pérdidas económicas para la compañía si el precio estimado es menor que el coste de producción, o la pérdida del cliente si el precio es demasiado elevado.

Existen distintas aproximaciones a la estimación de presupuestos de fabricación. En este capítulo se ha revisado el marco teórico de la estimación de costes y precios, así como la investigación relevante en el área de la implementación de sistemas inteligentes que tratan de mejorar los resultados de los sistemas tradicionales. Con este propósito se han repasado investigaciones relevantes en la definición de modelos de estimación y la creación de sistemas inteligente para el cálculo de presupuestos en compañías de fabricación de piezas mecanizadas.

Hvam et al. (2006), entre otros autores, han sugerido que, a pesar de la importancia que tiene una estimación fiable, los ingenieros a cargo de esta tarea tienen otras muchas actividades que desempeñar en la fábrica. En algunos casos esto puede implicar retrasos de semanas en la elaboración del presupuesto.

En el estudio realizado, se ha observado que la mayor parte de los sistemas convencionales (aquellos que no utilizan técnicas de sistemas basados en conocimiento) se basan en un modelo analítico. Sin embargo el número de sistemas que combinan varios métodos es también notable. Con respecto a la integración de los sistemas convencionales en la web, encontramos que, aproximadamente, apenas un 26 % de los sistemas estudiados están integrados en la web y dichos sistemas han sido desarrollados principalmente durante los últimos años.

Frente a los sistemas convencionales, los sistemas basado en conocimiento implican la representación del conocimiento del experto y su aplicación a la estimación del precio para la elaboración de un presupuesto. Como en el caso de los sistemas convencionales, en los sistemas basados en conocimiento estudiados prevalece la utilización de modelos analíticos. Sin embar-

go, en este caso, la escasez de información inherente a las primeras fases del desarrollo de un producto es paliada por medio del conocimiento del experto. En los sistemas basados en conocimiento también predomina la utilización de los métodos analógico e intuitivo. En estos sistemas destacan la utilización de sistemas basados en reglas, razonamiento basado en casos y aproximaciones basadas en redes de neuronas artificiales. Sin embargo, al igual que en el caso de los sistemas basados en reglas, el porcentaje de sistemas basados en web es muy inferior a los que se podrían denominar *de escritorio*.

A la luz de este estudio se puede concluir que los modelos basados en la experiencia constituyen un área abierta de investigación, refrendada en el hecho de que no parece que haya un modelo o un sistema aplicado con rotundidad en este dominio (Xie et al. 2003). Niazi et al. (2006) hacen hincapié en que los sistemas expertos aplicados a la estimación se basan principalmente en conocimientos teóricos considerando en menor medida conocimientos prácticos, e insisten en el potencial de la investigación en este sentido. Además, la representación del conocimiento del experto no solo permite la creación de sistemas de estimación, sino que favorece la permanencia del conocimiento en la organización. Este conocimiento puede ser utilizado en la formación de nuevos expertos, haciendo a las compañía menos dependientes de personas concretas.

Una clara conclusión es que la combinación de varios métodos de estimación es la aproximación ideal para obtener resultados satisfactorios (Duverlie and Castelain 1999) (Layer et al. 2002) (Lan and Ding 2007), y la combinación de técnicas basadas en conocimiento con un método cuantitativo más detallado es una aproximación que puede ofrecer resultado prometedores.

Sin embargo, todo el trabajo que se está desarrollando en la implementación de métodos de estimación no se está traduciendo en un número de Sistemas Web elevado. La integración en la web de este tipo de sistemas facilita el acceso rápido y desde cualquier lugar al conocimiento, además de las ventajas citadas por Grove (2000) y Duan et al. (2005) para los sistemas basados en conocimiento basados en web.

Tras analizar los métodos empleados y los sistemas existentes, se ha apreciado un denominador común en ellos: los sistemas inteligentes basados en la web estudiados no están basados en un modelo específico, sino que se implementan utilizando lenguajes tradicionales (CLISP, Prolog, NexpertObject, etc.) y programando por otra parte la interacción web. En el apartado referido a las ontologías se aprecia claramente esta tendencia, representando por un lado el conocimiento sobre los procesos a partir de ontologías específicas e implementando por otro lado la interacción web y la conexión con los servicios necesarios, pero dejando a un lado el punto de vista del experto (Dietze et al. 2008). Siguiendo con el estudio de Duan et al. (2005) son necesarios nuevos modelos que adapten la representación del conocimiento con la web. Además estos mo-

delos deben incorporar capacidad para representar acceso a información de otras aplicaciones, pues también forma parte del conocimiento de la compañía.

Un modelo que integre la capacidad de representar el conocimiento relativo al proceso, en esta investigación referido a la elaboración de presupuestos, integrado con la web y que pueda ser automatizado, puede proveer un marco para la evolución hacia una integración de sistemas basados en conocimiento con la web con menos requisitos de desarrollo. En el ámbito concreto de la estimación de presupuestos, un modelo de estas características puede contribuir a la representación de conocimientos más prácticos, basados en la experiencia, frente a la tendencia marcada por Niazi et al. (2006), aprovechando las ventajas anteriormente descritas de la integración con la web.

2.8. Resumen

Como se puede observar existen distintas líneas de investigación abiertas que abordan la estimación de precios de fabricación, tanto a nivel de modelos de estimación, como a nivel de aplicaciones convencionales y de sistemas basados en conocimiento. Sin embargo no parece que haya un sistema ampliamente aceptado o que tenga una gran aplicación en el dominio (Xie et al. 2003). El estudio de Niazi et al. (2006) remarca que los sistemas expertos aplicados a la estimación de costes están enfocados a la formalización de técnicas extraídas principalmente de libros de texto en lugar de encapsular el conocimiento práctico del experto, por lo que la investigación en este sentido tiene un gran potencial. Se trata por tanto de un área que está abierta a la aportación de nuevas propuestas.

Además, la evolución de las comunicaciones y de los mercados ha llevado a las compañías a integrar sus aplicaciones en Internet para aprovechar sus ventajas competitivas. También los sistemas basados en conocimiento deben adaptarse para operar en la Web, pues las técnicas tradicionales pueden no ser tan efectivas en dicho entorno (Duan et al. 2005).

Si bien las líneas analizadas en este capítulo muestran un acercamiento al cálculo de costes y precios para la elaboración de presupuestos y las primeras fases de diseño, no se ha encontrado ningún sistema basado en conocimiento aplicado al dominio que incorpore un modelo de representación específico que combine el conocimiento práctico del experto con las ventajas de la integración con la web.

Si se observa la representación desde el punto de vista de la representación de procesos, la investigación actual está centrada en la representación de tareas por medio de servicios web, así como en la composición y orquestación de dichos servicios. Sin embargo no siempre es

posible el acceso a un servicio web y, en ocasiones, es posible que no exista un servicio web disponible para una determinada tarea. En ocasiones se obtienen mejores resultados dando soporte a la ejecución de tareas por parte de las personas que automatizando por completo las mismas (Nurminen et al. 2003). Muchas de estas aproximaciones se centran en definición y descripción de servicios y tareas, dejando a un lado el punto de vista del experto (Dietze et al. 2008).

Como se detalla en el capítulo siguiente, la presente investigación tratará de cubrir los aspectos no contemplados hasta el momento por los sistemas analizados, ofreciendo un modelo integral y una implementación final basada en el mismo.

En este capítulo se plantean las hipótesis de las que parte la investigación, a partir de las conclusiones que se desprenden del estudio del estado del arte realizado en el capítulo anterior. A partir de las hipótesis planteadas, la solución propuesta es un modelo de representación del conocimiento para capturar la experiencia de un estimador y que permita reproducirlo automáticamente, permitiendo la intervención tanto de personas como de otras aplicaciones.

3.1. Hipótesis de Investigación

El proceso de elaboración de un presupuesto para atender la solicitud de un cliente, implica el conocimiento del proceso de fabricación de cada pieza, la capacidad para determinar su coste, así como de los costes adicionales, y añadir al coste total el margen de beneficio que estime conveniente. Esta tarea, por tanto, solo puede ser realizada por una persona con alta cualificación y experiencia. El problema reside en la falta de expertos para realizar estos presupuestos. Esto provoca que el personal cualificado emplee gran cantidad de tiempo realizando presupuestos en lugar de trabajar en otras actividades importantes para la empresa.

Las actuales necesidades de información de las fábricas requieren por un lado la integración de aplicaciones heterogéneas que dan soporte al proceso de fabricación, y por otro la inclusión de elementos que faciliten la utilización de la web, que faciliten el acceso remoto a la información y el desarrollo de sistemas de comercio electrónico. Duan et al. señalan que las técnicas de representación del conocimiento tradicionales pueden no ser tan efectivas en el entorno web (Duan et al. 2005). No es suficiente el desarrollo de un sistema basado en conocimiento, sino que el modelo de representación que sirve de base a dicho sistema debe permitir la integración con el resto de sistemas de la fábrica y facilitar el acceso a la información a través de una interfaz web. Además, dicho modelo debe facilitar la representación del conocimiento de una forma natural e intuitiva

Como se ha descrito hasta el momento los modelos están basados en conceptos teóricos y matemáticos dejando a un lado la experiencia (Niazi et al. 2006). Un modelo basado en la experiencia no solo puede ayudar a obtener buenos resultados en la estimación, sino que puede además ayudar a formar a nuevos expertos y conservar el conocimiento en la compañía. Las múltiples relaciones entre los parámetros de fabricación de una pieza y su influencia en el precio, hace que se compliquen los cálculos (Jahna-Shahi et al. 2001), tanto a nivel de definición del modelo como a nivel de implementación del mismo. Tal y como afirma Niazi et al. (2006), la visión del experto facilitaría la resolución de este problema. La combinación de varias técnicas de estimación es la aproximación ideal para obtener resultados satisfactorios (Duverlie and Castelain 1999), (Layer et al. 2002) y (Lan and Ding 2007), por lo que la combinación de un modelo basado en la experiencia, junto con un modelo cualitativa que represente los cálculos que realiza habitualmente el experto de una fábrica puede dar óptimos resultados.

Las hipótesis de las que parte la presente tesis doctoral son las siguientes:

1. El modelo propuesto puede representar el conocimiento del experto sobre el proceso de estimación de presupuestos.
2. El modelo propuesto puede representar el acceso conocimiento contenido en otras aplicaciones.
3. El modelo propuesto puede representar el intercambio de información entre agentes externos y el motor encargado de automatizar el proceso para la resolución del problema.
4. Un proceso representado a través del modelo propuesto se puede reproducir a través de un motor, al que se denominará motor de solución.
5. Un proceso automatizado puede obtener resultados similares a los de un experto.

3.2. Modelo Propuesto

El objetivo principal de la investigación es definir un modelo de representación del conocimiento que permita la representación de un proceso basado en la experiencia, incluyendo la representación de la interacción con otros sistemas y personas, y que permita automatizarlo por medio de un sistema software. Esta representación se enfoca a la estimación de precios, pero claramente es ampliable a otros procesos de decisión.

El modelo debe permitir representar la forma en que el experto toma las decisiones durante el proceso de estimación de precios. Pero la representación en el sistema basado en conocimiento no

se limita exclusivamente a reglas que el experto utiliza en su proceso de decisión. El experto, en su trabajo diario, emplea una serie de herramientas que facilitan su labor. Así, por ejemplo, para realizar las estimaciones de tiempo puede emplear una hoja de cálculo que recoja las fórmulas habituales y automatice los cálculos, y para conocer parámetros concretos de fabricación puede utilizar bases de datos.

El conocimiento que se encuentra en las aplicaciones que dan soporte al experto es fundamental para la estimación. Existen dos alternativas principales para adquirir este conocimiento. Una de ellas consiste en obtener toda la información y representarla en la base de conocimiento. Esta es una alternativa que, si bien aglutina toda la información dentro del sistema, crea ciertas complicaciones en su construcción y mantenimiento, pues es necesario representar en el sistema basado en conocimiento la información que se encontraba en otra aplicación. La segunda alternativa consiste en obtener directamente el conocimiento de la fuente a la que pertenece (base de datos, servicio web u otra aplicación) cuando sea preciso. Si se considera que las aplicaciones que se emplean en las fábricas pueden interactuar con otros sistemas, se llega a la conclusión de que la mejor solución no consiste en capturar el conocimiento y representarlo en la aplicación, sino que el sistema debe ser capaz de interactuar con dichas aplicaciones.

Dotar al modelo de la capacidad de representar la interacción con las aplicaciones que utiliza el experto, es una de las soluciones a la problemática identificada en la incorporación de la informática, la inteligencia artificial e internet en la industria: la integración con los sistemas existentes. Así por ejemplo, el modelo no solo debe representar el conocimiento acerca de un cálculo concreto, sino que recoge dónde se encuentra la hoja de cálculo, qué datos necesita y qué información le reporta.

La representación del conocimiento por medio de reglas de tipo *If <condición> then <acción>*, ha sido reconocida como la forma más sencilla para que un experto exprese su conocimiento (González and Dankel 1993). Efectivamente, cualquier experto que sea consultado sobre su proceso de trabajo expresará sus distintas formas de actuar ante un problema con fórmulas de tipo *si - entonces*. Sin embargo, cuando una persona expresa su conocimiento en forma de reglas, existe un contexto de fondo sin el cual la regla no tiene validez. Por ejemplo, si un experto en la fabricación de piezas actúa de acuerdo a una regla *“Si la superficie circular es mayor de 60 cm entonces utilizar la maquina 2”*, dicha regla solo tiene sentido en el contexto de trabajo con una pieza circular. El diccionario de la Real Academia de la Lengua Española¹, define una situación como el *“conjunto de factores o circunstancias que afectan a alguien o algo en un determinado momento”*.

¹<http://www.rae.es/>

El modelo propuesto se basa en que la resolución de un problema se puede plantear como el paso de una situación en la cuál hay un problema que se desea resolver, a otra en la que se ha alcanzado una solución para el mismo. El paso de la situación inicial a la situación en la que el problema se ha resuelto, supone pasar por un conjunto de situaciones intermedias en las que progresivamente se va aproximando a la solución, por medio de la toma de decisiones que modifican las circunstancias que rodean al problema. Desde este punto de vista, una situación implica también una acción que el experto realiza para resolver un problema.

Cada situación se caracteriza por un conjunto de hechos anteriores que han motivado la llegada a la situación actual. Además, en cada una de las situaciones por las que pasa el experto para resolver el problema, necesita realizar acciones que le permitan obtener más información para poder avanzar hacia la solución. Una vez que ha realizado dicha acción han cambiado las condiciones del entorno, por lo que se llegaría a una nueva situación.

Por tanto el modelo de representación propuesto se basa en:

1. Representar los distintos contextos que pueden existir dentro de un problema a resolver.
2. Representar, dentro de cada contexto, las distintas situaciones que se pueden dar en el proceso de resolución del problema.
3. Representar la acción a realizar en cada situación concreta por medio de un mensaje, que permita la interacción con agentes externos.
4. Para cada situación, en función del contexto en el que se encuentre, representar por medio de reglas las distintas decisiones que pueden llevarse a cabo.
5. Representar las distintas aplicaciones que el experto puede utilizar para resolver el problema, y definir la forma en que se interactúa con ellas.
6. Un motor capaz de evaluar las distintas situaciones e intercambiar información con el usuario y con otras aplicaciones a través de la Web. Dicho motor está integrado con un servidor web.

3.3. Resumen

En esta sección se han planteado las hipótesis de investigación que han guiado la realización de la presente tesis doctoral. Estas hipótesis se resumen en la capacidad de un modelo de representación para: representar el conocimiento del experto, representar el conocimiento de otras

aplicaciones, representar la intervención de personas y otras aplicaciones en el proceso, automatizar el proceso representado y obtener resultados similares a los del experto cuyo conocimiento se está representando.

Como respuesta a las hipótesis planteadas se introduce un modelo de representación del conocimiento basado en la identificación de las diferentes situaciones que se pueden plantear a lo largo del proceso de solución de un problema, las acciones a realizar en cada una de ellas y las posibles decisiones que puede tomar el experto.

Capítulo 4

Modelo de Representación del Conocimiento

En este capítulo se desarrolla un modelo de representación del conocimiento basado en situaciones que trata de cubrir las carencias encontradas en el estudio del estado de la cuestión. En primer lugar se realiza una definición formal del modelo de representación del conocimiento, para definir posteriormente una ontología que describe los conceptos del modelo y sus relaciones. A partir de este modelo es posible representar el conocimiento acerca del proceso de resolución de un problema, que podrá ser interpretado por un motor para reproducirlo de forma automática.

4.1. Consideraciones Previas

Antes de definir formalmente los distintos conceptos que conforman el modelo de representación, es necesario considerar el punto de vista desde el que se va a abordar la solución de un problema. Para ello, en primer lugar se van a enumerar un conjunto de consideraciones generales previas, que van a servir para comprender los distintos conceptos que se van a utilizar para definir el modelo de representación del conocimiento. Seguidamente se procederá a la conceptualización, definiendo el conjunto de los conceptos principales para la propuesta de representación, junto con sus propiedades y las relaciones que hay entre ellos. Para esta conceptualización se empleará un lenguaje de ontología (OWL-DL) como medio para expresar las características del modelo.

La semántica se completa en el capítulo 5 dedicado al algoritmo de solución, donde se detalla la interpretación que se hace de cada uno de los conceptos del modelo propuesto.

Uno de los objetivos del modelo de representación propuesto es representar el conocimiento de un experto en un ámbito, concretamente la elaboración de presupuestos. Como se ha mostrado en el capítulo 2, existen diversas aproximaciones para la representación del conocimiento, por ejemplo empleando marcos, objetos o reglas. En esta investigación se va a plantear la representación desde el punto de vista del proceso de resolución del problema que sigue el experto. Cuando el experto está resolviendo un problema realiza una serie de tareas aplicando sus co-

nocimientos, de forma que, tras ejecutarlas, consigue un objetivo. Una vez que el conocimiento relativo al problema y su resolución ha sido representado, un sistema experto puede reproducir el proceso para obtener un resultado similar. Esta aproximación va a proporcionar un modelo que permita plasmar la experiencia de expertos en el dominio de la elaboración de presupuestos (Niazi et al. 2006) y que facilite la posterior implementación por medio de tecnologías web (Duan et al. 2005).

Se va a definir *agente* como aquella entidad que interviene en el proceso de resolución de un problema realizando las tareas que sean necesarias. Cuando el proceso lo está realizando una persona, el *agente* es el propio experto, mientras que cuando el proceso lo está realizando un sistema experto, el *agente* es un programa.

A partir de esta definición simple del concepto de *agente*, en procesos más complejos se puede dar el caso en el que existan varios *agentes* interviniendo en el proceso de resolución. Así cuando el proceso de resolución lo está realizando una persona, puede ejecutar un programa de ordenador que realice un conjunto de tareas cuyo resultado acerque a la solución del problema. En este caso, además del *agente* que lleva a cabo el proceso de resolución, existe un segundo *agente* software que ayuda a alcanzar el objetivo. Por tanto, se van a distinguir dos clases de *agente*:

1. **Agente principal.** Es el agente que se encarga de guiar la resolución del problema.
2. **Agente secundario.** Es el agente que realiza tareas que ayudan al *agente principal* a alcanzar la solución del problema. El *agente secundario* solo interviene en el proceso cuando el *agente principal* lo requiere.

El modelo de representación propuesto parte de la definición del proceso de resolución de un problema como una sucesión de situaciones en las que un agente realiza una acción concreta. A partir de cada acción realizada, el proceso cambia, bajo una serie de condiciones establecidas, a una nueva situación ante la que es necesario realizar una nueva acción.

Bajo esta descripción, cada uno de los pasos dentro de un proceso definido está determinado por una situación, es decir, por un conjunto de condiciones del contorno que determinan el estado del problema que se está resolviendo. Por condiciones de contorno se entiende el conjunto de datos que son relevantes para la resolución del problema en un momento determinado. Para determinar la situación en la que se encuentra el problema, y el paso que se debe dar consiguientemente, el *agente principal* debe tener constancia de cuáles son las condiciones de contorno del problema y ser capaz de evaluarlas para tomar una decisión.

A partir de estas consideraciones previas, la definición de los elementos de representación se va a realizar en dos niveles. El primer nivel, que se llamará nivel formal, describirá mediante fórmulas los conceptos principales, así como sus relaciones. Sin embargo, la posterior automatización de un proceso, requiere considerar un conjunto de requisitos. Estas consideraciones, concretan el modelo conceptual en un segundo nivel, que se denominará modelo conceptual.

4.2. Modelo Formal

A continuación se definen de modo formal los conceptos del modelo.

4.2.1. Definición inicial del proceso de solución

El proceso de solución de un problema se plantea como un conjunto de situaciones que pueden darse en un problema determinado junto con las heurísticas que sirven para determinar en qué situación se encuentra el problema en un momento determinado. Formalmente, se define el proceso de resolución de la siguiente forma:

$$P = \{S, s_i, H, A, \delta, V\}$$

Donde:

- S es un conjunto de situaciones que se pueden dar en el proceso de resolución de un problema.
- s_i es la situación inicial en la que se encuentra el problema en el momento de ser resuelto, tal que $s_i \in S$.
- H es el conjunto de datos relevantes para el problema.
- A es el conjunto de acciones que se pueden llevar a cabo durante el proceso de resolución del problema.
- δ es la función de transición que determina el paso de una situación a otra.
- V es un conjunto de verificaciones que se pueden realizar durante el proceso de resolución del problema.

Como se definirá posteriormente, una situación representa el estado del entorno del problema en un instante determinado. El problema queda caracterizado por el conjunto de situaciones que se pueden producir desde el momento en que se plantea el mismo hasta el momento en que ha quedado solucionado. Para que el proceso de resolución quede definido, es necesario determinar la situación en la que se encuentra el problema en el momento inicial i (situación s_i).

La evaluación del entorno para determinar cuándo se cambia de una situación a otra queda definida por medio de la función de transición δ . Esta función tomará en consideración la situación en la que se encuentra el problema en un instante determinado, evaluará un conjunto de datos relevantes para dicha situación y determinará cuál es la nueva situación en la que se encuentra el problema.

Las siguientes secciones describen detalladamente los conceptos que se acaban de introducir.

4.2.2. Hecho

Un hecho representa un dato del entorno del problema que es relevante para el proceso de resolución del mismo.

Al conjunto de todos los hechos h_i se le denomina conjunto H .

4.2.3. Acción

Una acción a_i representa las distintas tareas que deben realizar los agentes a lo largo del proceso que se está representando. Una acción se realiza en el marco de una situación concreta, bajo unas condiciones de contorno determinadas.

Ante el conjunto de hechos de partida en una determinada situación H_t , una acción modifica dicho conjunto, añadiendo, eliminando o modificando valores a los hechos del mismo, de forma que se genera un nuevo conjunto $a_i(H_t) \rightarrow H_{t+1}$.

4.2.4. Verificación

Una verificación representa un conjunto de condiciones que determinan si en una situación concreta, tras ejecutar la acción que tiene asociada, es posible continuar con el proceso de solución. Una verificación $v_i(H_t)$ producirá un valor verdadero si cumplen las condiciones de v_i , o falso en caso contrario. Asociado a las condiciones a comprobar, la verificación tiene un valor que determina si el proceso puede continuar o no.

Formalmente:

$$v_k = \{P_k, \langle \text{continue} | \text{stop} \rangle\}$$

Donde:

- P_k es un conjunto de condiciones que se evalúan sobre un conjunto de hechos H_t tras la ejecución de una acción a_j .
- $\langle \text{continue} | \text{stop} \rangle$ indica si el proceso puede continuar o detenerse en caso de que las condiciones se cumplan.

Una condición se cumple si $\forall p \in P_k, p(H_t) = \text{true}$.

La verificación comprueba que los resultados de una acción son correctos (puede significar que la acción se ha realizado correctamente o que los resultados son adecuados para continuar adelante), y que las condiciones de contorno son las adecuadas para poder continuar con el proceso.

Por tanto, las verificaciones se realizan siempre tras ejecutar una acción determinada, y antes de comprobar las decisiones asociadas a la estado actual. De su resultado depende que el proceso continúe o, por el contrario, se detenga.

4.2.5. Decisión

La función δ que determina la transición entre una situación y otra se materializa por medio de las decisiones. Una decisión se define como un conjunto de condiciones asociadas a la transición del estado del problema a una nueva situación $s_f \in S$. Por tanto el conjunto δ se define como el conjunto de todas las decisiones que se puedan dar. Para que se pueda producir el paso de una situación s_i a una situación s_f , es necesario que se cumplan un conjunto de condiciones que estarán asociadas a la decisión. De esta forma, la decisión queda completamente definida de la siguiente forma:

$$\delta_k = \{P_k, s_f | s_f \in S\}$$

Donde:

- P_k Representa un conjunto de premisas o condiciones, expresadas en función de fórmulas lógicas sobre los hechos del entorno del problema, que deben cumplirse para que el estado del problema cambie la situación $s_f \in S$.
- s_f Es la situación de destino. Es la situación en la que se encuentra el problema si se cumplen las condiciones del conjunto P_k .

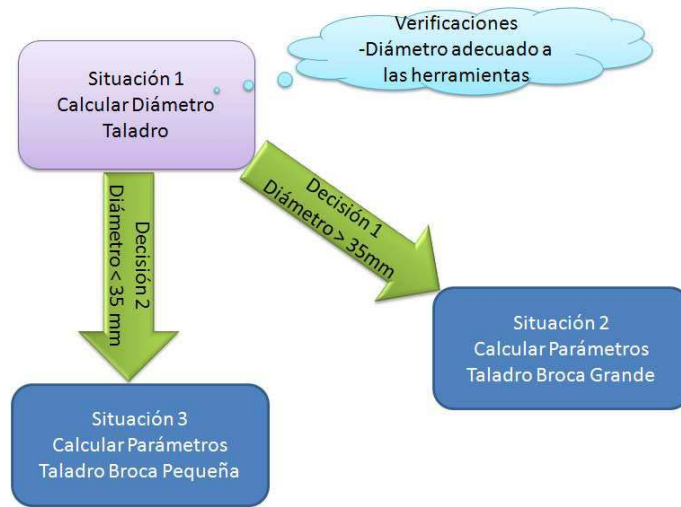


Figura 4.1: Ejemplo del concepto decisión.

De forma que en una situación s_a , que tenga asociada la decisión δ_k , en la que se ejecuta una acción $a_j(H_t) \rightarrow H_{t+1}$, la decisión $\delta_k(H_{t+1}) \rightarrow s_f$ se activa si $\forall p \in P_k, p(H_{t+1}) = true$.

Las decisiones determinan las condiciones que deben darse en una determinada situación para pasar al siguiente paso en el proceso de solución del problema.

Por medio del conjunto de decisiones asociadas a una situación, se representa las distintas alternativas que el proceso puede tomar tras realizar una acción en un contexto determinado. La figura 4.1 muestra un ejemplo de funcionamiento de las decisiones.

Sea un contexto $C_{piezasredondas}$ relativo a la fabricación de piezas con forma redondeada, y una situación $S_{seleccionaladro}$ en la que se determinan los parámetros referidos a la geometría de la pieza en función de las características requeridas (recogidas en el conjunto H_t). En esta situación, la acción que se debe realizar es determinar el diámetro del taladro que se puede utilizar para fabricar la pieza en función del diámetro de la misma. Suponiendo que existen dos taladros posibles en función de si el diámetro de la pieza es inferior o no a 35 mm, la situación tendría dos decisiones asociadas. La decisión δ_1 que llevaría a la situación $S_{taladroGrande}$ en el caso de que el diámetro de la pieza sea igual o superior a 35 mm, y la decisión δ_2 que llevaría a la situación $S_{taladroPequeno}$ en el caso de que el diámetro de la pieza sea inferior a 35 mm.

4.2.6. Situación

Una situación se define como el completo estado del universo en un instante determinado (McCarthy and Hayes 1969). Dado que para la resolución del problema no todos los elementos

del universo son relevantes, partiendo de la definición de McCarthy and Hayes (1969), para el modelo propuesto se define una situación mediante un conjunto de hechos relevantes que determinan el entorno del problema en un momento determinado.

Ante las condiciones del entorno y los objetivos planteados para el proceso, en una situación es necesario realizar una acción que alcance la solución del problema o bien que lleve a otra situación en la que la solución del problema se encuentre más próxima. La acción que se lleva a cabo en una situación cambia el entorno del problema asignando o modificando el valor de un conjunto de hechos.

Tras la ejecución de una acción en una situación determinada, puede ser necesario verificar que ésta se ha realizado correctamente, pues la acción ha podido generar resultados imprevistos. Si el resultado de la acción ha sido satisfactorio es necesario evaluar las nuevas condiciones de contorno que rodean el problema, por medio de un conjunto de reglas de decisión asociadas a la situación, que determinarán cuál es la nueva situación en la que se encuentra el problema.

Formalmente, una situación en un instante t s_t queda definida de la siguiente forma:

$$s_t = \{H_t, a_s, H_{t+1}, V_s, D_s\}$$

- H_t es un conjunto de hechos que se han producido anteriormente, tal que $H_t \subseteq H$.
- a_s una acción a ejecutar en la situación, tal que $a_s \in A$.
- H_{t+1} un conjunto de hechos que van a tomar un valor al ejecutarse la acción a_s , tal que $H_{t+1} \subseteq H$.
- V_s es un conjunto de verificaciones a realizar, donde $V_s \subseteq V$.
- D_s es un conjunto de decisiones asociadas a la situación, donde $D_s \subseteq \delta$.

4.2.7. Contexto

Para definir un proceso de decisión simple, es suficiente con describir las distintas situaciones que se pueden plantear, las acciones a tomar en cada una de ellas y las condiciones que gobiernan el paso de una situación a otra, hasta conseguir la resolución final de un problema. Sin embargo, en procesos complejos, las situaciones que se pueden plantear pueden requerir un comportamiento distinto, en función del estado en el que se encuentre el problema.

Cuando se describe cada paso de un proceso, se caracteriza por un conjunto de hechos que se han producido previamente y un conjunto de decisiones, que han motivado que se haya llegado a ese punto. Se trata de un contexto explícito basado en unos hechos y unas decisiones concretas.

Pero existe otra parte del contexto que está formada por reglas no expresadas de forma explícita, que se emplean para la resolución del problema. Estas heurísticas están expresadas en la forma en que se valoran las distintas situaciones, e influyen en la forma de resolver el problema. En el modelo de representación, este contexto implícito se plasma en la forma en que una acción se lleva a cabo en una situación concreta: la misma acción ejecutada en contextos distintos puede tener un comportamiento diferente, es decir, que las decisiones que se pueden tomar y las verificaciones ante una misma situación son dependientes del contexto. Por tanto, por medio del contexto se está plasmando el criterio que utiliza el experto para determinar las decisiones que se pueden tomar en una situación.

El contexto es el conjunto de condiciones implícitas que rodean al proceso de solución en un instante determinado y que influyen en las decisiones que se pueden tomar en un momento concreto.

$$c_i = \{S_{c_i}, s', B_{c_i}\}$$

Donde:

- S_{c_i} , es el conjunto de situaciones que se pueden producir en un determinado contexto.
- $s' \in S_{c_i}$, es la situación inicial en la que se encuentra el problema cuando se accede al contexto C .
- B_{c_i} está formado por tuplas de tipo $\langle s_j, \delta_j, V_j \rangle \mid s_j \in S_{c_i}, \delta_j \subseteq \delta, V_j \subseteq V$. Además, hay que notar que para que $\forall d \in \delta_j, d(H_t) \rightarrow s_t \mid s_t \in S_{c_i}$ o bien $d(H_t) \rightarrow c_t \mid c_t \in C$. Es decir que cada una de las tuplas del conjunto B_{c_i} va a relacionar, para un contexto determinado, una situación con un conjunto de decisiones y un conjunto de verificaciones que serán específicas para dicho contexto, por lo que se está definiendo realmente un comportamiento para una situación en un contexto determinado. Esto supone una evolución el concepto de situación, pues las verificaciones y decisiones asociadas a la misma pasan a depender del contexto, quedando plasmadas en el comportamiento. Además, como se desprende de la segunda parte de esta definición, todas las decisiones relacionadas con la situación en ese contexto, llevarán a situaciones de ese mismo contexto, o bien a otro contexto nuevo.
- Para todo $s \in S_{c_i}$ existirá *una y solo una* tupla en el conjunto B_{c_i} . De esta forma se evita la ambigüedad de comportamiento en un mismo contexto.

De esta forma podemos decir que cada situación en un contexto requerirá un comportamiento, que estará definido por un conjunto de verificaciones y un conjunto de decisiones.

La definición de contextos se puede ver también como la definición de distintos subproblemas en los que se puede descomponer un problema más general, de la misma forma en que una persona organizaría su conocimiento. Por ejemplo, supóngase una fábrica que dispone de tres tipos diferentes de máquinas que se pueden utilizar para mecanizar piezas, y que cada una de las máquinas requiere un proceso diferente de fabricación que hace necesario considerar parámetros distintos para la estimación del precio. Sin embargo, hay un conjunto de situaciones y acciones que van a ser comunes a los tres tipos de máquina, como puede ser, por ejemplo, la situación en la que se dispone de las características básicas de la pieza y es necesario determinar los parámetros relativos a la geometría (diámetro, longitud, forma, ...) de la pieza que se desea fabricar. La situación es similar para las tres máquinas, y la acción a realizar es la misma (decidir los parámetros), pero a partir de estos parámetros la situación siguiente será diferente en función de la máquina, pues cada una requerirá una configuración diferente. Por medio de los contextos, se puede definir por separado el proceso de decisión concreto para cada máquina, aprovechando situaciones y acciones comunes, y reduciendo el número de decisiones que hay que manejar en cada momento.

4.2.8. Resumen de definiciones

En las anteriores subsecciones se han identificado los distintos conceptos y se han ido refinando hasta llegar al concepto de *contexto*. La aparición del término contexto supone una adaptación de los conceptos anteriores que quedan definidos de la siguiente forma:

DEFINICIÓN. 1. Un **hecho** h_i representa un dato del entorno del problema que es relevante para el proceso de resolución del mismo.

Al conjunto de todos los hechos h_i se le denomina H . Cada estado del proceso $H_t \subseteq H$ está caracterizado por el conjunto de hechos relevantes para dicho estado.

DEFINICIÓN. 2. Una **acción** a_i representa una tarea que debe ser ejecutada en el proceso. Al conjunto de todas las acciones a_i se le denomina conjunto A .

Una acción se ejecuta en un determinado estado t del proceso, definido por un conjunto de hechos (H_t). La ejecución de la acción cambia el estado del proceso, añadiendo, eliminando o modificando los valores de los hechos, obteniendo un nuevo conjunto $a_i(H_t) \rightarrow H_{t+1}$.

DEFINICIÓN. 3. Una **verificación** $v_k = \{P_k, \langle \text{continue} | \text{stop} \rangle\}$ es un conjunto de expresiones lógicas P_k que se evalúan sobre un conjunto de hechos H_t tras la ejecución de una acción a_j , de

forma que si $\forall p \in P_k, p(H_t) = \text{true}$, la verificación se cumple y el proceso de razonamiento puede continuar o detenerse en función de lo expresado en la verificación.

DEFINICIÓN. 4. Una **situación** $s_t \in S$ se define como $s_t = \{H_t, a_s, H_{t+1}\}$, donde $H_t \subseteq H$ es un conjunto de hechos que se han producido anteriormente, $a_s \in A$ es una acción que se lleva a cabo, $H_{t+1} \subseteq H$ es un conjunto de hechos que van a modificar su valor $a_s(H_t) \rightarrow H_{t+1}$.

DEFINICIÓN. 5. Una **decisión** $\delta_k = \{P_k, s_f | s_f \in S\}$ es un conjunto de expresiones lógicas P_k que se evalúan sobre un conjunto de hechos $H_t \subseteq H$ en una determinada situación s_i , de forma que $\delta_k(H_t) \rightarrow s_f$ (el problema cambia a la situación s_f) si $\forall p \in P_k, p(H_t) = \text{true}$.

DEFINICIÓN. 6. Un **contexto** $c_i \in C$ se define como $c_i = \{S_{c_i}, s', B_{c_i}\}$, donde $S_{c_i} \subseteq S$ es el conjunto de situaciones que se pueden producir en el contexto c_i , $s' \in S_{c_i}$ es la situación inicial en la que se encuentra el contexto, y B_{c_i} es el conjunto de comportamientos que pueden tener las situaciones en un contexto.

DEFINICIÓN. 7. Se define **comportamiento** de una situación s_i en un contexto $c_j | s_i \in S_{c_j}$ (conjunto de situaciones de c_j), como una tupla formada por $\langle s_i, \delta_{ij}, V_{ij} \rangle | \delta_{ij} \subseteq \delta, V_{ij} \subseteq V$, donde δ_{ij} es un conjunto de decisiones que afectan a la situación s_i en el contexto c_j , y V_{ij} es el conjunto de verificaciones que se deben realizar para la situación s_i en el contexto c_j . Se debe cumplir que $\forall d \in \delta_{ij}, d(H_t) \rightarrow s_t | s_t \in S_{c_j}$ o bien $d(H_t) \rightarrow c_t | c_t \in C$.

DEFINICIÓN. 8. Un **proceso de solución** se caracteriza por $P = \{C, c', S, A, H, V, D\}$, donde C es un conjunto de contextos que se pueden distinguir en el problema a solucionar, c' es el contexto inicial en el que se encuentra el problema cuando comienza el proceso de solución, S es un conjunto de situaciones que pueden ocurrir en el proceso de solución del problema, A es un conjunto de posibles acciones que se pueden llevar a cabo a lo largo del proceso de resolución del problema, H es un conjunto de hechos relevantes para el problema, V es un conjunto de verificaciones que se pueden realizar tras una determinada acción y D es un conjunto de decisiones que sirven para evaluar la situación en la que se encuentra el problema en un momento determinado del proceso de solución.

4.3. Relación entre los Conceptos del Modelo Formal

De la definición de conceptos anterior se desprenden las relaciones entre conceptos que se muestran en la figura 4.2. A continuación se explican cada una de las relaciones.

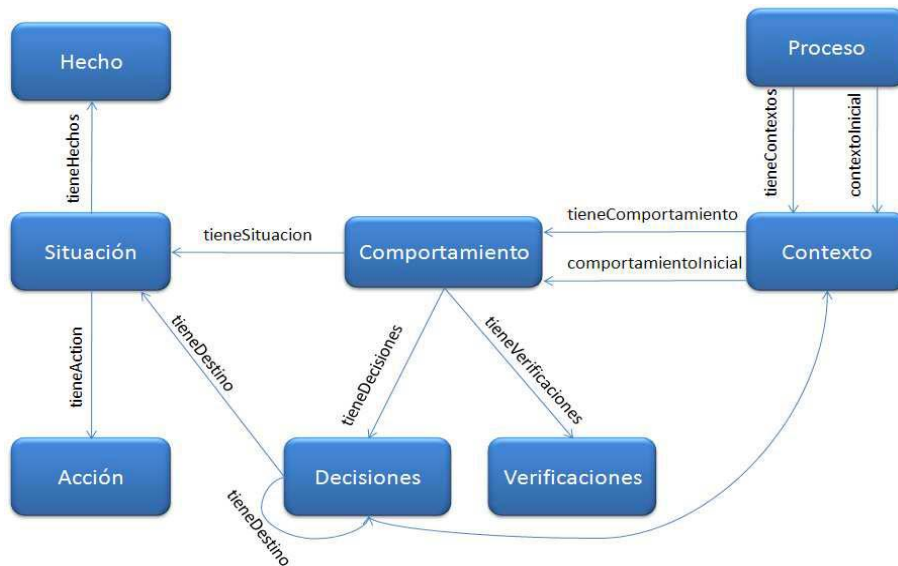


Figura 4.2: Conceptos del modelo de representación.

- En un proceso se pueden diferenciar distintos contextos.
- Cuando un proceso comienza, se encuentra en un contexto que se denomina contexto inicial.
- Cada contexto tiene un conjunto de situaciones que se relacionan entre sí, por medio de las decisiones.
- Una situación está definida por unas condiciones de contorno determinadas, dentro de un contexto, ante las que hay que realizar una acción. Por tanto una situación representa una actividad que hay que realizar en función del estado en que se encuentre el proceso.
- Una acción es una actividad que debe realizarse en un instante concreto del proceso. El modo en que se realiza de cada acción está determinado por la situación en la que se encuentra el proceso en el momento de llevarla a cabo.
- Una situación en un contexto determinado tiene asociado un único **comportamiento**.
- Un comportamiento se compone de un conjunto de **decisiones** y un conjunto de **verificaciones**, que deben evaluarse para la situación y contexto actuales.
- Una **decisión** tiene un destino, que sirve para determinar cuál es la siguiente situación en la que se encuentra el proceso de solución, y por ende, la siguiente actividad a realizar.

El destino puede ser una situación del perteneciente al contexto actual del problema, otra decisión asociada a la situación actual del problema, o un nuevo contexto. Si el destino es otra decisión, el destino será el de esta segunda decisión, siempre y cuando se cumplan sus condiciones. Cuando el destino es un nuevo contexto, la situación de destino será la situación inicial para dicho contexto.

- Las condiciones en las que se encuentra el proceso en un instante concreto se determinan a través de un conjunto de datos relevantes para el problema que se denominan **hechos**. Cada situación tiene asociado un conjunto de hechos cuyo valor puede cambiar al realizar la actividad que tiene asociada.

4.4. Extensión del Modelo: Hacia la Automatización

A partir de la definición formal del modelo es necesario plantear concretamente como se van a representar cada uno de los conceptos expuestos, su semántica y sus relaciones, para permitir modelar el conocimiento acerca de un proceso de resolución de un problema concreto. La definición formal del modelo permite la representación de un proceso de solución, pero la puesta en operación de un sistema que sea capaz de reproducir dicho proceso requiere una serie de consideraciones adicionales.

Dentro de la definición de un proceso se pueden distinguir dos elementos principales:

1. Las acciones que deben ejecutar para llevar a cabo el proceso.
2. Los agentes que intervienen en la ejecución de dichas acciones.

Para comprender la semántica de los elementos que se van a modelar es necesario explicar primero qué tipo de acciones se van a modelar y qué van a representar cada uno de los elementos propuestos.

4.4.1. Agentes que intervienen en un proceso

Los distintos tipos de acción que se van a representar van a estar condicionados por los agentes que van a intervenir en el proceso. El primer punto a destacar es que el agente principal de un proceso definido con el modelo propuesto va a ser un motor de solución (el software capaz de interpretar y ejecutar procesos representados a partir del modelo propuesto). Este motor, a partir de la definición del proceso, lo reproducirá ejecutando las acciones necesarias. Las características del motor de solución se describen en el capítulo 5.

Como se indicó al comienzo del capítulo, en un proceso pueden intervenir distintos agentes, que son los encargados de realizar las actividades necesarias para cumplir con los objetivos de dicho proceso. Cuando se considera un proceso desde el punto de vista de su automatización total o parcial se distinguen tres tipos de agente diferentes:

- El sistema de control. En un proceso automatizado, el sistema de control se encarga de gestionar cada uno de los pasos que se deben seguir en el proceso. Es el agente principal del proceso, pues, mantiene el estado en el que se encuentra el proceso. El sistema de control puede ser una guía para los agentes que intervienen en el proceso, un intermediario entre ellos o una parte activa del mismo realizando algunas actividades por sí mismo.
- Una persona. Realiza un trabajo que no puede ser realizado de forma automática por otro agente. Este trabajo puede ser una verificación, un trabajo manual, la activación de un proceso externo, la respuesta a unas preguntas, o simplemente una confirmación de seguridad. En este caso el sistema de control debe ser capaz tanto de enviar información que la persona pueda interpretar, como de recibirla para conocer el resultado de la operación y continuar con el proceso.
- Un sistema externo. Se trata de un sistema, que puede ser hardware o software, externo al sistema de control. Este sistema se encarga de ejecutar la actividad correspondiente, en función de unos parámetros de operación que reciba.

En el modelo formal de representación no se han considerado los agentes que intervienen en el proceso. Sin embargo, la forma definitiva que adopte una acción en una situación concreta, va a depender del agente que interviene en la misma. Esta definición de los agentes que intervienen en el proceso va a condicionar el modelo conceptual en base a las restricciones que se detallan a continuación:

1. Una acción se va a representar por medio de un mensaje. Este mensaje debe estar en un formato que sea válido tanto para la comunicación con personas como con otros sistemas.
2. Cuando la acción es realizada por el sistema de control (acción interna), es necesario utilizar una interfaz para establecer la comunicación con el sistema externo encargado de ejecutarla. Esta interacción debe estar recogida en el modelo.
3. Además de las acciones descritas, también es posible que en una situación concreta no sea necesario realizar ninguna actividad directa, sino que simplemente sean necesarias algunas comprobaciones antes de continuar con el proceso.

A partir de estas consideraciones, se refina la definición de los conceptos ya expuestos para obtener un modelo capaz de representar procesos de solución de problemas.

4.4.2. Acciones que pueden realizarse en un proceso

Definidos los tipos de agente que pueden intervenir en un proceso, se pueden distinguir distintos tipos de acción en función del agente que las lleve a cabo. Las diferencias entre los distintos tipos de acción se encuentran en las características particulares del agente que las va a realizar.

- **Acción Externa.** Una acción externa es una acción que debe ser realizada por un agente externo. Para poder comunicarse con el agente externo, el motor de solución necesita enviar un mensaje al exterior. Con este mensaje indica al agente que debe realizar una actividad concreta, quedando a la espera de una respuesta para continuar con el proceso.
 - El mensaje estará en formato XML.
 - Es posible utilizar una hoja de estilo para modificar el mensaje.
- **Acción Interna.** Se va a denominar acción interna a una acción que es realizada por el motor que controla la ejecución del proceso. En este caso no se va a enviar un mensaje, pues no es necesario notificar a otros agentes la actividad a realizar. En su lugar, el sistema de control deberá instanciar una interfaz con el sistema hardware o software correspondiente.
 - Es necesario representar un conjunto de parámetros que se utilizarán.
 - Cuando la acción es interna, los parámetros que se utilizan para su ejecución serán distintos en función de la situación y el contexto al que pertenece.
 - Una acción realizada por el sistema de control puede ser utilizada para generar un mensaje que se envíe a otro agente externo.

4.5. Modelo Conceptual

El modelo conceptual representa la definición concreta del modelo formal para permitir la representación del conocimiento a partir de los conceptos definidos anteriormente. El modelo conceptual está basado en el modelo formal, considerando los supuestos y restricciones que determina la definición de un sistema de decisión capaz de controlar un proceso representado en base a este modelo descritos en el apartado anterior.

Para describir el modelo se ha elegido utilizar un lenguaje ontológico: OWL-DL. El motivo de elegir una ontología es que va a permitir representar el conjunto de conceptos que conforman el modelo propuesto, sus propiedades y sus relaciones.

Dados los propósitos de la presente tesis doctoral, se ha seleccionado el lenguaje OWL dado que es la actual recomendación de la W3C y es ampliamente utilizado por la comunidad que se dedica a la web semántica. Este lenguaje está basado en XML lo que facilita el intercambio de información y la utilización por sistemas software. Existen tres sublenguajes de OWL:

- OWL-Lite para clasificaciones y restricciones simples.
- OWL-DL, basado en lógica descriptiva (DL: Description Logics) que proporciona computabilidad (todas las conclusiones son computables) y decibilidad (se puede computar en un tiempo finito). Para conseguir ambos objetivos impone algunas restricciones que pueden afectar a la expresividad.
- OWL-Full, que proporciona máxima expresividad, pero no garantiza que la ontología representada sea computable.

El sublenguaje seleccionado para describir el modelo es OWL-DL pues, al proporcionar computabilidad y decibilidad, abre las posibilidades en el futuro de ampliar el modelo con posibles inferencias.

Sin embargo, como se ha comentado, la expresividad de OWL-DL está sujeta a limitaciones. Para expresar axiomas y restricciones es necesario, en ocasiones, utilizar reglas adicionales sobre la ontología. En la descripción del modelo, estas reglas se van a expresar de acuerdo con SWRL (Semantic Web Rule Lenguaje) (Horrocks et al. 2004) que combina OWL y RuleML. Estas reglas tienen el formato

$$\textit{Antecedente} \implies \textit{Consecuente}$$

de forma que si el antecedente se cumple, el consecuente debe cumplirse también. Cuando el antecedente es vacío se considera que el consecuente debe cumplirse siempre, mientras que si el consecuente está vacío el antecedente no debe cumplirse nunca.

Según la definición de SWRL, toda interpretación de una ontología debe cumplir que todo enlace que satisface el antecedente, satisface también el consecuente. Por tanto determinadas restricciones que no podrán ser expresadas mediante los mecanismos de OWL serán expresadas en base a la sintaxis expresada en (Horrocks et al. 2004)

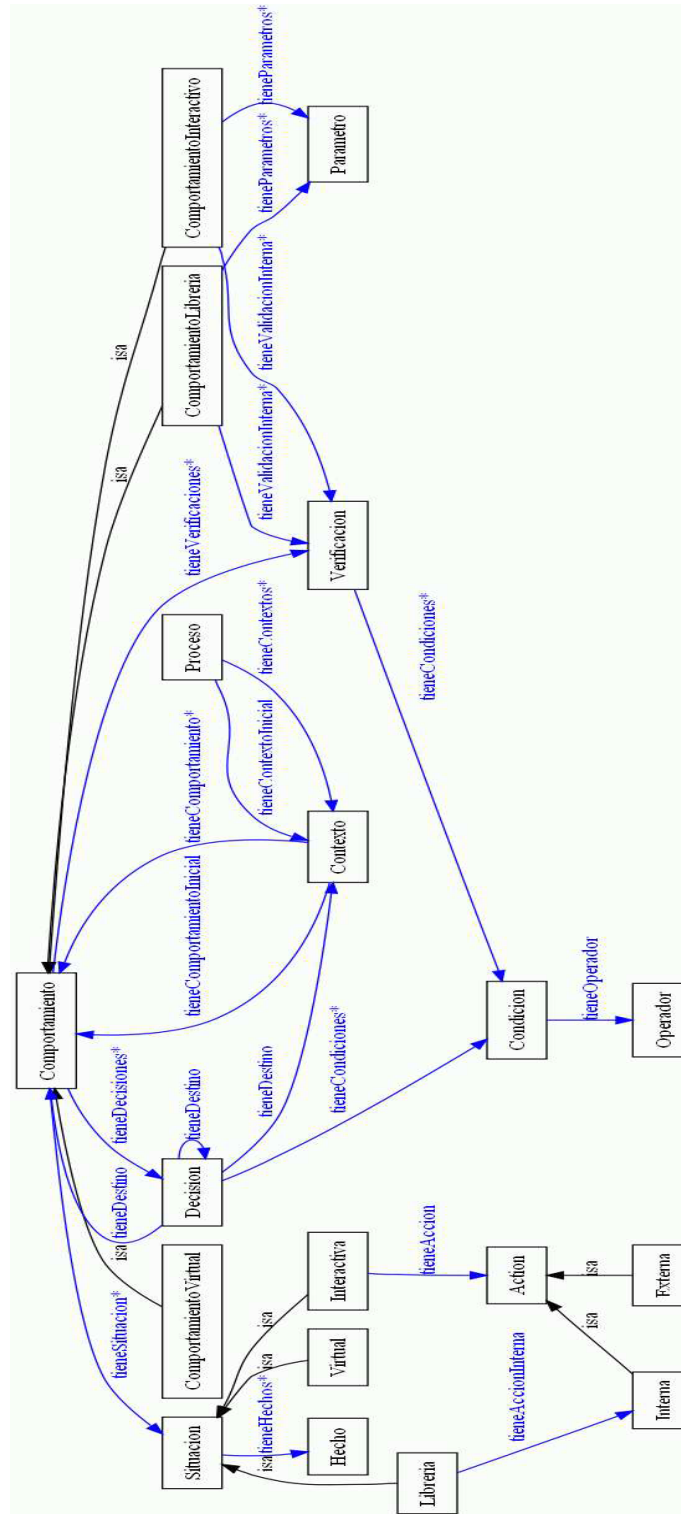


Figura 4.3: Modelo conceptual.



Figura 4.4: Relación entre proceso, contexto y comportamiento.

La figura 4.3 muestra la relación entre los conceptos principales del modelo de representación propuesto.

A continuación se describen los conceptos que ha sido necesario evolucionar para permitir que el modelo sea automatizable.

4.5.1. Proceso

Representa un proceso de solución de un problema. Un proceso puede tener asociados diferentes contextos. Para cumplir con los requisitos impuestos por la definición 8, a continuación se describen las propiedades referidas a contexto. Los conjuntos S, A, H, V, δ , se van a deducir siguiendo las relaciones a partir del contexto, como se puede observar en la figura 4.4.

- Propiedad: `tieneContextos`. Representa el conjunto de contextos que tiene asociados el proceso.
 - *Cardinalidad* ≥ 1 .
 - `allValues from Contexto`.
- Propiedad: `contextoInicial`. Representa el contexto inicial del Proceso.
 - *Cardinalidad* = 1. Un proceso debe siempre tener un contexto inicial.
 - **Functional**. Un proceso solo puede tener un contexto inicial.
 - `subPropertyOf tieneContextos`. Es decir, que uno de los contextos asociados a un proceso mediante la propiedad `tieneContextos`, debe ser el contexto inicial del proceso.

4.5.2. Hecho

Un hecho representa información relevante del entorno del problema para poder solucionarlo. Los hechos pueden tomar un valor por medio de la ejecución de una acción asociada a una situación, o bien a través de los operadores que se utilizan para evaluar las condiciones asociadas a un determinado comportamiento.

- Propiedad: *esMultivaluado*. indica si el operador admite que los operandos que evalúa sean multivaluados.
 - *Cardinalidad* = 1. Un operador debe indicar siempre si admite o no valores multivaluados.
 - *Funcional*. Solo puede tener un valor asociado.
 - *Tipo boolean*. Si toma el valor verdadero el operador admite operandos multivaluados.

- Propiedad: *valorPorDefecto*. Define el valor por defecto que puede tener un hecho.
 - *Axioma*:

$$\text{Hecho}(?h1) \wedge \text{esMultivaluado}(?h1, \text{false}) \rightarrow (\text{valorPorDefecto} \leq 1)(?h1).$$

Si un hecho no es multivaluado, como máximo solo puede tener un valor por defecto. En caso de ser multivaluado no hay restricción de cardinalidad, pues podría tener varios valores, uno o ninguno.
 - *Funcional*. Esta propiedad solo puede tener un valor asociado.
 - *Tipo string*.

4.5.3. Acción

Como se describe en el modelo formal, una acción a_i representa una actividad que debe realizarse en una situación s_t para modificar las condiciones de contorno H_t del problema, de forma que $a_i(H_t) \rightarrow H_{t+1}$. Por ejemplo, si se está representando un proceso de fabricación, una acción puede ser activar la línea de producción, consultar la base de datos de materiales o calcular las revoluciones de una operación de taladro.

Como se ha indicado en el apartado 4.4.2, una actividad puede ser realizada de forma automática por el motor de solución (acción interna) o por una entidad externa (acción externa). La figura 4.5 muestra los dos tipos de acción posibles. Las flechas sin etiquetar denotan la relación de tipo es-un (jerarquía), que se utilizan en el resto del capítulo.

Las siguientes subsecciones detallan las características de cada una de ellas.

Acción externa

Representa una actividad del proceso que es realizada por un agente externo al sistema de control. Como se ha descrito, una acción interna se va a concretar a través de un mensaje que se

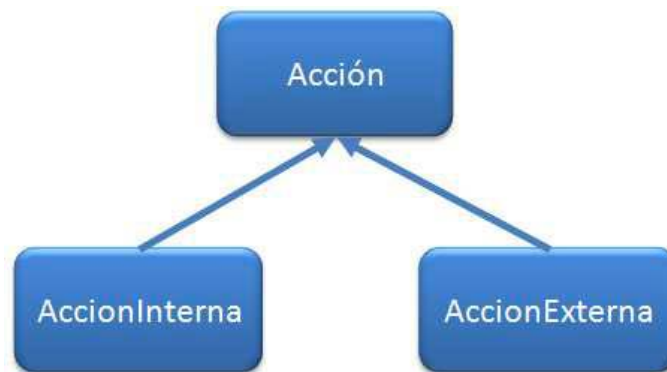


Figura 4.5: Distintos tipos de acción.

envía al agente que la va a ejecutar. Al existir dos tipos de agente externo, persona o sistema, el mensaje que se representa debe ser lo suficientemente flexible para facilitar la comunicación con ambos.

Para concretar el modelo, se ha decidido que el formato de mensaje sea XML. De esta forma, el mensaje es una cadena de caracteres con un formato flexible que permite tanto la comunicación con personas, a través de un navegador, como con otros sistemas, a través de un formato XML definido.

- Propiedad: pathMensaje. Representa el mensaje que se enviará para llevar a cabo la acción.
 - *Cardinalidad* = 1. Una acción externa debe siempre tener un mensaje asociado.
 - Funcional. Una acción externa solo puede tener un mensaje asociado.
 - Tipo string.

Acción interna

Una acción interna es aquella que es realizada por el motor de solución que gestiona el conocimiento del proceso. La acción interna se traduce en la llamada a otra aplicación para que realice la acción necesaria, pasándole un conjunto de parámetros asociados al comportamiento. Como resultado de esta llamada, el motor de solución obtiene un resultado que produce una modificación en las condiciones de contorno del problema.

- Propiedad: nombreObjeto. Representa la interfaz que se debe instanciar para que el motor de solución pueda realizar la llamada interna.
 - *Cardinalidad* = 1. Una acción externa debe siempre tener un mensaje asociado.

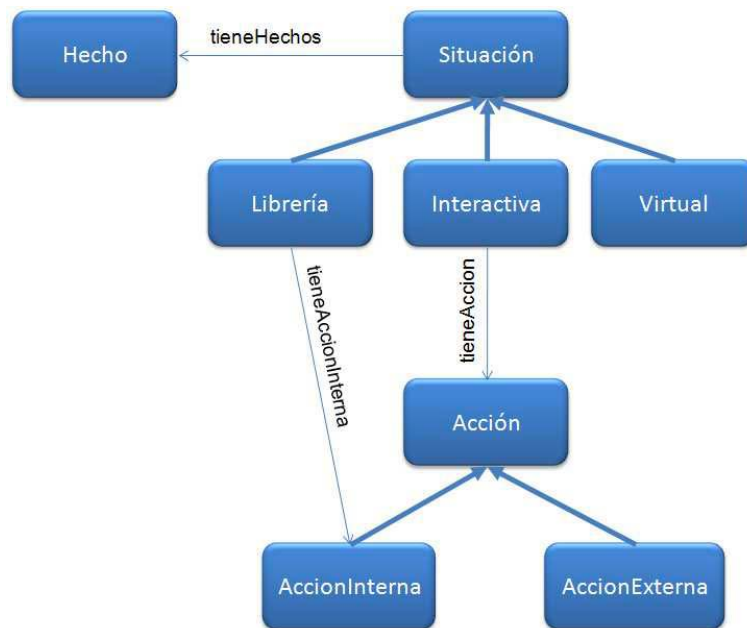


Figura 4.6: Clases de situación en función de las acciones.

- Funcional. Una acción externa solo puede tener un mensaje asociado.
- Tipo string.

4.5.4. Situación

En la propuesta que se presenta en esta tesis en una situación determinada, caracterizada por un conjunto de hechos relevantes que se han producido previamente, es necesario llevar a cabo una acción para avanzar hacia la solución del problema. La figura 4.6 muestra la estructura jerárquica del concepto Situación en función de las acciones que tiene asociadas.

- Propiedad: tieneHechos. Representa la lista de hechos relevantes que pueden tomar valor en la situación. El conjunto de hechos que se han producido anteriormente está implícito en el conjunto de situaciones previas que se han seguido hasta llegar a la situación actual.
 - No hay restricción de cardinalidad.
 - allValuesFrom Hecho. Solo puede tomar valores de tipo Hecho.

Los dos tipos de acción que se distinguen en el apartado anterior, van a condicionar tres tipos de situaciones distintos, que se detallan a continuación.

Situación automática o librería

Representa una situación en la que es el sistema el que debe realizar una acción determinada. Esta acción se realiza a través de una librería externa que permite interactuar con otros sistemas a través de unos parámetros.

Las propiedades específicas de una situación automática son:

- Propiedad: `tieneAccionInterna`. Representa la acción interna que tiene asociada. .
 - *Cardinalidad* = 1. Una librería siempre tiene una acción interna asociada.
 - `allValuesFrom Interna`. Solo puede tomar valores de tipo acción Interna.

El resto de propiedades necesarias para una acción interna está determinado por el contexto en el se produzca la situación.

La figura 4.7 muestra el funcionamiento de este tipo de situaciones:

1. El sistema de control envía un conjunto de parámetros a una aplicación externa a través de una interfaz.
2. La aplicación realiza un conjunto de operaciones y devuelve a través de dicha interfaz un conjunto de resultados que serán tratados como hechos.
3. El resultado es evaluado y continua el proceso de resolución del problema

Situaciones interactivas

Representa una situación en la que es necesaria una acción de una entidad externa al sistema, de la cual se espera una respuesta. Se concreta en el envío de un mensaje al exterior en formato XML del que se espera una respuesta determinada, que condicionará el paso a una situación distinta. Este es el caso de una pregunta que debe ser respondida por una persona, o un mensaje enviado a otro sistema que debe responder con el resultado de una operación concreta.

La figura 4.8 muestra el funcionamiento de este tipo de acciones. El mensaje es enviado al agente externo y éste devuelve un conjunto de hechos que reflejarán el cambio en las condiciones de contorno del problema.

Las propiedades específicas de una situación interactiva son:

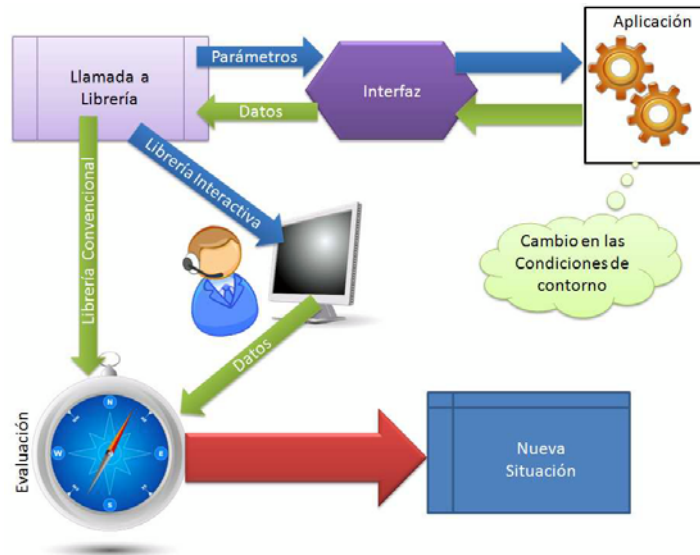


Figura 4.7: Esquema de funcionamiento de una situación automática.



Figura 4.8: Esquema de funcionamiento de una situación interactiva.

- Propiedad: *tieneAccion*. Representa la acción que está asociada a la situación. La acción puede ser tanto interna como externa, es decir, que el mensaje que se envía en esta situación puede ser resultado de una llamada a una librería externa (acción externa), como el contenido de un fichero (acción interna).
 - *Cardinalidad* = 1. Una situación interactiva debe tener una acción asociada.
 - *Functional*. Una situación interactiva solo puede tener una acción asociada.
 - *allValuesFrom Acción*. Solo puede tener relacionadas acciones a través de esta propiedad.

- Propiedad: *hojaDeEstilo*. Representa un fichero que contiene una hoja de estilo a aplicar sobre el mensaje XML de la acción que tiene asociada.
 - No tiene restricción de cardinalidad. Una situación externa puede tener o no una hoja de estilo asociada.
 - *Functional*. Como máximo solo puede tener una hoja de estilo asociada.
 - Tipo *string*.

Un caso especial se produce cuando el resultado de una acción interna es el mensaje que se utiliza para una acción externa. Por ello la acción de una situación interactiva puede ser tanto interna como externa. En este caso se le denomina también **librería interactiva**.

La figura 4.7 muestra el funcionamiento de una acción que es a la vez interna y externa. La semántica de este caso particular es la siguiente:

1. El sistema de control realiza una acción interna.
2. Se verifican los resultados por medio de verificaciones.
3. El resultado obtenido lo envía a un agente externo.
4. El agente externo realiza la acción y responde al sistema de control.
5. El sistema de control evalúa el resultado y continúa el proceso de resolución.

Situación virtual

Una situación virtual es aquella que no tiene ninguna acción que realizar. En este tipo de situaciones solo se comprobarán las verificaciones y las decisiones que estén asociadas al comportamiento para el contexto concreto en el que se encuentren.

4.5.5. Contexto

Representa cada uno de los contextos que se pueden dar dentro de un proceso determinado. Tal y como se ha indicado en la definición 6, un contexto va a componerse de varios comportamientos, uno de los cuáles será el comportamiento inicial. El comportamiento, va a caracterizar la forma en que se van a tomar las decisiones para una situación concreta dentro del contexto, por lo que a través de la relación con los diferentes comportamientos se representa el conjunto de situaciones S_{c_i} para cada contexto c_i . Así mismo, a través de la propiedad `tieneComportamientoInicial`, se consigue obtener también $s' \in S_{c_i}$.

- Propiedad: `tieneComportamientos`. Representa el conjunto de comportamientos que están asociados el proceso.

- *Cardinalidad* ≥ 1 .
- `allValues from Comportamiento`.
- `inverseFunctional`. Un comportamiento solo puede pertenecer a un contexto.
- Axioma:

$$\text{Contexto}(?c1) \wedge \text{tieneComportamiento}(?c1, ?b1) \wedge \text{tieneComportamiento}(?c1, ?b2) \wedge \text{tieneSituacion}(?b1, ?s) \wedge \text{tieneSituacion}(?b2, ?s) \rightarrow \langle \text{vacío} \rangle$$

La misma situación no puede tener más de un comportamiento en un contexto. El consecuente vacío implica que no se puede cumplir nunca esta condición (esta condición describe el caso en que para un contexto $c1$ hay dos comportamientos $b1$ y $b2$ para la misma situación s).

- Propiedad: `comportamientoInicial`. Representa el comportamiento inicial del Proceso.
 - *Cardinalidad* = 1. Un contexto debe siempre tener un comportamiento inicial, que determinará cuál es la situación inicial del contexto.
 - `Functional`. Un contexto solo puede tener un comportamiento inicial.
 - `subPropertyOf tieneComportamiento`. Es decir, que uno de los comportamiento asociados a un proceso mediante la propiedad `tieneComportamiento`, debe ser el comportamiento inicial del proceso.

4.5.6. Comportamiento

Tal y como indica la definición 7, el comportamiento determina un conjunto de características concretas para una situación, en función del contexto en el que se produce. En particular, el

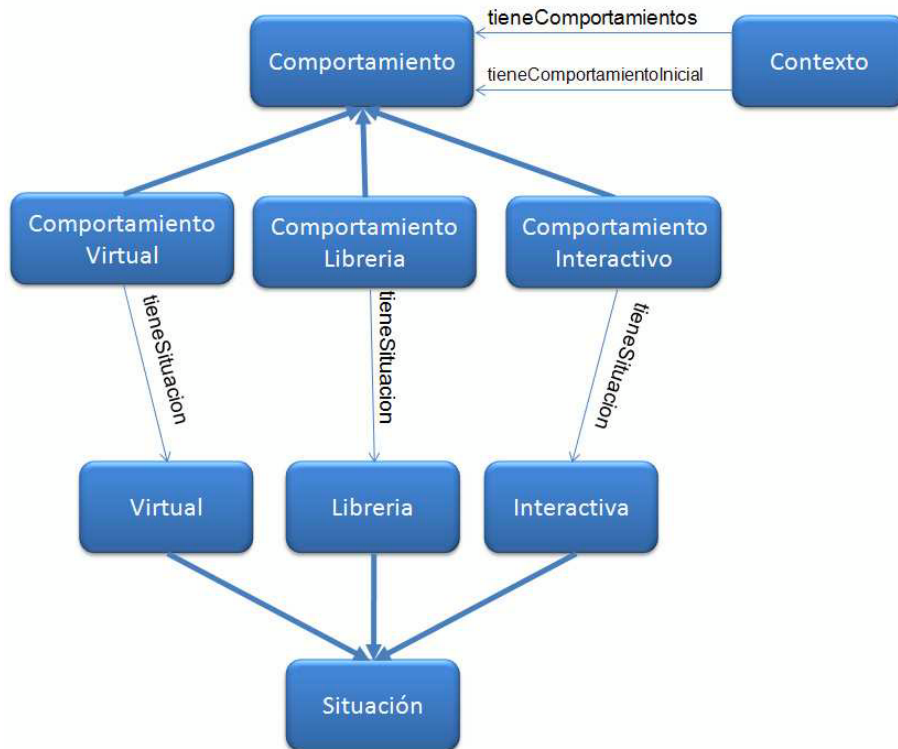


Figura 4.9: Tipos de comportamiento en función de las situaciones.

comportamiento de una situación en un contexto concreto va a estar formado por un conjunto de verificaciones y un conjunto de decisiones que se pueden tomar. Así, dependiendo del contexto en el que se produzca la situación, las comprobaciones a realizar y las decisiones a tomar pueden ser diferentes.

Dado que cada situación que se puede plantear en el proceso tiene características diferentes, se van a distinguir tres tipos de comportamiento: comportamiento virtual, comportamiento librería y comportamiento interactivo. La figura 4.9 identifica como cada uno de los tipos de comportamiento se relaciona con los distintos tipos de situación que se pueden modelar. En la figura 4.3 esta relación aparece simplificada por motivos de claridad.

Las características comunes a todos los comportamientos son:

- **Propiedad: evaluarTodasDecisiones.** Indica que para el comportamiento de la situación en el contexto actual, se deben evaluar todas las decisiones que tiene asociada. Esto significa que el motor de solución, tras comprobar todas las decisiones, pasará por todas las situaciones de aquellas que se cumplan (ver capítulo 5). Esto implica que cada situación puede tener una estrategia diferente de activación de reglas.

- *Cardinalidad* = 1. Todos los comportamientos deben especificar si se evaluarán todas las decisiones o no.
- *Functional*. Esta propiedad solo puede tener un y solo un valor.
- *Tipo boolean*. Si la propiedad toma el valor verdadero el motor de solución evaluará todas las decisiones y visitará todas las situaciones relacionadas con aquellas que se cumplan. Si toma el valor falso, el motor de solución continuará por la situación de la primera decisión que se cumpla.
- *Propiedad: tieneDecisiones*. Lista de decisiones que están asociadas al comportamiento de una situación en un contexto determinado.
 - No tiene restricciones de cardinalidad. Un comportamiento puede tener de 0 a n decisiones asociadas. Cuando no tiene decisiones, se dice que la situación es terminal para el contexto en el que se encuentra.
 - *allValuesFrom Decision*. Todos los valores de la lista deben ser de tipo *Decision*.
- *Propiedad: tieneVerificaciones*. Lista de verificaciones que se deben realizar tras realizar la acción que tiene asociada la situación (ver capítulo 5).
 - No tiene restricciones de cardinalidad. El comportamiento de una situación en un contexto determinado puede no requerir ningún tipo de verificación.
 - *allValues from Verificación*.

Además de las propiedades comunes, cada tipo de comportamiento va a tener unas características propias, en función del tipo de situación que tenga asociado. La figura 4.10 muestra las relaciones que van a caracterizar cada uno de los tipos de comportamiento.

Comportamiento virtual

Es el comportamiento asociado a una situación virtual. Este tipo de comportamiento no tiene ninguna propiedad adicional, pues las situaciones virtuales no tienen una acción que realizar.

- *Propiedad: tieneSituación*. Situación virtual que está asociada al comportamiento para un contexto determinado.
 - *Cardinalidad* = 1. Un comportamiento virtual tiene siempre asociada una situación virtual.

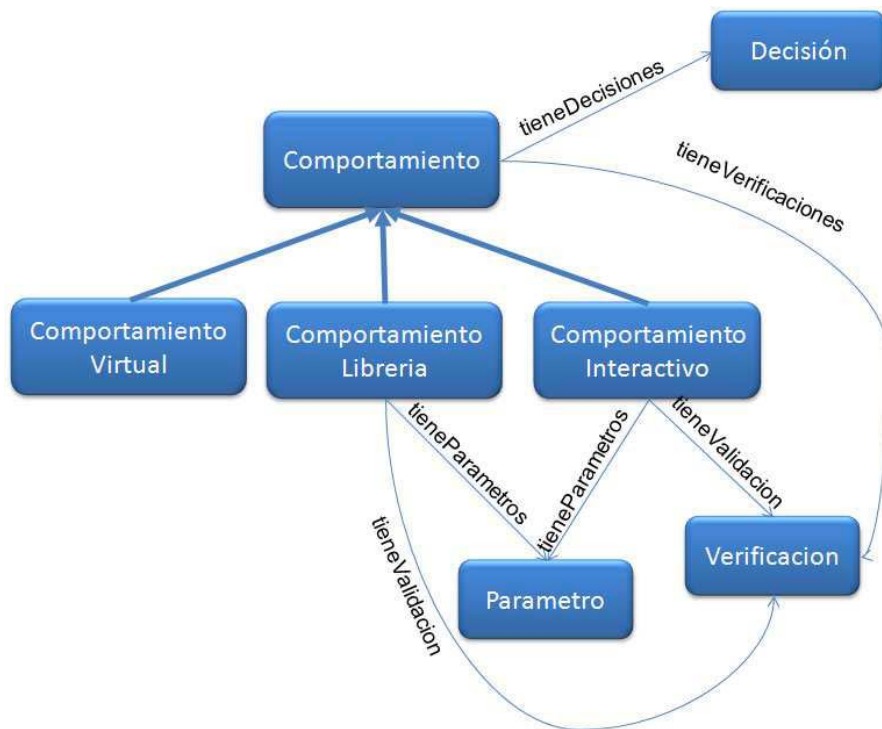


Figura 4.10: Relaciones de los distintos tipos de comportamiento.

- **Functional.** Un comportamiento virtual tiene una y solo una situación virtual asociada.
- **allValuesFrom Virtual.** Un comportamiento virtual solo puede estar asociado a situaciones de tipo virtual.

Comportamiento interactivo

El comportamiento interactivo determina las características que tendrá una situación interactiva, es decir, aquella que conlleva una acción que se es ejecutada por un agente externo.

- **Propiedad: tieneParámetros.** Lista de parámetros que tiene la situación interactiva. Los parámetros para la situación interactiva solo serán necesarios en el caso de que el mensaje sea generado a partir de una acción interna.
- **Axioma:**

$$\begin{aligned}
 & ComportamientoInteractivo(?x1) \wedge tieneSituacion(?x1, ?s1) \\
 & \wedge tieneAccion(?s1, ?a1) \wedge Externa(?a1) \rightarrow (tieneParametros = 0)(?x1).
 \end{aligned}$$

Cuando el comportamiento interactivo tiene asociado una situación que implica una

acción externa, la cardinalidad de la propiedad *tieneParametros* será 0, pues las acciones externas no requieren parámetros. En caso contrario no existen restricciones de cardinalidad, pues la acción interna puede tener o no parámetros.

- InverseFunctional. Un parámetro solo puede aparecer en un comportamiento. De esta forma se evitan efectos cruzados.
 - allValuesFrom Parametro. Todos los valores de esta propiedad son de tipo parámetro.
- Propiedad: *tieneSituación*. Situación interactiva para la que se define el comportamiento en un contexto determinando.
- *Cardinalidad* = 1. Un comportamiento debe tener una situación asociada.
 - Functional. Un comportamiento tiene asociada una y solo una situación.
 - allValuesFrom Interactiva. Todas situación asociada a un comportamiento interactivo deben ser una situación interactiva.
- Propiedad: *tieneValidacionInterna*. Lista de verificaciones que se deben realizar tras ejecutar una acción interna para la situación y contexto que determina el comportamiento.
- Axioma:

$$\text{ComportamientoInteractivo}(?x1) \wedge \text{tieneSituacion}(?x1, ?s1) \\ \wedge \text{tieneAccion}(?s1, ?a1) \wedge \text{Externa}(?a1) \rightarrow (\text{tieneValidacionInterna} = 0)(?x1).$$

Cuando el comportamiento interactivo tiene asociado una situación que implica una acción externa, la cardinalidad de la propiedad *tieneValidacionInterna* será 0, pues las acciones externas no requieren parámetros. En caso contrario no existen restricciones de cardinalidad, pues la acción interna puede tener o no parámetros.
 - allValuesFrom Verificación. Todos los valores de esta propiedad son de tipo Verificación.

Comportamiento librería

El comportamiento librería caracteriza a las situaciones automáticas en un contexto determinado. Las propiedades específicas de este tipo de comportamiento son:

- Propiedad: *tieneSituación*. Situación automática asociada al comportamiento de tipo librería.

- *Cardinalidad* = 1. Un comportamiento de tipo librería tiene siempre asociada una situación interna.
 - *Functional*. Un comportamiento de tipo librería tiene una y solo una situación interna asociada.
 - *allValuesFrom Interna*. Un comportamiento librería solo puede estar asociado a situaciones de tipo interna.
- Propiedad: *tieneParámetros*. Lista de parámetros que tiene la situación interactiva. Los parámetros para la situación interactiva solo serán necesarios en el caso de que el mensaje sea generado a partir de una acción interna.
 - No tiene restricciones de cardinalidad. Una situación interna puede tener o no asociados un conjunto de parámetros.
 - *InverseFunctional*. Un parámetro solo puede aparecer en un comportamiento. De esta forma se evitan efectos cruzados.
 - *allValuesFrom Parametro*. Todos los valores de esta propiedad son de tipo parámetro.
 - Propiedad: *tieneValidacionInterna*. Lista de verificaciones que se deben realizar tras ejecutar una acción interna para la situación y contexto que determina el comportamiento.
 - No tiene restricciones de cardinalidad. Una situación interna puede tener o no asociados un conjunto de verificaciones.
 - *allValuesFrom Verificación*. Todos los valores de esta propiedad son de tipo Verificación.

4.5.7. Parámetro

Este concepto representa cada uno de los parámetros que se enviará a a una acción interna en función del contexto en que se encuentre. Las propiedades de un parámetro son:

- Propiedad *nombreParametro*: contiene el nombre del parámetro que se va a enviar.
 - *Cardinalidad* = 1. El nombre del parámetro debe tener un valor asociado.
 - *Functional*. Un parámetro tiene un y solo un nombre.
 - Tipo *string*.

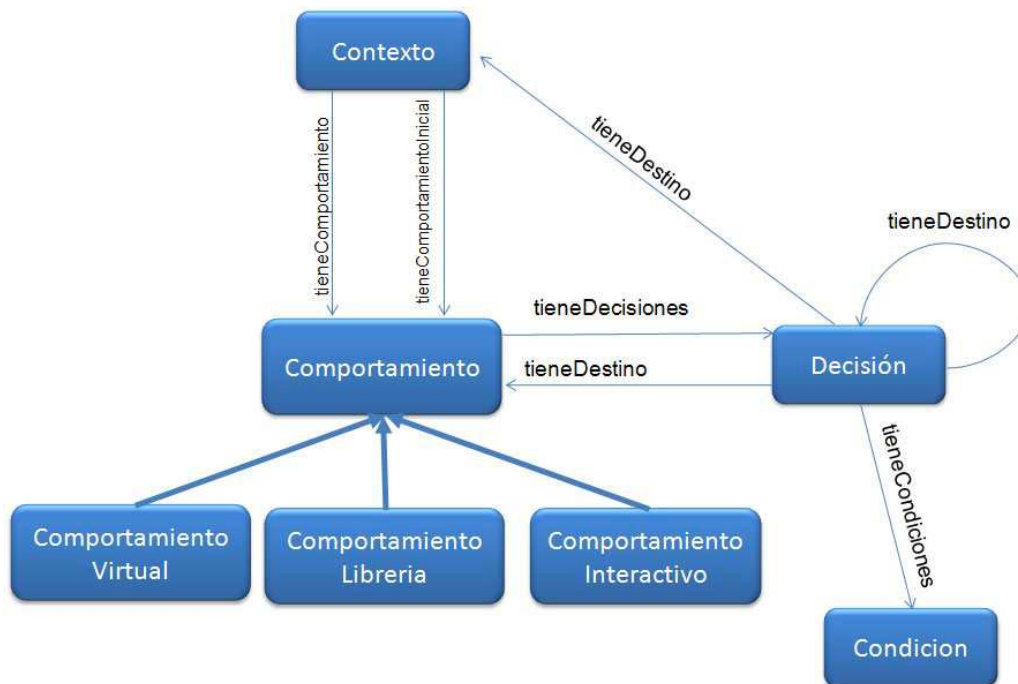


Figura 4.11: Relaciones del concepto decisión.

- Propiedad valorParametro: contiene el valor del parámetro que se va a enviar.
 - Cardinalidad = 1. El valor del parámetro debe tener un valor asociado.
 - Funcional. Un parámetro tiene un y solo un valor.
 - Tipo string.

4.5.8. Decisiones

Según la definición formal de decisión (definición 5), una decisión es un conjunto de condiciones que, en caso de cumplirse, determinan cuál es la siguiente situación en la que se encuentra el problema. Con los conceptos de comportamiento y decisión hay que considerar que una decisión puede determinar la siguiente situación a través de tres posibles tipos de destino (figura 4.11):

1. Una situación del mismo contexto en el que se encuentra el problema, por lo tanto, debido a la cardinalidad de las relaciones, para simplificar las restricciones se ha establecido enlace con el comportamiento correspondiente. Cuando el destino de una decisión es un comportamiento, la situación en la que se encuentra el problema es la asociada a dicho

comportamiento. Es necesario hacer notar que de acuerdo con la definición de decisión, el contexto del comportamiento de destino debe ser el mismo que el contexto actual.

2. Un contexto (se produce un cambio de contexto en el problema). Cuando el destino de una decisión es un contexto, la siguiente situación es la situación asociada al comportamiento inicial de dicho contexto.
3. Una decisión. La decisión de destino debe pertenecer al mismo comportamiento de la decisión de origen. Cuando el destino es una decisión, la decisión destino se evalúa y, en caso de cumplirse, el destino es el que ésta marque. En caso de que no se cumpla se asume que la decisión de origen no se ha cumplido.

En base a las restricciones anteriores, las propiedades de una decisión son las siguientes:

- Propiedad: *tieneCondiciones*. Representa la lista de condiciones que deben cumplirse para que se pueda dar por buena la decisión.
 - *Cardinalidad* ≥ 1 . Una decisión debe tener al menos una condición asociada.
 - *allValuesFrom Condición*. Solo puede tomar valores de tipo Condición.
- Propiedad: *tieneDestino*. Representa la lista de condiciones que deben cumplirse para que se pueda dar por buena la decisión.
 - *Cardinalidad* = 1. Una decisión debe tener siempre un destino.
 - *Functional*. Una decisión tiene un y solo un destino.
 - *allValuesFrom Decision \sqcup Comportamiento \sqcup Contexto*. Solo puede tomar valores de tipo Decision, Comportamiento o Contexto.
 - Axiomas: es necesario verificar dos condiciones, de acuerdo con las restricciones de los tipos de destino mencionados:
 - $Contexto(?c) \wedge tieneComportamientos(?c, ?b) \wedge tieneDecision(?b, ?d) \wedge tieneDestino(?d, ?b2) \wedge Comportamiento(?b2) \rightarrow tieneComportamientos(?c, ?b2)$.
 Si una decisión pertenece a un comportamiento de un contexto concreto y su destino es otro comportamiento, entonces el comportamiento de destino debe pertenecer al mismo contexto.

- $Contexto(?c) \wedge tieneComportamientos(?c, ?b) \wedge tieneDecisiones(?b, ?d) \wedge tieneDestino(?d, ?d2) \wedge Decision(?d2) \rightarrow tieneDecisiones(?c, ?d2)$.

Si una decisión tiene como destino otra decisión, ambas deben pertenecer al mismo comportamiento. Esto puede resultar útil para evaluar decisiones de prioridad menor en caso de que se cumplan unas determinadas condiciones.

4.5.9. Verificación

De acuerdo con la definición de verificación (definición 3), una verificación es un conjunto de condiciones que se van a evaluar tras la ejecución de una acción. Dada la diversidad de acciones que se pueden producir, hay dos momentos en los que se puede realizar una verificación. La figura 4.10 muestra las relaciones del concepto Verificación en función del tipo de comportamiento al que están asociado.

- Justo antes de evaluar las decisiones asociadas al comportamiento de la situación para el contexto actual. Estas verificaciones representan una comprobación general del estado del contorno del problema antes de continuar adelante con el proceso de resolución.
- Justo después de ejecutar una acción interna. En adelante este tipo de verificaciones se denominarán *validación*. Estas validaciones representan las comprobaciones que hay que realizar justo después de llamar a otra aplicación, para asegurar que el resultado ha sido el adecuado.

Existen dos motivos para separar en dos las verificaciones. El primero es el de la claridad, pues se separan las comprobaciones internas de las comprobaciones sobre el estado general del problema. El segundo es que existe un caso en el que es obligada esta separación: cuando el resultado de una acción interna es el mensaje que utiliza la situación interactiva. En este caso particular, las validaciones se comprueban justo después de la acción interna, mientras que las verificaciones se comprueban cuando se ha recibido la respuesta del agente externo.

Cuando una verificación se cumple es necesario tomar algún tipo de acción sobre el proceso: detenerlo o continuar con el proceso de resolución. En caso de detenerse, el proceso de resolución regresará a la última situación válida por la que se haya pasado, enviando un mensaje de error. En caso de continuar, se evaluarán las decisiones asociadas al comportamiento de la situación para el contexto actual, y se continuará con la siguiente situación, enviando además el mensaje de error.

Las propiedades que definen una verificación son:

- Propiedad: *tieneCondiciones*. Representa la lista de condiciones que deben cumplirse para que se pueda dar por buena la decisión.
 - *Cardinalidad* ≥ 1 . Una verificación debe tener al menos una condición asociada.
 - *allValuesFrom* Condición. Solo puede tomar valores de tipo Condición.
- Propiedad: *accionVerificacion*. Cuando una verificación se cumple pueden tomarse dos acciones: detener el proceso regresando a la última situación válida o continuar adelante.
 - *Cardinalidad* = 1. Una decisión debe tener la acción a tomar definida.
 - Valores *owl:oneOf*{“parar”, “continuar”}. Los valores que puede tomar son “parar”, deteniendo el proceso de resolución, o “continuar”, continuando el proceso de decisión.
- Propiedad: *mensajeError*. Mensaje o código de error que se envía en caso de que la verificación se cumpla .
 - *Cardinalidad* = 1. Una decisión debe tener un mensaje definido.
 - Tipo *string*.

4.5.10. Operador

Un operador es un objeto que va a realizar un conjunto de operaciones sobre un conjunto de parámetros (operandos) que recibirá, devolviendo verdadero si se satisfacen los objetivos que tiene marcados, o falso en caso contrario. Un ejemplo sencillo es el operador de igualdad. Recibirá dos parámetros, los comparará, y devolverá verdadero si son iguales y falso en caso contrario.

Las propiedades de un operador son:

- Propiedad: *admiteMultivaluados*. indica si el operador admite que los operandos que evalúa sean multivaluados.
 - *Cardinalidad* = 1. Un operador debe indicar siempre si admite o no valores multivaluados.
 - *Functional*. Solo puede tener un valor asociado.
 - Tipo *boolean*. Si toma el valor verdadero el operador admite operandos multivaluados.

- Propiedad: esOperadorBinario. Indica si el operador admite dos operandos.
 - *Cardinalidad* = 1. Un operador debe indicar siempre si admite o no dos operandos.
 - *Functional*. Solo puede tener un valor asociado.
 - Tipo boolean. Si toma el valor verdadero el operador admite dos operandos, mientras que si toma el valor falso solo admite un operando.
- Propiedad: objetoOperador. Identificador del objeto que el motor de solución debe instanciar para realizar la operación.
 - *Cardinalidad* = 1. Un operador debe indicar siempre el objeto que necesita ser instanciado.
 - *Functional*. Solo puede tener un valor asociado.
 - Tipo string.
- Propiedad: pathOperador. Ruta donde se encuentra el fichero que contiene el objeto que representa el operador.
 - *Cardinalidad* = 1. Un operador debe indicar siempre la ruta del fichero que contiene el objeto a instanciar.
 - *Functional*. Solo puede tener un valor asociado.
 - Tipo string.

4.5.11. Condición

Una condición representa una expresión lógica que puede evaluarse para determinar si es verdadera o falsa. Para poder evaluar una condición es necesario determinar la forma en que se puede expresar la misma. Se ha decidido representar la condición como una combinación entre un primer operando, un operador y un segundo operando. De esta forma se pueden crear nuevos operadores que se utilizarán en las condiciones que se definan.

La figura 4.12 muestra las relaciones del concepto Condición, con los conceptos Verificación, Decisión y Operador.

Los operandos pueden ser multivaluados, es decir, que cada operando puede ser una lista de valores. Una condición debe tener al menos definidos el primer operando y el operador. Si el operador es binario, la condición deberá además tener definido el segundo operando. Finalmente, si el operador admite multivaluados, los operandos pueden tener una cardinalidad mayor que uno.

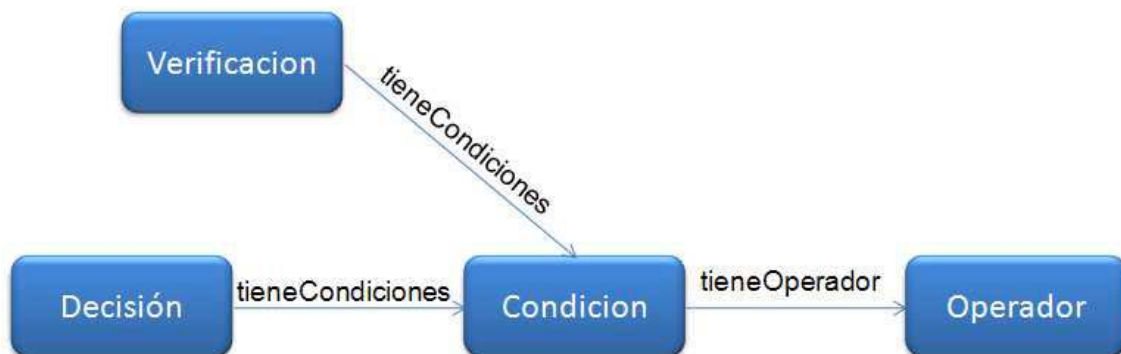


Figura 4.12: Relaciones del concepto condición.

- Propiedad: primerOperando. Lista de cadenas de caracteres que pueden formar el primer operando.
 - Cardinalidad ≤ 1 . Por defecto, una condición debe tener definido al menos un primer operando.
 - Axioma:

$$\text{Condicion}(?c1) \wedge \text{tieneOperador}(?c1, ?o1) \wedge \text{admiteMultivaluados}(?o1, \text{false}) \rightarrow (\text{primerOperando} = 1)(?c1).$$
 Si el operador de una condición no admite multivaluados, entonces el primer operando debe tomar un y solo un valor.
 - Tipo string.
- Propiedad: segundoOperando. Lista de cadenas de caracteres que pueden formar el segundo operando de una condición.
 - Axioma:

$$\text{Condicion}(?c1) \wedge \text{tieneOperador}(?c1, ?o1) \wedge \text{admiteMultivaluados}(?o1, \text{false}) \wedge \text{esOperadorBinario}(?o1, \text{true}) \rightarrow (\text{segundoOperando} = 1)(?c1).$$
 Si el operador de una condición no admite multivaluados y es binario, entonces el segundo operando debe tomar un y solo un valor.
 - Axioma:

$$\text{Condicion}(?c1) \wedge \text{tieneOperador}(?c1, ?o1) \wedge \text{admiteMultivaluados}(?o1, \text{true}) \wedge \text{esOperadorBinario}(?o1, \text{true}) \rightarrow (\text{segundoOperando} \geq 1)(?c1).$$
 Si el operador de una condición no admite multivaluados y es binario, entonces el segundo operando puede tomar uno o varios valores.

- Axioma:

$$\text{Condicion}(?c1) \wedge \text{tieneOperador}(?c1, ?o1) \wedge \text{esOperadorBinario}(?o1, \text{false}) \\ \rightarrow (\text{segundoOperando} = 0)(?c1).$$

Si el operador de una condición no es binario, entonces el segundo operando no puede tomar ningún valor.

- Tipo string.
- Propiedad: tieneOperador. Operador que debe ser utilizado para evaluar la condición.
 - *Cardinalidad* = 1. Una condición debe tener siempre un operador asociado.
 - *Functional*. Solo puede tener un operador asociado.
 - *allValues from Operador*. Todos los valores relacionados deben ser de tipo Operador.

4.6. Resumen

En este capítulo se ha definido un modelo de representación del conocimiento basado en la representación de las diferentes situaciones que se pueden producir a lo largo del proceso de resolución de un problema. Cada situación tiene asociada una acción a realizar, que representa la acción que el experto realiza en dicha situación. Esta acción puede representar un mensaje que se envía al agente externo para que realice una tarea y envíe la información del resultado al sistema (situación interactiva), o una llamada a otro sistema para que realice una operación determinada y envíe la información resultante (situación automática). Las comprobaciones que el experto realiza sobre las diferentes acciones se representan por medio de verificaciones, mientras que la valoración que el experto realiza para determinar cuál es la siguiente situación en la que se encuentra el problema se representa por medio de decisiones. La definición de contextos permite la reutilización de las situaciones definidas en diferentes condiciones del contorno.

El capítulo siguiente completa la semántica del modelo con la descripción del algoritmo de interpretación de un proceso definido a partir del modelo propuesto.

En el capítulo anterior se ha presentado un modelo de representación, expresado a través un lenguaje ontológico. A partir de este modelo es posible representar procesos de resolución de problemas, identificando contextos, situaciones, acciones y decisiones. Este capítulo propone un algoritmo para automatizar un proceso definido en base al modelo propuesto. Además se define e implementa la arquitectura de un motor de solución capaz de interpretar un proceso expresado de acuerdo con dicho algoritmo. Este motor de solución se utilizará posteriormente para validar la aplicación del modelo a la estimación de precios de fabricación.

5.1. Introducción

Una vez que se ha representado el conocimiento acerca del proceso de solución del problema, éste puede ser utilizado para transmitir el conocimiento entre personas de una o varias organizaciones. Sin embargo el conocimiento representado de acuerdo con unas reglas y con un sentido definido, es susceptible de ser utilizado por un sistema para poder tomar decisiones.

El motor de solución es el encargado de interpretar la información que se ha representado por medio del modelo propuesto para resolver un problema, interpretando las distintas situaciones que se pueden plantear, ejecutando las acciones asociadas a cada situación y evaluando las distintas decisiones que determinan la siguiente situación en la que se encuentra el problema.

El motor de solución actuará como un sistema basado en conocimiento a partir de un proceso de decisión representado utilizando el modelo propuesto. Este motor debe enviar los mensajes correspondientes a las situaciones interactivas, ejecutar las llamadas correspondientes a las situaciones automáticas o librerías, evaluar las verificaciones que se han planteado en cada una de las situaciones posibles y evaluar las decisiones que sea posible tomar para cada situación y contexto en la que se encuentre el problema.

Tras la definición del modelo conceptual en un lenguaje ontológico, existen dos alternativas para la definición de un motor de solución:

1. Utilizar un razonador capaz de interpretar la definición realizada, como Jena, Racer o Pellet, y combinarlo con otras tecnologías para el desarrollo de sistemas inteligentes como Jess.
2. Definir una nueva arquitectura para la interpretación del conocimiento modelado.

El lenguaje OWL-DL ha sido empleado como un medio para expresar la definición de los distintos elementos de modelado y sus restricciones de acuerdo a un estándar. Sin embargo tras modelar el proceso es necesario interpretarlo y ejecutarlo de acuerdo con la semántica de cada uno de los elementos, es decir, que en el modelo se especifican los elementos que se pueden utilizar y como se pueden relacionar entre sí para definir un proceso, pero su comportamiento queda fuera del alcance de la definición, por lo que no puede ser automáticamente deducido a partir de la interpretación del mismo con un razonador. Por ejemplo, que la acción externa suponga enviar al exterior el mensaje asociado a la misma, no queda explicitado en el modelo, por lo que es necesario definir detalladamente como se va a interpretar cada uno de los elementos propuestos. Además, la posibilidad de definir interfaces con otras aplicaciones, nuevos operadores, la inclusión dentro de los operandos de las condiciones de valores de hechos o expresiones en lenguajes que puedan ser interpretados para, por ejemplo, realizar cálculos, conlleva la inclusión de cada vez más funcionalidades a un sistema existente que pudiera servir como base.

Por los motivos expuestos, se ha tomado la decisión de definir e implementar un motor de solución propio para el modelo propuesto. De esta forma se ha conseguido la flexibilidad suficiente para poder definir el comportamiento de cada uno de los elementos de acuerdo con la idea original, sin tener que restringir el comportamiento a las posibles limitaciones de los sistemas existentes.

5.2. Algoritmo de Solución: Control y Ejecución de un Proceso

Brachman and Levesque (2004) afirman que la definición más representativa de un sistema de producción es *un sistema de razonamiento de encadenamiento hacia adelante que utiliza reglas denominadas reglas de producción (o simplemente producciones), como representación del conocimiento general*. Los sistemas de producción mantienen una memoria de trabajo, también llamada base de hechos, en la que se van almacenando las distintas aserciones que se producen durante la operación del sistema. Una regla de producción tiene la forma “If condición then acción”.

El motor de solución va a partir de la definición de un sistema de producción, adaptado para tomar las reglas a ejecutar a partir de las situaciones y contextos por los que vaya pasando el problema que se desea resolver.

El proceso de solución va a representar el conjunto de pasos que debe dar el motor de solución para resolver un problema determinado. Las diferentes situaciones que pueden surgir durante el proceso están relacionadas entre sí por medio de decisiones. Este conjunto de situaciones y decisiones puede ser interpretado como un grafo dirigido de situaciones, que lleva desde una situación inicial hasta una situación final, en la que se llega a una conclusión, a través de un conjunto de decisiones.

El motor de solución debe mantener el control del estado en el que se encuentra el proceso en un momento determinado para poder tomar las decisiones oportunas. Mantener el control del proceso de solución implica:

1. Para poder controlar el proceso, el motor de solución debe tener una memoria de trabajo en la que almacene el contexto y la situación actuales, los hechos que se han producido hasta el momento (conjunto H_t) y las decisiones que están pendientes de evaluar, si las hubiera.
2. Evaluar la situación en la que se encuentra el problema en un momento determinado.
3. Decidir la tarea que debe llevarse a cabo para aproximarse a la solución.
4. Ejecutar la tarea por sí mismo, o comunicarse con un *agente secundario* que la realice.

El motor de solución actuará como *agente principal* del proceso, será el encargado de gestionar la conversación con el resto de agentes y guiar el proceso, a partir de la definición del mismo.

En el escenario que se muestra en la figura 5.1, el motor de solución se comunica con los agentes externos por medio de mensajes. Con estos mensajes cede el control del proceso a uno de estos agentes y espera la recepción de un mensaje de respuesta para continuar adelante con el proceso. Cuando el sistema actúa como agente principal, es él mismo el que ejecuta la acción sin ceder el control del proceso.

La ejecución de un proceso se inicia con la llegada de un mensaje de activación del exterior, y debe concluir con un mensaje final de respuesta, en el que se informa acerca del resultado del proceso. En función de las necesidades del proceso, puede ser necesario intercambiar más mensajes con este agente externo para generar los hechos necesarios para alcanzar la solución del problema (situaciones interactivas). En función del rol que asuma el motor de solución con respecto al agente externo que desencadena una sesión, se van a distinguir dos tipos de proceso:

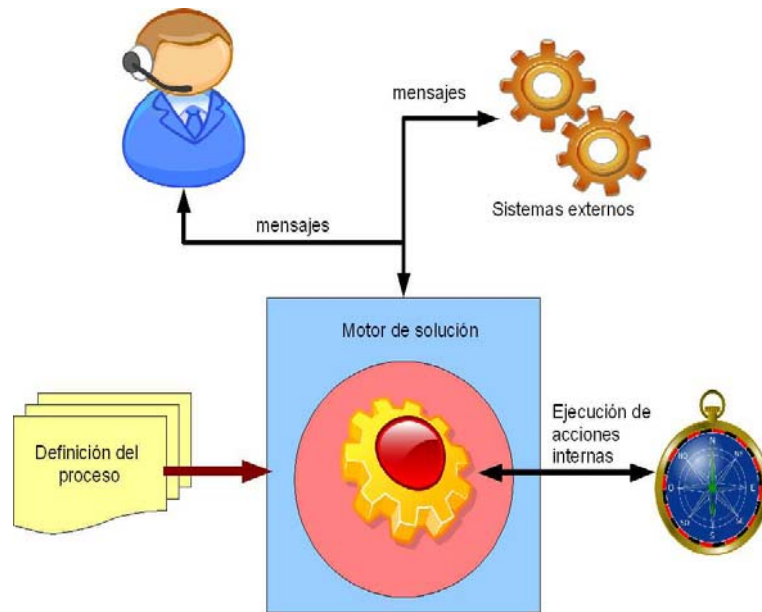


Figura 5.1: Arquitectura de alto nivel.

- Procesos guiados (figura 5.2). En este caso, el encargado de llevar a cabo las acciones es el agente externo que inicia el proceso. El motor de solución se encarga de controlar el proceso indicando cuáles son los pasos que se deben dar a continuación, a través de situaciones interactivas. En estos procesos guiados el motor de solución puede llevar a cabo alguna actividad que facilite el trabajo de las personas que intervienen en el proceso. Ejemplos de este tipo de proceso son los procesos de soporte a la toma de decisiones.
- Procesos automatizados (figura 5.3). En este caso es el motor de solución el encargado de realizar las acciones. Cuando el motor recibe el mensaje de activación del proceso, se desencadena una secuencia de situaciones virtuales que, a través de las acciones que conllevan, van obteniendo información y cambiando las condiciones de contorno hasta que se alcanza una conclusión. El proceso finaliza con un mensaje de respuesta al agente que inició el proceso.

Una sesión iniciada por un agente externo, es el proceso completo seguido desde la activación del proceso hasta la situación en la que se ha llegado a una conclusión acerca del problema que se está tratando. Al comienzo de una sesión, el motor de solución debe establecer el contexto inicial y la situación inicial asociada a dicho contexto, a partir del conocimiento modelado.

El motor de solución puede gestionar varias sesiones simultáneamente, por lo que en la memoria de trabajo deberá tener asociados a cada sesión, el contexto en el que se encuentra, la

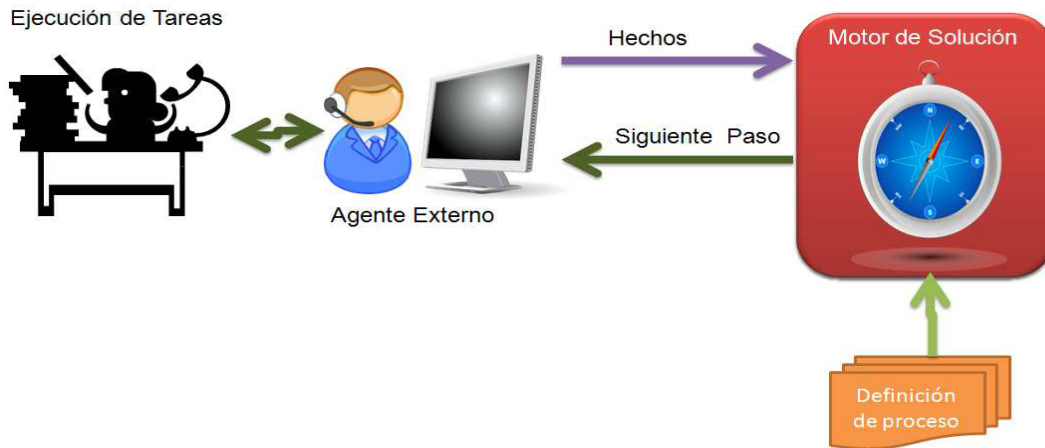


Figura 5.2: Procesos de solución guiados.

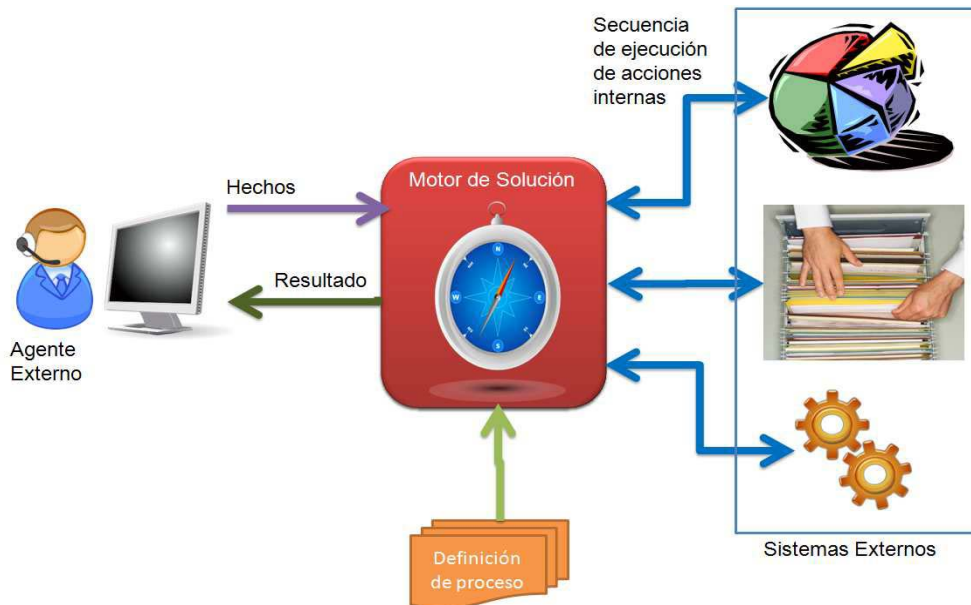


Figura 5.3: Procesos de solución automatizados.

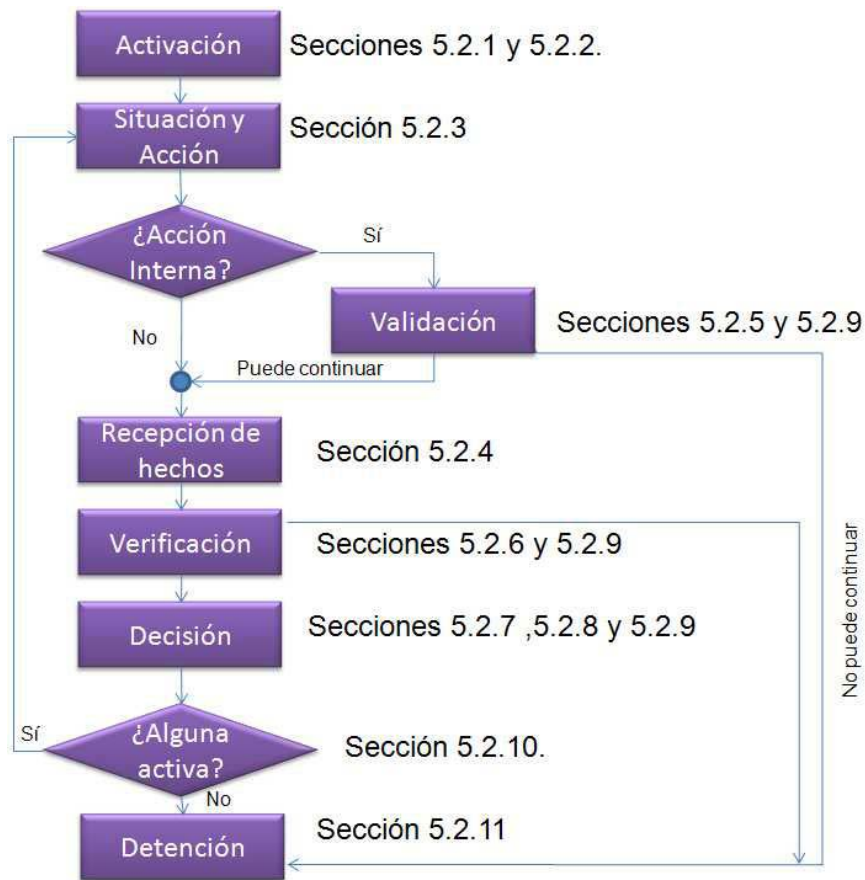


Figura 5.4: Algoritmo de control del motor de solución.

situación que se está produciendo, el conjunto de hechos que tiene en ese momento, y el conjunto de reglas que pueden activarse, en función de la estrategia de resolución que adopte en cada situación.

Cada una de las fases dentro del flujo de control de un proceso de solución conlleva varios pasos. La figura 5.4 muestra de forma esquemática los pasos que se han establecido para interpretar un proceso definido a partir del modelo propuesto. Las siguientes subsecciones describen cada uno de estos pasos llevados a cabo por el motor de solución durante el proceso de solución.

5.2.1. Activación de un proceso de solución

El motor de solución necesita una señal para comenzar a reproducir el proceso de resolución del problema. El proceso se inicia con la llegada de un mensaje de inicio procedente de un agente externo.

Cuando el motor de solución recibe el mensaje inicial, asigna un identificador de sesión al proceso. En todos los intercambios de mensajes con un agente externo a través de situaciones interactivas, el número de sesión será incluido dentro del mensaje. La respuesta del agente externo debe incluir también este identificador para permitir al motor de solución determinar cuál de los procesos activos es el que está manejando en ese instante.

5.2.2. Contexto y situación inicial

Tras la asignación de un código de sesión, el motor de solución busca en la definición del proceso el contexto y situación iniciales. En el modelo conceptual, el contexto inicial está identificado por medio de la propiedad *tieneContextoInicial* de un Proceso.

Para el contexto inicial del proceso, el modelo conceptual define la situación inicial a través de la propiedad *tieneComportamientoInicial*. Esta propiedad identifica el comportamiento que va a estar relacionado con la situación inicial por medio de la propiedad *tieneSituación*.

Una vez localizados, determina la acción asociada a la situación inicial y la ejecuta de la siguiente forma:

- Si la situación inicial es interactiva, el motor de solución incorpora al mensaje el número de sesión y lo envía al agente externo que inició el proceso para que éste realice las tareas necesarias. La acción asociada a la situación está identificada en el modelo conceptual por medio de la propiedad *tieneAccion* del objeto de tipo situación interactiva.
- Si la situación inicial es una librería interactiva, el motor de solución instanciará la interfaz externa y realizará una llamada a la aplicación enviando los parámetros que determine el comportamiento de la situación para el contexto en el que se encuentre. El resultado de esta llamada se enviará al agente externo que inició el proceso, junto con el identificador del proceso.
- Si la situación inicial es una librería convencional en una situación que no es interactiva, el motor de solución instanciará la interfaz externa y realizará una llamada a la aplicación enviando los parámetros que estén incluidos en el comportamiento de la situación para el contexto actual. El resultado de la llamada se incorporará a la memoria de trabajo, asociado al identificador de la sesión en la que se ha realizado la llamada.
- Si la situación inicial es virtual, no se ejecutará ninguna acción y continuará con el siguiente paso del proceso de decisión.

En la sección “Situaciones y acciones” se va a describir con mayor detalle como procesa el motor de solución las acciones asociadas a cada situación.

5.2.3. Situaciones y acciones

En el modelo formal se define una situación como un conjunto de hechos relevantes que determinan el entorno del problema en un momento determinado. A nivel de interpretación de un proceso, el conjunto de hechos que se han producido en las situaciones anteriores se encuentra en la memoria de trabajo del motor de solución, por lo que este conjunto de hechos que determinan la situación está implícito en la sucesión de situaciones por las que se han pasado hasta llegar a la situación actual.

Por otro lado una situación, tal y como se indica en su definición, va a incluir una acción que modifique dicho entorno para avanzar hacia la solución del problema.

Cuando el motor de solución determina la situación en la que se encuentra el problema, recupera de la definición del proceso la acción asociada a la misma y la ejecuta. En función del tipo de acción, el comportamiento del motor de solución va a ser diferente.

En el modelo conceptual se han especificado tres tipos de situación en función de la acción que tengan asociada. A continuación se describe el comportamiento del motor de solución para cada uno de ellos.

Situaciones interactivas

Las situaciones interactivas son aquellas que implican una acción que debe ser realizada por un agente externo distinto del motor de solución. Este tipo de acciones requiere en primer lugar la capacidad de intercambiar mensajes con el agente externo, es decir, que el motor de solución debe poder enviar un mensaje al agente externo y recibir de él una respuesta con los hechos que se han producido fruto de la acción realizada.

El motor de solución debe también intercambiar información con el agente que inició el proceso, en respuesta a uno de los mensajes enviados por dicho agente externo. El caso de que la interacción se produjera con un agente externo distinto del que inició la conversación supone que, en lugar de enviar el mensaje como respuesta a uno de los mensajes recibidos, la iniciativa de la comunicación la toma el motor de solución. Esta iniciativa pasa por abrir una interfaz de comunicación con el agente, enviar el mensaje y esperar respuesta, para lo cuál sería necesario utilizar una acción interna.

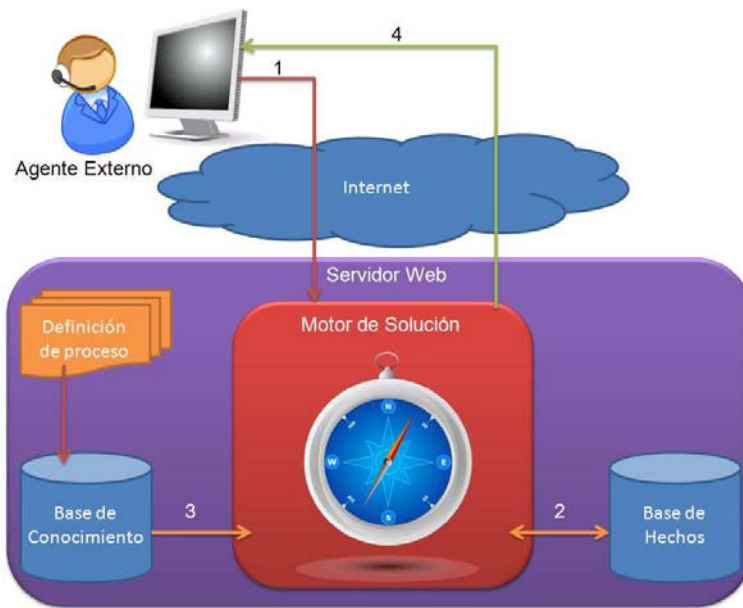


Figura 5.5: Comportamiento implementado para situaciones interactivas.

El procesamiento de una situación interactiva por parte del motor de solución es el siguiente (figura 5.5):

1. El motor de solución recibe un mensaje del agente externo.
2. El motor de solución comprueba en la base de hechos el estado del proceso y almacena los hechos que ha recibido, si procede (ver apartado 5.2.4).
3. El motor de solución recupera de la definición del proceso la acción asociada a la situación y contexto actuales.
4. Si la acción asociada a la situación es una acción externa, el motor de solución envía el mensaje al agente que inició el proceso.
5. Si la acción asociada a la situación es una acción interna, el motor de solución ejecuta la llamada a la aplicación externa (ver apartado 5.2.3), y el resultado de la llamada lo envía al agente externo como si fuera el mensaje de una acción externa.
6. Tras enviar el mensaje al agente externo, el motor de solución espera la llegada de los nuevos hechos que se hayan producido fruto de la acción realizada.

En la implementación concreta del motor de solución se pueden incluir el tratamiento de los mensajes enviados al exterior, como el procesamiento de los mensajes en formato XML.

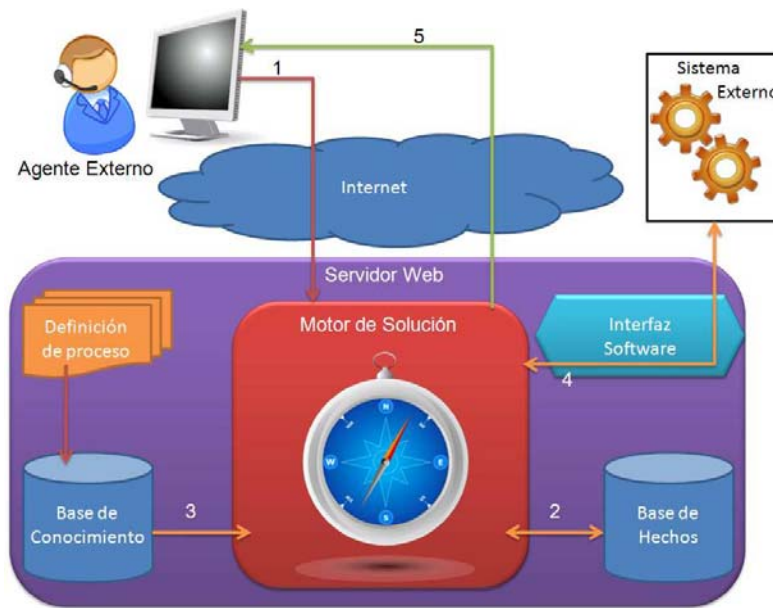


Figura 5.6: Comportamiento de situaciones automáticas.

Situaciones automáticas

Una situación automática es aquella que conlleva una acción que debe ser realizada por el motor de solución. Ejemplos de este tipo de acciones puede ser la consulta a una base de datos, recuperando en forma de hechos información necesaria para el proceso de decisión, o consultas a hojas de cálculo, por ejemplo para realizar cálculos complejos que ya están recogidos en una hoja de cálculo, aprovechando este conocimiento para facilitar el proceso de decisión.

En el comportamiento asociado a la situación automática para el contexto actual, se encuentran definidos los parámetros que se deben utilizar en la llamada a la acción interna. Los pasos que sigue el motor de solución son (ver figura 5.6):

1. El motor de solución recibe un mensaje del agente externo.
2. El motor de solución comprueba en la base de hechos el estado del proceso y almacena los hechos que ha recibido, si procede (ver apartado 5.2.4).
3. El motor de solución recupera de la definición del proceso la acción interna asociada a la situación automática actuales. Recupera también del comportamiento asociado a la situación y contexto actuales los parámetros de llamada correspondientes a la situación. El motor de solución, para cada parámetro, sustituye el nombre de los hechos por su valor

correspondiente. En caso de tratarse de un atributo multivaluado genera tantos parámetros como valores tenga el hecho.

4. El motor de solución instancia la interfaz con la aplicación externa y ejecuta una función de la interfaz, enviando los parámetros que había recuperado del comportamiento asociado a la situación y contexto actuales.
5. Tras esta ejecución, el motor de solución recibirá como resultado un conjunto de parámetros que contendrán los hechos, tal y como se describe en el apartado 5.2.4.

Situaciones virtuales

Son las situaciones en las que no es necesario realizar ninguna acción. En este caso el motor de solución continúa con el algoritmo de resolución como si se hubiera ejecutado correctamente una acción, pasando directamente a la evaluación de las verificaciones.

5.2.4. Recepción de nuevos hechos

Como resultado de la ejecución de una acción de cualquier tipo, se va a producir un cambio en las condiciones de contorno del problema. Este cambio se va a ver materializado en el cambio de los valores para un conjunto de hechos que van a estar asociados a la situación y contexto en el que se ha realizado la acción. En el modelo formal, este cambio se describía de la siguiente forma:

$$a_i(H_t) \rightarrow H_{t+1}$$

El conjunto de hechos H_t es un conjunto definido de forma implícita para cada una de las situaciones por las que va pasando el proceso de solución. Este conjunto es el formado por todos los hechos que se han producido a lo largo de las situaciones anteriores, y se encuentra almacenado en la memoria de trabajo del motor de solución, asociados al número de sesión que identifica la ejecución concreta del proceso. Además, en el conjunto de hechos H_t se incluyen los valores iniciales que se hayan establecido para aquellos hechos que aún no hayan recibido valor por una acción del proceso. En el modelo propuesto, los valores iniciales están asociados a cada hecho a través de la propiedad *tieneValorInicial*.

Estos nuevos hechos han podido producirse de dos formas distintas:

- Por medio de una situación interactiva. El sistema recibe un mensaje del agente externo que contiene, junto al identificador de sesión que permite identificar el proceso que se

está resolviendo, un conjunto de pares (nombre, valor), que representan los hechos que se han generado.

- Por medio de una situación automática. El motor de solución instancia una interfaz con otra aplicación y envía un conjunto de parámetros a la misma. Como resultado recibe de dicha aplicación un conjunto de pares (nombre, valor) que incorporará a la memoria de trabajo asociado al proceso actual.

Una tercera forma de modificar el estado de la memoria de trabajo es utilizando operadores específicos en las condiciones de las decisiones. Esta asignación se produce en un momento posterior del proceso de resolución, que será la “Evaluación de condiciones”.

5.2.5. Validación de librerías

Una validación se define en el modelo conceptual como un conjunto de condiciones que deben cumplirse tras la llamada a una aplicación externa. El objetivo de las validaciones es comprobar que la llamada se ha realizado correctamente y que los resultados obtenidos son válidos para las causas que motivaron la llamada al sistema externo.

El motor de solución realizará las validaciones justo después de recibir la respuesta de la aplicación externa a través de la interfaz software. El motivo de realizar esta comprobación separada de las verificaciones convencionales, es que la acción interna puede ser empleada para crear un mensaje utilizado por una situación interactiva. En este caso, si no se comprueba la validez de la acción inmediatamente, podría suponer el envío de datos incorrectos al agente externo, quedando desvirtuado el resultado del proceso.

En el modelo conceptual, las validaciones se representan por medio de objetos de tipo Verificación. Esto es así debido a que tienen una función similar: la comprobación de unas condiciones que, en caso de cumplirse, implicarán un mensaje de aviso y, adicionalmente, la posibilidad de detener el proceso.

Conceptualmente las validaciones de librería están asociadas al comportamiento de la situación dentro de un contexto determinado. Es lógico, dado que la misma llamada en contextos diferentes podría tener unas condiciones de validez distintas.

El proceso de validación de una librería es el siguiente:

1. El motor de solución recupera el conjunto de validaciones que están relacionadas con el comportamiento de la situación actual para el contexto actual.
2. Para cada validación, el motor de solución evalúa las condiciones que tiene asociada.

3. En caso de cumplirse todas las condiciones de una validación, el motor de solución almacena el mensaje asociado. Si la validación implica la detención del proceso, el motor de solución activa un flag para evitar que el proceso siga adelante, pero continúa evaluando el resto de validaciones.
4. Se repite el proceso para cada una de las validaciones.
5. Si alguna de las validaciones que se hayan cumplido implica no poder continuar, supone la detención del proceso y el envío de un mensaje al agente externo. La detención del proceso de decisión se trata en el apartado 5.2.10.
6. Si alguna validación se ha cumplido, pero ninguna implica la detención del proceso, el motor de solución continuará adelante manteniendo los mensajes asociados a las validaciones que se han cumplido para enviarlo al agente externo en la siguiente situación interactiva.

En caso de poder continuar adelante con el proceso, el siguiente paso a dar por el motor de solución puede ser:

- Si se trata de una situación interactiva que implicaba una acción interna, el motor de solución envía el resultado de la acción al agente externo.
- Si se trata de una situación interna, el motor de solución continúa evaluando las verificaciones asociadas a la situación para el contexto actual.

5.2.6. Verificaciones

Una verificación se define en el modelo conceptual como un conjunto de condiciones que se van a evaluar tras la ejecución de una acción. Más concretamente, estas comprobaciones se realizan en el momento en que se han recibido todos los hechos que haya generado la acción asociada a una situación. A diferencia de las validaciones, las verificaciones necesitan tener todos los hechos disponibles pues, mientras una validación comprueba que la llamada a una aplicación externa se ha realizado correctamente, la verificación comprueba que la información sobre el problema recogida en los hechos es correcta y consistente para poder continuar con el proceso de solución.

Una verificación está representada en el modelo conceptual por un conjunto de condiciones que, en caso de cumplirse, determinan una inconsistencia que implica el envío de un mensaje y la posibilidad de detener el proceso de solución.

El proceso de verificación es el siguiente:

1. El motor de solución recupera el conjunto de verificaciones que están relacionadas con el comportamiento de la situación actual para el contexto actual.
2. Para cada verificación, el motor de solución evalúa las condiciones que tiene asociada.
3. En caso de cumplirse todas las condiciones de una verificación, el motor de solución almacena el mensaje asociado. Si la verificación implica la detención del proceso, el motor de solución activa un flag para evitar que el proceso siga adelante, pero continúa evaluando el resto de verificaciones.
4. Se repite el proceso para cada una de las verificaciones asociadas a la situación para el contexto actual.
5. Si alguna de las verificaciones que se hayan cumplido implica no poder continuar, supone la detención del proceso y el envío de un mensaje al agente externo. La detención del proceso de decisión se trata en el apartado 5.2.10.
6. Si alguna verificación se ha cumplido, pero ninguna implica la detención del proceso, el motor de solución continuará adelante manteniendo el mensaje asociado a las validaciones que se han cumplido para enviarlo al agente externo en la siguiente situación interactiva.

Si las verificaciones no han determinado la detención del proceso de solución, es posible continuar con el siguiente paso del algoritmo. El motor de solución continuará evaluando las decisiones que están asociadas a la situación para el contexto actual.

5.2.7. Decisiones

Tras comprobar por medio de las verificaciones que las condiciones de contorno del problema tienen un valor adecuado para continuar con el proceso de solución, el motor de solución evalúa las reglas de decisión que determinarán cuál es la siguiente situación en la que se encuentra el proceso.

En el modelo conceptual, las decisiones se expresan como un conjunto de condiciones y un destino en caso de que estas condiciones se cumplan. La sección 5.2.9 detalla el proceso de evaluación de las condiciones asociadas a una decisión.

Las decisiones van a representar el conocimiento necesario para determinar cuál es el estado en el que se encuentra el problema y cuál es la siguiente acción que debe realizarse.

5.2.8. Resolución de conflictos y pila de decisiones

Uno de los problemas que deben resolver los sistemas basados en reglas es la resolución de conflictos, es decir, determinar la regla que se ejecuta en el caso de que haya varias que estén en condiciones de hacerlo, esto es, que se cumplan todas sus condiciones.

En el modelo de representación, las reglas se expresan en forma de decisiones. Una decisión se activa cuando las condiciones asociadas a misma se cumplen (ver apartado 5.2.9). Cuando hay varias decisiones activas es necesario determinar cuál es la regla que se debe ejecutar.

Las estrategias clásicas para resolver conflictos son (Brachman and Levesque 2004):

- *Según el orden.* Se ejecuta la primera regla aplicable en el orden en que se han presentado. Este es el tipo de estrategia que utiliza Prolog y es uno de los más comunes.
- *Según la especificidad.* Se selecciona aquella regla cuyas condiciones son más específicas. Se dice que un conjunto de condiciones es más específico que otro, cuando las condiciones que satisfacen al otro son un subconjunto del primero.
- *La más reciente.* Selecciona la regla aplicable basándose en cuándo ha sido utilizada. Existen diferentes versiones de esta estrategia, desde la más recientemente creada o modificada hasta la utilización de la que menos se ha utilizado.
- *No repetición.* No se selecciona una regla que se acaba de aplicar, con los mismos valores para sus variables.

El modelo propuesto va a representar un proceso de solución de problema, en el que cada paso se materializa en una situación con una acción a tomar. Las decisiones van a determinar cuál es el siguiente paso a dar. Cuando se va a especificar el siguiente paso a dar se pueden dar dos posibles estrategias:

- En primer lugar puede suceder que los pasos a dar a continuación sean excluyentes, es decir, que llegada una situación solo sea posible ir a una de las posibles situaciones que siguen a continuación. En este caso una sola regla es la que debe activarse. Por tanto se establece que las reglas se ejecuten en el orden que establece el experto, siendo la regla que se activa la que primero se cumple. En el modelo conceptual esta estrategia se activa asignando el valor *false* a la propiedad *evaluarTodasLasDecisiones* del objeto comportamiento que relaciona una situación con un contexto.

- La segunda estrategia consiste en la posible bifurcación del proceso de solución en varias situaciones a partir de la actual. En este caso es necesario evaluar todas las decisiones asociadas a una situación en un contexto determinado, y activar todas en un orden determinado. Esta opción se refleja en el modelo conceptual a través de la propiedad *evaluarTodasLasDecisiones* con valor *true* para objeto comportamiento que relaciona una situación con un contexto.

Como se ha indicado la estrategia de selección de las reglas activas se establece para cada situación en cada uno de los contextos que puedan aparecer, por medio de la propiedad *evaluarTodasLasDecisiones* del objeto comportamiento. Esto supone que cada situación puede tener una estrategia de activación de reglas diferente. Incluso la misma situación en contextos diferentes podría tener estrategias de activación distintas.

Cuando la estrategia de activación marca que deben activarse todas las decisiones que se cumplan para una determinada situación, el motor de solución debe recordar las decisiones que tiene pendientes para evaluarlas en el momento preciso. Este momento puede ser inmediatamente, o cuando se haya llegado a una situación final siguiendo otras decisiones.

La opción de evaluar las decisiones pendientes antes que el resto de decisiones, se descarta por la semántica que se está representando en el modelo. Dado que se está representando un proceso de solución basado en una secuencia de situaciones que se van sucediendo, permitir que se evalúen primero las decisiones pendientes supone dejar en suspenso las decisiones de la situación que se acaba de evaluar. Como la situación conlleva una acción, esta estrategia implicaría que las consecuencias de la última acción no serían evaluadas inmediatamente, sino que la siguiente acción en el orden lógico determinado por el experto, quedaría en suspenso mientras se van ejecutando el resto de decisiones pendientes. Si se ve el proceso representado como un árbol que se bifurca en función de las decisiones, esta estrategia supondría hacer un recorrido primero en anchura.

La opción adoptada consiste en tratar las decisiones pendientes con una pila de decisiones. Esta opción permite que, cuando la estrategia sea la de ejecutar todas las decisiones que se activan, se introducen en la pila y se selecciona la primera. El proceso de decisión continúa como si solo se hubiera activado esta regla hasta que se llega a una situación en la que no hay más decisiones que tomar. En ese momento el motor de solución toma de la pila la siguiente decisión. Si se ve el proceso representado como un grafo que se bifurca en función de las decisiones de cada situación, esta estrategia supondría hacer una exploración primero en profundidad.

En resumen, la estrategia de activación por defecto es la activar de la primera regla que se cumpla. Cuando se establece que la estrategia a seguir sea la de ejecutar todas las decisiones que

se activen, se almacenan en una pila y se continúa como si se tratara de la estrategia por defecto con la primera decisión de la pila que se cumpla. Cuando se llega una situación en la que no hay más decisiones que tomar, el motor de solución selecciona de la pila la siguiente decisión y continúa el proceso hasta que se llegue a una situación final y no haya más decisiones pendientes.

5.2.9. Evaluación de condiciones

Las decisiones, validaciones y verificaciones se basan en la evaluación de un conjunto de condiciones por parte del motor de solución. A nivel de evaluación, los tres objetos se van a tratar de la misma forma, por lo que en esta sección se hará referencia a los mismos por el nombre de “Reglas”.

En el modelo conceptual, se descompone cada condición en un primer operando, un operador y un segundo operando. Esta descomposición permite representar tantos operadores como se considere necesarios, permitiendo definir nuevos operadores en función de las necesidades del problema que se está resolviendo.

El motor de solución debe resolver las siguientes cuestiones a la hora de evaluar las condiciones.

- Los operandos que se definen en las condiciones deben poder incluir nombres de hechos que sean sustituidos por su valor correspondiente en tiempo de ejecución.
- Un operando puede contener además el nombre de una regla: en caso de cumplirse el nombre de la regla se sustituye por el valor “1”, y en caso contrario por el valor “0”. Esta capacidad de incluir reglas en los operandos permite la definición de expresiones lógicas complejas basadas en los resultados de otras reglas.
- Por defecto todas las condiciones deben cumplirse para que se cumpla la *Regla* que se está evaluando. Sin embargo, es posible definir grupos de condiciones que se tratarán entre sí como un OR. Por tanto, para que se cumpla una regla es suficiente con que se cumplan las condiciones de uno de los grupos OR. Para que un grupo OR se cumpla, todas las condiciones del grupo OR deben cumplirse.
- Los operandos pueden ser expresiones en un lenguaje de programación. Esto va a permitir realizar operaciones matemáticas o lógicas sobre los operandos, antes de ser evaluados por el operador.

- El operador está expresado en forma de un objeto que debe ser instanciado para obtener un resultado.

Cuando llega el momento de evaluar una *regla* el motor de solución sigue los siguientes pasos:

1. El motor de solución recupera de la base de conocimiento el conjunto de condiciones asociado a la decisión.
2. El motor de solución toma la primera condición.
3. El motor de solución analiza el primer operando, sustituyendo los nombres de los hechos por el valor que tiene en la memoria de trabajo para la sesión actual. Si el hecho tiene un valor multivaluado, el motor de solución crea una lista de valores que constituirá el primer operando. Si el operando contiene una referencia a una regla, esta se evalúa y se sustituye en el operando por el valor “1” si se cumple o “0” en caso contrario.
4. Si el primer operando debe ser evaluado con el lenguaje JavaScript, el motor de solución ejecuta una función que devuelve el resultado de evaluar el operando como una expresión JavaScript. En caso de ser multivaluado, evaluará como una expresión JavaScript cada uno de los elementos que componen la lista del operando.
5. Si la condición tiene dos operandos, el motor de solución repite los pasos 2,3 y 4 para el segundo operando.
6. El motor de solución recupera de la base de conocimiento el operador asociado a la condición que está evaluando.
7. El motor de solución instancia el operador, pasándole como parámetros los operandos que previamente ha preparado.
8. El operador devuelve el valor “true” si los operadores cumplen los requisitos del operador, o “false” si los operadores no los cumplen.
9. Si el operador ha devuelto el valor “true”, el motor de solución recupera de la base de conocimiento la siguiente condición asociada a la decisión y lo evalúa volviendo al paso 2 del proceso.
10. Si ya no hay más condiciones, el motor de solución considera que la decisión se ha cumplido si todas las condiciones se han cumplido.

11. Si el operador ha devuelto el valor “false”, la decisión ya no se va a cumplir.

Una característica especial del proceso de evaluación es que se pueden definir operadores que creen o modifiquen valores para hechos en la memoria de trabajo para la sesión activa. Esto supone que durante la evaluación de condiciones asociadas a una regla se puede cambiar el estado del problema, permitiendo al motor de solución actuar sobre el entorno del problema.

En el capítulo 6 se detalla el conjunto de operadores por defecto que se han implementado para la evaluación del modelo.

5.2.10. Detención del proceso

Un proceso de solución puede detenerse por tres causas:

- Se ha cumplido una validación que implica la detención del proceso. Este caso se da cuando se ha realizado una acción interna, y el resultado de la misma ha producido resultados incorrectos o inesperados. En este caso, si la Validación que se ha cumplido tiene definida en su atributo *accionVerificacion* el valor *stop*, el proceso se detiene y no continúa con la evaluación de las decisiones.
- Se ha cumplido una verificación que implica la detención del proceso. Este caso se produce cuando, tras recibir todos los hechos resultado de la acción asociada a una situación, por medio de las verificaciones se comprueba que los valores de los hechos asociados al proceso no tienen un estado consistente o contienen valores incorrectos o incoherentes. En este caso, si el atributo *accionVerificacion* de la Verificación que se ha cumplido tiene el valor *stop*, el proceso se detiene y no continúa con la evaluación de las decisiones.
- En una situación concreta, a pesar de que las acciones se hayan llevado a cabo correctamente y el estado de la memoria de trabajo sea correcto, puede darse el caso de que no haya ninguna decisión que se cumpla o no tenga ninguna decisión asociada.

En cualquiera de los tres supuestos en los que el proceso de solución se detiene, el motor de solución restaura el estado del proceso a la última situación antes de que se produjera la detención.

Dadas las características de las situaciones automáticas y virtuales, en las que el proceso de decisión continúa, la situación en la que se restaura el sistema debe ser una situación interactiva: de esta forma se puede enviar un mensaje identificando que el proceso ha quedado detenido, y se evita entrar en un bucle infinito de intentos fallidos de continuar el proceso. Por tanto la

situación a restaurar es la última situación interactiva que se haya producido antes de producirse la detención del proceso.

5.2.11. Selección de la siguiente situación

Una vez que se han evaluado el conjunto de decisiones que están asociadas al comportamiento de una situación para el contexto actual, el motor de solución debe decidir la siguiente situación de acuerdo con la estrategia que se haya establecido para la situación (ver apartado 5.2.8).

Cuando el motor de solución ha determinado la siguiente decisión que se va a activar, ésta indica cuál es la siguiente situación en la que se encuentra el proceso de decisión. Como se describe en el modelo conceptual, una decisión puede tener tres posibles destinos:

1. Un comportamiento perteneciente al contexto actual del proceso. En este caso el motor de solución establece en el estado del proceso que la nueva situación es la que está relacionada con el comportamiento de destino.
2. Un contexto. En este caso el motor de solución establece como la situación actual, la situación inicial del contexto de destino. Además, el motor de solución actualiza el contexto actual para que sea el contexto destino de la decisión que se ha activado.
3. Una decisión. En este caso el motor de solución debe evaluar la decisión de destino. En caso de no cumplirse asume que la decisión de la que partió tampoco se ha cumplido y continúa con el proceso de selección de la siguiente regla activa. En caso de cumplirse, la nueva situación será el destino de esta última decisión, de acuerdo con las opciones 1 y 2. Es posible encadenar varias decisiones a fin de establecer comprobaciones complejas.

Tras determinar a partir de las decisiones la nueva situación y contexto en que se encuentra el proceso de solución del problema, el motor de solución actualiza en la memoria de trabajo asociada a la sesión actual la situación y contexto, procediendo a la ejecución de la acción asociada a la misma, tal y como se describe en el apartado 5.2.3.

5.3. Resumen: Semántica del Modelo Propuesto

Dado que la descripción de los diferentes contextos, situaciones y comportamientos son realizados desde el punto de vista de un experto en el dominio que se está representando, la primera

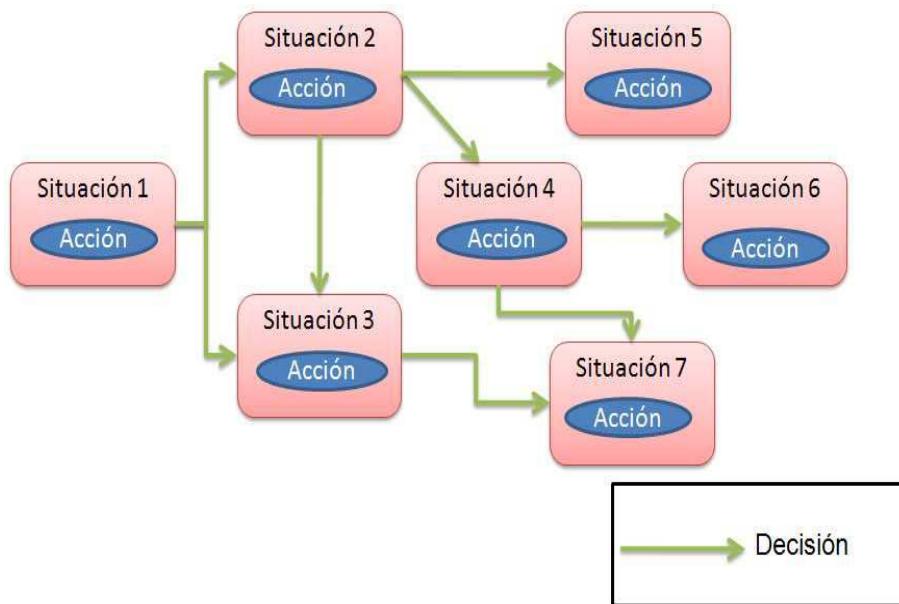


Figura 5.7: Esquema de modelado como red de situaciones simple.

consideración a realizar es que los distintos elementos deben interpretarse desde el punto de vista de una entidad que está llevando a cabo el proceso, en este caso el motor de solución.

A partir de las definiciones de los distintos conceptos que conforman el modelo de representación (ver capítulo 4) y la descripción del algoritmo para la interpretación del mismo, es posible detallar la semántica completa de cada uno de los elementos que componen el modelo. A continuación se resume el significado de todos los conceptos a partir de su interpretación por el algoritmo de solución.

Una visión general de la representación de un proceso de solución se puede observar en la figura 5.7, que muestra el conjunto de situaciones como un grafo en el que cada nodo es una situación que se puede producir, en la que se lleva a cabo una determinada acción. Cada situación está conectada con otras situaciones por medio de un arco, que está representado en el modelo por medio de las decisiones. Las decisiones van a recoger las condiciones que determinan el paso a una u otra situación a partir de una situación dada.

La figura 5.8 muestra cada una de las situaciones dentro del contexto al que pertenecen. De esta forma se está representando una red de contextos, en el que cada contexto agrupa un conjunto de situaciones que tienen una relación lógica entre sí y que pueden producirse dentro de un conjunto de condiciones dadas. Gráficamente también se puede observar como el contexto puede representar un subproblema dentro del problema global cuyo proceso de resolución se está modelando.

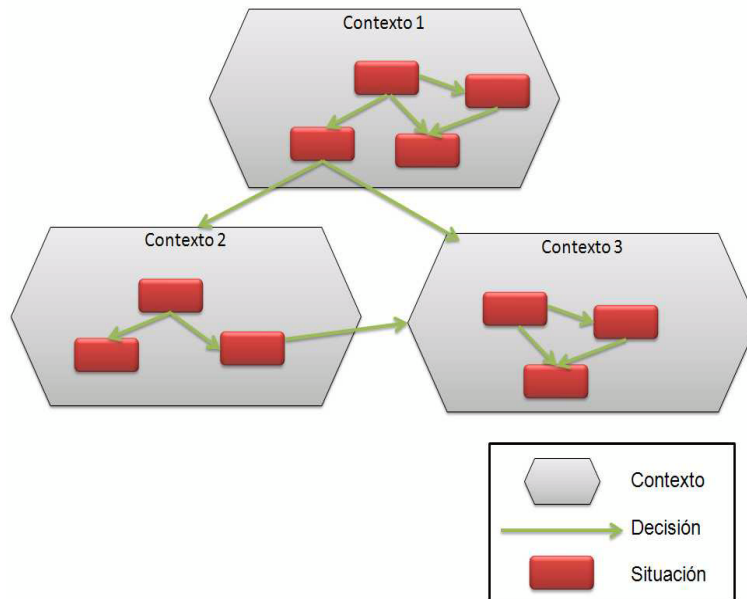


Figura 5.8: Esquema de modelado como red de contextos.

Las situaciones interactivas van a implicar intercambio de información entre el agente externo que inicia el proceso y el agente principal, en este caso el motor de solución. Esta interacción debe traducirse en el intercambio de mensajes, de forma que al mensaje de inicio de un proceso enviado por el agente externo debe suceder al menos un mensaje indicando la conclusión del proceso de resolución lanzado. Sin embargo, como muchos sistemas basados en conocimiento, es necesario que un usuario envíe información adicional que el motor de solución pueda utilizar para tomar decisiones. Las situaciones interactivas servirán para que el agente externo pueda enviar información adicional a petición del sistema.

Por tanto, las situaciones interactivas van a modelar una conversación entre el motor de solución y un agente externo, en la que a cada mensaje recibido por el motor de solución debe suceder una respuesta. Esta respuesta puede ser una conclusión o una petición de más datos.

Sin embargo, esta respuesta no tiene por qué ser inmediata. Tras recibir un mensaje del agente externo, el motor de solución puede realizar una secuencia de acciones internas sin la intervención del agente externo, por medio de las situaciones automáticas, y responder al agente externo cuando haya llegado a una conclusión o cuando necesite más información.

Una situación automática va a estar vinculada a una acción interna que realiza el motor de solución, sin intervención de un agente externo. De la misma forma que el experto utiliza otras aplicaciones, el motor de solución utiliza las acciones internas para representar la llamada a otra aplicación a fin de obtener un resultado que utilizar en el proceso de decisión.

Como se ha expuesto, las situaciones interactivas van a representar la conversación con el agente que ha iniciado el proceso, puesto que si es el motor de solución el que debe tomar la iniciativa de conversar con otros agentes lo haría a través de una interfaz software, que está representado por medio de las situaciones automáticas.

Los hechos van a tener una doble visión. A nivel conceptual, los hechos van a estar relacionados con las situaciones para delimitar aquéllos que van a ser modificados en dicha situación. Por tanto el conjunto de hechos asociado a una situación será el conjunto de hechos que la situación va a modificar en un momento determinado. Esto resulta especialmente útil para, definir las comprobaciones que deben realizarse en cada situación. Pero en tiempo de ejecución, los hechos van a representar información que recibe el motor de solución, bien por medio del intercambio de información con un agente externo, o bien como resultado de la ejecución de una acción interna o de una decisión. El conjunto de hechos que determina la llegada a una situación (formalmente el conjunto H_t) va a estar representado a través la base de hechos que utiliza el motor de solución para evaluar las decisiones a tomar. Efectivamente, en la base de hechos están todos los hechos que se han producido hasta el momento, y que, a través de las decisiones han determinado la llegada a la situación actual.

Tras la ejecución de una acción interna puede ser necesario realizar un conjunto de comprobaciones para determinar si se ha realizado correctamente.

Por otro lado, una vez que se han recibido todos los hechos puede ser necesario realizar comprobaciones adicionales para decidir si se puede continuar adelante con el proceso de solución: son las llamadas verificaciones. El caso particular que determina la separación entre las verificaciones y las validaciones es el de las llamadas a librería que se utilizan para componer un mensaje que enviar a en una situación interactiva: en este caso las validaciones se realizan tras ejecutar la acción interna, y las verificaciones cuando se ha recibido la respuesta del agente externo.

Finalmente las decisiones van a representar las reglas que el experto sigue en una situación determinada para decidir en qué estado se encuentra el problema. En el modelo se representa como un conjunto de condiciones que debe evaluar el motor de solución, y un destino que determinará la siguiente situación del problema, en caso de que las condiciones se cumplan. Este destino puede ser una situación, un contexto nuevo u otra decisión. Cuando el destino es un contexto, la situación en la que se encuentre el problema será la situación inicial asociada a dicho contexto.

En cuanto a la representación de condiciones y operadores, la definición independiente de estos elementos en el modelo de representación va a flexibilizar las evaluaciones que debe realizar el motor de solución.

5.4. Propuesta de Arquitectura para un Motor de Solución

Para poder reproducir un proceso de solución representado a través del modelo propuesto en la presente tesis doctoral, de acuerdo con el algoritmo de decisión descrito, es necesario contar con un motor de solución capaz de interpretar una definición de proceso. Este motor de solución debe implementar la capacidad de intercambiar información con una persona (situaciones interactivas) y con otras aplicaciones (situaciones automáticas o librerías).

Para permitir que la definición de un proceso de solución del problema, representado a través del modelo propuesto, pueda ser automatizada, es necesario considerar los siguientes elementos:

1. En el proceso pueden intervenir uno o varios agentes. Estos agentes pueden ser personas u otros sistemas software que intercambian información entre sí para llevar a cabo el proceso. En ocasiones pueden intervenir tanto personas como sistemas. Por ello es necesario considerar que la automatización de un proceso va a implicar el intercambio de mensajes entre los distintos agentes que intervienen.
2. Además, tal y como se ha descrito anteriormente, es necesario un sistema de control, en este caso el motor de solución, que gestione la ejecución automática del proceso. Debe ser capaz de intercambiar mensajes con el resto de agentes que intervienen en el proceso. Este motor de solución va a condicionar parte de la información que deberá representar cada elemento del modelo.
3. Un medio de transmisión que permita el intercambio de mensajes entre el motor de solución que controla del proceso y los agentes que intervienen en dicho proceso.

La figura 5.9 muestra el esquema básico de la arquitectura propuesta:

1. Una base de conocimiento que contiene la definición del proceso de solución basado en el modelo propuesto.
2. Una base de hechos actúa como memoria de trabajo, almacenando el estado de cada uno de los procesos que está controlando el motor de solución para cada usuario.
3. Un motor de solución capaz de controlar el proceso y tomar decisiones para resolver el problema.
 - Interpretar los procesos de solución definidos. Esto implica identificar las situaciones por las que va pasando el problema mientras se está resolviendo, dar los pasos

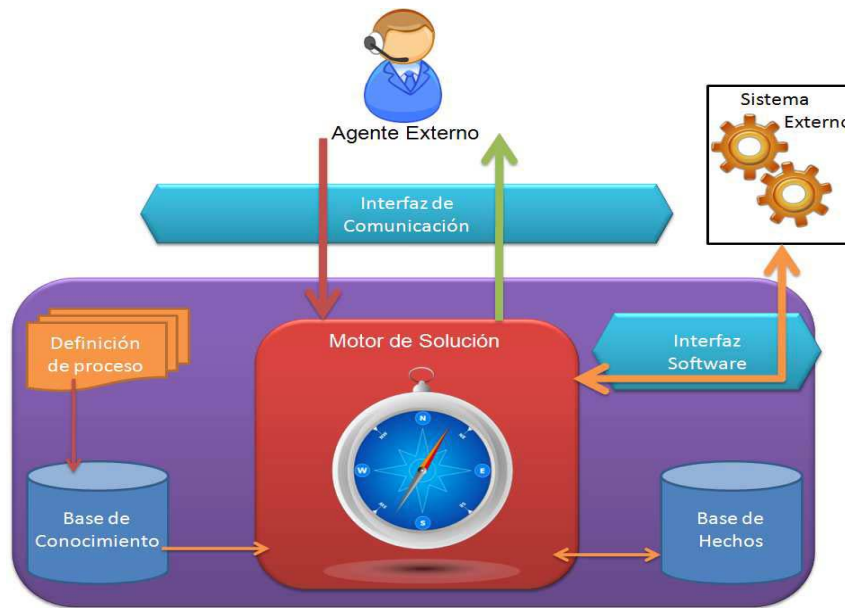


Figura 5.9: Arquitectura propuesta del motor de solución.

necesarios para ejecutar las acciones que van a estar asociadas a cada situación y contexto, y evaluar las distintas decisiones y verificaciones que asociadas a dicho par situación-contexto.

- Controlar varios procesos simultáneamente. El control de varios procesos simultáneamente implica que cada uno de los procesos que se están llevando a cabo, debe ser identificado de forma única. Además la memoria de trabajo donde se almacenan las instancias de los hechos que se van produciendo en un determinado proceso deben quedar asociados al mismo, evitando interferencias con el resto de procesos activos.
- Enviar y recibir mensajes a través de una interfaz que permita intercambiar información con personas. En un proceso en el que deben intervenir personas, el motor de solución debe poder enviar los mensajes necesarios para que la persona pueda llevar a cabo las acciones correspondientes a cada situación. Además, una vez que la persona haya realizado una acción, debe poder responder al motor enviándole el valor de los hechos que se han producido en dicha acción. Por medio de esta capacidad, las acciones interactivas del modelo propuesto quedan implementadas.
- Enviar y recibir datos a través de una interfaz que permita intercambiar información con otras aplicaciones. De la misma forma que un experto puede abrir una hoja de cálculo o consultar una base de datos, el motor de solución debe ser capaz de inter-

cambiar información con otras aplicaciones para poder llevar a cabo.

- Instanciar y ejecutar los operadores necesarios para evaluar las condiciones que se hayan definido para validaciones, verificaciones y decisiones.
- Activar las decisiones de acuerdo a las posibles estrategias descritas.

4. Una interfaz de comunicaciones que permita enviar y recibir mensajes de personas.

5. Una interfaz que permita intercambiar información con otros sistemas

La arquitectura que se plantea es la de un sistema reactivo desde el punto de vista de la activación del proceso, es decir, que es necesaria la intervención exterior, a través de un mensaje inicial, para iniciar un proceso de solución. Esta activación es llevada a cabo por un agente externo y como respuesta recibirá un mensaje asociado a una acción. Sin embargo, desde el punto de vista del proceso de solución, el sistema es interactivo pues se produce un intercambio de mensajes entre los distintos agentes que intervienen en el proceso de resolución y el motor de solución.

A continuación se describen los distintos elementos que componen la arquitectura propuesta.

5.4.1. Motor de Solución

El motor de solución es el componente principal de la arquitectura propuesta. Va a implementar el algoritmo de resolución presentado en el apartado 5.2 y debe estar integrado con el resto de componentes de la arquitectura.

La función principal del motor de solución es la de tomar decisiones a partir de la definición del proceso y de los datos que conoce acerca del mismo. Para ello, además de implementar el algoritmo descrito, debe integrarse con el resto de componentes. Esta integración se define en las siguientes secciones.

Para la validación de la arquitectura, se ha implementado un motor de solución aplicando tecnología ActiveX de Microsoft.

5.4.2. Interfaz de Comunicación

Para la interfaz de comunicación se ha seleccionado la web, es decir, que el motor de solución va a estar integrado en un servidor web, y la comunicación con el agente externo se va a realizar a través del protocolo HTTP.

Las ventajas de esta decisión son las siguientes:

1. No es necesario definir un protocolo de comunicación, sino que la comunicación se realiza sobre un protocolo ampliamente utilizado.
2. Como los mensajes están basados en XML, el envío a través de HTTP va a facilitar la visualización de los mensajes por parte de personas a través de un navegador.
3. Al integrar el motor de solución en un servidor web, se van a poder procesar varias peticiones simultáneamente, con conexiones que va a gestionar el servidor.

El inconveniente principal es que el protocolo HTTP no tiene estado, es decir, que no recuerda información de mensajes que se hayan producido previamente. Por tanto el motor de solución deberá gestionar las distintas sesiones que los agentes externos establezcan para lanzar procesos de solución.

El motor de solución recibirá por tanto los mensajes del agente externo a través de un mensaje HTTP que llegará al servidor Web. Esto va a implicar dos puntos fundamentales:

- Los hechos van a llegar al sistema a través los parámetros del protocolo HTTP, bien a través de envíos con el método GET o bien a través del método POST. En el método GET los parámetros están incluidos en la URL de la petición al servidor, mientras que en el método POST los parámetros viajan en el cuerpo del mensaje HTTP que se envía.
- La interfaz del motor de solución se va a componer de un conjunto de páginas que se ejecutan en el servidor, que van a procesar los mensajes para responder con el resultado que genere dicho motor.

Un mensaje HTTP puede ser generado por un navegador, o bien componerse por medio de una implementación ad-hoc, bien programado directamente o bien empleando las diferentes API que proporcionan los lenguajes de programación como Java o lenguajes de la familia .NET, pero también puede generarse por medio de peticiones de páginas web (a través de hipervínculos o de formularios). Estas dos alternativas van a posibilitar que puedan establecer una comunicación con el sistema tanto personas (a través de navegador) como otros sistemas (programando una interfaz de comunicación). En este último caso, sería necesario que el sistema que establece la comunicación fuera capaz de interpretar los mensajes que recibe del motor de solución.

Como respuesta a los mensajes, el motor de solución enviará el mensaje asociado a la acción en un paquete de respuesta a la petición HTTP que llegó con los parámetros.

Para validar la arquitectura, el sistema ha implementado el intercambio de información con el agente externo a través de un navegador Web. La figura 5.10 muestra el esquema de la propuesta

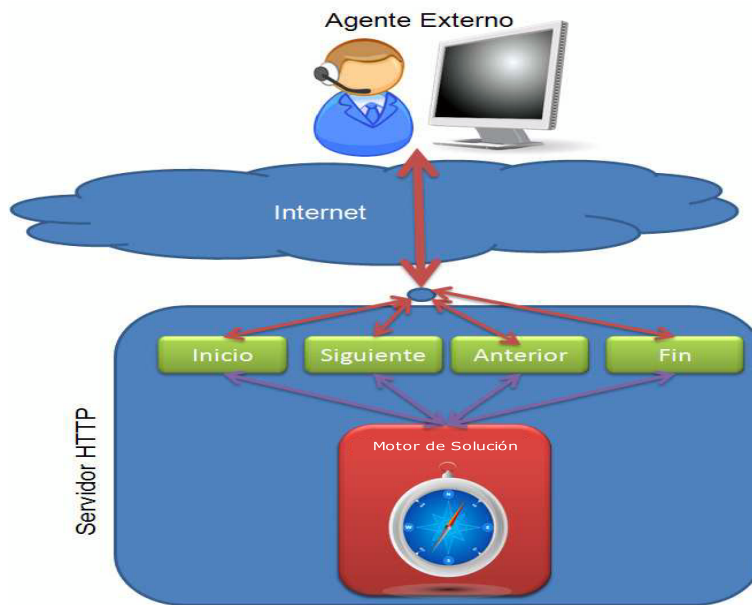


Figura 5.10: Esquema del interfaz de comunicación.

de implementación de la arquitectura para el motor de solución. El motor de solución estará ubicado en un servidor Microsoft Internet Information Server, y las páginas que se utilizarán para establecer la comunicación serán de tecnología ASP.

El motor de solución enviará a través de cada una de estas páginas un mensaje de respuesta a cada una de las peticiones que ha ido recibiendo. Este mensaje debe corresponderse con una situación interactiva.

5.4.3. Interfaz Software

Para que el motor de solución pueda acceder a otras aplicaciones e intercambiar parámetros para realizar acciones internas, es necesario definir una interfaz software que le permita enviar parámetros y recibir resultados. El motor de solución instanciará la interfaz correspondiente según indique la acción interna que está ejecutando en cada momento.

Llegados a este punto, la definición concreta de la interfaz con otras aplicaciones pasa necesariamente por el desarrollo de la misma, pues cada aplicación puede tener un método distinto de comunicación.

Para validar la arquitectura propuesta se ha definido una interfaz ActiveX sobre la que se implementarán distintas clases que permitirán la comunicación. Una clase de interfaz debe tener los siguiente atributos y procedimientos:

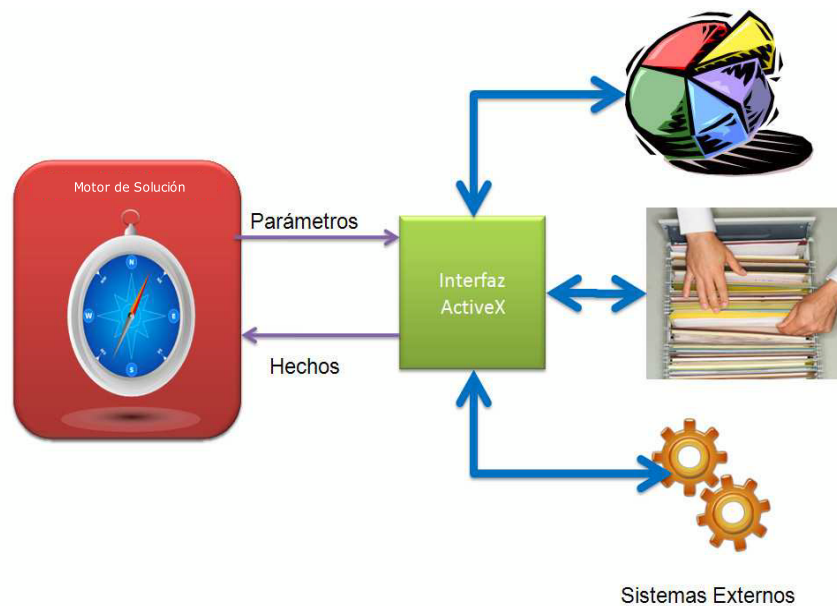


Figura 5.11: Esquema de la interfaz de software.

- Propiedad paramIn: lista de tuplas (nombre,valor), donde nombre y valor son cadenas de caracteres que representan los parámetros que van a ser necesarios para realizar la operación.
- Propiedad paramOut: lista de tuplas (nombre,valor), donde se guardarán los hechos resultantes de la ejecución de la aplicación.
- Procedimiento Execute(): desencadena el proceso que se haya establecido.

Cuando es necesario comunicarse con una aplicación, se definirá un componente ActiveX que siga las reglas establecidas. El motor de solución instanciará como un objeto genérico la interfaz, le enviará los parámetros y ejecutará una función llamada (ver figura 5.11).

La definición de esta interfaz software va a permitir no solo la definición de interfaces con otras aplicaciones, sino la definición de clases que, aunque no impliquen la llamada a otras aplicaciones, realicen un procesamiento de los parámetros que reciban como entrada, y produzcan un resultado de salida. Esto amplía enormemente la gama de acciones que puede llevar a cabo el motor de solución.

Para la validación del modelo en la estimación de precios se han definido dos clases para comunicar con sendas aplicaciones externas:

- Conexión con Excel. Clase que va permitir la comunicación con excel, permitiendo cargar

una hoja de cálculo, introducir valores en las celdas y recuperar el valor de las celdas que se desee. Esto va a permitir al motor de solución acceder al conocimiento que un experto en estimación de precios tiene almacenado en las hojas Excel con las que trabaja.

- **Conexión con Bases de Datos.** Clase que va a permitir la comunicación con bases de datos, permitiendo ejecutar sentencias de recuperación y actualización. Esto va a permitir al motor de solución el conocimiento que el experto en estimación de precios tiene almacenado en las distintas bases de datos con las que trabaja.

5.4.4. Base de Conocimiento

La base de conocimiento almacena la definición de los procesos representados a partir del modelo propuesto. Existen dos alternativas para implementar una base de conocimiento que recoja la definición del modelo propuesto:

1. Utilizar una aplicación basada en ontologías, como Jena, Pellet o Racer, que permita acceder a la definición de una ontología en OWL y realizar operaciones sobre ellas.
2. Utilizar un sistema gestor de bases de datos.

A la hora de decidir cómo se va a almacenar el proceso modelado es necesario también considerar que no solo hay que almacenar los datos referidos a la representación de un proceso de solución de un problema, sino que además va a ser necesario almacenar el estado de cada una de las sesiones que se establezcan con el motor de solución para resolver un problema determinado. El motor de solución debe ser capaz de almacenar la información de cada sesión y recuperarla de forma eficiente para poder tomar una decisión en tiempo de ejecución.

Existen diversos motivos para almacenar una ontología en un sistema gestor de bases de datos relacional (Astrova et al. 2007):

- **Utilización de información anterior.** La utilización de una base de datos relacional permite integrar la información de la ontología con datos anteriores almacenados en otras bases de datos.
- **Aplicaciones antiguas.** Cuando se almacena la ontología en una base de datos relacional, se facilita el acceso a aplicaciones antiguas que funcionan con bases de datos relacionales.
- **Grandes ontologías:** los sistemas gestores de bases de datos relacionales son capaces de manejar grandes cantidades de información, siendo adecuadas para el almacenamiento de ontologías con un gran número de instancias.

A partir de las consideraciones realizadas, se ha decidido representar la ontología que define el modelo en forma de una base de datos relacional. De esta forma se utiliza una única interfaz de datos tanto para la definición del proceso como para el almacenamiento de las distintas instancias del mismo que se generan para cada sesión que se establece con el motor de solución. Además de esta forma se previenen problemas de rendimiento y escalabilidad en caso de que el número de procesos y de instancias se haga elevado (Astrova et al. 2007, Trinkunas and Vasilecas 2007, Motik et al. 2007, Jeong et al. 2006, Vysniauskas and Nemuraite 2006).

La base de datos contendrá definiciones de procesos basadas en el modelo cuyas reglas se han expresado por medio de la ontología propuesta en el capítulo 4. La definición de los distintos elementos que constituyen el proceso de resolución de un problema es introducido en la base de datos a través de una herramienta de modelado, presentada en el capítulo 6.

5.4.5. Base de Hechos

Como se ha descrito en el algoritmo de control del proceso, el motor de solución debe mantener una memoria de trabajo en la que se almacene la situación y contexto en la que se encuentra el proceso de solución en un momento dado, así como el conjunto de hechos que definen el estado del proceso en ese instante. La base de hechos representa la memoria de trabajo del sistema. En esta base de hechos se almacenarán los hechos que se han producido en los distintos procesos que están ejecutándose en un momento dado.

La implementación propuesta integra el motor de solución en un servidor web para permitir la recepción y envío de mensajes con agentes externos a través del protocolo HTTP. Esto va a suponer que distintos agentes van a poder iniciar una sesión con el motor de solución y resolver problemas para los que hay conocimiento representado. Sin embargo, el protocolo HTTP es un protocolo sin estado, es decir, que en el intercambio de mensajes entre el agente externo y el motor de solución no queda almacenado el estado de la comunicación. Como tampoco parece una solución óptima que el motor de solución mantenga en memoria principal el estado de cada una de las sesiones, las soluciones para mantener una memoria de trabajo que pueda emplearse para la solución pasa por dos alternativas:

- Una solución basada en *cookies*, bien a través de variables de sesión o bien a través de los objetos del servidor sobre el que se implementa el motor.
- Una solución basada en una base de datos en la que se almacenen las distintas sesiones y el estado en que se encuentran.

Se ha optado por la solución basada en una base de datos, a fin de mantener unificadas y bajo control del motor de solución tanto la definición del proceso como los datos asociados a las distintas ejecuciones que se realizan del mismo.

5.5. Detalles de Implementación

Se ha implementado un motor de solución a partir de la arquitectura propuesta. Este motor de solución implementa el algoritmo descrito en este capítulo. A continuación se describen los aspectos más relevantes de esta implementación.

5.5.1. Interfaz HTTP y control del proceso

El control del proceso que deber realizar el motor de solución pasa por implementar el algoritmo de decisión descrito en el presente capítulo. El motor de solución implementado se basa en el intercambio a través del protocolo HTTP de mensajes con un agente externo a través del navegador.

En la arquitectura propuesta se ha presentado un motor de solución incorporado en un servidor web para intercambiar información a través de él con el agente externo. El control del proceso requiere las siguientes operaciones:

- Iniciar un proceso. Realiza las operaciones necesarias para iniciar una nueva sesión en la que se resolverá un problema. El motor de solución, además de seguir con el algoritmo propuesto, debe almacenar en la memoria de trabajo (base de hechos) el identificador de sesión y los valores iniciales, antes de responder con la situación inicial.
- Continuar con el proceso. Cuando se ha ejecutado una acción, el proceso debe continuar adelante. Cuando la situación es interactiva, la continuación del proceso depende de la recepción de un mensaje del agente externo. Una vez recibida la información resultante de la acción asociada a la situación y contexto actuales, el motor de solución pasa a la siguiente situación en función de las decisiones definidas.
- Volver al paso anterior. En ocasiones cuando se detecta un error, cuando ha cambiado alguna condición del problema o cuando los resultados no son los deseados, puede ser necesario volver intencionadamente a un paso anterior para actuar de una forma diferente. En este caso, el motor de solución restaura el estado del problema a la última situación

interactiva, actualizando los valores del contexto actual, así como los valores de los hechos que existían en ese momento.

- Dar por terminado el proceso. En un momento determinado el agente externo puede dar por finalizado el proceso. Esto supone el cierre de la sesión del proceso de solución. En este caso el motor de solución muestra un mensaje al agente externo que puede ser personalizado (ver apartado 5.5.2).

Para establecer la comunicación con el exterior, el motor de solución se integra con el servidor Microsoft Internet Information Server. El algoritmo de decisión del motor de solución se ha implementado como parte de una librería ActiveX que puede ser instanciada desde páginas activas en el servidor web.

La ejecución de las funciones asociadas al control del proceso se basa en el intercambio de mensajes via HTTP con el agente externo, y se implementa a través de 5 archivos de tecnología Active Server Pages (asp) de Microsoft:

- *Página inicio.asp*: recibe la petición de inicio de un proceso de solución. Esta página activa el motor de solución y responde a la petición con el envío del mensaje asociado a la situación inicial (ver algoritmo en 5.2.1).
- *Página siguiente.asp*: recibe la respuesta del agente externo a un mensaje anterior. Esta página toma del mensaje HTTP las variables que van constituir los hechos y los pasa al motor de solución para que decida cuál es la siguiente situación (ver algoritmo en 5.2.4).
- *Página anterior.asp*: recibe un mensaje del agente externo para volver atrás en el proceso de solución, regresando a la situación interactiva inmediatamente anterior y restableciendo al estado correspondiente la memoria de trabajo para la sesión.
- *Página inicioSiguiente.asp*: inicia una sesión con el motor de solución, recibiendo directamente los hechos relativos a la situación inicial. Este caso se da cuando se accede al sistema a través de una página web que proporciona la posibilidad de enviar la información antes de iniciar la sesión con el motor de solución.
- *Página fin.asp*: recibe un mensaje del agente externo para finalizar el proceso de solución que se estaba ejecutando.

Como se ha explicado, cada una de estas páginas implementa un paso diferente del algoritmo de solución. Todas ellas instancian una clase que representa el motor de solución, para permitir

el procesamiento de los hechos recibidos y la decisión de la siguiente situación. El motor de solución responderá, a través de cada página, al agente externo con la acción asociada a una situación interactiva, o con un mensaje final en caso de que se haya finalizado la sesión.

5.5.2. Extensiones de los conceptos de la ontología

Tal y como se ha descrito en el modelo de representación (capítulo 4), algunos atributos son propios del tratamiento que se da por parte del motor de solución a los mensajes.

El motor de solución proporciona una librería en javascript que puede ser utilizada para la creación de botones que envíen al motor de solución el contenido de los formularios de una página HTML que actúe como acción externa. A partir de esta librería se pueden crear ficheros en formato HTML que contengan un conjunto de imágenes o botones, de forma que el motor de solución lo pueda incluir automáticamente en todas las acciones externas. De esta forma se facilita la creación de las acciones externas, pues los botones se definen una sola vez, establecen una imagen común para todas las situaciones y son gestionados por el motor de solución.

Por otra parte, el tratamiento que realiza el motor de solución de las acciones y situaciones que se van sucediendo, requiere un conjunto de propiedades que van a facilitar la definición de elementos comunes durante el proceso de decisión. A continuación se describen las extensiones que se han realizado sobre el modelo inicial propuesto.

Proceso

El concepto proceso ha sido extendido en la implementación del motor de solución para soportar la definición de parámetros específicos y la definición de valores por defecto para determinadas propiedades.

- Propiedad: pathBotones. Establece un fichero con los botones que se utilizarán por defecto en las situaciones interactivas. Estos botones envían la información de los formularios a la página asp correspondiente.
 - No tiene una restricción de cardinalidad.
 - Funcional. Solo puede tener un valor asociado.
 - Tipo string. La cadena representa el fichero que contiene los botones predeterminados.

- Propiedad: ttl. Establece el tiempo de espera durante el cuál, en caso de no recibir ningún hecho, el motor de solución considerará abierta la sesión. En caso de superarse este tiempo de espera sin recibir ningún hecho de la sesión, ésta se considera cerrada y se detendría el proceso de decisión.
 - *Cardinalidad* = 1. El proceso debe tener siempre identificado el tiempo de espera.
 - *Functional*. Solo puede tener un valor asociado.
 - *Tipo int*. Número de segundos que se mantendrá una sesión activa sin recibir respuesta.
- Propiedad: pathPredeterminado. Establece el nombre y la ruta que tendrá la carpeta que contiene todos los recursos asociados al proceso que se está representando. Estos recursos pueden ser ficheros que representan las acciones externas, imágenes utilizadas en dichos ficheros, bases de datos, hojas de cálculo, etc.
 - *Cardinalidad* = 1. El proceso debe tener siempre definida una ruta predeterminada.
 - *Functional*. Solo puede tener un valor asociado.
 - *Tipo string*. Ruta de la carpeta predeterminada.
- Propiedad: xslPorDefecto. Establece una hoja de estilo predeterminada para aplicar a los ficheros XML asociados a acciones externas en situaciones interactivas. Si la situación interactiva determina que hay que utilizar el estilo por defecto, el motor de solución aplicará la hoja de estilo al fichero XML antes de procesarlo y enviarlo al agente externo.
 - No tiene restricción de cardinalidad.
 - *Functional*. Solo puede tener un valor asociado.
 - *Tipo string*. Ruta y nombre del fichero XSL que se emplea por defecto.
- Propiedad: cssPorDefecto. Establece la hoja de estilo CSS que por defecto se utilizará en los ficheros HTML. Si la situación interactiva determina que se utilice el estilo por defecto, le motor de solución incluirá los estilos en el fichero HTML antes de procesarlo y enviarlo al agente externo.
 - No hay restricción de cardinalidad.
 - *Functional*. Solo puede tener un valor asociado.
 - *Tipo string*. Ruta y nombre del fichero CSS que se emplea por defecto.

- Propiedad: `accionFinal`. Establece un mensaje que se utilizará cuando el agente externo finalice una sesión. Este mensaje se enviará como notificación de que la sesión se ha cerrado correctamente.
 - No hay restricción de cardinalidad.
 - Functional. Solo puede tener un valor asociado.
 - Tipo string. Ruta y nombre del fichero que contiene la acción externa.
- Propiedad: `mensajeTransiciones`. Cuando el intercambio de información con el agente externo es a través de un navegador web, establece un mensaje que visualiza en el navegador el agente externo mientras el motor de solución toma las decisiones.
 - No hay restricción de cardinalidad.
 - Functional. Solo puede tener un valor asociado.
 - Tipo string. Ruta y nombre del fichero que contiene un mensaje en formato HTML para mostrar mientras el motor de solución toma decisiones.
- Propiedad: `procesarTags`. Corrige las etiquetas en los ficheros asociados a las acciones externas, cuando se han sustituido los valores `>` y `<`.
 - *Cardinalidad* = 1.
 - Functional. Solo puede tener un valor asociado.
 - Tipo boolean. Si el valor es true, en el proceso de las acciones externas, el motor de solución sustituirá los caracteres `>` por `>` y `<` por `<`.
- Propiedad: `usarFrames`. Cuando el intercambio de información con el agente externo es a través de un navegador web, incluye el contenido de las acciones externas en un frame del navegador para ocultar de la barra de navegación las URL de los recursos.
 - *Cardinalidad* = 1
 - Functional. Solo puede tener un valor asociado.
 - Tipo boolean. Si toma el valor true se utilizarán frames en la navegación, mientras que si toma el valor false, no se utilizarán.
- Propiedad: `optimizarPDA`. Deshabilita el javascript en el control interno de las situaciones interactivas para adaptarlas a la navegación en los navegadores de dispositivos móviles.

- *Cardinalidad* = 1.
- *Functional*. Solo puede tener un valor asociado.
- Tipo boolean. Si toma el valor true, deshabilita el control con JavaScript, mientras que si toma el valor false el control con JavaScript está habilitado.

Situaciones interactivas

Tal y como se ha indicado, la integración en el servidor del motor de solución implica el envío de los hechos correspondientes a las acciones externas a través de formularios. El motor permite la utilización de ficheros con la definición de botones que envíen automáticamente la información. Además, en la extensión del concepto de proceso se han incluido valores por defecto tanto para los ficheros de botones como para las hojas de estilo.

Para permitir que las situaciones interactivas representen estos datos, se ha extendido su definición incorporando las siguientes propiedades:

- Propiedad: *pathBotones*. Representa un fichero que contiene botones que se pueden incluir dentro del código XML que se envía al exterior. Estos botones son utilizados por el sistema de control para manejar el intercambio de información con el usuario.
 - No tiene restricción de cardinalidad. Una situación externa puede tener o no un fichero con botones asociados.
 - *Functional*. Como máximo solo puede tener un fichero de botones asociado.
 - Tipo string.
- Propiedad: *usarBotonesPorDefecto*. Indica que en la acción asociada a la situación, se emplearán los botones predeterminados para el proceso.
 - *Cardinalidad* = 1. Todas las situaciones interactivas deben especificar si utilizarán o no los botones por defecto para el proceso.
 - *Functional*. Esta propiedad solo puede tener un y solo un valor.
 - Tipo boolean.
- Propiedad: *usarEstiloPorDefecto*. Indica que en la acción asociada a la situación, se utilizará la hoja de estilo predeterminada para el proceso.
 - *Cardinalidad* = 1. Todas las situaciones interactivas deben especificar si utilizarán o no la hoja de estilo predeterminada para el proceso proceso.

- **Functional.** Esta propiedad solo puede tener un y solo un valor.
- **Tipo boolean.**

Comportamiento

Las situaciones interactivas a través de un navegador web, basan el envío de información en los formularios. En un proceso de decisión, es posible regresar a una situación interactiva anterior para modificar los datos que se introdujeron. Así mismo, por la definición del proceso, puede ser necesario regresar a la misma situación varias veces. En estos casos, es útil que en el formulario correspondiente aparezcan los valores que proporcionó anteriormente el agente externo.

En la implementación del motor de solución, se ha contemplado esta posibilidad, almacenando en la memoria de trabajo los hechos recibidos de cada acción externa, asociados a su correspondiente formulario. Esto, unido al procesamiento de las situaciones que el motor de solución realiza previamente al envío del mensaje al agente externo, va a posibilitar que se restaure la información cuando sea necesario.

Para definir cuándo es necesario restaurar los datos en los formularios, se ha extendido el modelo con dos propiedades nuevas en el comportamiento. Estas opciones se han incluido en el comportamiento interactivo para posibilitar que se pueda restaurar o no los datos de la situación en función del contexto.

- **Propiedad: *resturarAlVolver*.** Cuando la acción asociada es una página Web, los hechos que genera la acción están asociados a un formulario. Esta propiedad indica que si en el proceso de decisión se ha llegado a esta situación como un paso atrás, los valores del formulario se restaurarán con los valores que se introdujeron la última vez que se produjo esta situación para el proceso actual.
 - *Cardinalidad* = 1. El comportamiento interactivo siempre determinará si hay que restaurar o no la información de los formularios cuando se vuelva a la situación.
 - **Functional.** Esta propiedad solo puede tener un y solo un valor.
 - **Tipo boolean.**
- **Propiedad: *restaurarALLegar*.** En ocasiones se puede volver a la misma situación varias veces conforme se avanza en la solución de un problema. Esta propiedad indica que se restaurarán los valores de los formularios que contienen los hechos de la acción externa, a partir de los valores que se introdujeron la última vez que se produjo.

- *Cardinalidad* = 1. El comportamiento interactivo indicará siempre si se debe restaurar el valor al regresar a la misma situación en el mismo contexto.
- *Functional*. Esta propiedad solo puede tener un y solo un valor.
- Tipo boolean.

Condición

El motor de solución será el encargado de evaluar las condiciones que están asociadas a los distintos tipos de reglas que se pueden modelar (validación, verificación o decisión).

En ocasiones es necesario también realizar operaciones sobre cada uno de los operandos, por ejemplo sumas, restas, etc. Por ello se ha recogido la opción de que cada uno de los operandos pueda ser evaluado por separado utilizando javascript.

Para la implementación del motor de solución se han incorporado al modelo las siguientes propiedades para cada condición:

- Propiedad: *evaluarPrimerOperando*. Indica si el primer operando debe ser evaluado con javascript antes de evaluar la condición completa.
 - *Cardinalidad* = 1. Una condición debe indicar siempre si el primer operando debe ser evaluado con javascript.
 - *Functional*. Solo puede tener un valor para esta propiedad.
 - Tipo boolean. Si toma el valor true, el primer operando será evaluado como una expresión javascript antes de evaluar la condición.
- Propiedad: *evaluarSegundoOperando*. Indica si el segundo operando debe ser evaluado con javascript antes de evaluar la condición completa.
 - *Cardinalidad* = 1. Una condición debe indicar siempre si el segundo operando debe ser evaluado con javascript.
 - Axioma:

$$\text{Condicion}(?c1) \wedge \text{tieneOperador}(?c1, ?o1) \wedge \text{esOperadorBinario}(?o1, \text{false}) \rightarrow (\text{evaluarSegundoOperando} = 0)(?c1).$$
 Si el operador de una condición no es binario, no tiene sentido indicar que el segundo operando se evalúe con javascript.
 - *Functional*. Solo puede tener un valor para esta propiedad.

- Tipo boolean. Si toma el valor true, el segundo operando será evaluado como una expresión javascript antes de evaluar la condición.

5.5.3. Tags de control en las acciones externas

Cuando se ejecuta la acción asociada a una situación interactiva, el motor de solución recuperará el código XML asociado a la situación y lo enviará al agente externo. Sin embargo, puede ser necesario incluir información relativa a hechos y reglas que estén representados en el sistema. Por este motivo, el motor de solución realizará un procesamiento de las acciones antes de enviarlas al agente externo. Para que en el procesamiento de las páginas se puedan sustituir hechos y reglas dentro de las acciones será necesario utilizar unas etiquetas especiales:

- `<AVI_EXP>expresion </AVI_EXP>`. Donde:
 - $expresion ::= <@ | \#><“var.” | “reg.”> < identificador >$.
 - $identificador ::= < a..z|A..Z|0..9|_ - > +$.

Cuando el motor de solución encuentra la etiqueta `<AVI_EXP>`, significa que dentro de ella encontrará una expresión que va a implicar el nombre hechos o reglas que deben ser sustituidos por su valor. Por tanto, el motor de solución evaluará el contenido de la etiqueta de la siguiente forma:

- Si la expresión comienza por una almohadilla # , el motor de solución buscará el nombre de todos los hechos (comienzan por "var.") y reglas (comienzan por reg.") contenidos en la etiqueta y los sustituirá por su valor: los hechos los sustituirá por el valor que tengan en la base de hechos, mientras que las reglas las sustituirá por 1 si se cumplen y 0 en caso contrario.
 - Si la expresión comienza por arroba @ , el motor de solución sustituirá también los hechos y las reglas que encuentre dentro de la etiqueta, pero a continuación evaluará el resultado como una expresión en JavaScript.
- `<AVI_INCLUDE_FILE> nombre_fichero </AVI_INCLUDE_FILE>`. Cuando el motor de solución encuentra esta etiqueta dentro de la página, interpretará que el contenido es el nombre de un fichero. El motor de solución incluirá el contenido del fichero indicado en lugar de la etiqueta.

- `<AVI.LOAD.LIB> nombre_libreria</AVI.LOAD.LIB>` Cuando el motor de solución encuentra esta etiqueta dentro de una página ejecuta la acción interna indicada antes de continuar procesando el contenido. El motor de solución asume que la acción interna está asociada al contexto actual de la sesión para recuperar el comportamiento. Tras la ejecución de la llamada interna, se han generado nuevos hechos que pueden ser utilizados en el resto de la página.

5.6. Resumen

En este capítulo se ha descrito el algoritmo para interpretar los procesos representados a partir del modelo propuesto en el capítulo 4. En el detalle de cada paso se especifica la función de cada uno de los elementos de representación y cómo son interpretados, completando la semántica del modelo propuesto. El proceso modelado se puede resumir en una red de contextos y situaciones que encadenan las acciones a realizar, a través de decisiones, hasta que se alcanza una solución. Cada situación en un contexto determinado tiene una determinada estrategia de selección de las decisiones que deben ejecutarse y que el motor de solución debe implementar. En base al algoritmo propuesto se presenta una arquitectura que se ha implementado para evaluar el modelo y su aplicación al dominio de la estimación de presupuestos.

El motor de solución debe tener acceso a la definición de un proceso de solución representada en base al modelo propuesto. Para ello es necesario crear los objetos de conocimiento necesarios y relacionarlos entre sí, de acuerdo con las normas establecidas. El capítulo siguiente describe una herramienta de modelado que se ha implementado para facilitar la adquisición del conocimiento.

En este capítulo se describe una herramienta desarrollada para aplicar el modelo de representación del conocimiento propuesto en el capítulo 4. La herramienta permite representar los distintos elementos aportando una forma de introducir el conocimiento sencilla, abstrayendo de la complejidad de la definición detallada del modelo.

6.1. Descripción General

Para dar soporte a la aplicación del modelo de representación propuesto, es necesario disponer tanto de una herramienta que facilite la representación a partir de los elementos propuestos, como de un motor de solución capaz de interpretar dicha representación. El motor de solución ya ha sido descrito en el capítulo 5.

La herramienta de modelado tiene los siguientes objetivos:

1. Facilitar la creación de elementos de representación del conocimiento, permitiendo definir sus atributos de forma sencilla.
2. Crear automáticamente las relaciones entre los elementos, cuando son definidos.
3. Almacenar las definiciones de procesos de solución en la base de conocimiento utilizada por el motor de solución.
4. Abstracter de conceptos complejos como el comportamiento, que a nivel de usuario está implícito en la relación entre una situación y el contexto al que pertenece.
5. Comprobar automáticamente las restricciones impuestas por el modelo, para evitar inconsistencias y facilitar la resolución de problemas.
6. Visualizar el grafo de decisión conforme se va creando.
7. Probar los distintos elementos como si estuvieran en el entorno de operación real.

Además de las facilidades para representar el conocimiento, abstrayendo al usuario de las restricciones que impone el modelo, destaca la capacidad de la herramienta para comprobar las restricciones expresadas en la definición del modelo.

Las siguientes secciones describen como la herramienta de modelado representa los distintos elementos.

6.2. Organización General de la Herramienta de Modelado

La herramienta de modelado va a permitir la creación de bases de conocimiento basadas en la definición del modelo de representación. Como se ha descrito en la sección 5.4.4 los elementos representados se van a almacenar en una base de datos para facilitar el tratamiento de los elementos de modelado por el motor de solución.

Con la herramienta de modelado se van a poder definir los distintos elementos de conocimiento que representan el proceso de decisión del experto. Conforme se van creando objetos y se relacionan entre sí, estos pueden ser visualizados en un árbol que representa el grafo de decisión con el que se está trabajando.

Junto con el grafo de decisión, la herramienta de modelado se compone de un conjunto de pestañas que van a permitir la creación, modificación y eliminación de elementos de conocimiento. Cada una de las pestañas va a gestionar un elemento de representación del modelo.

Durante el proceso de captura y representación del conocimiento, es interesante comprobar que se están obteniendo los resultados esperados. Una opción de traza va a permitir reproducir el proceso como si se tratara de un sistema en funcionamiento, comunicándose con el motor de solución. La traza va a mostrar información acerca de las decisiones que toma el motor de solución a partir del conocimiento representado, facilitando la identificación y corrección de los posibles problemas y errores que se puedan cometer.

Las siguientes secciones describen en mayor detalle las funcionalidades y ventajas de la herramienta de modelado propuesta

6.3. Proceso de Solución

El concepto más general del modelo de representación es el de “Proceso”. El proceso va a contener el conocimiento relativo al proceso de resolución de un problema (o proceso de decisión) concreto. El conocimiento acerca de un proceso se va a gestionar en una base de datos.

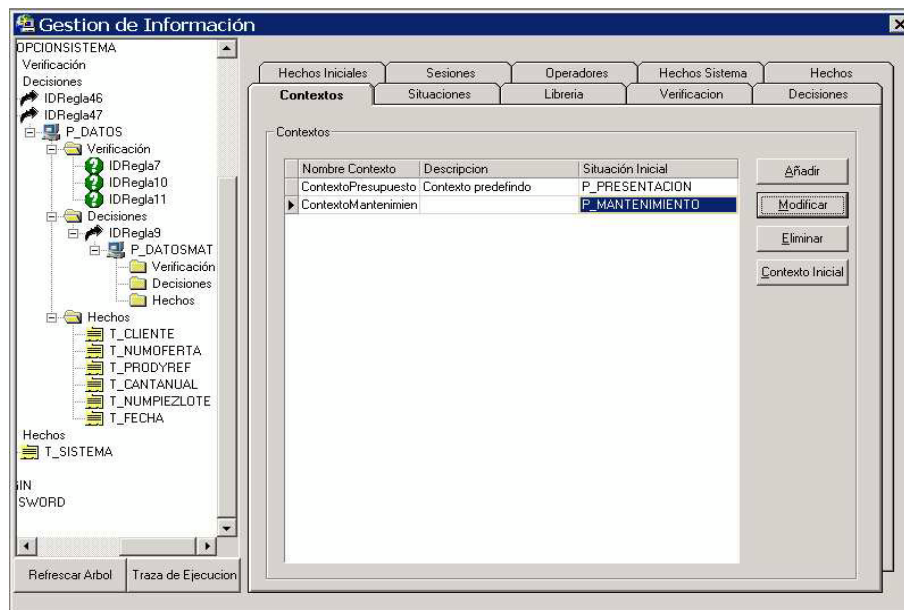


Figura 6.1: Pantalla principal de la herramienta de modelado.

La herramienta de modelado va a permitir la creación de bases de datos basadas en el modelo de representación, cada una de las cuales contendrá la información relativa a un proceso de solución.

Tanto la creación como la carga de un proceso ya existente se realiza a través de sendas opciones del menú principal.

Una vez que se ha creado o cargado una base de conocimiento con la representación de un proceso, se accede a la pantalla principal de la herramienta (figura 6.1). Como se ha descrito, esta pantalla se compone de una vista en forma de árbol que representa el grafo de decisión que forman los elementos del modelo, que se sitúa en el lado izquierdo de la ventana. En el lado derecho un conjunto de pestañas va a permitir introducir la información de cada uno de los elementos que necesarios para representar el conocimiento.

El grafo se representa en forma de árbol para simplificar el tratamiento. Por ello, al hacer clic en cada uno de los elementos del árbol, se va desplegando mostrando las distintas situaciones, contextos, verificaciones y decisiones que se pueden producir. Cuando se selecciona un elemento del árbol, aparecen los datos relativos a dicho elemento en la pestaña correspondiente.

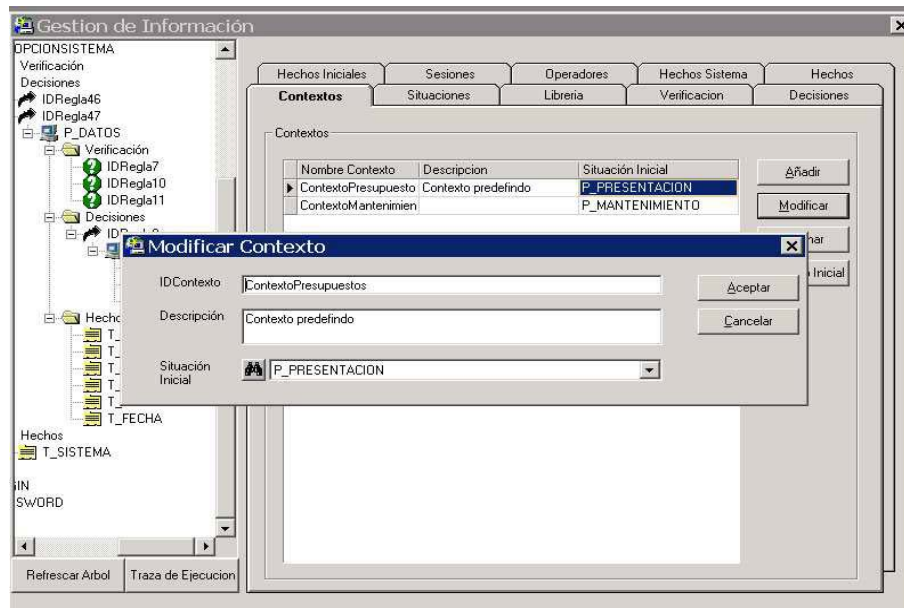


Figura 6.2: Gestión de contextos con la herramienta de modelado.

6.4. Representación de Contextos

A nivel práctico, un contexto va a ser un agrupador de situaciones relacionadas. El concepto de contexto facilitará que las mismas situaciones tengan diferentes comportamientos en función del contexto al que pertenece. La pestaña “Contextos” va a facilitar la creación de los contextos que sean necesarios a partir de un identificador y una breve descripción, permitiendo la creación, modificación y eliminación de los mismos (figura 6.2).

Los datos que solicita la aplicación para la creación de un contexto son:

- Nombre del contexto (sin espacios). Este nombre debe ser único.
- Descripción opcional para la mejor comprensión del contexto.
- Situación inicial del contexto. El usuario debe seleccionar la situación inicial para el contexto. Si la situación no pertenece al contexto, automáticamente queda relacionada con el mismo.

El sistema relaja la restricción de crear la situación inicial antes de crear el contexto, para facilitar la creación en primer lugar de un contexto que luego pueda acoger a la nueva situación. Sin embargo, no permitirá la ejecución del proceso mientras no se cumplan todas las restricciones definidas en la ontología que describe el modelo de representación.

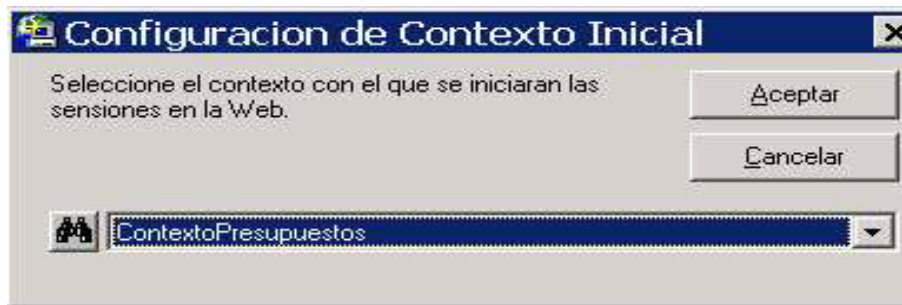


Figura 6.3: Selección del contexto inicial del proceso.

Pulsando el botón “Contexto Inicial” se permite la selección del contexto inicial del proceso que se está representando (figura 6.3):

6.5. Representación de Situaciones y Acciones

La base sobre la que se sustenta el modelo de representación del conocimiento propuesto son las situaciones y las acciones. Las situaciones van a condicionar la acción a llevar a cabo en un momento determinado y, junto al contexto en que se produzcan, determinarán el comportamiento de dicha acción.

En el modelo de representación propuesto existen tres tipos de situaciones: situaciones interactivas, situaciones automáticas y situaciones virtuales. El tipo de acción a tomar en cada una es el elemento diferenciador entre un tipo de situación y otras. Por ello, a nivel de la herramienta de modelado y de cara a la introducción del conocimiento por personas poco experimentadas, la gestión de situaciones y acciones se realiza de forma unificada a través de dos pestañas: situaciones y librerías.

A continuación se describe como crear los distintos tipos de situaciones y acciones a través de la herramienta propuesta.

6.5.1. Situaciones interactivas

Una situación interactiva es aquella que va a suponer una acción que debe realizar el agente externo que está interviniendo en el proceso. Dado que es necesaria la comunicación con dicho agente, y que el medio de transmisión elegido es la web, la acción externa se modela como un mensaje en formato XML. Este mensaje puede estar asociado un fichero (acción externa), u obtenerse a través de una llamada a un agente software (acción interna).

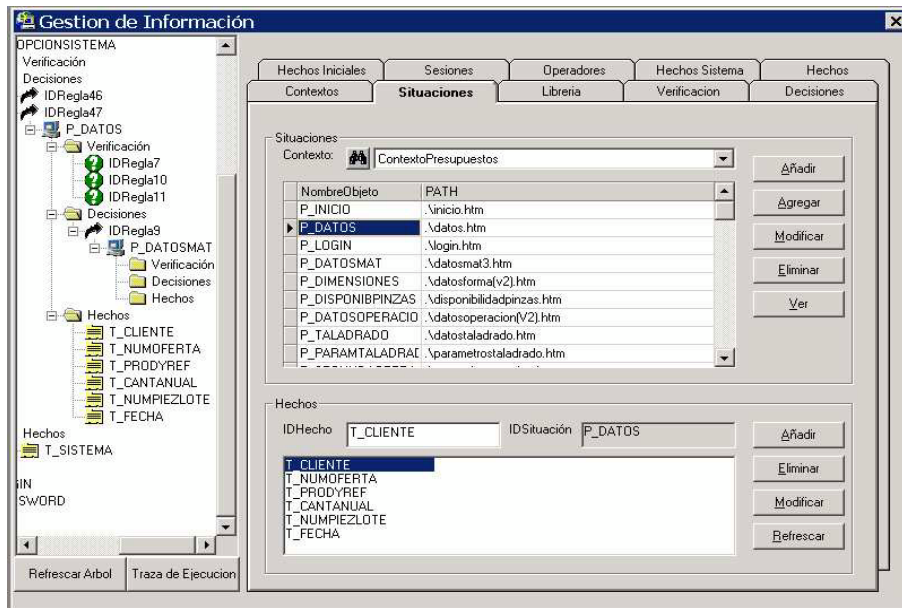


Figura 6.4: Gestión de situaciones interactivas con la herramienta de modelado.

La gestión de situaciones interactivas con acciones externas se tratan a través de la pestaña “Situaciones”. El tratamiento de las situaciones interactivas a partir de acciones internas, se trata en la sección 6.5.2, junto con las situaciones automáticas.

La pestaña “Situaciones” va a mostrar el conjunto de todas las situaciones interactivas basadas en acciones externas que se han definido para el contexto seleccionado (figura 6.4).

Como se aprecia en la figura 6.4, en la parte superior de la pestaña un desplegable permite seleccionar el contexto para el cuál se está creando la situación interactiva. Justo bajo el desplegable de selección de contextos, una rejilla de datos muestra el listado de situaciones, junto con la ruta de la acción externa asociado.

Un conjunto de botones situados a la derecha de la rejilla de situaciones, va a permitir la gestión de las mismas:

- El botón “Añadir” permite la creación de una situación interactiva con una acción externa asociada, definiendo el comportamiento para el contexto seleccionado. La nueva situación queda automáticamente asociada al contexto que estaba seleccionado.
- El botón “Agregar” permite seleccionar una situación ya existente para agregarla al contexto seleccionado. En este caso se crea el comportamiento que debe tener la situación para el contexto seleccionado.

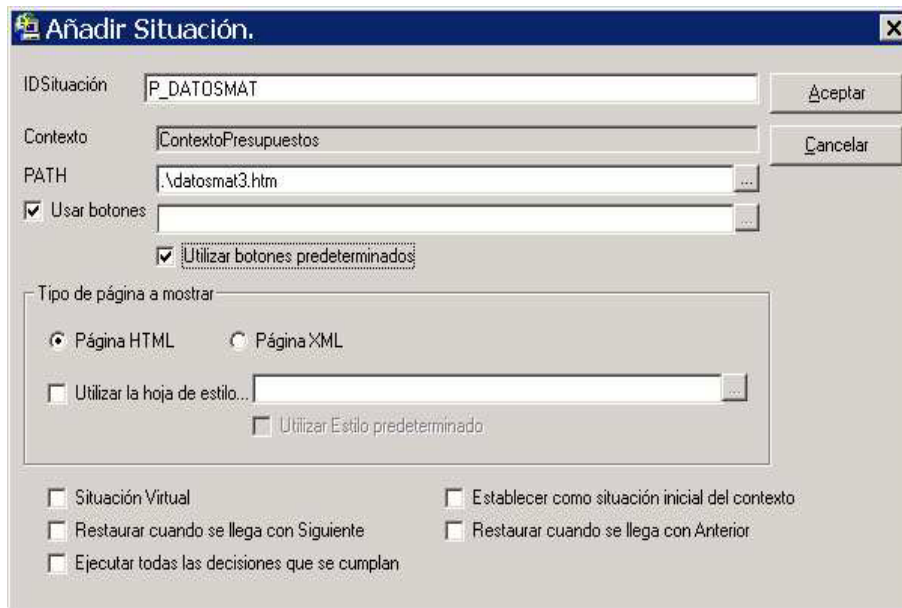


Figura 6.5: Vista de la ventana para crear o modificar una situación interactiva.

- El botón “Modificar” permite la modificación de la situación seleccionada en la rejilla, tanto a nivel de su definición como su comportamiento para el contexto seleccionado.
- El botón “Eliminar” elimina la situación seleccionada del contexto actual, es decir, que se elimina el comportamiento de la situación para el contexto actual. Esta eliminación es a nivel de contexto, pues la situación puede estar asociada a otros contextos definidos en la base de conocimiento. Si la situación no está asociada a más contextos, la herramienta da la opción de eliminarla completamente de la base de conocimiento.

La figura 6.5 muestra la ventana en la que se puede crear o modificar una nueva situación. La ventana se divide en tres partes diferenciadas. En la parte superior se introduce el nombre de la situación, se selecciona el fichero que contiene el mensaje asociado a la acción externa que debe ejecutarse en esta situación y se definen los botones que se utilizarán cuando el agente externo reciba el mensaje. Estos botones son información adicional para el motor de solución. Los botones van a emplearse para enviar la información al motor de solución desde el navegador en que el agente externo visualiza el mensaje. La opción de definir los botones por separado facilita emplear los mismos botones en todas las situaciones definiéndolos en un único fichero.

La parte central de la ventana permite introducir información acerca del tipo de acción que se está definiendo. Es habitual que el intercambio de información con un agente externos se realice a través de páginas HTML. Por eso se permite definir si el mensaje será una página HTML o

será XML puro, y poder asociar en cualquier caso una hoja de estilo. Esto facilitará que todos los mensajes tengan una apariencia similar, utilizando la misma hoja de estilo.

En la parte inferior se puede determinar si la situación es virtual (no tiene una acción asociada): este detalle se explica en la sección 6.5.3. El resto de la parte inferior de la ventana es la correspondiente al comportamiento de la situación para el contexto actual:

1. Establecer como situación inicial del contexto, indica que la situación será la situación inicial para el contexto seleccionado. Esto actualizará la información del contexto.
2. Restaurar cuando se llega con siguiente y restaurar cuando se llega con anterior. Estas opciones indican al motor de solución que, cuando se llegue a una situación interactiva de la que se dispongan de valores previos para los hechos que contiene, debe enviar esta información para que el agente externo la pueda ver en el formulario. Esto facilita enormemente la navegación, pues cuando el agente externo regrese a una situación que ya ha visitado, bien porque se haya establecido un bucle (llega a través de la siguiente situación), o bien porque haya decidido volver atrás (llega a través de anterior), encontrará los datos que introdujo en los formularios. De esta forma solo tendrá que modificar los valores que considere convenientes, en caso de ser necesario.
3. Ejecutar todas las directrices de negocio que se cumplan. Indica al motor de solución la política de resolución que debe emplear para las decisiones de la situación que se está gestionando para el contexto seleccionado. Si esta opción no está marcada, la política consistirá en ejecutar la decisión que primero se cumpla. En caso de estar marcada, el motor de solución ejecutará todas las decisiones que se cumplan para la situación, permitiendo la bifurcación del proceso de decisión en varios caminos.

Si la opción seleccionada es la de agregar una situación ya existente al contexto seleccionado, la herramienta da la opción de seleccionar en un desplegable la situación a agregar (figura 6.6). Una vez seleccionada, solo se pueden modificar las opciones relativas al comportamiento en el contexto seleccionado, es decir, se puede establecer la situación como la inicial del contexto, indicar que se restaure la información de los formularios al llegar con siguiente o con anterior o cambiar la política de resolución de conflictos.

La opción para modificar se realiza en una ventana análoga a la de creación de una nueva situación. Cuando la situación pertenece a varios contextos y se modifica alguna propiedad genérica de la situación o de la acción externa, la herramienta muestra un mensaje de advertencia antes de guardar las modificaciones realizadas.

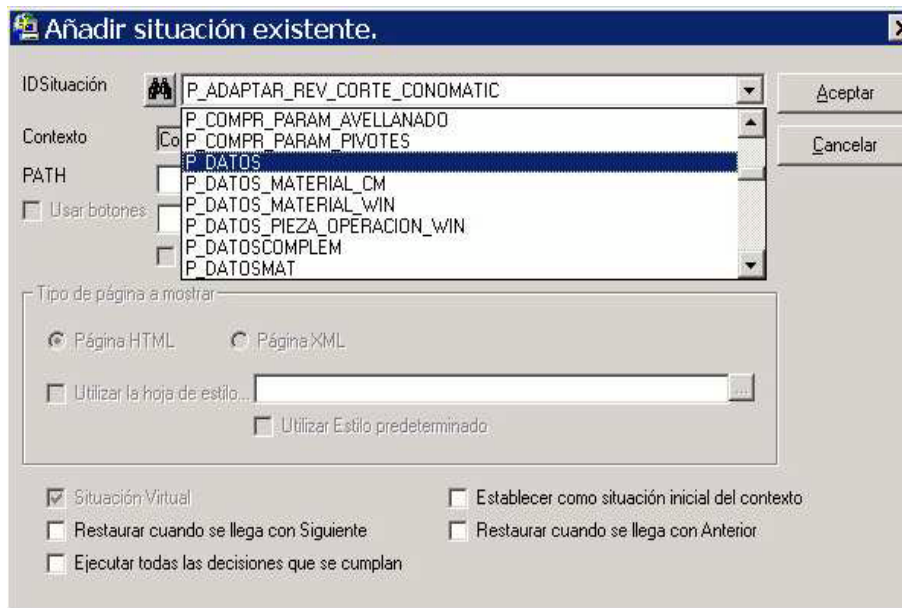


Figura 6.6: Vista de la ventana para añadir una situación interactiva existente a un contexto.

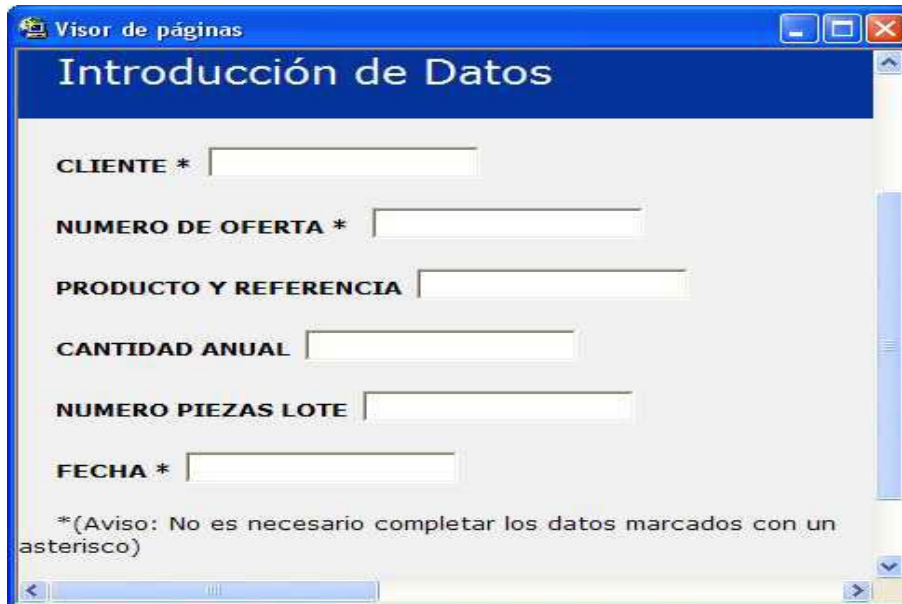
El botón “Ver” dentro de la pestaña “Situaciones” permite visualizar en un navegador el mensaje asociado a la situación seleccionada (figura 6.7).

En la parte inferior de la pestaña “Situaciones” se pueden gestionar los hechos asociados a la situación (figura 6.8). Aquí se dan las opciones para añadir, eliminar o modificar los hechos asociados a una situación. Por último, el botón “Actualizar” buscará en el código de la acción externa asociada a la situación seleccionada (siempre que sea una página HTML) el nombre de todos los hechos contenidos en formularios web, cargándolos en la base de conocimiento asociados a la mencionada situación.

6.5.2. Situaciones automáticas o librerías

Las situaciones automáticas o librerías son aquellas que van a implicar una acción interna (realizada por el motor de solución) en la que no se va a producir intercambio de información con el agente externo del proceso. La pestaña “Librerías” se encarga de la gestión de este tipo de situaciones (figura 6.9). Además, en esta gestión se incluyen las acciones internas que se utilizan en situaciones interactivas que obtienen el mensaje por medio de una acción interna. Esta agrupación se debe a que en ambos casos es necesario gestionar los parámetros de las llamadas internas.

La estructura de la pestaña “Librerías” es similar a la pestaña “Situaciones”. En la parte



Visor de páginas

Introducción de Datos

CLIENTE *

NUMERO DE OFERTA *

PRODUCTO Y REFERENCIA

CANTIDAD ANUAL

NUMERO PIEZAS LOTE

FECHA *

*(Aviso: No es necesario completar los datos marcados con un asterisco)

Figura 6.7: Visor de una acción externa en una situación interactiva.



Hechos

IDHecho IDSituación

- T_CLIENTE
- T_NUMOFERTA
- T_PRODYREF
- T_CANTANUAL
- T_NUMPIEZLOTE
- T_FECHA

Figura 6.8: Gestión de los hechos asociados a una situación.

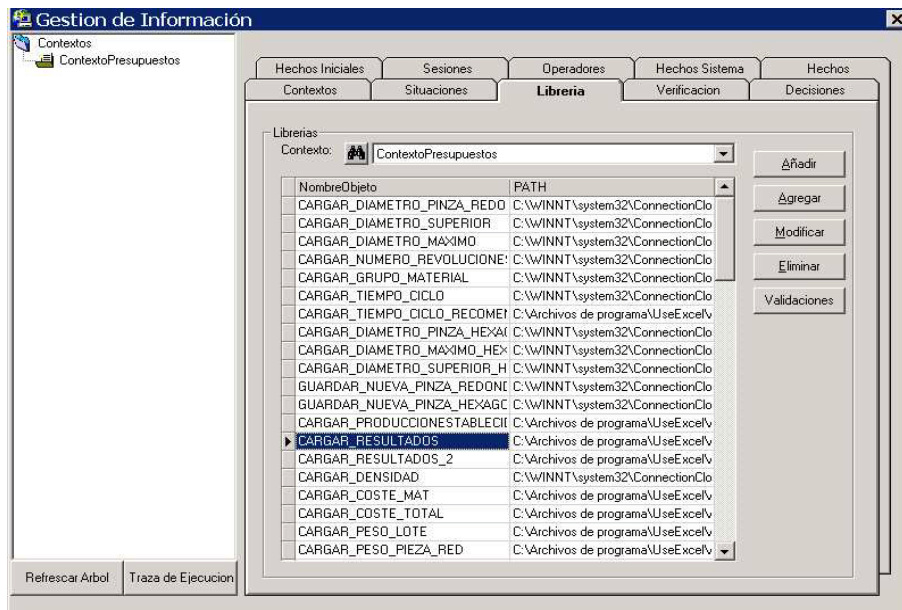


Figura 6.9: Gestión de situaciones automáticas con la herramienta de modelado.

superior un desplegable permite seleccionar el contexto para el que se van a tratar las situaciones automáticas. Bajo el desplegable una rejilla de datos muestra el conjunto de librerías disponibles con los dos datos principales que van a identificar la situación y la acción interna: el nombre de la situación y el fichero que contiene la interfaz para que el motor de solución pueda llamar a otras aplicaciones.

Junto a la rejilla de datos, un conjunto de botones permite añadir, eliminar o modificar librerías del contexto seleccionado.

El botón añadir permite crear una nueva situación automática con una acción interna asociada. La figura 6.10 muestra la ventana que se utiliza para añadir o modificar los datos de una situación automática.

La organización es similar a la ventana que permite la creación de situaciones interactivas (figura 6.5). En la parte superior se introduce el identificador de la situación, el fichero que contiene la interfaz de comunicación y el nombre del objeto que deberá instanciar el motor de solución para ejecutar la acción interna.

En la parte central de la ventana, se encuentra la información relativa a los botones, la hoja de estilo y el comportamiento de la situación. Si la situación automática se marca como virtual, significa que el resultado de la llamada interna no debe ser enviado al agente externo. En este caso se deshabilitan las opciones relativas a los botones y la hoja de estilo. Si la situación automática no se marca como virtual, significa que el resultado de la llamada interna será enviado al agente

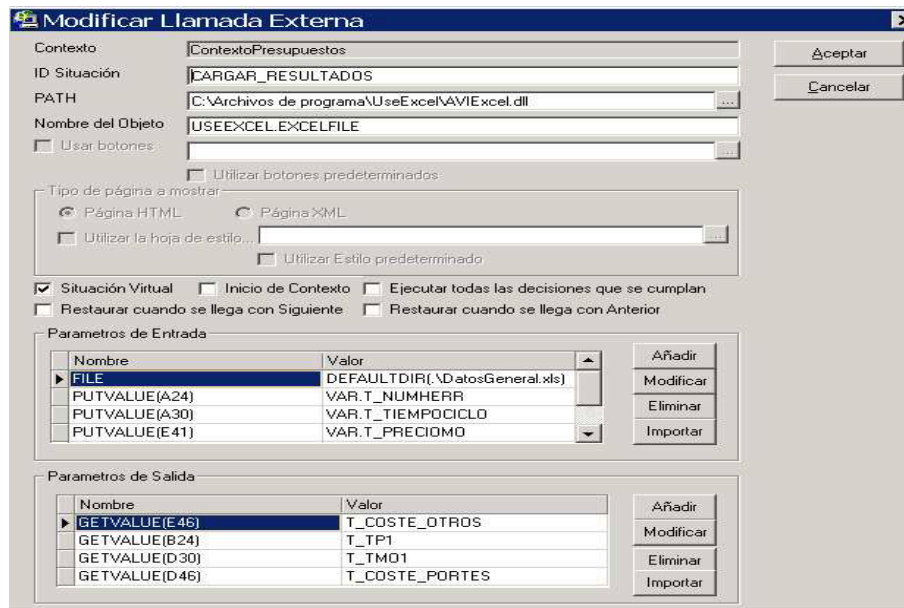


Figura 6.10: Vista de la ventana para crear o modificar una situación automática.

externo como si se hubiera leído del fichero correspondiente a una acción externa. También se pueden seleccionar las opciones correspondientes para seleccionar la política de resolución de conflictos en las decisiones y, en caso de no ser virtual, para restaurar los valores cuando se regrese a la situación).

En la parte inferior se ofrecen dos rejillas de datos para la creación de parámetros de entrada y de salida. Estos parámetros son dependientes del contexto en el que se encuentra la situación automática.

Los parámetros de entrada son aquellos que necesita la acción externa para cumplir con su cometido, mientras que los parámetros de salida son aquellos que va a devolver la librería al motor de solución para que continúe con el proceso. El formato de los parámetros es de tipo <nombre,valor>, donde el contenido de nombre y valor es propio de la librería a la que se está llamando. Junto a las rejillas de datos aparecen los botones para añadir, eliminar o modificar un parámetro, más un botón “Importar” adicional que permite hacer una copia de los parámetros de otra librería. Esta última opción es especialmente útil cuando la librería se utiliza en contextos diferentes pero con parámetros muy parecidos o iguales.

Cuando en la pestaña “Librerías” se pulsa el botón modificar, se carga en una ventana los datos de la situación automática junto con los datos específicos de su comportamiento para el contexto seleccionado, permitiendo la modificación de los valores que sean necesarios.

Al igual que en el caso de las situaciones interactivas, el botón “Agregar” permite seleccionar

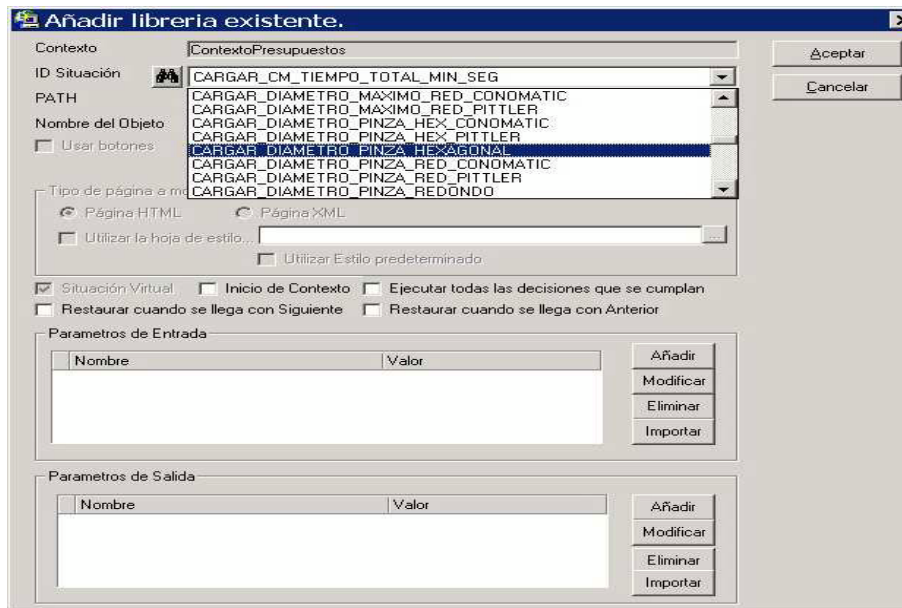


Figura 6.11: Vista de la ventana para añadir una situación automática existente a un contexto.

una situación automática ya existente para vincularlo al contexto seleccionado. En este caso, de la misma forma que se ha explicado para las situaciones interactivas (ver apartado 6.5.1), una vez seleccionada la situación automática, el usuario debe introducir los valores relativos al comportamiento de la situación para el contexto al que se está vinculando (figura 6.11). Estos valores son: la estrategia de resolución de conflictos (opción “Ejecutar todas las decisiones que se cumplan”), establecer como la situación inicial del contexto, restaurar cuando se llega con siguiente o anterior y el conjunto de parámetros de entrada y de salida.

El botón eliminar de la pestaña “Librerías” borra la situación automática seleccionada del contexto actual. En caso de que la librería no se encuentre relacionada con ningún otro contexto, la herramienta da la opción de eliminarla definitivamente.

Finalmente, el botón “Validaciones” dentro de la pestaña “Librerías” muestra la ventana que permitirá gestionar las validaciones correspondientes a la situación automática seleccionada para el contexto actual (figura 6.12).

La gestión de validaciones es similar a la gestión de verificaciones que se detalla en el apartado 6.6. La diferencia se encuentra en la relación que tienen a nivel del modelo con el comportamiento de la situación y el momento en que el motor de solución realiza las comprobaciones. Como se ha explicado en la definición del modelo y, posteriormente, en la descripción del algoritmo para interpretarlo, las validaciones se comprueban justo después de ejecutar la acción interna. De esta forma, si el resultado de la acción debe enviarse al agente externo (librería interactiva),

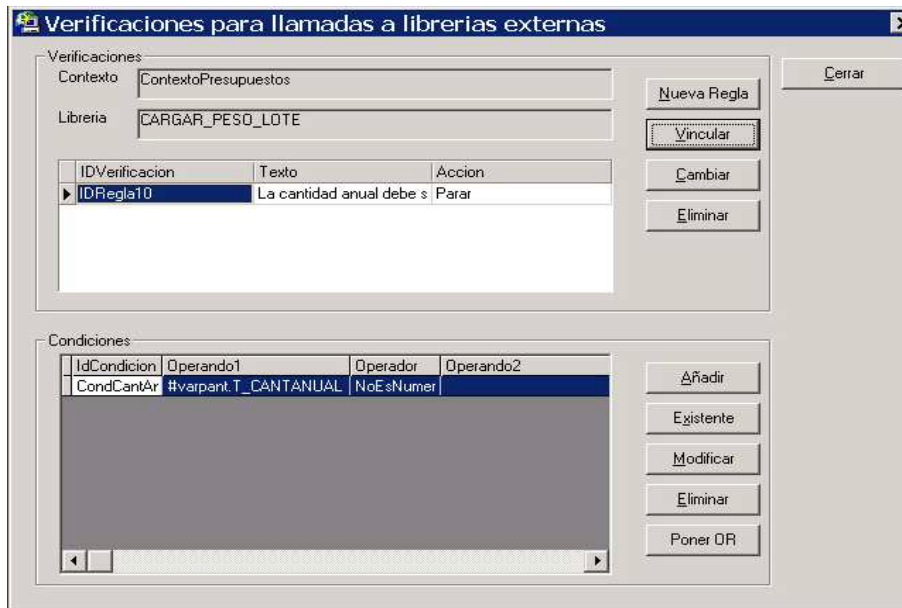


Figura 6.12: Vista de la ventana para gestionar las validaciones de una situación automática.

se ha podido comprobar previamente si el resultado de la acción era correcto.

6.5.3. Situaciones virtuales

Una situación virtual es aquella que no tiene asociada ninguna acción. El sentido de esta situación es el de agrupar verificaciones y decisiones que, por claridad en la definición del proceso, no interesa incluir en otras situaciones. También representa la situación en que un experto no tiene una acción concreta que llevar a cabo, más que razonar acerca de los datos que tiene hasta el momento para resolver el problema.

Para crear una situación virtual, desde la pestaña “Situaciones” se crea una nueva situación, y en la ventana de creación hay que seleccionar la opción “Situación virtual” (ver figura 6.5). Cuando se selecciona esta opción se deshabilitan las opciones relativas a la acción asociada a la situación (nombre del fichero que contiene el mensaje, botones y hoja de estilo), y del comportamiento de las situaciones interactivas (restaurar cuando se llega con anterior o siguiente).

Como se ha explicado en el apartado 6.5.2, cuando al crear una situación automática, se selecciona la opción “Situación virtual” (figura 6.10), significa que el resultado de la acción interna asociada a la situación no se enviará al agente externo.

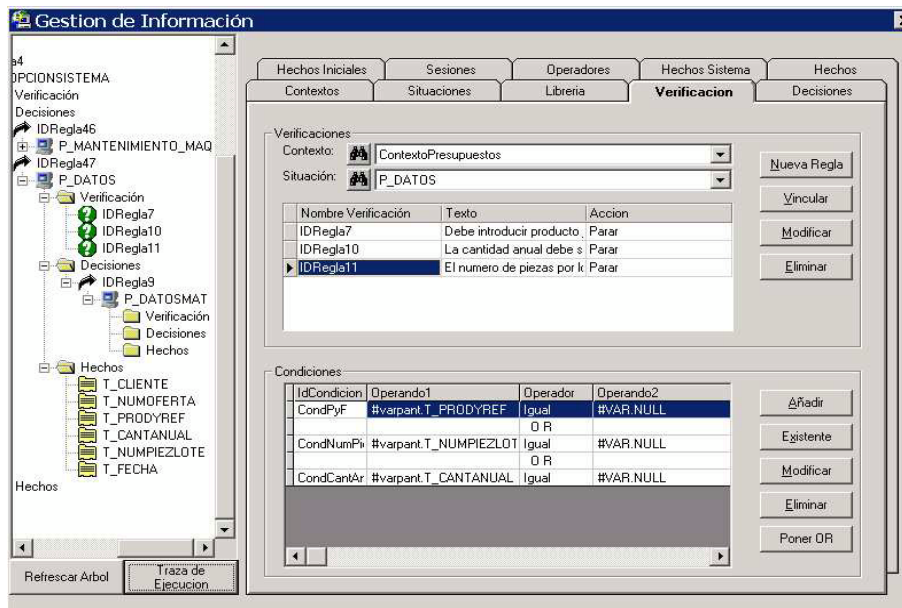


Figura 6.13: Pestaña para gestionar las verificaciones de una situación.

6.6. Representación de Verificaciones

Las verificaciones, como se describe en el modelo conceptual, se van a encargar de comprobar antes de evaluar las decisiones, que los hechos tienen un valor adecuado para continuar adelante con el proceso de decisión.

La pestaña “Verificaciones” (figura 6.13) se compone de dos desplegables donde se puede seleccionar el contexto y la situación para la que se quieren definir las verificaciones (al seleccionar el contexto y la situación queda determinado el comportamiento). Como se explica en el capítulo 4, las verificaciones de una situación dependen del contexto en el que se encuentre, por ello en primer lugar es necesario seleccionar la combinación de ambos.

Una vez seleccionados el contexto y la situación para la que se van a definir las verificaciones, en la pestaña “Verificaciones” se distinguen dos partes claramente diferenciadas. En la parte superior una rejilla de datos se utiliza para gestionar la información general acerca de la verificación: nombre de la verificación, mensaje a mostrar y acción a tomar. En la parte inferior se encuentra la zona de gestión de condiciones. Cuando se selecciona una verificación en la rejilla de verificaciones, en la parte inferior se cargan las condiciones asociadas a la misma para la situación y contexto seleccionados. La gestión de condiciones se trata en la sección 6.8.

Junto a la rejilla de las verificaciones se encuentran cuatro botones para añadir, agregar una verificación ya existente, modificar o eliminar una verificación.

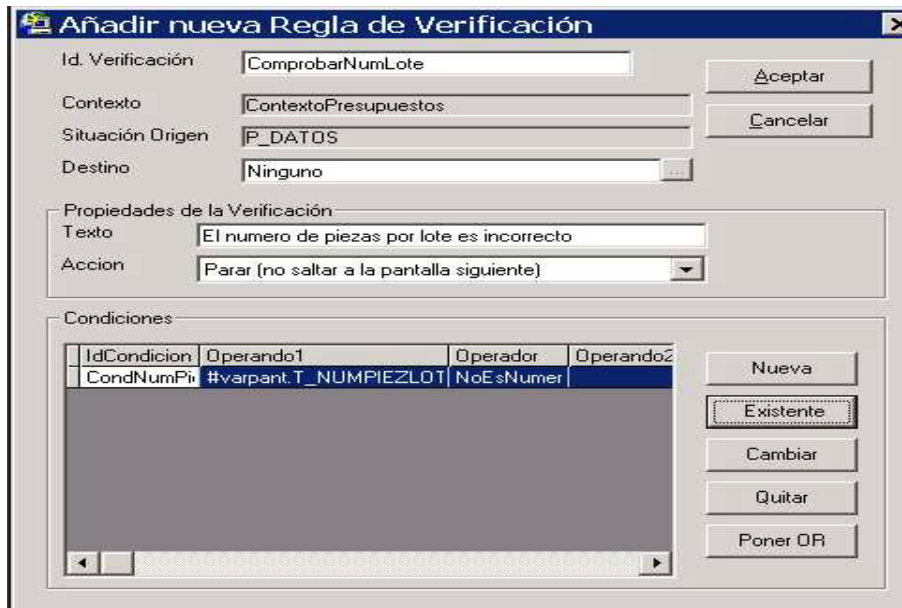


Figura 6.14: Ventana para crear una nueva verificación.

El botón añadir muestra la ventana para la creación de una nueva verificación (figura 6.14).

Para crear una verificación se introduce un nombre identificador único de la verificación, para facilitar su utilización en otros comportamientos. A continuación se muestra la situación y contexto para el que se está creando la nueva verificación. La opción “Destino” es propia de las decisiones, por lo que para las verificaciones esta opción está deshabilitada. En la zona central de la ventana de creación, se encuentran las propiedades específicas de la verificación: el mensaje que debe mostrarse en caso de que las condiciones se cumplan, y la acción a tomar (parar o continuar mostrando el mensaje). En la parte inferior de la ventana aparece una zona de gestión de condiciones, que va a permitir la creación de las condiciones asociadas a la verificación. La gestión de condiciones se explica en la sección 6.8. Una vez que se han asignado los valores para las propiedades de la verificación y se han creado las condiciones que deben cumplirse, se pulsa el botón aceptar para crear la verificación en la base de conocimiento, quedando asociada al comportamiento que se había seleccionado previamente.

El botón modificar muestra la ventana para la creación de una nueva verificación (figura 6.15). La ventana para modificar una verificación contiene los datos relativos a las propiedades específicas de la verificación, pero no incluye la gestión de condiciones. Esto se debe a que la gestión de las condiciones de la verificación se pueden hacer desde la pestaña (figura 6.13).

El botón agregar, permite seleccionar una verificación que ya existía previamente para vincularla al comportamiento determinado por la situación y contexto actuales (figura 6.16). De esta

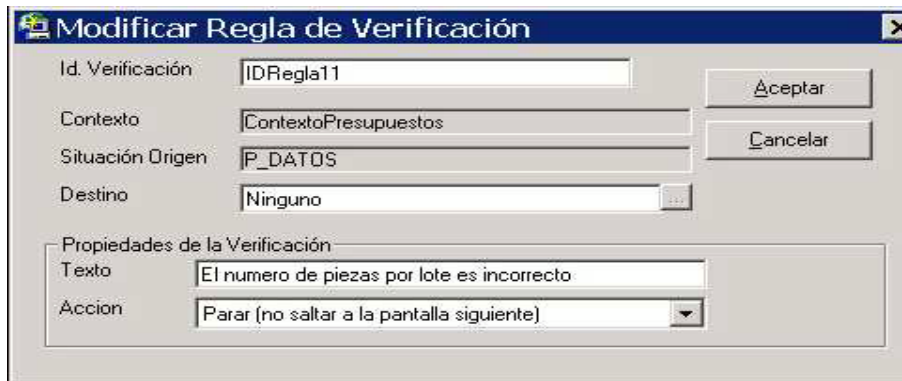
Una ventana de diálogo con el título "Modificar Regla de Verificación". Contiene los siguientes campos: "Id. Verificación" con el valor "IDRegla11"; "Contexto" con el valor "ContextoPresupuestos"; "Situación Origen" con el valor "P_DATOS"; "Destino" con el valor "Ninguno"; "Propiedades de la Verificación" que incluye un campo "Texto" con el valor "El numero de piezas por lote es incorrecto" y un campo "Acción" con el valor "Parar (no saltar a la pantalla siguiente)". Hay botones "Aceptar" y "Cancelar" a la derecha.

Figura 6.15: Ventana para modificar una verificación.

forma, es posible utilizar una misma verificación en diferentes situaciones.

El botón eliminar, borra la verificación del comportamiento asociado a la situación y contexto actuales. En caso de que la verificación no esté siendo utilizada en otra situación, la herramienta da la opción de eliminarla definitivamente de la base de conocimiento.

6.7. Representación de Decisiones

Las decisiones representan el conocimiento del experto para decidir cuál es la siguiente situación en la que se encuentra el problema, tras haber llevado a cabo una determinada acción.

Una decisión va a estar siempre relacionada con el comportamiento de una situación en un contexto determinado. En la pestaña “Decisiones” dos listas desplegables permiten seleccionar el contexto y la situación para la que se van a gestionar las decisiones (ver 6.17).

Además de los desplegables, la pestaña para gestionar las decisiones se divide en dos partes. En la zona superior se encuentra una rejilla de datos que muestra la información relativa a las decisiones que pertenecen al comportamiento de la situación elegida en el contexto seleccionado. La zona inferior gestiona las condiciones asociadas a la decisión, tal y como se explica en la sección 6.8.

Para la gestión de decisiones, en la parte superior de la pestaña aparecen cuatro botones que permiten añadir, modificar o eliminar decisiones para el comportamiento asociado a la situación y contexto seleccionados.

El botón “Añadir” muestra la ventana para la creación de una nueva decisión en la situación y contexto seleccionados (figura 6.18).

En la parte superior de la ventana, el usuario debe introducir un nombre identificador único

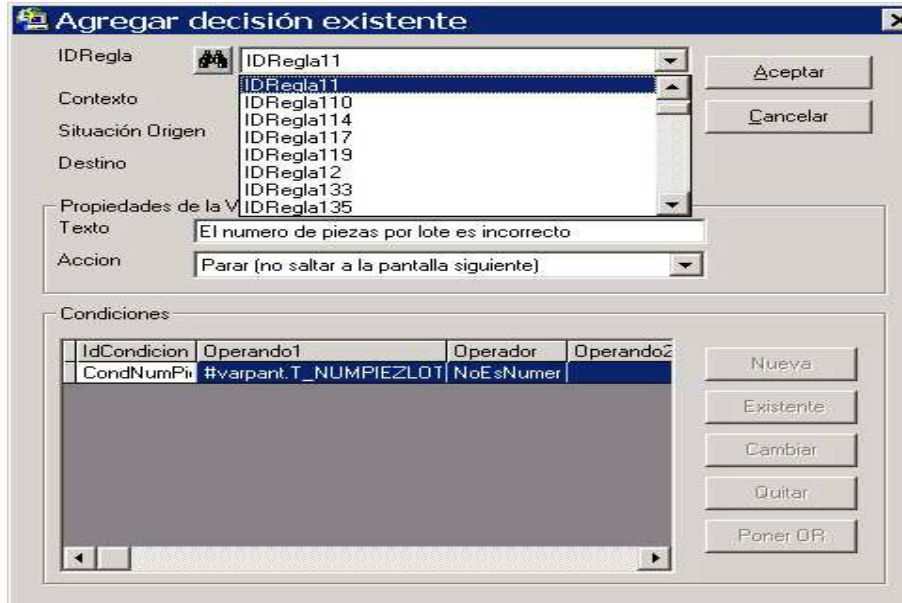


Figura 6.16: Ventana para añadir una verificación existente.

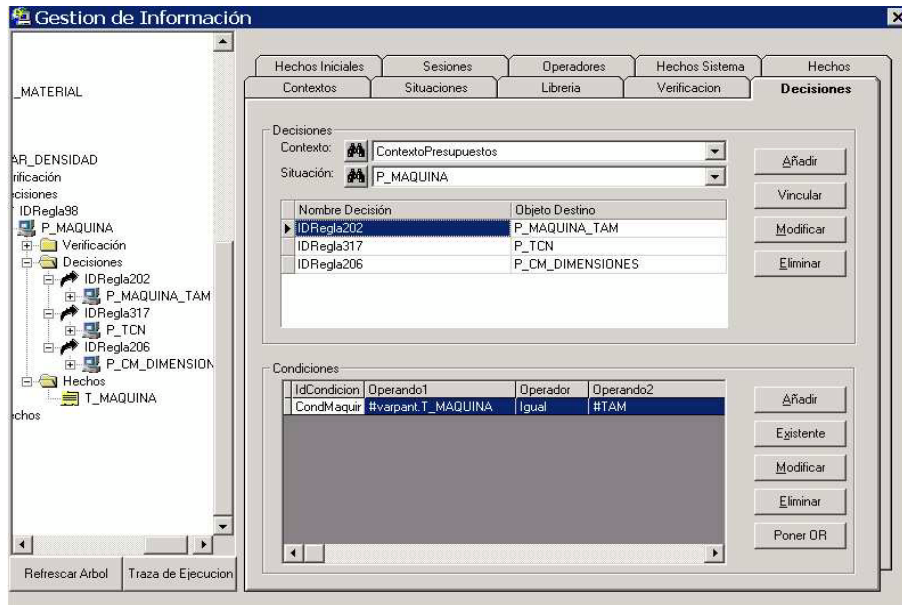


Figura 6.17: Vista de la pestaña para gestionar las decisiones para una situación.

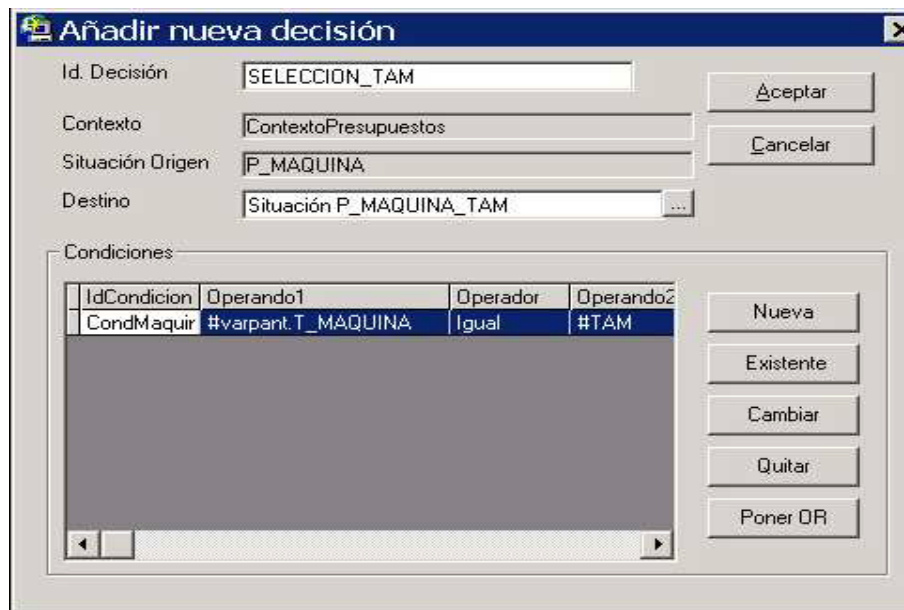


Figura 6.18: Ventana para crear una nueva decisión para una situación.

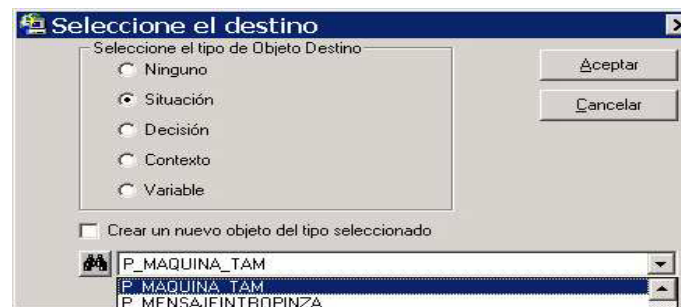


Figura 6.19: Ventana para seleccionar el destino de una decisión.

para la decisión. Bajo este nombre se muestran la situación y contexto para la que se está creando la decisión. Seguidamente el usuario debe seleccionar el destino de la decisión en caso de cumplirse. Esta selección se realiza a través de la ventana que aparece al pulsar el botón “...” (figura 6.19).

El destino de una decisión debe ser una situación. Como se explica en el capítulo relativo al motor de solución, este destino puede ser una situación del contexto actual, un contexto (el destino sería la situación inicial del contexto) u otra decisión (el destino sería el destino de la decisión en caso de que se cumplieran sus condiciones). La herramienta de modelado, a partir de la implementación realizada del motor de solución, permite que el destino sea el nombre de un hecho. El hecho debe contener el nombre de la situación de destino. Esto permite definir *destinos*

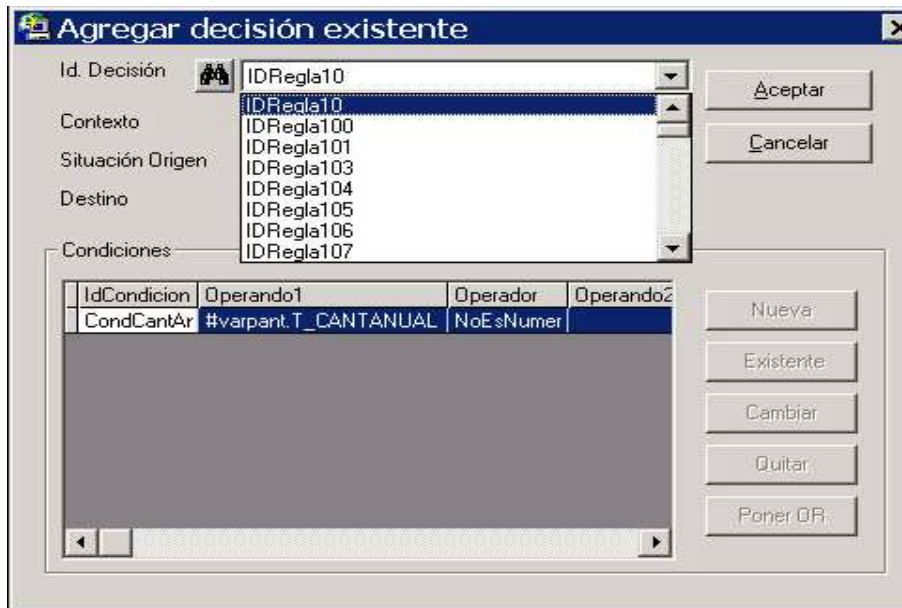


Figura 6.20: Ventana para seleccionar una decisión existente para una situación.

dinámicos para las decisiones.

En la parte inferior de la ventana se encuentra una zona de gestión de condiciones, que permite definir las condiciones que deben cumplirse para que se active la decisión. La gestión de condiciones se detalla en el apartado 6.8.

El botón “Vincular” permite seleccionar una decisión ya existente para vincularla a la situación y contexto seleccionados (figura 6.20). En esta ventana no se va a poder modificar ninguna propiedad de la decisión, pues no hay ningún elemento que sea dependiente de la situación y contexto.

El botón “Modificar” de la pestaña de gestión de decisiones permite cambiar las propiedades fundamentales de una decisión (figura 6.21). Al igual que en el caso de las verificaciones, la modificación de las condiciones asociadas se realiza a través de una rejilla.

Finalmente el botón “Eliminar”, borra la relación existente entre el comportamiento de la situación para el contexto seleccionado. Esta eliminación no quita la decisión de la base de conocimiento pues puede pertenecer a otros comportamientos. En caso de que la decisión no aparezca en ningún otro comportamiento, la herramienta da la opción de eliminar definitivamente la decisión de la base de conocimiento.

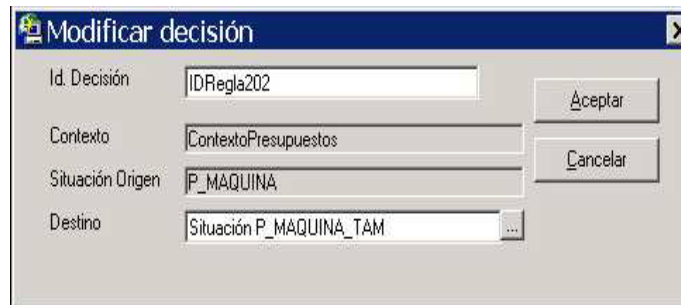


Figura 6.21: Ventana para modificar una decisión en situación y contexto seleccionados.

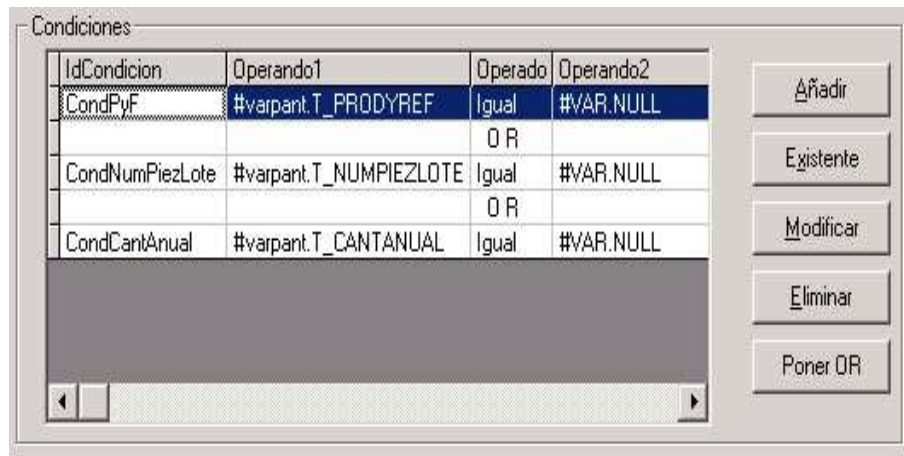


Figura 6.22: Zona de gestión de condiciones.

6.8. Representación de Condiciones

La gestión de condiciones es común a los tres tipos de reglas que se pueden definir con la herramienta de modelado: validaciones, verificaciones y decisiones. En las distintas pestañas relacionadas con la gestión de reglas se ha observado que todas comparte una zona común para la definición de reglas, que se muestra en la figura (6.22).

La gestión de condiciones sigue con la filosofía del resto de objetos de conocimiento, es decir, una rejilla de datos que se actualiza, en este caso al seleccionar una validación-verificación-decisión, y un conjunto de botones que permiten su gestión.

El botón “Añadir” permite añadir una nueva condición. Para añadir una condición, en primer lugar es necesario seleccionar un elemento de la rejilla correspondiente a decisiones, validaciones o verificaciones, y pulsar el botón “Añadir”. En este momento, el usuario debe seleccionar si desea añadir la condición antes o después de la posición seleccionada. Esta selección es relevante,

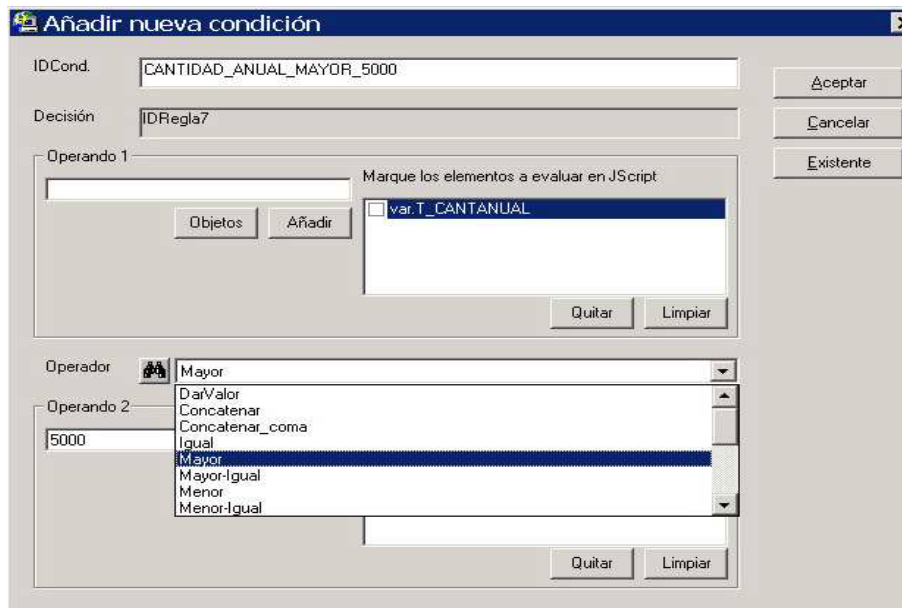


Figura 6.23: Ventana para crear una condición.

pues el orden en que se evalúa la condición puede determinar que se llegue al resultado de forma más rápida.

La ventana “Añadir condición” (figura 6.23) permite definir una condición introduciendo un nombre identificador único.

La creación de una condición implica la definición de dos operandos y la selección de un operador. El primer operando debe tener al menos un valor, mientras que el segundo puede tener cero, uno o varios valores, en función del operador que se esté utilizando para la condición. El valor de cada operando puede ser un literal, un número, el nombre de un hecho, el nombre de una regla, o una expresión que se pueda evaluar en JavaScript.

La definición del contenido de un operando se realiza sobre una caja de texto. Para diferenciar los nombres de los hechos y las decisiones de los literales, se utiliza una sintaxis especial, para que el motor de solución, en tiempo de ejecución, sea capaz de sustituir el nombre del objeto correspondiente por un valor concreto a utilizar.

- Los hechos que están almacenados en la base de hechos, deben utilizar la siguiente sintaxis: “var.” + <nombre del hecho> + <espacio>.
- Los hechos no se almacenan en la base de hechos hasta que no se han ejecutado las verificaciones. Además puede ser necesario distinguir entre los hechos que se han recibido de la acción que se acaba de ejecutar, de los hechos que se habían producido anteriormente.

Los hechos que se acaban de recibir, y que aún no están almacenados en la base de hechos deben referenciarse de acuerdo con la siguiente sintaxis: “varpant.” + <nombre del hecho> + <espacio>.

- Cuando el contenido de un hecho está en formato XML, se pueden realizar consultas adicionales sobre el mismo, sustituyendo el nombre de la variable por el contenido en formato XML o por un valor concreto, resultado de una consulta XPATH:
 - Para obtener el XML resultante de una consulta XPATH sobre una variable que contiene información XML, se empleará la siguiente sintaxis: “VARXML.” + < nombre del hecho > + < expresión XPATH >. En la expresión XPATH se sustituirán los separadores (“/”) por puntos (“.”).
 - Para obtener el valor de una consulta XPATH aplicada sobre el XML contenido en un hecho, se empleará la siguiente sintaxis: “VALXML.” + < nombre del hecho > + < expresión XPATH >. En la expresión XPATH se sustituirán los separadores (“/”) por puntos (“.”). Este resultado no contendrá ninguna etiqueta XML, solamente recuperará el contenido de las etiquetas (o atributos) seleccionados mediante la expresión XPATH.
- Las decisiones se pueden referenciar dentro de un operando siguiendo la siguiente sintaxis: “REG.” + < nombre de la decisión > + <espacio>. El motor de solución sustituirá el nombre de la decisión por el valor “1” si la decisión se cumple o por “0” en caso contrario.
- También puede recuperarse un valor basado en operaciones estadísticas sobre valores anteriores de un hecho determinado. Es lo que se ha denominado *experiencia*. Para ello se empleará la siguiente sintaxis: “exp.” + <nombre del hecho> + “.” + <valor a recuperar> + <espacio>. El valor a recuperar puede ser:
 - **avg**. Recupera la media aritmética de los valores que haya tomado el hecho para todas las sesiones anteriores.
 - **sum**. Suma de todos los valores que haya tomado el hecho para todas las sesiones anteriores.
 - **count**. Número de valores que ha tomado el hecho en todas las sesiones anteriores.
 - **stdev**. Desviación típica de los valores que ha tomado el hecho en las sesiones anteriores.

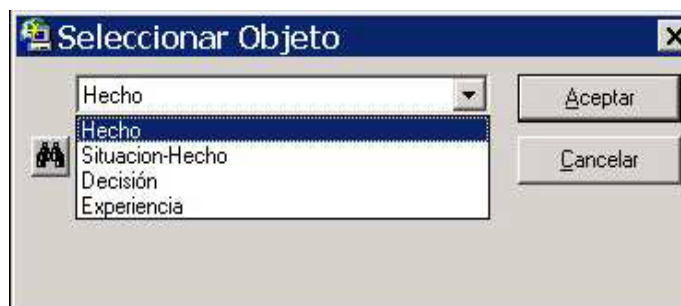


Figura 6.24: Selección de un objeto de conocimiento para un operando.

- **max.** Valor máximo de los que ha tomado el hecho en las sesiones anteriores.
 - **min.** Valor mínimo de los que ha tomado el hecho en las sesiones anteriores.
 - **var.** Varianza de los valores que ha tomado el hecho en las sesiones anteriores.
- El operando puede ser también una operación en JavaScript. Para ello se escribe en el cuadro de texto del operador una expresión de este lenguaje. Pueden utilizarse referencias a objetos de conocimiento dentro de la expresión, pues sus valores se sustituirán antes de evaluar la expresión.

Para facilitar la utilización de objetos de conocimiento en la definición de operadores, pulsando el botón “Objetos” aparece una ventana que facilita la selección del tipo de objeto de conocimiento que se quiere utilizar (figura 6.24). Cuando se pulsa el botón aceptar, aparece en el cuadro de texto del operando la referencia al objeto con la sintaxis correcta.

Tras la definición de los operandos es necesario seleccionar un operador. Por defecto se ofrecen un conjunto de operadores que se pueden emplear para la evaluación de condiciones. Estos operadores son:

- “Alguno contenido”. La condición se cumple si alguno de los valores del primer operando está contenido en el conjunto de valores del segundo operando.
- “Concatenar”. Concatena en una cadena de caracteres el conjunto de valores que contiene el segundo operando, y lo asigna a un hecho que se especifica en el primer operando. La condición se cumple si se consigue realizar la operación correctamente.
- “Contenido”. La condición se cumple si todos los valores del primer operando están contenidos en el conjunto de valores del segundo operando.

- “Dar valor”. Asigna a un hecho indicado en el primer operando el valor o valores que se especifican en el segundo operando. La condición se cumple si se consigue realizar la operación correctamente.
- “Distinto”. La condición se cumple si el valor del primer operando es diferente del valor del segundo operando.
- “Es cierto”. La condición se cumple si el valor del primer operando es el valor booleano “true”. Este operador no admite segundo operando.
- “Es número”. La condición se cumple si el valor del primer operando es numérico. Este operador es especialmente útil para las verificaciones. No admite un segundo operando.
- “Es texto”. La condición se cumple si el valor del primer operando es de tipo texto. Este operador es útil para las verificaciones. No admite un segundo operando.
- “Igual”. La condición se cumple si el valor del primer operando es igual al valor del segundo operando.
- “Mayor”. La condición se cumple si el valor del primer operando es mayor que el valor del segundo operando.
- “Mayor-igual”. La condición se cumple si el valor del primer operando es mayor o igual que el valor del segundo operando.
- “Menor”. La condición se cumple si el valor del primer operando es menor que el valor del segundo operando.
- “Menor-igual”. La condición se cumple si el valor del primer operando es menor o igual que el valor del segundo operando.
- “No contenido”. La condición se cumple si el conjunto de valores del primer operando no está contenido en el segundo.
- “No es cierto”. La condición se cumple si el valor del primer operando tiene el valor “false”. No admite un segundo operando.
- “No es número”. La condición se cumple si el valor del primer operando no es un número. No admite un segundo operando.

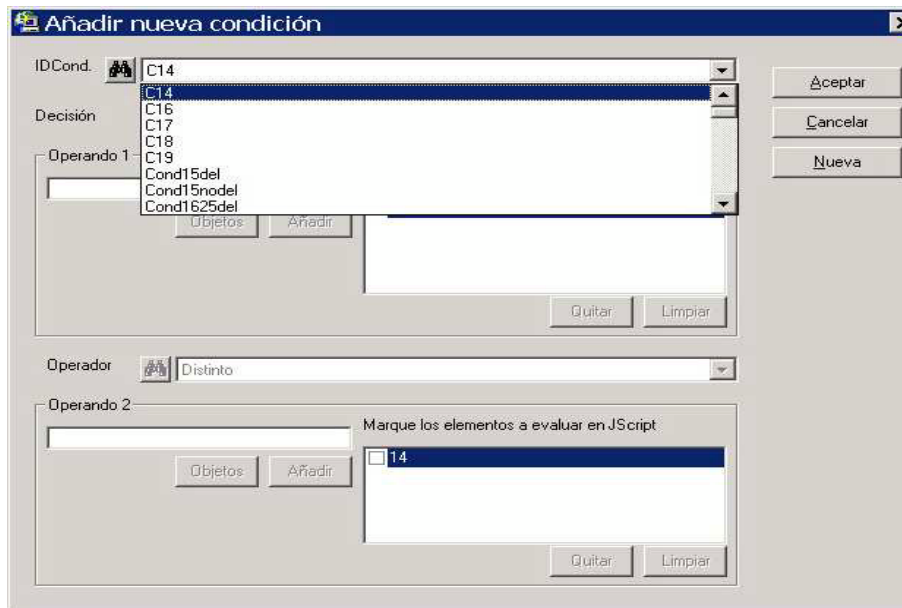


Figura 6.25: Selección de una condición existente.

- “Or”. La condición se cumple si uno de los dos operandos toma el valor booleano “true”.

El botón “Modificar” junto a la rejilla de condiciones, permite modificar la condición que se ha seleccionado, de forma análoga a la ventana del botón “Añadir”.

El botón “Existente” permite seleccionar una condición ya existente (figura 6.25). Al seleccionar una condición se pueden ver sus características, pero no se pueden modificar. Al pulsar aceptar, la condición queda asociada a la regla (decisión, verificación o validación) indicada. Esto facilita la reutilización de condiciones en varias reglas, cuando sea necesario.

El botón “Eliminar” desvincula la condición seleccionada de la regla con la que se está trabajando. Si el elemento seleccionado es un “Or”, se unifican las condiciones de los dos grupos de condiciones que separaba dicho “Or”.

El botón “Or” añade un operador Or a continuación de la posición seleccionada. De esta forma para que se cumpla la regla seleccionada basta con que se cumplan las condiciones de uno de los bloques separados por condiciones Or.

6.9. Representación de Hechos

Los hechos se pueden ver desde una doble perspectiva: a nivel de definición del proceso o a nivel de ejecución del proceso. A nivel de ejecución, los hechos definidos van a tomar un valor

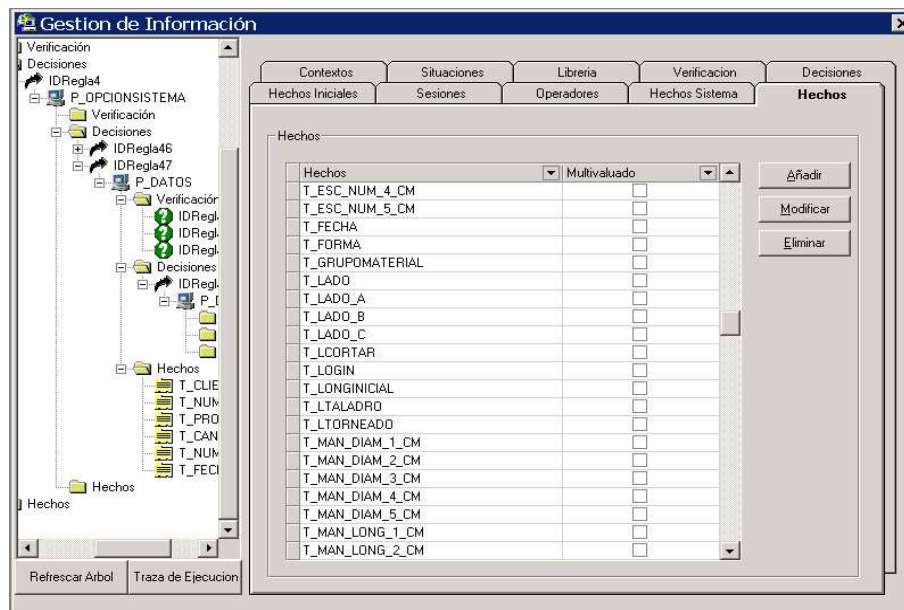


Figura 6.26: Pestaña para la gestión de hechos.

concreto para el proceso que se está ejecutando y permanecen almacenados en la base de hechos del motor de solución, en forma de tuplas ($\langle \text{nombre del hecho} \rangle, \langle \text{valor} \rangle$) asociadas a una sesión.

A nivel de definición, los hechos son información que se espera recibir para poder tomar decisiones. Estos hechos se definen a través de un nombre y una definición del tipo de hecho (multivaluado o no). La pestaña “Hechos” (figura 6.26), permite la definición de hechos que no van a estar relacionados directamente con una situación.

En la pestaña de gestión de hechos una rejilla muestra el conjunto de hechos que se han definido hasta el momento, junto con la indicación del tipo de hecho. Como es habitual, junto a la rejilla, tres botones permiten crear nuevos hechos, modificarlos o eliminarlos.

El botón “Añadir” un hecho muestra la ventana de creación de un nuevo hecho (figura 6.27). En esta ventana se define el nombre del hecho que se quiere crear y el tipo (multivaluado o no). La ventana de modificación (botón “Modificar”) es análoga a la anterior. Finalmente el botón “Eliminar” elimina de la base de conocimiento el hecho seleccionado en la rejilla de datos.

En la especificación del modelo, los hechos pueden tener un valor inicial determinado por el experto. Este valor se especifica en la pestaña “Hechos iniciales” (figura 6.28). En esta pestaña una rejilla de datos muestra el conjunto de hechos que tienen definido un valor inicial. Junto a la rejilla de datos, el conjunto de botones que permiten la gestión de los valores iniciales de los hechos.

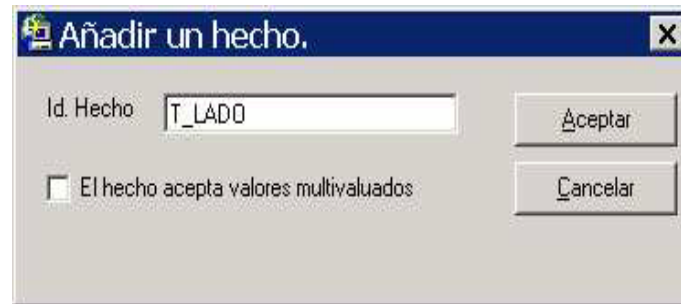


Figura 6.27: Ventana para la creación de un hecho.

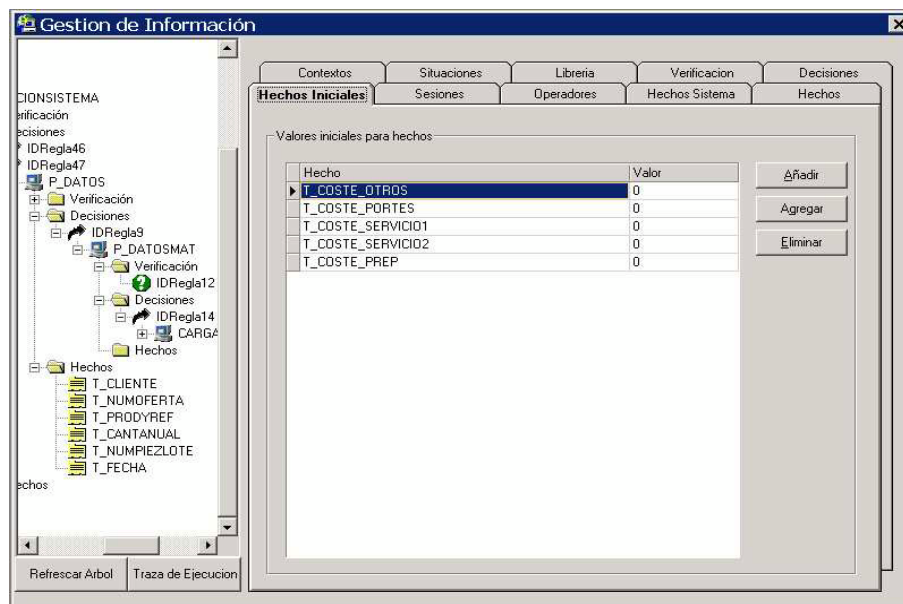


Figura 6.28: Pestaña para la gestión de valores iniciales.

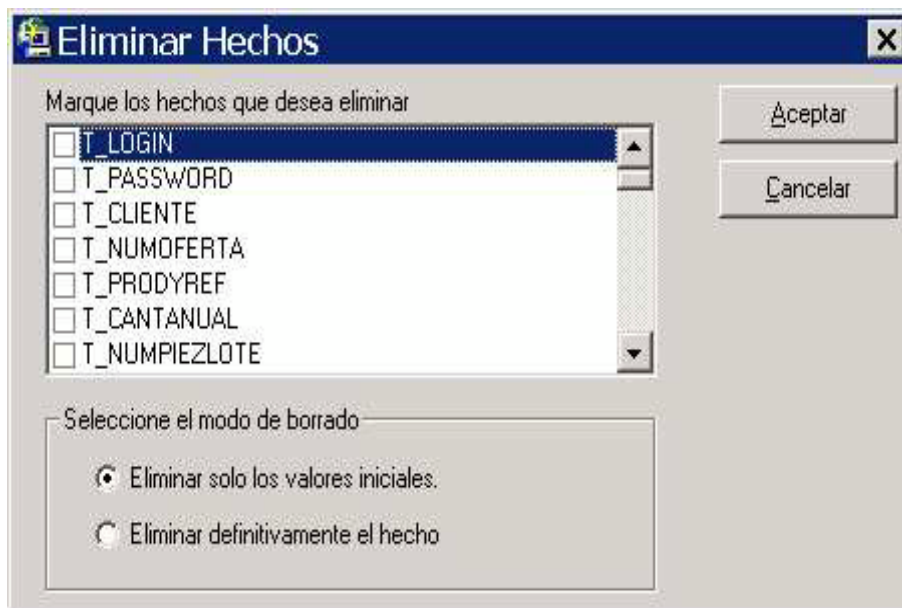


Figura 6.29: Ventana para eliminar un conjunto de hechos.

El botón añadir, permite añadir un nuevo hecho junto con su correspondiente valor inicial. El botón agregar, añade un valor inicial para un hecho ya existente seleccionado. El botón eliminar muestra la ventana para eliminación de valores iniciales.

La ventana de eliminación de valores iniciales permite la selección de aquellos hechos que quieran eliminarse (figura 6.29). Para los hechos seleccionados se ofrecen dos posibles opciones. La primera consiste en eliminar solo los valores iniciales. La segunda eliminará definitivamente tanto el hecho como los valores iniciales de la base de conocimiento.

6.10. Representación de Operadores

El modelo de representación define un objeto de conocimiento que representa los operadores que pueden ser empleados en la definición de condiciones. En el apartado anterior se ha descrito un conjunto de operadores predefinidos por la herramienta de modelado e implementados por el motor de solución, que facilitan la definición de condiciones comunes. A partir de estos operadores predefinidos se puede cubrir un amplio espectro de necesidades. Sin embargo, en ocasiones es necesario definir nuevos operadores en función del dominio del proceso que se está representando.

Se ha definido una interfaz que debe cumplir un operador, de forma que cuando el motor de solución detecta que el operador utilizado en una condición no se encuentra entre los predefini-

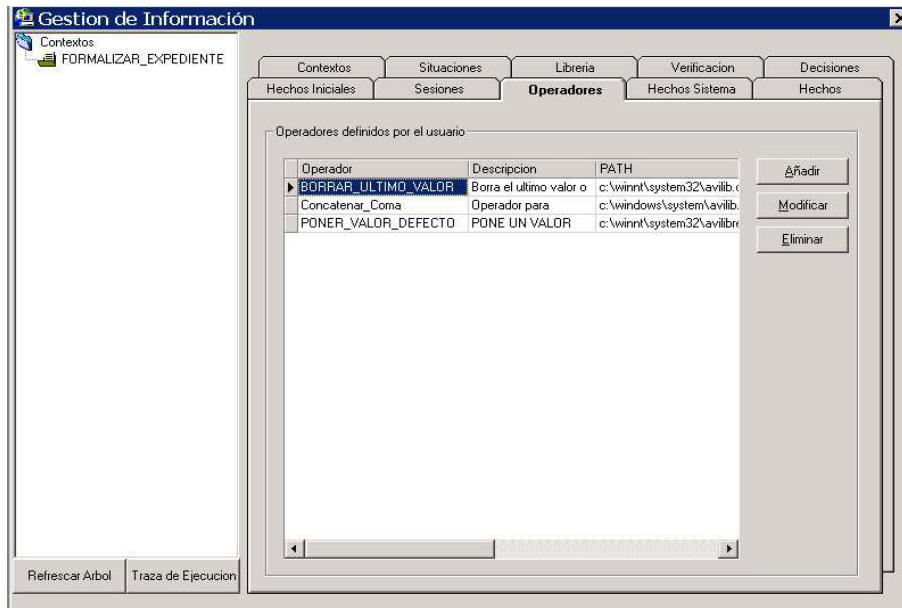


Figura 6.30: Pestaña para la gestión de operadores.

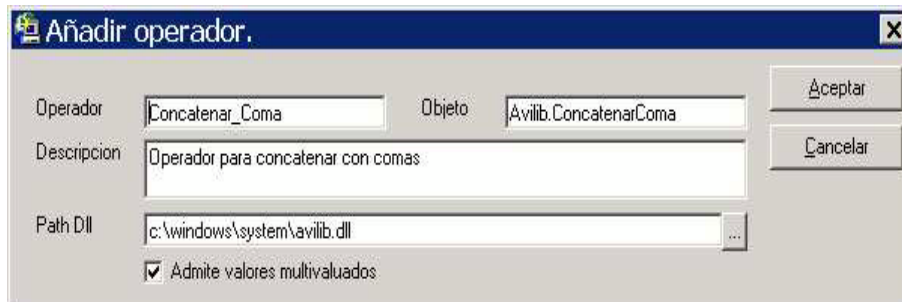


Figura 6.31: Ventana para crear un nuevo operador.

dos, instanciará el operador y ejecutará un método determinado que devolverá si la condición se cumple o no se cumple a partir de los operandos definidos.

La pestaña “Operadores” muestra en una rejilla de datos el conjunto de operadores que ha definido el usuario (figura 6.30). Junto a la rejilla, como en el resto de pestañas, existe un conjunto de botones para añadir, modificar o eliminar un operador.

El botón “Añadir” permite crear un nuevo operador que pueda ser utilizado en las condiciones (figura 6.31). Los datos para el operador son:

- Operador. Es el nombre identificador único del operador. Este nombre es el que se selecciona en la ventana para crear una condición.

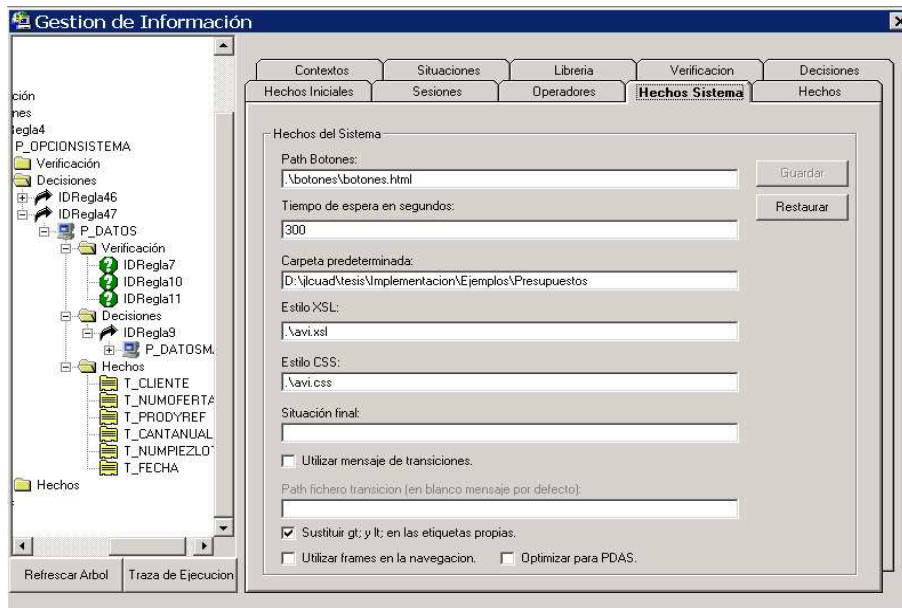


Figura 6.32: Configuración general del proceso.

- Objeto. Para que el motor de solución pueda instanciar el operador durante el proceso de solución, es necesario determinar el nombre completo del objeto (de la forma NombrePaquete.Clase, donde NombrePaquete es el nombre del contenedor del objeto, y Clase es el nombre de la clase a instanciar).
- Descripción. Breve descripción del operador.
- PathDll. Ruta completa de la librería que contiene la clase del operador.
- Admite Multivaluados. Indica si el operador admite valores multivaluados o no.

El botón “Modificar” muestra una ventana análoga a la de añadir, que permite modificar las propiedades del operador seleccionado.

El botón “Eliminar” elimina el operador seleccionado de la base de conocimiento.

6.11. Representación de Parámetros del Proceso

El motor de solución implementado admite un conjunto de valores por defecto que facilitan el control del proceso de decisión. La pestaña “Hechos del sistema” (figura 6.32) facilita la definición de estos parámetros.

Los parámetros que se pueden configurar son:

- **Path Botones.** Establece un fichero que contiene los botones predeterminados que se utilizarán en las situaciones interactivas. Los botones son imágenes o elementos de tipo botón en un formulario que se incluyen en las acciones externas, de forma que al pulsar sobre ellos envía la información contenida en la acción externa desde el navegador al motor de solución. Los botones predeterminados se incluyen en las acciones externas siempre que así se indique en la definición de la misma, facilitando la gestión en la medida en que los mismos botones se pueden utilizar en varias situaciones con solo definirlos en esta pestaña.
- **Tiempo de espera en segundos.** El motor de solución está integrado en un navegador web. La gestión de sesiones que utiliza el servidor utiliza un tiempo limitado para mantener la información. El tiempo máximo que puede tardar una sesión en recibir nueva información puede ser configurado desde esta caja de texto.
- **Carpeta predeterminada.** Cuando son muchas las páginas web definidas, es habitual que se encuentren dentro de la misma carpeta. La carpeta predeterminada permite localizar los ficheros que se encuentren en la misma sin tener que escribir la ruta completa. Para hacer referencia a la carpeta predeterminada es suficiente con sustituir la ruta completa por “.\”
- **Estilo xsl.** Define una hoja de estilo que se aplicará por defecto en los mensajes basados en páginas XML, siempre que así se indique en la situación interactiva. Esto facilitará la definición de acciones externas basadas en XML con apariencia similar.
- **Estilo css.** define una hoja de estilo CSS que se aplicará por defecto en los mensajes basados en páginas HTML, siempre que así se indique en la definición de la situación interactiva.
- **Situación final.** Define un mensaje que se enviará al agente externo cuando éste seleccione finalizar el proceso de solución.
- **Utilizar mensaje en las transiciones.** Define un mensaje que se mostrará al agente externo mientras se procesan los hechos en el servidor. Si esta opción es seleccionada, se puede establecer un fichero que contenga una página web con este mensaje. Si no se especifica ningún fichero, se mostrará el mensaje “Cargando, espere por favor”.
- **Sustituir gt; lt; en etiquetas propias.** En ocasiones, algunos editores de páginas web, cuando no conocen un tag determinado, sustituyen los delimitadores < y > por > y < respectivamente. Cuando se utilizan los tags propios del motor de solución, esto puede provocar que no se sustituyan correctamente los valores deseados. Marcando esta opción

los mensajes asociados a las acciones externas se procesan previamente para corregir este problema.

- Utilizar frames en la navegación. Esta opción indica al motor de solución que realice el intercambio de información con el agente externo a través de un frame en su navegador.
- Optimizar para PDAs. La utilización de botones en las acciones externas suele implicar la utilización de JavaScript. Cuando el acceso al motor de solución se realiza a través de un dispositivo móvil, las características de los navegadores pueden limitar la utilización de este lenguaje. Cuando la opción optimizar para PDAs es seleccionada, no se emplearán botones ni elementos JavaScript para enviar los hechos al motor de solución, sino que incluirá elementos POST en el formulario que contiene los hechos.

6.12. Gestión de Sesiones

Como se ha indicado en el capítulo 5, el motor de solución está integrado en un servidor web. Esto implica que va a poder atender simultáneamente varios procesos de decisión y que, además, va a ser necesario que la base de hechos, que constituye la memoria de trabajo del motor, almacene de forma permanente la información sobre el estado de cada una de las sesiones. Por este motivo, la base de hechos se almacena en una base de datos, en la que, para cada sesión, se almacena los distintos contextos, situaciones por los que va pasando y los valores que han tomado los hechos a lo largo del proceso de decisión.

La pestaña “Sesiones” va a facilitar la gestión de la base de hechos, así como facilitar consultas para obtener información adicional acerca de los distintos procesos que se han llevado a cabo (figura 6.33). Como las bases de datos constituyen un histórico de las sesiones que se han mantenido con el motor de solución, se utiliza el término *histórico* para simplificar la nomenclatura.

En la parte superior de la pestaña muestra un filtro simple de las sesiones que se van a mostrar en la rejilla de datos. El filtro puede ser entre un rango de fechas, en función del valor de una variable o una combinación de ambos.

En la rejilla de datos se carga la información relativa a las sesiones que se han producido hasta el momento. Los datos de las sesiones aparecen agrupados por el identificador de sesión. Para cada una de las sesiones se muestra una fila por cada uno de los hechos que se han recibido. Cada fila contiene:

1. Orden de navegación. Indica el número de orden en que se ha producido el valor del hecho que se está mostrando.

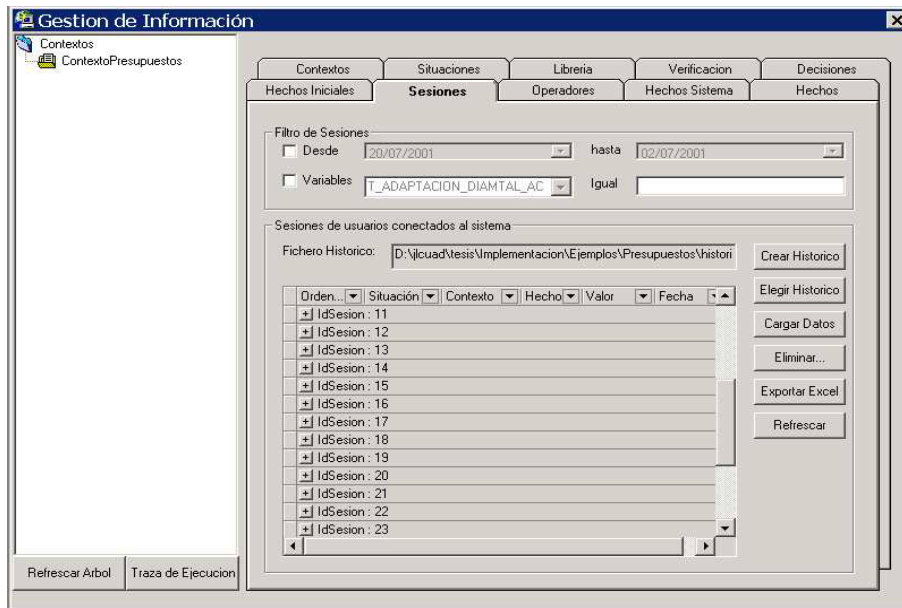


Figura 6.33: Pestaña para la gestión de sesiones del motor de solución.

2. Contexto. Nombre del contexto en el que se ha producido el hecho.
3. Situación. Nombre de la situación en la que el hecho ha tomado el valor.
4. Hecho. Nombre del hecho que ha tomado valor en la situación y contexto indicados.
5. Valor. Valor que ha tomado el hecho. En caso de tratarse de un hecho multivaluado, en la rejilla aparecerán para el mismo hecho tantas filas con el mismo número de orden, situación y contexto como valores diferentes tenga asignados.
6. Fecha en la que se produjo la sesión.

Con esta información es posible reproducir el proceso que ha seguido un agente externo desde el comienzo hasta el final, permitiendo estudiar posibles problemas o tendencias en la resolución del problema.

La rejilla permite, haciendo clic en cada columna, filtrar el contenido que se está mostrando, facilitando así al gestor del conocimiento la obtención de los datos que considere más relevantes.

Los botones que hay junto a la rejilla permiten la gestión de esta información. El botón “Crear Histórico” permite la creación de una base de datos ACCESS que pueda ser utilizada por el motor de solución como base de hechos. El botón “Seleccionar Histórico” permite seleccionar una base de datos ACCESS existente con información acerca de las distintas sesiones. El botón “Cargar

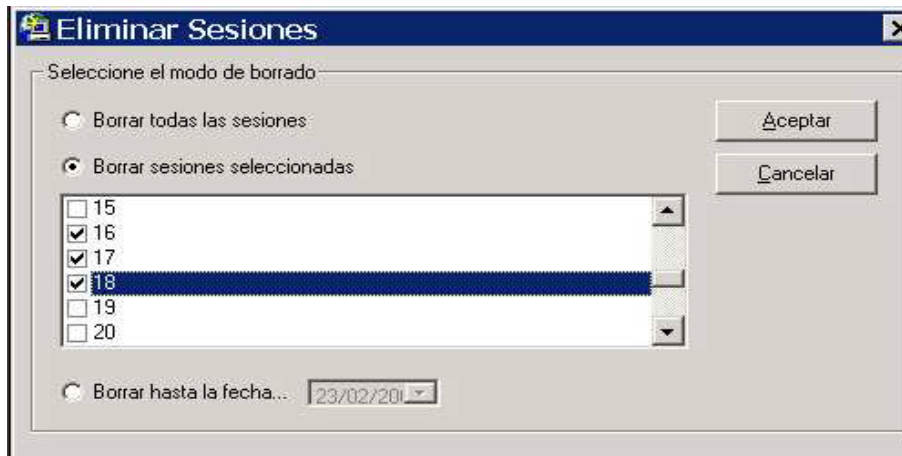


Figura 6.34: Ventana para eliminar sesiones del motor de solución.

Datos” lee el contenido de la base de datos seleccionada y lo carga en la rejilla de datos aplicando los filtros que se hayan establecido. El botón “Eliminar” permite eliminar todas las sesiones, las sesiones seleccionadas o las sesiones hasta una fecha determinada (figura 6.34). El botón “Exportar Excel” guarda en formato Microsoft Excel la información de las sesiones, facilitando la creación de gráficas y análisis más detallados sobre la información generada durante el proceso de solución. Finalmente el botón “Refrescar” actualiza el contenido de la rejilla de datos, opción útil cuando se está siguiendo un proceso mientras se está ejecutando.

6.13. Traza de Ejecución

Mientras se representa el conocimiento del experto, o bien una vez que se han creado todos los objetos de conocimiento, es necesario comprobar que el proceso se reproduce de la forma prevista. Para ello es necesario, no solo poner en operación el motor de solución, sino que, además, es necesario obtener información detallada acerca de la situación y contexto en el que se encuentra el proceso en un instante determinado, los hechos que ha recibido el sistema y las decisiones que se están evaluando.

Para esta labor, la herramienta de modelado proporciona una funcionalidad denominada traza de ejecución. La traza de ejecución reproduce las condiciones de funcionamiento del motor de solución, sin necesidad de utilizar un servidor web.

Cuando se lanza la traza de ejecución, en primer lugar se da la opción de utilizar un fichero de configuración en formato XML. Este fichero de configuración puede servir para modificar determinados parámetros del sistema sin necesidad de modificar la base de conocimiento. Tras se-

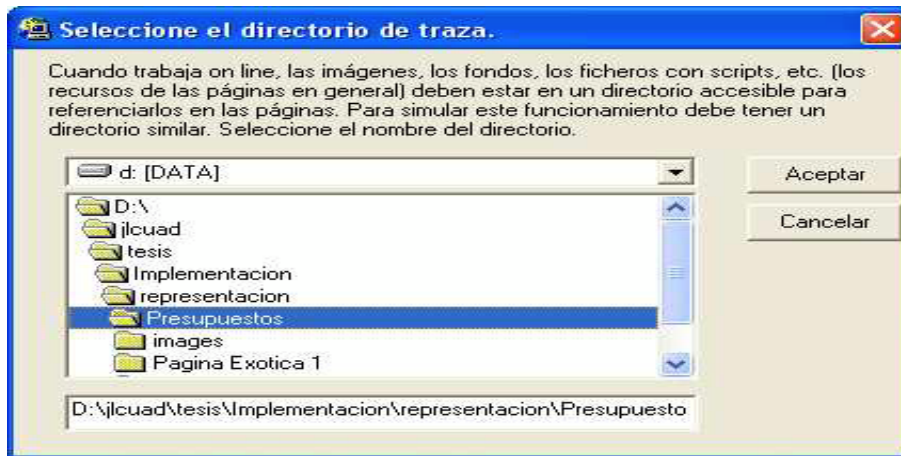


Figura 6.35: Selección del directorio a utilizar en la traza de ejecución.

leccionar un fichero de configuración (si fuera necesario), es necesario seleccionar un directorio que simule el directorio del servidor web donde se encuentra publicado el proceso representado. En este directorio es donde se encontrarán todas las referencias a imágenes y otros elementos (flash, vídeos, etc.) que sean necesarias en las páginas web que se utilicen en las situaciones interactivas.

El directorio para la traza suele ser el directorio por defecto utilizado en la configuración del proceso (ver apartado 6.11). Sin embargo se da la opción de seleccionar un directorio diferente.

Tras la selección del directorio a emplear en la simulación, se muestra la ventana de la traza (figura 6.36). En la parte superior de la ventana tres botones simulan los botones empleados en el navegador para el envío de información. El botón con una flecha hacia el lado izquierdo representa la petición de volver a la situación anterior. El botón con una flecha hacia la derecha envía los valores de los hechos representados en la situación que se está visualizando. El botón “Stop” detiene a traza de ejecución para poder realizar ajustes en la base de conocimiento.

En la parte central de la ventana, un navegador muestra la página correspondiente a la situación interactiva actual. En esta ventana se puede asignar un valor a los elementos de formulario que constituyen los hechos que luego se enviarán al sistema. Además en esta ventana se puede apreciar el formato que visualizará un agente externo cuando acceda al proceso a través de internet. Cuando se pulsa el botón con una flecha hacia la derecha en la zona superior, se recorren todos los formularios de la página y se añade la información de control para enviarla al motor de solución.

El motor de solución envía mensajes durante el proceso de decisión, indicando cada paso que da en la evaluación de la nueva situación. Estos mensajes se pueden visualizar en la parte inferior



Figura 6.36: Ventana de la traza de ejecución.

de la ventana, de forma que se puede comprobar el orden en que se van dando los pasos, y se puede comprobar si los resultados del proceso son acordes al conocimiento que se ha recibido. El log completo se puede guardar desde el menú “Traza → Guardar Log”.

Si durante el proceso de depuración se detecta la necesidad de realizar algún cambio en la base de conocimiento, se puede detener el proceso pulsando el botón “Stop”. Una vez pulsado este botón, se habilita el editor de conocimiento para poder realizar los cambios pertinentes. Una vez realizados dichas modificaciones en la base de conocimiento se puede regresar a la ventana de la traza y, pulsando un botón con texto “Go” que habrá sustituido al botón de “Stop” se podrá continuar con la depuración del proceso.

Para volver a la situación anterior, es necesario pulsar el botón con una flecha hacia la izquierda. Este botón realiza un llamada al motor de solución, indicando que debe regresar a la situación interactiva anterior a la actual. El motor de solución restaura los valores al estado anterior y muestra la situación interactiva anterior, retrocediendo en el proceso.

6.14. Resumen

En este capítulo se describe la funcionalidad de una herramienta de modelado que se ha desarrollado para aplicar el modelo de representación del conocimiento propuesto. La herramienta se compone de un conjunto de editores de conocimiento que permiten la creación de dos distintos

elementos de acuerdo al modelo propuesto, una vista de árbol que permite visualizar la estructura que toma el proceso representado conforme se van introduciendo nuevos elementos de conocimiento, y, finalmente, de una traza que facilita la depuración de los procesos de decisión que se definen a partir de la herramienta.

Capítulo 7

Representación del Proceso de Elaboración de Presupuestos

En este capítulo se describe la aplicación del modelo de representación del conocimiento propuesto al proceso de elaboración de presupuestos para una empresa que se dedica a la fabricación de piezas mecanizadas. A través de los distintos elementos del modelo se representan los pasos que el experto da para la definición del precio y el conocimiento acerca de cómo se estima el precio en la compañía: que parámetros se tienen en cuenta y cómo se calcula el coste. El proceso es interpretado con el motor de solución propuesto en el capítulo 5 para reproducirlo de forma automática.

7.1. Introducción

En los capítulos anteriores se han definido los elementos que componen el modelo de representación propuesto, así como el algoritmo a seguir para interpretar el modelo y reproducirlo empleando un motor de solución. Además se ha descrito una herramienta para representar el conocimiento a partir del modelo propuesto.

A partir del modelo, el motor de solución y la herramienta de modelado, es posible modelar el conocimiento de un experto en la elaboración de presupuestos en una fábrica de piezas mecanizadas.

Las siguientes secciones describen cómo se ha representado el conocimiento de un experto para la estimación de presupuestos en una fábrica de piezas mecanizadas.

7.2. Metodología

Para la definición del modelo de estimación de presupuestos, se han seguido los siguientes pasos:

1. Identificación de los diferentes contextos que se pueden producir en el problema de la estimación de presupuestos en la fábrica.
2. Identificación de las distintas situaciones que se pueden dar durante el proceso de resolución del problema.
3. Clasificación del tipo de acción que debe tomar el motor de solución en cada una de las situaciones propuestas.
4. Determinación de los hechos relevantes en cada una de las situaciones.
5. Definición de las verificaciones y decisiones que se deben realizar en cada situación.
6. Creación de los mensajes para cada una de las situaciones interactivas.
7. Creación de las acciones internas necesarias.
8. Representación con la herramienta de modelado.
9. Pruebas y puesta en marcha.

La captura de conocimiento e implementación a partir del modelo propuesto en esta tesis, se ha llevado a cabo el marco de una fábrica de piezas mecanizadas, que responde a las características de una empresa de fabricación versátil. Este trabajo se llevó a cabo en el marco del proyecto Fernandez Heijnen (2004) y López-Cuadrado et al. (2005), Lopez-Cuadrado et al. (To appear)

En un paso previo se había identificado la necesidad de la compañía de liberar la carga de trabajo de sus expertos en la elaboración de presupuestos. Un sistema basado en conocimiento supondría por una parte mantener el conocimiento de sus expertos en la compañía, y por otra una ayuda para que otros empleados con menos experiencia pudieran hacerse cargo de la estimación. Por tanto, desde el principio se planteó la representación de un proceso guiado (ver 5.2), que sirviera de asistente para los empleados en la elaboración de un presupuesto. Por tanto, condicionados por el tipo de proceso se identificaron los agentes que iban a intervenir en el proceso:

- El agente externo va a ser un empleado de la fábrica. Por medio de los mensajes irá proporcionando al motor de solución los datos necesarios para realizar la estimación.
- Agente Software: Microsoft Excel. El motor de solución tendrá que intercambiar información con Microsoft Excel para realizar los cálculos de tiempos y costes que el experto de la fábrica realiza habitualmente.

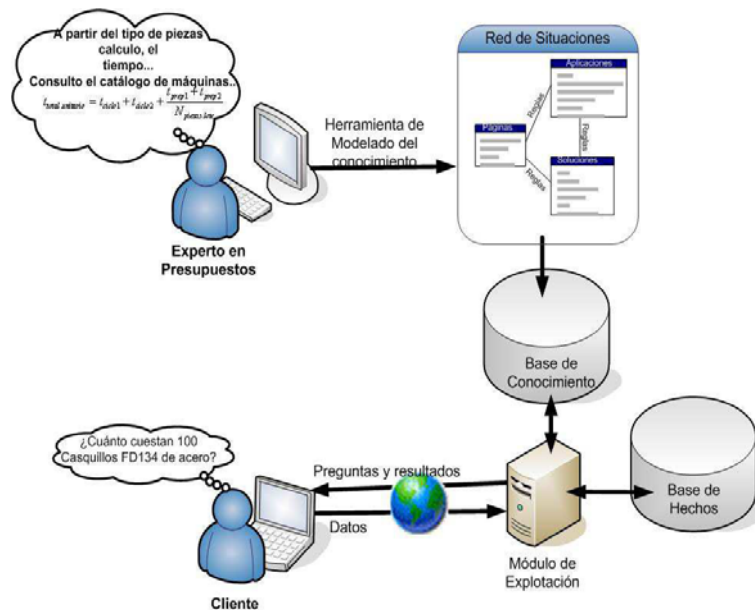


Figura 7.1: Modelado y automatización de la estimación.

- Agente Software: Microsoft Access. El motor de solución necesita conocer distintos parámetros relacionados con el proceso de fabricación que se encuentran almacenados en una base de datos. Por tanto deberá hacer las consultas necesarias para obtener esta información.

La figura 7.1 muestra como a través de la herramienta de modelado se ha definido una red de situaciones que representa el proceso de solución al problema de la estimación de presupuestos, a partir del conocimiento del experto. Este conocimiento se almacena en una base de conocimiento creada de acuerdo con el modelo de representación del conocimiento propuesto. Un agente externo enviará al motor de solución los datos de los que disponga para la elaboración del presupuesto de fabricación, y el motor de solución irá solicitando más información e informando de los resultados a los que ha llegado.

Dadas las características del trabajo, también se restringió el número de máquinas para las que se iba a capturar el conocimiento: tres tornos automáticos multihusillo.

En la primera fase se llevaron a cabo entrevistas con el experto de la fábrica para conocer las necesidades de la fábrica y cuál es el proceso de estimación que sigue el experto. En estas entrevistas se identificaron, de acuerdo a la forma de razonar del experto, las distintas situaciones que se podían producir durante su estimación y se redujo el problema a un solo contexto, más otro que serviría para gestionar datos de las bases de datos de piezas y parámetros de fabricación.

Tras la identificación inicial de las situaciones se estudió cuál sería la acción asociada a cada una de las situaciones y cuál sería el conjunto de hechos que se generaría en cada una de ellas. Esta decisión está condicionada por el tipo de proceso que se quiere crear, pues al crear un asistente el intercambio e información se realizará sobre todo por medio de situaciones interactivas que guiarán al agente externo y solicitarán información necesaria para llevar a cabo la estimación. Las acciones internas se utilizarán para obtener los parámetros de fabricación y realizar los cálculos necesarios de tiempos y costes.

El análisis de cada situación y su acción asociada, conlleva la definición de las verificaciones que es necesario realizar, y cuáles son las posibles decisiones que se pueden llevar a cabo. Estas decisiones son expresadas de forma natural por el experto en forma “si (condiciones) entonces (siguiente paso)”, y recogidas a través de la herramienta de modelado.

Para cada una de las situaciones interactivas identificadas se creó una página HTML como mensaje asociado. En cada una de estas páginas los hechos se representan como elementos de un formulario que es enviado al motor de solución para que los recoja y continúe con el proceso.

Para las acciones internas se utilizaron dos librerías como interfaz de los dos agentes software que intervienen en el proceso de estimación: Excel y Access. La librería de conexión con Excel recibirá como parámetro la hoja de cálculo que debe ser utilizada, las celdas que deben recibir un valor (junto con el valor) y las celdas de las que se recuperará un valor como respuesta (junto con el nombre del hecho en el que debe almacenarse). La librería de conexión con Access recibirá como parámetro la base de datos a la que debe conectarse, el nombre de la tabla sobre la que debe actuarse, los parámetros de la consulta a ejecutar (inserción, modificación, actualización o borrado) y los valores a recuperar (si los hubiera) junto con el nombre del hecho en que se almacenará cada valor. En la definición de cada una de las situaciones automáticas, se definen los parámetros asociados al contexto en que se producen.

En paralelo a la captura del conocimiento y la definición de las acciones a realizar, se fue modelando el proceso por medio de la herramienta de representación del conocimiento, pudiendo ir paso a paso comprobando el efecto de cada acción en las distintas situaciones que se van creando.

Una vez finalizada la definición del proceso de estimación del experto, se puso en marcha en la compañía y se comprobaron junto con el experto un conjunto de casos de prueba para comprobar los resultados obtenidos. La sección “Evaluación” concreta los resultados de las pruebas realizadas y detalla las consecuencias sobre las hipótesis de investigación planteadas al comienzo de esta tesis.

Las siguientes secciones resumen el conocimiento del proceso adquirido y cómo es representado a través del modelo propuesto para la estimación de precios para piezas que se fabriquen con tornos automáticos multihusillo.

7.3. Descripción del Proceso de Estimación

El primer paso para representar el proceso de estimación que sigue el experto de la fábrica en cuestión, consistió en determinar los pasos que da para calcular un precio. Además resultaba de interés conocer no solo el proceso en sí, sino el contexto en el que se produce, esto es, en el proceso RFQ (ver apartado 2.3). Este contexto en el que se lleva a cabo el proceso es relevante a la hora de determinar la información relevante para decidir el resultado, así como los datos relevantes que deben resultar de la estimación para que sea útil para el proceso.

El proceso completo seguido por la compañía seleccionada para el caso de estudio es el siguiente:

1. La compañía recibe una petición de presupuesto sobre una pieza, en la que se incluyen datos identificativos de la misma. En ocasiones se disponen de planos adicionales o piezas de muestra.
2. A partir de la petición del cliente, se define el proceso necesario para fabricar la pieza. En esta definición se identifican las operaciones necesarias, las máquinas que van a intervenir y se estima el tiempo y el coste total de fabricación. Estos cálculos suelen ser aproximados. A partir de estas estimaciones se responde al cliente con un presupuesto indicando el número de referencia, cantidad anual, tamaño de los lotes, material y precio.
3. En caso de que el cliente acepte el presupuesto se acuerdan las fechas de entrega y se formaliza el pedido.
4. Cuando se recibe el pedido se elabora una hoja detallada con las máquinas y la preparación necesaria. Para cada lote a fabricar se especifica detalladamente el proceso que se estimó en la elaboración del presupuesto. La documentación que generan estas actividades se gestiona por medio de una aplicación informática.
5. Una vez se han cumplimentado los pasos anteriores, puede comenzar la fabricación de las piezas.

La tarea de determinar el proceso y estimar el tiempo y coste de fabricación cuando se recibe una petición de presupuesto es realizada por un experto de la fábrica, en base a sus conocimientos. En las estimaciones de tiempo se consideran los tiempos de preparación, y en los costes se tienen en cuenta los costes del material, portes embalajes y otros servicios adicionales.

Las siguientes subsecciones resumen el conocimiento capturado del experto acerca del proceso de estimación para la fabricación con tornos automáticos multihusillo, y cómo se ha representado a partir del modelo de representación propuesto.

7.4. Aplicación del Modelo

Las situaciones interactivas van a permitir al agente externo enviar y recibir información del motor de solución: de esta forma se va a proporcionar al motor la información necesaria para calcular el precio a ofrecer. Por otro lado, las situaciones automáticas van a permitir la realización de los cálculos necesarios durante el proceso de estimación, así como acceder a las bases de datos que van a contener los parámetros de fabricación que deben ser tenidos en cuenta para calcular el precio. Finalmente, las situaciones virtuales agruparán decisiones que, por simplicidad de representación, serán separadas de otras situaciones. Estas situaciones van a representar el conocimiento del experto acerca de los datos que son necesarios para estimar el precio y las operaciones que deben realizarse durante el proceso.

Las situaciones se van a conectar entre sí por medio de decisiones. Las decisiones van a representar el conocimiento del experto acerca de cómo actuar en cada uno de los pasos del proceso y cuál es el siguiente paso que debe darse.

En la definición del proceso de solución se han identificado dos contextos diferentes. Un contexto relativo a la estimación de precio y un segundo contexto relativo a la inclusión de nueva información relativa a distintos accesorios empleados en la mecanización de una pieza que puedan influir en la determinación de un precio. Concretamente la información relativa a estos accesorios permite al motor de solución determinar el tiempo de fabricación y proporcionar al agente externo información acerca de los parámetros que deben ser tenidos en cuenta durante el proceso de fabricación.

El contexto relativo a la información de accesorios se compone de un conjunto de situaciones interactivas en las que el usuario introduce los datos requeridos acerca de nuevas pinzas que se puedan utilizar en los tornos. Tras introducir estos datos, que son el torno para el que se da de alta la pinza, el tipo de pinza (redonda o hexagonal) y su diámetro, se verifica que los datos son correctos y se guardan en la base de datos de la compañía en una situación automática, por medio



Figura 7.2: Grafo común de decisión.

de la acción interna para acceder a bases de datos.

A continuación se describe la representación del proceso de estimación de precios.

Contexto de estimación de precios

El objetivo del contexto de estimación de precios es el de proporcionar al agente externo la información necesaria para proporcionar una respuesta a una petición de oferta recibida. Además, cuando el experto realiza la estimación del precio también produce información que, en caso de que el cliente acepte el precio, será necesaria para el proceso de fabricación. Por tanto además de generar información para la estimación de precio, como resultado del proceso representado, el motor de solución determinará parámetros recomendados para las operaciones a realizar.

La fábrica estudiada dispone de diversos tipos de máquina que pueden ser empleadas en la fabricación de piezas. Estas máquinas son tornos automáticos multihusillo centros de mecanizado y máquinas de control numérico (CNC).

La figura 7.2 muestra la estructura general de las situaciones que se pueden dar en el contexto de la determinación del precio de fabricación. El grafo de situaciones tiene tres alternativas claramente definidas: una para la estimación para la fabricación con tornos, otra para la fabricación con centro de mecanizado y otra para la fabricación con máquinas de control numérico. Para la validación sólo se han tomado en cuenta los tornos automáticos multihusillo.

A continuación se describen las distintas situaciones y las decisiones que se han representado.

Información general

Independientemente de las piezas a fabricar y del proceso de fabricación, el experto requiere un conjunto de datos relativos a la petición de oferta que se ha recibido. Esta es la información

relativa al cliente y a la solicitud que se está tratando.

Por tanto, la primera situación es la que representa el comienzo del proceso de estimación, en la que el experto adquiere la información general acerca de la petición de oferta. Se va a tratar de una situación interactiva, en la que el agente externo proporcionará esta información obtenida a través de la solicitud del cliente. Los hechos asociados a esta situación interactiva son:

- Nombre del Cliente.
- Número de Oferta.
- Producto y referencia.
- Cantidad anual.
- Numero de piezas del lote.
- Fecha del pedido.

La acción externa asociada a la situación interactiva es una página web que contiene un formulario en el que se solicitan al agente externo los valores para los hechos reseñados anteriormente, tal y como se muestra en la Figura 7.3. Estos hechos son utilizados más adelante para calcular el coste de cada pieza y el coste total de cada uno de los lotes que se vayan a producir.

Tras esta situación, el experto necesita conocer el tipo de material que se debe utilizar para fabricar la pieza solicitada, por lo que se crea una decisión para llevar a la situación de selección de material.

Selección de máquina a utilizar

El siguiente paso del experto es determinar cuál es la máquina que se va a emplear para la mecanización de la pieza, pues el proceso de fabricación, y, por consiguiente, los componentes del coste, serán diferentes en función de la máquina que se emplee.

Es el agente externo el que va a decidir la máquina a utilizar durante el proceso de fabricación. Por tanto el momento de seleccionar la maquinaria a utilizar se va a representar por medio de una situación interactiva. La acción externa asociada a la situación interactiva será una página web, en la que el agente externo seleccionará, dentro de un formulario, la máquina a emplear en el proceso. Para el prototipo se han tenido en cuenta dos tipos de máquina, los tornos automáticos multihusillo y los centros de mecanizado.

Figura 7.3: Datos generales para el presupuesto.

En esta situación hay dos posibles decisiones a tomar. En caso de seleccionar la fabricación con tornos automáticos multihusillo, la siguiente situación debe ser la selección del torno a utilizar. En caso de seleccionar la fabricación con un centro de mecanizado, la siguiente situación debe llevar a la selección del material a utilizar, puesto que solo hay un centro de mecanizado disponible.

Selección de un torno

Si el agente externo ha seleccionado un proceso de fabricación con tornos, la figura 7.4 muestra cuál es el grafo de decisión a seguir a continuación. Las cajas azules representan situaciones interactivas donde el agente externo va a enviar información acerca del proceso, mientras que las cajas verdes van a representar conjuntos de situaciones virtuales y de situaciones automáticas que van a representar la forma de actuar del experto por medio de cálculos externos y consultas.

Hay tres posibles tornos que pueden ser utilizados en la fábrica:

1. Torno Ruso Wainer D.40.
2. Torno Plitter 2" D.50.
3. Torno Conomatic 2"5/8 D.67.5.

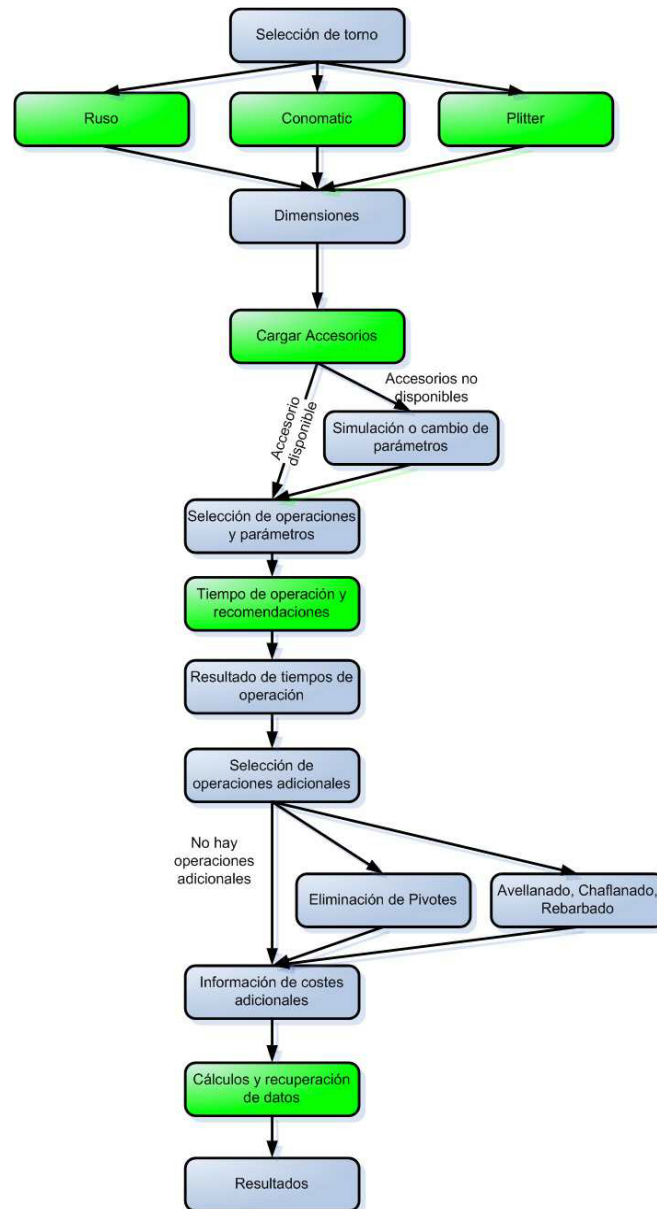


Figura 7.4: Grafo de situaciones para la estimación con tornos.

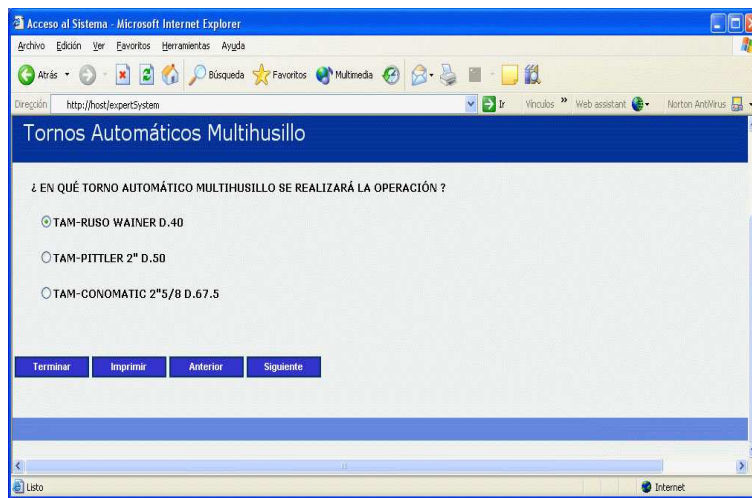


Figura 7.5: Selección del torno a utilizar.

La selección del torno a utilizar se va a realizar por medio de una situación interactiva, en la que el agente externo seleccionará el torno que se va a utilizar. Para ello, la acción externa asociada a la situación va a ser una página HTML con un formulario en el que se va a seleccionar una de las tres opciones que hay. La figura 7.5 muestra la acción definida para la selección del torno.

El hecho con el torno seleccionado será empleado posteriormente para determinar los parámetros de fabricación que son dependientes de la máquina y que van a influir en el tiempo de fabricación y en el coste final de la pieza.

Una vez que se conoce el torno en que se va a trabajar, el experto solo determina una decisión posible: la de conocer el material a emplear durante el proceso de fabricación.

Selección de material

Tras recibir la información general y la máquina que se va a emplear durante el proceso de fabricación, es necesario conocer el material a utilizar en la fabricación de la pieza. Este material va a afectar al tratamiento posterior en el proceso de fabricación y al coste final.

En la representación del proceso, esto se va a traducir en una situación interactiva, en la que el agente externo debe seleccionar un material de los distintos materiales disponibles en la fábrica. La acción externa asociada a esta situación interactiva es una página web en la que se muestra un formulario con un listado de materiales disponibles en la fábrica (figura 7.6).

Cuando el agente externo selecciona el material en la página web, se envía al motor de solución el código del material para que lo almacene como un hecho y lo tenga en cuenta durante el

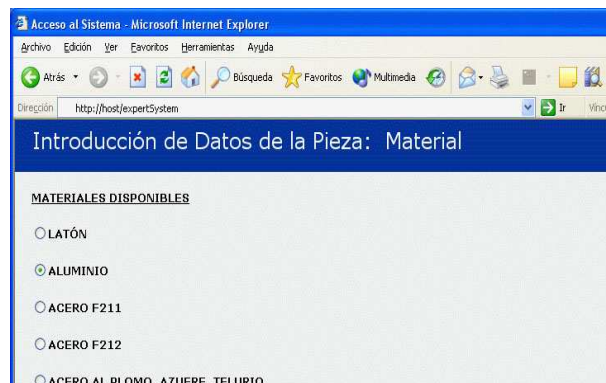


Figura 7.6: Selección del material.

proceso de estimación.

El experto de la fábrica, a la hora de calcular los parámetros y tiempos de fabricación se basa en las características de cada material. Sin embargo, el experto utiliza parámetros idénticos para varios materiales debido a su similitud de comportamiento ante la mecanización. Son los llamados grupos de materiales.

La decisión del grupo de materiales es propia del experto. Por este motivo se ha representado este conocimiento en lugar de solicitarlo al agente externo a través de una situación interactiva.

Para representar los distintos grupos de materiales se creó una base de datos que contiene los distintos grupos de material que tiene identificado el experto, así como el código de los materiales que corresponden a cada grupo. Como se ha indicado, el criterio de agrupación de los materiales es su similitud de comportamiento en la mecanización.

La selección del grupo de material se lleva a cabo en una situación automática, es decir, que el agente externo no va intervenir y, de hecho, no tendrá conocimiento de este paso. La situación automática “Cargar grupo de material” tiene asociada una acción interna de conexión a una base de datos. Va a recibir como parámetros de entrada:

- el nombre de la base de datos a consultar,
- el nombre de la tabla de la que se obtiene la información,
- el nombre del campo que representa el código del material asociado a un grupo de materiales,
- y el código del material cuyo grupo se desea conocer.

Como parámetros de salida devolverá un hecho con el código del grupo de material.

El hecho con el valor del grupo de material será empleado posteriormente en el proceso de estimación para la determinación de los parámetros de corte.

Tras la situación en la que determina el grupo de material al que pertenece el material con el que debe fabricarse la pieza, el experto a continuación necesita conocer la densidad del material, que también será necesaria a lo largo de la estimación. Por tanto solo existe una decisión en esta situación, que es la de pasar a la consulta de la densidad del material.

La situación “Cargar Densidad” es nuevamente una situación automática. La densidad del material es un conocimiento que se encuentra en una base de datos, por lo que no es necesario que esta información sea proporcionada por el agente externo. Nuevamente la acción asociada a esta situación es la librería de conexión a la base de datos. Los parámetros de entrada para esta situación van a ser:

- el nombre de la base de datos a consultar,
- el nombre de la tabla de la que se obtiene la información,
- el nombre del campo que representa un material,
- y el código del material cuyo grupo se desea conocer.

Como parámetros de salida, la consulta a la base de datos producirá un hecho con la densidad asociada al material. Esta densidad será empleada posteriormente en los parámetros de fabricación.

Una vez que se ha ejecutado la acción asociada a la situación automática, hay dos posibles decisiones que se pueden tomar. Si la máquina seleccionada era un torno automático multihusillo, la siguiente situación requerirá la obtención de los datos acerca de la geometría de la pieza que influyen en la fabricación con tornos automáticos multihusillo. Si la máquina seleccionada es un centro de mecanizado, la siguiente situación recogerá los parámetros de la geometría de la pieza necesarios para esta máquina. La separación en dos situaciones diferentes de la recogida de parámetros se debe a que éstos son diferentes en función de la máquina que se esté utilizando.

Dimensiones de la pieza para el torno

En esta situación, en la que se conoce la máquina a utilizar y el material necesario para fabricar la pieza, el experto necesita la información relativa al bloque de material que se va a utilizar para fabricar la pieza. En el caso de los tornos automáticos multihusillo, este bloque puede ser una barra de forma redonda o hexagonal. Además de conocer la forma de la barra

Figura 7.7: Forma y dimensiones de la pieza.

de material, el experto necesita conocer la longitud de la barra y el diámetro de la misma para, posteriormente, determinar las pinzas que pueden ser utilizadas para su sujeción. Además el diámetro de la pieza siempre será igual o inferior al diámetro de la barra de la que se parte.

La información acerca de la barra de metal (forma, longitud y diámetro), será proporcionada por el agente externo. Por tanto, se representa por medio de una situación interactiva, a través de una acción externa. Esta acción externa será una página web con un formulario en el que el agente externo seleccionará la forma de la barra de metal, e introducirá su longitud y diámetro. Estos hechos se enviarán al motor de decisión y se almacenarán en la memoria de trabajo para posteriormente determinar los parámetros de fabricación (figura 7.7).

En esta situación, tras recibir los hechos de la acción externa, existen tres posibles decisiones a tomar. Estas decisiones están condicionadas por el tipo de torno que se ha seleccionado anteriormente:

1. Si la barra es redonda y el torno es Russo, la situación siguiente buscará las pinzas de sujeción para piezas redondas del torno Russo.
2. Si la barra es hexagonal y el torno es Russo, la situación siguiente buscará las pinzas de sujeción para piezas hexagonales del torno Russo.
3. Si la barra es redonda y el torno es Plitter, la situación siguiente buscará las pinzas de sujeción para piezas redondas del torno Plitter.
4. Si la barra es hexagonal y el torno es Plitter, la situación siguiente buscará las pinzas de sujeción para piezas hexagonales del torno Plitter.

5. Si la barra es redonda y el torno es Conomatic, la situación siguiente buscará las pinzas de sujeción para piezas redondas del torno Conomatic.
6. Si la barra es hexagonal y el torno es Conomatic, la situación siguiente buscará las pinzas de sujeción para piezas hexagonales del torno Conomatic.

En la implementación inicial se crearon situaciones distintas para recuperar la información de cada tipo de barra y torno utilizado. Sin embargo dada la flexibilidad en la definición de parámetros se sustituyeron por una única situación. Para ello cada una de las anteriores decisiones, en caso de cumplirse, asigna a un hecho un código que identifica la tabla de una base de datos de donde se recuperan los códigos de las piezas.

Selección de pinzas de sujeción para un torno

El proceso de mecanización utilizando tornos automáticos multihusillo requieren la utilización de pinzas. La pinza a emplear para la fabricación va a depender del torno a utilizar y de las dimensiones de la pieza. Las pinzas disponibles para cada uno de los tornos se encuentra en una tabla diferente de una base de datos. El nombre de dicha tabla a consultar ha sido asignado a un hecho durante la evaluación e las decisiones explicadas en el apartado anterior.

Estas situaciones automáticas van a recibir los siguientes parámetros de entrada:

- el nombre de la base de datos a consultar,
- el nombre de la tabla de la que se obtiene la información, que se encuentra almacenado en un hecho.
- el nombre del campo que almacena el tamaño de la pinza,

Los parámetros de salida son:

- el nombre del campo que almacena el tamaño de la pieza, junto con el nombre del hecho en el que lo almacenará el motor de solución.

El resultado de la llamada va a ser un hecho multivaluado que va a almacenar todos los diámetros de las pinzas disponibles para el torno y forma de barra seleccionados.

Tras recibir el hecho con el conjunto de los diámetros de las pinzas disponibles, la decisión a tomar consisten en comprobar si el diámetro de la barra a emplear está contenido en la lista de diámetros disponibles:

- Si el diámetro de la barra está contenido en la lista de diámetros disponibles, el proceso puede continuar con el siguiente paso que será “Determinar datos de operación para el torno”.
- Si no está contenido, la siguiente situación será “Ajustar pinzas disponibles para el torno”

Ajustar pinzas disponibles para el torno

Esta situación se produce cuando, trabajando con tornos automáticos multihusillo, no se ha encontrado una pieza del mismo tamaño que la barra de material que se va a utilizar en el proceso de fabricación. En este punto el experto valora tres opciones posibles:

1. Volver hacia atrás y seleccionar un nuevo diámetro para la barra de material.
2. Continuar con la estimación como si realmente hubiera una pinza del tamaño adecuado.
3. Utilizar una barra de tamaño mayor. En este caso se necesitará una operación adicional de torneado, para ajustar al diámetro original.

La opción dos es útil para el caso de clientes importantes, en los que la consecución del pedido haría factible la inversión en una nueva pinza para el torno. De esta forma se puede continuar con la estimación como si la pinza existiera, quedando en manos de la fábrica comprarla posteriormente.

La opción tres es útil cuando el diámetro de la barra es especial, por ejemplo 10.5 mm y el diámetro de la pinza disponible es ligeramente superior (11 mm). Posteriormente, para corregir el diámetro de la barra sería necesaria una operación adicional de torneado.

Por tanto, se trata de una situación interactiva, que va a tener una acción externa asociada. Esta acción externa será una página HTML que contendrá un formulario en la que el agente externo podrá seleccionar entre dos opciones: continuar como si existiera la pinza o utilizar una barra con el tamaño de pinza inmediatamente superior. La opción de volver atrás y seleccionar un diámetro diferente está implícita en el control del proceso del motor de solución.

Las opciones a tomar dependen del valor seleccionado en la acción externa:

- Si la opción seleccionada es continuar como si la pinza estuviera disponible. Una verificación informará al agente externo por medio de un mensaje, sin detener el proceso, de la conveniencia de incluir la nueva pinza en la base de datos de pinzas. La siguiente situación será “Determinar datos de operación de un torno”.

- Si la opción seleccionada es utilizar una barra de tamaño superior, la siguiente situación será “Seleccionar diámetro máximo”.

Seleccionar diámetro máximo

Si la decisión consiste en utilizar una barra de diámetro mayor al necesario por la pieza, se deberá hacer una operación de torneado adicional. En el proceso habitual, el experto comprueba que existe una pinza de tamaño superior.

Esta comprobación se va a llevar a cabo por medio de dos situaciones automáticas que van a representar la forma de proceder del experto.

La primera situación automática, llamada “Cargar diámetro máximo”, requiere la obtención del diámetro de la pinza más grande para el torno con el que se está trabajando. Este valor es necesario para comprobar que el diámetro de la barra no es superior al máximo tamaño de pinza disponible. Para obtener este valor, se realiza una llamada a la acción interna para consultar la base de datos.

Los parámetros de entrada son:

- el nombre de la base de datos a consultar,
- el nombre de la tabla de la que se obtiene la información, determinada en la situación relativa a las dimensiones de la pieza.
- la función de SQL que permite la recuperación del valor máximo de un campo, en este caso $\max(\text{Diámetro})$.

Como parámetros de salida:

- el nombre del campo en el que se recupera el diámetro máximo, junto con el nombre del hecho en el que lo almacenará el motor de solución.

Cuando se ha obtenido el máximo diámetro de pinza disponible se pueden tomar dos decisiones posibles:

1. Si el máximo diámetro de la pinza disponible es inferior al diámetro de la barra que se va a utilizar en la fabricación, no es posible utilizar un diámetro mayor. Por tanto la siguiente situación debe advertir de este problema, es la situación “Error de disponibilidad de pinzas”.

2. Si el diámetro máximo es superior al diámetro de la barra, sí va a existir una pinza de tamaño superior que se pueda utilizar para mecanizar la pieza. En ese caso la siguiente situación recuperará el tamaño de pinza inmediatamente superior, es la situación “Cargar diámetro superior de pinza”.

La situación “Error de disponibilidad” es una situación interactiva en la se informa al agente externo que no hay pinzas de tamaño superior disponibles. En este caso la acción externa es una página HTML con un mensaje informativo, por lo que no implica hechos adicionales. Solo tiene una decisión asociada, que implica el regreso en todo caso a la situación “Ajustar tamaño pinzas disponibles para un torno” para que el agente externo pueda seleccionar otra opción de ajuste.

La situación “Cargar diámetro superior de pinza” representa la selección del tamaño de pinza inmediatamente superior al tamaño del diámetro de la barra seleccionada. Esta situación va a ser automática, y va a tener asociada una acción interna para consultar la base de datos con los tamaños de pinza disponible. Los parámetros de entrada son:

- el nombre de la base de datos a consultar,
- el nombre de la tabla de la que se obtiene la información,
- función SQL para recuperar el valor mínimo de un campo: $\min(\text{diámetro})$,
- una condición SQL para recuperar los tamaños de pinza superiores al diámetro de la barra: $\text{diametro} > \text{var.T_DIAMETRO}$.

Como parámetros de salida:

- el nombre del campo en el que se recupera el diámetro mínimo, junto con el nombre del hecho en el que lo almacenará el motor de solución.

Tras obtener el tamaño de pinza inmediatamente superior el experto solo define una posible decisión que llevará a la situación en la que se definen los datos de las operaciones de mecanización a realizar (“Selección de operaciones y parámetros para un torno”).

Selección de operaciones y parámetros para un torno

A esta situación se llega cuando ya se ha determinado el material a utilizar y el tamaño tanto de la barra como de la pinza de sujeción para el torno automático multihusillo. En esta situación,

The screenshot shows a web form titled "Datos de Operación" (Operation Data). The form is designed to collect information about the operations required for a lathe. It includes a header section with the title, followed by a question "¿QUE OPERACIONES NECESITA?" (Which operations do you need?). Below this, there are three rows of radio buttons for selecting operations: "TALADRADO:" (Drilling) with "SI" (Yes) and "NO" (No) options; "TORNEADO:" (Turning) with "SI" and "NO" options; and "CORTE:" (Cutting) with "SI" and "NO" options. Below the radio buttons, there are five input fields for numerical values, each followed by "mm": "NÚMERO DE HERRAMIENTAS A UTILIZAR" (Number of tools to use), "DIÁMETRO DEL TALADRO" (Drill diameter), "LONGITUD A TALADRAR" (Drilling length), "LONGITUD A TORNEAR" (Turning length), and "LONGITUD A CORTAR *" (Cutting length). At the bottom of the form, there is a note: "(si no existe alguno de los parámetros, introducir el valor 0 en la casilla correspondiente)" (if none of the parameters exist, enter the value 0 in the corresponding box).

Figura 7.8: Situación interactiva: parámetros de operación para un torno.

el experto determina cuáles son las operaciones que es necesario llevar a cabo y los parámetros necesarios para cada una de ellas.

Esta información es determinada por un empleado de la fábrica, por lo que será el agente externo el que proporcione esta información. Así pues, se trata de una situación interactiva que ha recibido el nombre de “Selección de operaciones y parámetros”. La acción externa asociada a esta situación interactiva será una página HTML en la que se solicitará la siguiente información (ver figura 7.8):

- Es necesario taladrar.
- Es necesario tornear.
- Es necesario cortar.
- Número de herramientas que se van a utilizar.
- Longitud y diámetro del taladro, si es necesario.
- Longitud a tornear, si es necesario.
- Longitud a cortar, si es necesario.

Estos parámetros son almacenados en hechos, que se van a utilizar para determinar los parámetros de fabricación y el tiempo de proceso, que influirá directamente en el coste.

En esta situación, tras la ejecución de la acción interna, el experto solo considera una decisión a tomar, y es el cambio a la situación en la que se recuperan los parámetros básicos de corte para realizar cálculos posteriores.

Recuperar parámetros de corte

En esta situación, el experto determina los parámetros de corte para la operación que se va a realizar. Los parámetros de corte necesarios son el régimen de giro y el avance, y están relacionados con el tipo de material y el diámetro de la barra que se están utilizando. Este conocimiento está representado en una base de datos, en la que existe una tabla por cada tipo de material que distingue el experto. Cada tipo de material tiene unos parámetros de corte diferentes en función del diámetro. Los parámetros representados en la base de datos son para el torno Ruso, por lo que en caso de utilizar un torno diferente deben ser adaptados, tal y como se muestra en las decisiones asociadas a esta situación.

Como el conocimiento sobre los parámetros de corte está en una base de datos, la situación “Cargar parámetros de corte” es una situación automática, en la que la acción interna es la librería de consulta a la base de datos.

Los parámetros de entrada para esta situación son:

- el nombre de la base de datos a consultar,
- el nombre de la tabla de la que se obtiene la información. Como el nombre de la tabla varía en función del grupo de materiales al que pertenece el material que se está utilizando en la fabricación, el nombre de la tabla se ha asignado previamente en el hecho T_CODIGO_MAT_OPERACION,
- el nombre el nombre de los campos que se quieren recuperar: Avance y Revoluciones,
- La condición SQL que delimita el diámetro de la barra de material que se va a emplear en la fabricación “(VAR.T_DIAMETRO < *DiametroSuperior*)and(VAR.T_DIAMETRO ≥ *DiametroInferior*)”

Parámetros de salida:

- Nombre del hecho en el que se carga el valor del avance: T_AVANCE_CORTE.

- Nombre del hecho en el que se carga el valor de las revoluciones T_NUM_REV_CORTE.

Las decisiones que se pueden tomar en esta situación son:

- Si el torno es Ruso pasa a la situación virtual “P_FIN_REVOLUCIONES_CORTE”.
- Si el torno es Conomatic, pasa a una secuencia de situaciones automáticas que van a adaptar las revoluciones del torno Ruso recuperadas al torno Conomatic, consultando la base de datos. Tras estas situaciones automáticas, lleva a la situación virtual “P_FIN_REVOLUCIONES_CORTE”.
- Si el torno es Plitter, pasa a una secuencia de situaciones automáticas que van a adaptar las revoluciones del torno Ruso recuperadas al torno Plitter, consultando la base de datos. Tras estas situaciones automáticas, lleva a la situación virtual “P_FIN_REVOLUCIONES_CORTE”.

Fin de revoluciones de corte

Se trata de una situación virtual, que sirve de punto de unión del proceso tras las alternativas que se han producido para adaptar las revoluciones del torno Ruso a los tornos Conomatic y Plitter.

Las situaciones virtuales no tienen acción asociada, por lo que el proceso continúa evaluando las decisiones correspondientes. Esta situación virtual evalúa dos posibles decisiones:

- Si es necesaria una operación de corte, se pasa a la situación para calcular el tiempo de corte.
- Si no es necesaria la operación de corte, pero es necesaria la operación de torneado, pasa a la situación “Cargar Parámetros Torneado”
- Si no es necesaria la operación de corte, pero es necesaria la operación de taladrado, pasa la situación “Cargar revoluciones taladrado”.
- Si no es necesaria ninguna de las operaciones pasa a la situación de “Parámetros de Operación Recomendados”

Cargar parámetros de corte

Si se requiere la operación de corte, es necesario calcular el tiempo necesario para la operación en función de la longitud del corte y los parámetros revoluciones y avance obtenidos anteriormente.

Tras la entrevista con el experto, se determinó que la ecuación para calcular el tiempo de corte es la siguiente:

$$T_{cutting} = \left[\frac{(L_{cutting} + 7)60}{na} + 2,5 \right] F_{correction} \quad (7.1)$$

La longitud de corte ($L_{cutting}$) es conocida tras la situación “Selección de operaciones y parámetros para un torno”. A este valor se le añaden 7mm debido a que la herramienta comienza a trabajar un poco antes de tocar la pieza. A continuación, se multiplica por 60 para obtener el tiempo en segundos y se suman 2.5 segundos que es el tiempo estimado para que la pieza salga de la máquina. El término $F_{correction}$ es un factor de corrección cuyo valor estimado es 1.3 para considerar tiempos adicionales de la operación. El valor n son las revoluciones del torno y el valor a es el avance de la máquina que se han obtenido en situaciones anteriores.

Este conocimiento se ha representado en una hoja de cálculo, en la que se han habilitado celdas para introducir los tres parámetros necesarios, y una cuarta celda en la que se implementa la fórmula anterior, de forma que se pueda recuperar el resultado.

Para acceder a este conocimiento, esta situación debe ser automática, y la acción interna va a ser una llamada a la librería de consulta de hojas Excel. Los parámetros de entrada son:

- Hoja de cálculo donde están las formulas.
- Celda donde se debe introducir la longitud de corte y el valor de dicha longitud.
- Celda donde se debe introducir las revoluciones a las que girará el torno, y valor de dichas revoluciones.
- Celda donde se debe introducir el régimen de avance del torno, y el valor de dicho régimen.

Los parámetros de salida son:

- Hecho en el que se almacenará el tiempo de corte estimado (T_TIEMPO_CORTE), junto con la celda de la que se recuperará este valor.
- Hecho en el que se almacenará el tiempo de corte ajustado al valor de corrección (T_TIEMPO_CICLO_CORTE), junto con la celda de la que se recuperará este valor.

Las decisiones relacionadas con esta situación automática son las siguientes:

- Si es necesaria la operación de torneado, pasa a la situación “Cargar Parámetros Torneado”

- Si no es necesaria la operación de torneado, pero es necesaria la operación de taladrado, pasa la situación “Cargar revoluciones taladrado”.
- Si no es necesaria ninguna de las operaciones pasa a la situación de “Parámetros de Operación Recomendados”

Cargar parámetros torneado

Si se requiere el torneado, es necesario calcular el tiempo de torneado que será necesario. Tras consultar con el experto, la fórmula para obtener este tiempo es la siguiente:

$$T_{turning} = \left[\frac{(L_{turning} + 5)60}{na} + 2,5 \right] F_{correction} \quad (7.2)$$

Se puede observar que es análoga a la expresión para el tiempo de corte (ecuación 7.1), con la diferencia de que la longitud extra es ligeramente inferior (5 mm) por las características de la herramienta. El resto de factores tienen un significado similar al de la ecuación 7.1.

La forma de representar este conocimiento es análogo al tiempo de corte, es decir, por medio de una situación automática y una acción interna de acceso a la hoja de cálculo donde se representa la fórmula y se obtiene el resultado.

Las decisiones que se pueden tomar en esta situación son:

- Si es necesaria la operación de taladrado, pasa a la situación “Cargar parámetros taladrado”.
- Si no es necesaria la operación de taladrado, pasa a la situación “Parámetros de Operación Recomendados”.

Cargar parámetros taladrado

Si se requiere la operación de taladrado, el cálculo se realiza de forma análoga las operaciones anteriores. En este caso es necesario recuperar previamente al cálculo, las revoluciones recomendadas en función del torno a utilizar. Esto se realiza a través de una consulta a una base de datos de parámetros, que, para simplificar el proceso, se omite su detalle.

La fórmula para el cálculo del tiempo de taladro es representada nuevamente en una hoja de cálculo es la siguiente.

$$T_{drilling} = \left[\frac{(L_{drilling} + 0,3D_{drilling} + 5)60}{na} + 2,5 \right] F_{correction} \quad (7.3)$$

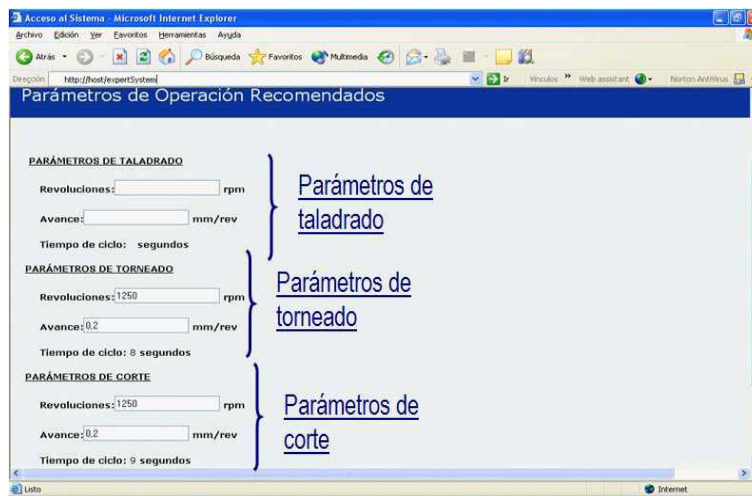


Figura 7.9: Parámetros recomendados.

En la expresión anterior, $L_{drilling}$ es la longitud del taladro a realizar, $D_{drilling}$ es el diámetro del taladro, n son las revoluciones recomendadas para el torno que se está empleando y el factor a es el factor de avance. El resto de parámetros es análogo a las ecuaciones 7.1 y 7.3.

Los parámetros de llamada la hoja de cálculo para la acción interna son análogos a las llamadas para calcular el tiempo de corte y el tiempo de torneado.

Dado que los cálculos de tiempo se han representado de forma secuencial, tras calcular el tiempo de taladrado ya se han determinado los tiempos y parámetros de las tres operaciones principales (si aplican). Por tanto solo hay una decisión posible en esta situación, que es pasar a la situación “Parámetros de Operación Recomendados”

Parámetros de operación recomendados

Esta situación no representa un paso que puede dar el experto. La justificación de esta situación proviene del hecho de que el tipo de proceso que se ha decidido representar es el de un asistente, en el que el motor de solución pregunta al agente externo y le va proponiendo resultados.

En esta situación se van a presentar al agente externo los parámetros y tiempos que acaba de calcular para las tres operaciones principales (ver figura 7.9), de forma que éste pueda comprobar su corrección y modificarlos en caso de que, basándose en su experiencia, lo considere conveniente.

Por tanto se trata de una situación interactiva. La acción externa asociada es una página HTML en la que se va a presentar dentro de un formulario los parámetros que se han calculado

en las operaciones descritas anteriormente. El agente externo va poder modificar estos valores y enviarlos al motor de solución.

Tras recibir los hechos, solo hay una posible decisión a tomar en cualquier caso, y es pasar a la situación “Ajustar Revoluciones”

Ajustar revoluciones

Tras los cálculos realizados en las operaciones anteriores se han determinado los regímenes de giro para las tres operaciones principales. Sin embargo, el torno solo puede tener un régimen de giro, al tiempo, por lo que las revoluciones a las que funcionará el torno estarán determinadas por la operación que requiera un menor número de revoluciones por minuto. Esto además, va a influir en el tiempo de operación.

Por tanto, en esta situación se calcula el régimen mínimo de revoluciones entre los asignados a las operaciones de taladrado, torneado y corte, y se calcula de nuevo el tiempo de operación de acuerdo con el nuevo régimen de giro.

Esta operación se realiza por medio de tres llamadas consecutivas a la hoja de cálculo que se empleó para el cálculo de los tiempos.

Las decisiones a tomar en esta secuencia de llamadas se basan en determinar cuál de las operaciones a realizar es la más lenta, pues este tiempo determinará el tiempo de ciclo de la máquina. El tiempo de ciclo en segundos es el tiempo de la operación más lenta.

El cálculo del tiempo de ciclo se realiza mediante una acción interna con una llamada a una hoja de cálculo que representa la siguiente fórmula.

$$T_h = \frac{T_s}{3600} \quad (7.4)$$

Donde T_s es el tiempo para mecanizar la pieza, expresado en segundos, y T_h es el mismo tiempo expresado en horas. Esta transformación se debe a que la estimación del experto se basa en el coste por horas.

El número de piezas por hora se representa también en la hoja de cálculo, por medio de la siguiente fórmula:

$$production = \frac{1}{T_h} \quad (7.5)$$

Estos cálculos se secuencian mediante decisiones simples que enlazan una situación interna con otra, hasta que se han obtenido los resultados necesarios, de la misma forma que el experto realiza los cálculos para su estimación.

Finalmente la decisión a tomar tras el último cálculo se basa en el hecho de que el proceso automatizado se comportará como un asistente para un empleado. La siguiente situación una vez calculados los tiempos de ciclo y es la determinación de las operaciones adicionales que deban ser realizadas.

Operaciones adicionales

Una vez se han determinado los parámetros de las operaciones principales que afectan al tiempo y el coste de la pieza a fabricarla, es necesario considerar otras operaciones adicionales que puede ser necesario a lo largo del proceso de fabricación. Las operaciones más relevantes que se pueden llevar a cabo son:

- Eliminación de pivotes.
- Avellanado.
- Rebarbado.
- Chaflanado.

En su proceso de estimación, el experto determina cuál o cuáles de estas operaciones van a ser necesarias en el proceso de mecanización, y calcula el tiempo de operación y preparación para cada una de ellas. Estos cálculos son muy subjetivos y difíciles de realizar, puesto que dependen en gran medida de la intervención de un operario para realizar las operaciones.

Las operaciones de avellanado, rebarbado y chaflanado se tratan como una única operación pues tienen características similares.

En primer lugar es necesario determinar las operaciones adicionales que se deben realizar. Esta información va a ser proporcionada por el agente externo que ha recibido el pedido, por lo que se ha representado por medio de una situación interactiva. Esta situación interactiva tiene asociada una acción externa en forma de página HTML, con un formulario en el que el agente externo seleccionará las operaciones necesarias (figura 7.10).

Las opciones disponibles son:

- Eliminación de pivotes.
- Avellanado, rebarbado y chaflanado. Se han agrupado por tratarse de operaciones similares.

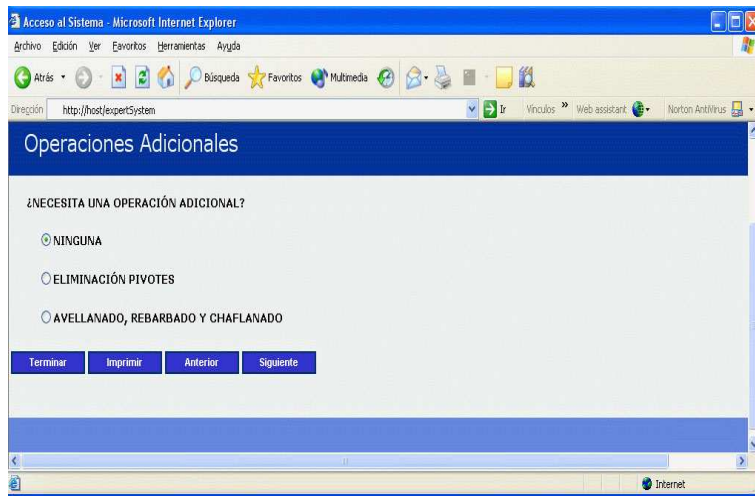
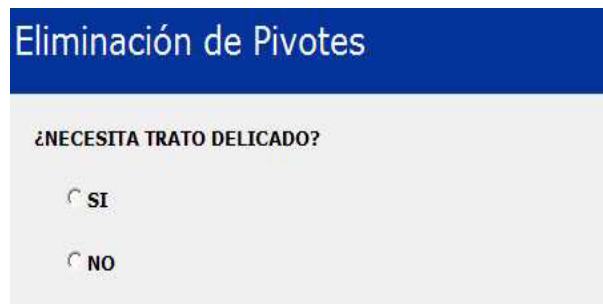


Figura 7.10: Selección de operaciones adicionales.

- No realizar ninguna operación adicional.

Cuando el agente externo ha seleccionado una opción, las decisiones van a representar el siguiente paso a dar en el proceso de estimación.

- Si es necesaria la eliminación de pivotes, la siguiente situación del proceso llevará a la determinación de los parámetros para la eliminación de pivotes.
- Si es necesario el avellanado, chaflanado o rebarbado y el tamaño del lote es superior a 5000 piezas, la operación se realizará por medio de un taladro manual. Se asigna a la máquina a utilizar para la operación de avellanado-chaflanado-rebarbado (hecho T_MAQUINA_SEG_OPER_A) el valor “TALADRO MANUAL”, se establece el tiempo de preparación de la operación de avellanado (hecho T_TP2_A) el valor 0.25 horas y la siguiente situación será “Parámetros de Avellanado”.
- Si es necesario el avellanado, chaflanado o rebarbado y el tamaño del lote es inferior a 5000 piezas la operación se realizará con una máquina especial. La siguiente situación será “Parámetros de Chaflanado”.
- Si no es necesaria ninguna operación más, la siguiente situación del proceso llevará a calcular los tiempos totales de las operaciones adicionales.

El formulario tiene un encabezado azul con el título "Eliminación de Pivotes". Debajo, en un fondo gris claro, se encuentra la pregunta "¿NECESITA TRATO DELICADO?". Hay dos opciones de radio: "SI" y "NO".

Eliminación de Pivotes

¿NECESITA TRATO DELICADO?

SI

NO

Figura 7.11: Parámetros de eliminación de pivotes.

Parámetros para la eliminación de pivotes

En esta situación se determinan los parámetros necesarios para la operación de eliminación de pivotes. Esta operación se puede llevar a cabo con una piedra o con una lija mecánicas.

Según el conocimiento del experto de la fábrica, el tiempo de preparación se puede considerar nulo, mientras que el tiempo de trabajo (incluyendo los periodos improductivos) va a depender del diámetro de la pieza y del tratamiento que necesite: si debe tener un tratamiento delicado o no determinará más o menos tiempo. Tratamiento delicado significa que la pieza no puede ser lanzada a un contenedor una vez terminado su procesamiento, sino que debe ser depositada manualmente para no dañarla.

Los datos relativos al diámetro de la pieza son conocidos, por lo que en esta situación es necesario conocer si el tratamiento de la pieza debe ser delicado o no. Este dato es conocido por el agente que ha recibido el pedido, por lo que esta situación se representa por medio de una situación interactiva.

La acción asociada a esta situación interactiva es una página HTML con un formulario en el que el agente externo seleccionará si el tratamiento debe ser delicado o no (figura 7.11)

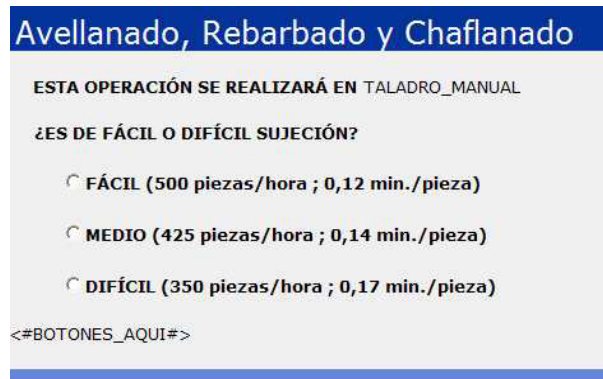
A partir de la selección del tipo de tratamiento, las decisiones que el experto toma son:

- Si el diámetro es menor que 16 mm, y el tratamiento es delicado, asignar al hecho correspondiente a la máquina a utilizar para la eliminación de pivotes (T_MAQUINA_SEG_OPER_P) el valor LIJA, y al hecho que almacena el número de unidades procesadas por hora para eliminación de pivotes (T_PRODUCION2_P) el valor 600. La siguiente situación es "Operaciones Adicionales" de nuevo para permitir la selección de una nueva operación adicional.
- Si el diámetro es menor que 16 mm, y el tratamiento es delicado, asignar al hecho correspondiente a la máquina a utilizar para la eliminación de pivotes (T_MAQUINA_SEG_OPER_P)

el valor LIJA, y al hecho que almacena el número de unidades procesadas por hora para eliminación de pivotes (T_PRODUCCION2_P) el valor 800. La siguiente situación es “Operaciones Adicionales” de nuevo para permitir la selección de una nueva operación adicional.

- Si el diámetro es mayor que 16 mm, pero inferior a 25 mm, y el tratamiento es delicado, asignar al hecho correspondiente a la máquina a utilizar para la eliminación de pivotes (T_MAQUINA_SEG_OPER_P) el valor PIEDRA, y al hecho que almacena el número de unidades procesadas por hora para eliminación de pivotes (T_PRODUCCION2_P) el valor 400. La siguiente situación es “Operaciones Adicionales” de nuevo para permitir la selección de una nueva operación adicional.
- Si el diámetro es mayor que 16 mm, pero inferior a 25 mm, y el tratamiento no es delicado, asignar al hecho correspondiente a la máquina a utilizar para la eliminación de pivotes (T_MAQUINA_SEG_OPER_P) el valor PIEDRA, y al hecho que almacena el número de unidades procesadas por hora para eliminación de pivotes (T_PRODUCCION2_P) el valor 500. La siguiente situación es “Operaciones Adicionales” de nuevo para permitir la selección de una nueva operación adicional.
- Si el diámetro es mayor que 25 mm, y el tratamiento es delicado, asignar al hecho correspondiente a la máquina a utilizar para la eliminación de pivotes (T_MAQUINA_SEG_OPER_P) el valor PIEDRA, y al hecho que almacena el número de unidades procesadas por hora para eliminación de pivotes (T_PRODUCCION2_P) el valor 300. La siguiente situación es “Operaciones Adicionales” de nuevo para permitir la selección de una nueva operación adicional.
- Si el diámetro es mayor que 25 mm, y el tratamiento no es delicado, asignar al hecho correspondiente a la máquina a utilizar para la eliminación de pivotes (T_MAQUINA_SEG_OPER_P) el valor PIEDRA, y al hecho que almacena el número de unidades procesadas por hora para eliminación de pivotes (T_PRODUCCION2_P) el valor 400. La siguiente situación es “Operaciones Adicionales” de nuevo para permitir la selección de una nueva operación adicional.

La decisión de la máquina a utilizar y el número de piezas por hora está basado en la experiencia del técnico de la fábrica.



Avellanado, Rebarbado y Chaflanado

ESTA OPERACIÓN SE REALIZARÁ EN TALADRO_MANUAL

¿ES DE FÁCIL O DIFÍCIL SUJECIÓN?

FÁCIL (500 piezas/hora ; 0,12 min./pieza)

MEDIO (425 piezas/hora ; 0,14 min./pieza)

DIFÍCIL (350 piezas/hora ; 0,17 min./pieza)

<#BOTONES_AQUI#>

Figura 7.12: *Parámetros de avellanado.*

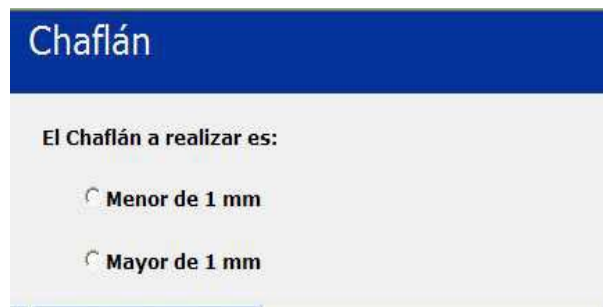
Parámetros de avellanado

En esta situación el experto necesita conocer la dificultad de sujeción de la pieza a mecanizar para determinar el número de piezas por hora que se pueden procesar, pues el avellanado se va a realizar por medio de un taladro manual.

La información acerca de la dificultad de sujeción la tiene que proporcionar el agente externo, por lo que esta situación va a ser interactiva. La acción externa asociada a la situación será una página HTML con un formulario en el que se va a seleccionar el grado de dificultad de sujeción de la pieza: fácil (500 piezas hora), medio (425 piezas/hora) o difícil (350 piezas/hora). La figura 7.12 muestra la página descrita.

Cuando ya se conoce la dificultad de sujeción, las decisiones que se pueden tomar son:

- Si la dificultad de sujeción es fácil, se asigna al hecho que determinará el número de piezas que se pueden avellanar a la hora (T_PRODUCCION2_A) el valor 500, y la siguiente situación será nuevamente la selección de operaciones adicionales, por si fuera necesario realizar otra operación más.
- Si la dificultad de sujeción es medio, se asigna al hecho que determinará el número de piezas que se pueden avellanar a la hora (T_PRODUCCION2_A) el valor 425, y la siguiente situación será nuevamente la selección de operaciones adicionales, por si fuera necesario realizar otra operación más.
- Si la dificultad de sujeción es difícil, se asigna al hecho que determinará el número de piezas que se pueden avellanar a la hora (T_PRODUCCION2_A) el valor 350, y la siguiente situación será nuevamente la selección de operaciones adicionales, por si fuera necesario realizar otra operación más.



The image shows a web form with a blue header containing the word 'Chaflán'. Below the header, the text 'El Chaflán a realizar es:' is followed by two radio button options: 'Menor de 1 mm' and 'Mayor de 1 mm'.

Figura 7.13: *Parámetros de chaflanado.*

El número de piezas que se pueden avellanar a la hora es necesario para determinar el tiempo total del proceso.

Parámetros de chaflanado

Cuando la operación implica el chaflanado, para determinar el tiempo de proceso es necesario conocer si el diámetro del chaflán a realizar es mayor o menor de 1 mm de profundidad. Según el experto de la fábrica, esto va a determinar la máquina con la que se va a realizar la operación.

La profundidad del chaflán es determinada por el agente externo, por lo que esta situación se modela como una situación interactiva. La acción externa es una página HTML con un formulario en la que el agente seleccionará entre dos opciones: tamaño superior a 1 mm. o inferior a 1 mm (figura 7.13).

Las decisiones que el experto toma en esta situación son:

- Si la profundidad del chaflán es inferior a 1 mm., se asigna al hecho que determinará el número de piezas que se pueden chaflanar a la hora (T_PRODUCCION2_A) el valor 1000, al hecho que indica el tiempo de preparación de la operación de chaflanado (hecho T_TP2_A, nótese que es el mismo que para el avellanado, pues se realiza una u otra en función del tamaño del lote) el valor 0.5 horas y al hecho que recoge la máquina que se va utilizar en la operación adicional (T_MAQUINA_SEG_OPER_A) el valor “Máquina Transfer” (la máquina que se va a emplear en la operación. Finalmente la siguiente situación será nuevamente la selección de operaciones adicionales, por si fuera necesario realizar otra operación más.
- Si la la profundidad del chaflán es superior a 1 mm., se asigna al hecho que determinará el número de piezas que se pueden chaflanar a la hora (T_PRODUCCION2_A) el

valor 200, al hecho que indica el tiempo de preparación de la operación de chaflanado (hecho T_TP2_A) el valor 0.75 horas (pues la máquina requiere más tiempo que en el caso de la decisión anterior) y al hecho que recoge la máquina que se va utilizar en la operación adicional (T_MAQUINA_SEG_OPER_A) el valor “Torno paralelo” (la máquina que se va a emplear en la operación. Finalmente la siguiente situación será nuevamente la selección de operaciones adicionales, por si fuera necesario realizar otra operación más.

Con los valores asignados a los hechos en las decisiones anteriores, el motor de solución podrá calcular posteriormente los tiempos totales de proceso y preparación de las operaciones adicionales.

Tiempo total de operaciones adicionales

Cuando ya conoce el tiempo total de cada una de las operaciones adicionales (eliminación de pivotes y avellanado-chaflanado-rebarbado), el experto calcula el tiempo total de estas operaciones adicionales para continuar con la estimación total del coste de fabricación.

El experto calcula el tiempo total de las operaciones adicionales por medio de la ecuación (7.6):

$$T_{extraoperation} = T_{pivotscleaning} + T_{csinking} \quad (7.6)$$

Donde $T_{extraoperation}$ es el tiempo total de las operaciones adicionales, $T_{pivotscleaning}$ es el tiempo necesario para la eliminación de pivotes de una pieza y $T_{csinking}$ es el tiempo de avellanado para una pieza, todos calculados en horas. En caso de que alguna de las operaciones adicionales no sea necesaria, tomará el valor 0.

Además, calcula el tiempo de preparación para las operaciones adicionales mediante la ecuación (7.7):

$$T_{extrapreparation} = T_{pivotsprep} + T_{csinkingprep} \quad (7.7)$$

Donde $T_{extrapreparation}$ es el tiempo total de preparación para las operaciones adicionales, $T_{pivotsprep}$ es el tiempo de preparación para la eliminación de pivotes y $T_{csinkingprep}$ es el tiempo de preparación para las operaciones de avellanado-chaflanado-rebarbado. Todos los tiempos están expresados en horas, y si alguna operación no es necesaria, toman el valor 0.

La realización de estos cálculos se representa por medio de una situación automática, llamada “Calcular tiempo operaciones adicionales”. Esta situación automática tiene una acción interna

que realiza una llamada a Excel para utilizar una hoja de cálculo que contiene las fórmulas indicadas.

Los parámetros de entrada son:

- Hoja de cálculo donde están las formulas.
- Celda donde se debe introducir el tiempo de eliminación de pivotes y el valor de dicho tiempo.
- Celda donde se debe introducir el tiempo de avellanado-chaflanado-rebarbado, y valor de dicho tiempo.
- Celda donde se debe introducir el tiempo de preparación de la operación de eliminación de pivotes, y el valor de dicho tiempo.
- Celda donde se debe introducir el tiempo de preparación de la operación de avellanado-chaflanado-rebarbado, y el valor de dicho tiempo.

Los parámetros de salida son:

- Hecho en el que se almacenará el tiempo total de las operaciones adicionales (T_TMO2), junto con la celda de la que se recuperará este valor.
- Hecho en el que se almacenará el tiempo total de preparación para las operaciones adicionales (T_TP2), junto con la celda de la que se recuperará este valor.

Tras el cálculo de los tiempos de las operaciones adicionales, el experto solo contempla una decisión, que consiste en calcular los pesos de las piezas y de cada lote para calcular los costes finales. Por tanto, la siguiente situación es “Calcular peso de las piezas”

Calcular peso de las piezas

Cuando ya se conocen las características del material a utilizar de las operaciones a realizar y de la pieza a fabricar, es necesario calcular el peso del material que se va a emplear para calcular el coste que implica para el proceso de fabricación.

El experto calcula el peso de forma diferente, en función de la forma de la barra de material que se emplea para la fabricación de la pieza. Esta barra puede ser redonda o hexagonal. En esta sección se explica como se ha representado este conocimiento de forma conjunta, pues la

diferencia entre los cálculos es un parámetro de una situación automática que se encarga de consultar la hoja de cálculo.

El peso bruto para barras redondas se calcula a partir de la ecuación (7.8):

$$W_{roundedpart} = \frac{\pi D^2}{4} L_{total} \frac{\rho}{10^6} \quad (7.8)$$

Donde $W_{roundedpart}$ es el peso total de la pieza redonda D es el diámetro de la pieza, L_{total} es la longitud de material necesario para fabricar la pieza y ρ es la densidad del material que se está utilizando. La densidad del material a utilizar se obtuvo al inicio del proceso. El peso total está expresado en Kilogramos (Kg).

La longitud de material para fabricar la pieza (L_{total}) se obtiene de la ecuación (7.9):

$$L_{total} = \max [(1,1L); (L + 6mm)] \quad (7.9)$$

El peso del material para las piezas de forma hexagonal se calcula por medio de la ecuación (7.10):

$$W_{hexagonalpart} = \frac{\sqrt{3}}{2} D^2 L_{total} \frac{\rho}{10^6} \quad (7.10)$$

Donde $W_{hexagonalpart}$ es el peso del material para la pieza de forma hexagonal, D representa la distancia entre los lados paralelos de la barra hexagonal, y ρ representa la densidad del material.

El peso neto de las piezas y el lote de piezas es necesario para conocer, por una parte, el coste del transporte de las piezas fabricadas y, por otro lado, para calcular el coste de la materia prima que se utilizará para fabricar la pieza.

El coste del transporte será determinado por el agente externo, mientras que el coste de la materia prima se calcula con la ecuación (7.11):

$$C_{Material} = W_{material} P_{material} \quad (7.11)$$

Donde $W_{material}$ es el peso del material necesario para la fabricación de la pieza (calculado con las ecuaciones anteriores), y $P_{material}$ es el precio por kilo de material.

El conocimiento relativo a estas fórmulas se representa en una hoja de cálculo. Para la realización de estas operaciones es necesaria una situación virtual que tenga asociada una acción interna para el acceso a la hoja de cálculo. Dado que son diferentes las operaciones, se han dividido las operaciones en varias situaciones automáticas, secuenciadas por medio de sus decisiones.

Los parámetros de entrada de las acciones internas son:

- Nombre de la hoja de cálculo a utilizar.
- Conjunto de los hechos necesarios para el cálculo del valor, junto con la celda de la hoja de cálculo donde debe introducirse.

Los parámetros de salida son:

- Conjunto de celdas cuyo valor es necesario recuperar y nombre del hecho en el que quedarán almacenados.

Tras obtener los resultados de estos cálculos, la decisión sugerida por el experto es pasar a realizar los cálculos finales, para lo cuál, en primer lugar debe obtener los datos referidos a los costes de mano de obra, adquisición de material y transporte.

Costes de mano de obra, material y transporte

Para poder calcular el coste de fabricación y, por extensión, el precio de la pieza que ha sido solicitada, el experto necesita conocer el coste por hora de la mano de obra, el precio del material necesario para la fabricación (expresado en precio por kg.) y el coste de portes y embalajes.

Esta información es conocida por la persona que está encargada de estimar el precio. En el coste de la mano de obra, la compañía incluye el margen de beneficio.

Por tanto, la captura de la información relativa a los costes necesarios para calcular el coste total del pedido, se va a representar por medio de una situación interactiva, asociada a una acción externa en forma de un mensaje HTML, que presentará un formulario para que el agente externo introduzca el coste por hora de la mano de obra, el precio por kg. del material, el coste de portes y embalajes por pieza y el coste de hasta dos servicios adicionales que deban ser tenidos en cuenta en la estimación del precio (figura 7.14).

Tras obtener los datos relativos a los costes a utilizar para calcular el precio final, el experto puede calcular el coste final de todos los componentes que afectan al precio de la pieza.

Cálculo de costes

Una vez que el experto ha estimado el tiempo de fabricación, las operaciones adicionales a realizar, los parámetros necesarios para fabricar la pieza, y conoce el coste de la mano de obra, de la materia prima, así como los costes adicionales, está en disposición de calcular el precio final de la pieza.

Con la información obtenida hasta el momento se puede determinar el tiempo efectivo de fabricación, es decir, lo que tarda en fabricarse la pieza desde que entra en el torno hasta que

Datos Complementarios

PRECIO DE LA MANO DE OBRA EUROS/HORA

PRECIO DEL MATERIAL EUROS/Kg.

COSTE DE PORTES Y EMBALAJES EUROS/PIEZA

Pesos para arandelas y casquillos sencillos:
(Peso Neto por pieza: #VAR.T_PESOPIEZANETO Kg.)
(Peso Neto del lote: #VAR.T_PESOLOTENETO Kg.)

COSTE DE OTROS SERVICIOS:

Servicio: , Coste de este servicio: EUROS/PIEZA

Servicio: , Coste de este servicio: EUROS/PIEZA

<#BOTONES_AQUI#>

Figura 7.14: Situación para obtener los costes unitarios.

sale terminada. Sin embargo, para calcular el coste total el experto también considera tanto el coste de la preparación de las máquinas que se van a utilizar, como el coste de preparación de las operaciones adicionales que sean necesarias. Con la ecuación (7.12) el experto calculará el tiempo de preparación del torno automático multihusillo que se empleará para la fabricación de la pieza:

$$T_{lathepreparation} = \max [((0,25N_{tools}) + 3); 6] \quad (7.12)$$

El tiempo de preparación del torno ($T_{lathepreparation}$) se determina a partir del número de herramientas que serán utilizadas durante el proceso de mecanización. Este es el tiempo que se tarda en preparar y ajustar las piezas al torno, y debe ser añadido al tiempo de mecanización propiamente dicho para obtener el tiempo final. El tiempo de preparación para cada una de las herramientas es de 0.25 horas, según la experiencia recogida en la fábrica. Este valor se multiplica por el número de herramientas que deben ajustarse (N_{tools}). A este valor se le añaden 3 horas, que es el tiempo que el experto considera necesario para montar y ajustar el torno. Como se puede apreciar en la fórmula, el tiempo mínimo que se determina es de 6 horas. Este es el tiempo mínimo de preparación de acuerdo con la experiencia del estimador.

Para calcular el coste total de la mano de obra por cada pieza, el experto multiplica el coste por hora de la mano de obra por el tiempo total de fabricación, sumando el tiempo de fabricación

efectivo más el tiempo de las operaciones adicionales:

$$C_{manpower} = (T_h + T_{extratotal})P_{mp} \quad (7.13)$$

Donde:

- T_h es el tiempo de fabricación en torno automático multihusillo, expresado en horas.
- $T_{extratotal}$ es el tiempo total de las operaciones adicionales necesarias para la fabricación (eliminación de pivotes o avellanado).
- P_{mp} es el coste por hora de la mano de obra.

Como se ha indicado anteriormente, además del coste por pieza de la mano de obra empleada para la fabricación efectiva de la pieza, el experto toma en consideración el coste por pieza de la mano de obra que se emplea en la preparación de los tornos. Este coste es calculado con la ecuación (7.14):

$$C_{preparation} = \frac{T_{lathepreparation} + T_{extrapreparation}}{N_{parts}} P_{mp} \quad (7.14)$$

Donde:

- $T_{lathepreparation}$ es el tiempo de preparación para el torno, expresado en horas.
- $T_{extrapreparation}$ es el tiempo de preparación para las operaciones adicionales, expresado en horas. Este coste se calcula en el bloque de situaciones relacionadas con las operaciones adicionales.
- N_{parts} Número de piezas que se tienen que mecanizar.

A partir de las ecuaciones anteriores, el experto calcula el coste total.

$$C_{total} = C_{manpower} + C_{material} + C_{preparation} + C_{additional} \quad (7.15)$$

Donde $C_{additional}$ está compuesto por $C_{transport}$ (coste de transporte) and $C_{additional}$ (coste de servicios) que han sido proporcionados en la situación “Costes de mano de obra, material y transporte”.

Con estos últimos cálculos, el experto ha conseguido determinar el precio final de la pieza para la que un cliente ha solicitado un presupuesto de fabricación.

Para representar los cálculos que se han expresado en este apartado, se van a encadenar un conjunto de situaciones automáticas que calcularán los tiempos de producción finales, y el precio final de la pieza en función de los costes obtenidos anteriormente. La acción interna asociada a estas situaciones automáticas será nuevamente la llamada a la hoja de cálculo con los parámetros ya explicados anteriormente en situaciones análogas.

El resultado de estas situaciones va a ser la aparición de un conjunto de hechos con todos los parámetros necesarios para la fabricación de la pieza, tiempos de fabricación de cada pieza y el coste de cada una. Como en la compañía estudiada el margen de beneficio está incluido en la mano de obra, el coste total es el precio final para la pieza.

En este punto la representación del conocimiento del experto acerca de la estimación del precio para mecanizar piezas en tornos automáticos multihusillo ha finalizado. Sin embargo, como se ha indicado en la descripción de la automatización de los procesos representados, para finalizar el proceso es necesario enviar un último mensaje interactivo al agente externo para comunicarle los resultados. Por tanto, tras la secuencia de situaciones automáticas se incluye una situación interactiva para mostrar los resultados finales de la estimación. Esto se refleja en una decisión cuyo destino es la situación “Resultados de la Estimación”.

Resultados de la estimación

Esta situación representa el final del proceso de estimación. Tras la determinación de los distintos parámetros, tiempos y costes de fabricación, en esta situación todos los datos relativos al precio y al proceso de fabricación son ahora conocidos.

Este caso se representa como una situación interactiva, en la que, por medio de una acción externa en forma de página HTML, se muestran al agente externo todos los resultados obtenidos a lo largo del proceso de estimación (figura 7.15).

A partir de estos resultados, el agente externo puede responder a la solicitud de oferta que ha recibido del cliente. Además con los datos proporcionados acerca del proceso de fabricación de la pieza, facilitará la elaboración del documento de definición del proceso, pues a través de las situaciones interactivas solicita parte de la información necesaria, proporcionando además parámetros de fabricación recomendados para que puedan ser incluidos en la misma.

A pesar de que esta situación es una situación final, pues el proceso de estimación ha finalizado, se ha incluido una decisión que para que tras esta última situación se regrese a la situación inicial. De esta forma se puede repetir el proceso de estimación para otro pedido, en caso de ser necesario.

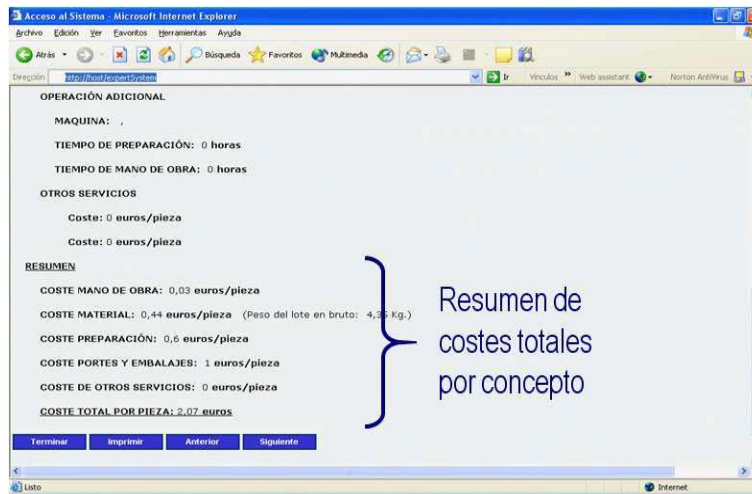


Figura 7.15: Resultados de la estimación.

7.5. Resumen

En esta sección explica como se aplicó el modelo de representación propuesto, para representar el conocimiento de un experto sobre el proceso a seguir para determinar el precio de fabricación de piezas mecanizadas. Como caso de estudio se ha elegido un proceso de fabricación a partir de un conjunto de máquinas y operaciones limitado pero representativo. En las distintas secciones se explica de forma sintética el conocimiento que se adquirió del experto de la fábrica, indicando los elementos del modelo de representación utilizados para representarlo.

A partir de la representación del conocimiento que se ha mostrado en este capítulo, se implantó en un servidor con el motor de solución propuesto y se realizaron un conjunto de pruebas para validar la solución propuesta. Los siguientes capítulos detallan los resultados obtenidos y las conclusiones que se han alcanzado con respecto a las hipótesis de partida.

Este capítulo describe el proceso de validación que se ha llevado a cabo a partir de las hipótesis planteadas. Para la validación se ha empleado la herramienta de modelado y el motor de solución descritos en los capítulos anteriores. A partir de estas implementaciones, se ha modelado el conocimiento sobre la estimación de precios y se ha comprobado la validez de las hipótesis en un entorno operativo real.

8.1. Metodología de Validación

Las hipótesis expuestas al comienzo de la investigación plantean un modelo de representación del conocimiento que pueda representar tanto el conocimiento del experto, como el conocimiento recogido en otras aplicaciones, así como la posibilidad de representar métodos de estimación detallados en combinación con la experiencia. Además, el modelo debe poder ejecutarse y ofrecer intercambio de información a través de la web con un usuario externo. Concretamente las hipótesis planteadas son:

1. El modelo propuesto puede representar el conocimiento del experto sobre el proceso de estimación de presupuestos.
2. El modelo propuesto puede representar el acceso conocimiento contenido en otras aplicaciones.
3. El modelo propuesto puede representar el intercambio de información entre agentes externos y el motor de solución para la resolución del problema.
4. Un proceso representado a través del modelo propuesto se puede reproducir a través de un motor de solución.
5. Un proceso automatizado puede obtener resultados similares a los de un experto.

La validación de las hipótesis planteadas se ha dividido en dos fases diferentes:

1. En una primera fase se ha validado la capacidad del modelo para representar los distintos tipos de conocimiento que se plantearon.
2. La segunda fase, denominada validación empírica, tiene, a su vez, un doble objetivo: en primer lugar validar la capacidad de automatización de un proceso modelado, intercambiando información a través de la web y de interfaces con otras aplicaciones, y, en segundo lugar, comprobando que los resultados que obtiene un proceso automatizado son aceptables con respecto al proceso de estimación del experto. Además, la segunda fase busca refrendar los resultados de la primera, pues comprobaría la utilidad del conocimiento representado.

Para la validación en ambas fases se ha representado el conocimiento del experto en una fábrica sobre la elaboración de presupuestos. Los pasos que se han seguido en el proceso de validación son:

1. Establecer una forma de implementar las definiciones realizadas con el modelo. Esta fase corresponde a la creación de la herramienta de representación y el desarrollo del motor de solución, pues son imprescindibles para poder validar el modelo propuesto.
2. Seleccionar un experto del dominio. Para validar las hipótesis era necesario representar el proceso de estimación de un experto y comprobar junto a él que se capturaba el conocimiento de forma adecuada y que se obtenían resultados aceptables. Como se ha descrito en el capítulo 7 se trabajó con el experto de una fábrica de piezas mecanizadas encargado de atender las peticiones de presupuesto de los clientes.
3. Modelar la experiencia de un estimador del dominio. A través de reuniones con el experto de la fábrica se representó el conocimiento acerca del proceso de estimación de presupuestos.
4. Puesta en marcha del proceso automatizado. En esta fase se implanta el motor de solución desarrollado para el modelo propuesto y se pone en marcha el proceso automatizado. Con ayuda del experto se comprueban los resultados del mismo.
5. Comprobación las hipótesis planteadas a partir de la implementación realizada. A partir de los resultados obtenidos se comprobó la validez de las hipótesis planteadas en la investigación.

Las siguientes secciones describen los pasos realizados y los resultados obtenidos.

8.2. Validación del Modelo de Representación

En esta primera fase de la validación se comprueban las siguientes hipótesis:

- **Hipótesis 1:** el modelo propuesto puede representar el conocimiento del experto sobre el proceso de estimación de presupuestos.
- **Hipótesis 2:** el modelo propuesto puede representar el acceso al conocimiento de aplicaciones externas.
- **Hipótesis 3:** el modelo propuesto puede representar el intercambio de información entre agentes externos y el motor de solución para la resolución del problema.

Hipótesis 1: Representación del conocimiento del experto

Para representar el conocimiento a partir del modelo propuesto, se contó con la colaboración de un experto de la oficina técnica de la fábrica, responsable de la elaboración de presupuestos, junto con dos jefes de sección de fabricación.

El proceso de representación fue evolutivo, combinando entrevistas con la representación del conocimiento a partir del modelo. El experto colaboró en la definición de la secuencia de pasos que da para establecer el presupuesto de fabricación, detallando los criterios que determinan cuál es el siguiente paso a realizar en el proceso, en función de las variables que intervienen en el problema.

A partir de esta información se definieron las distintas situaciones, con las acciones a llevar a cabo. A partir de los criterios para determinar el siguiente paso a dar en el proceso, se definieron las decisiones y los hechos asociados a cada situación.

Los elementos de conocimiento creados para la representación del conocimiento, se resumen en:

- 154 Situaciones
 - 53 situaciones interactivas, una de ellas librería interactiva.
 - 20 situaciones virtuales.
 - 81 situaciones automáticas.
- 36 verificaciones.
- 233 decisiones.

- 298 hechos.

El proceso representado en la validación está basado principalmente en la experiencia del estimador de la fábrica, complementada con un modelo detallado de estimación, basado en el proceso a realizar y en las características de la pieza. El conocimiento detallado se ha modelado por medio de situaciones automáticas. En el capítulo 7 se describe como se ha representado el conocimiento del experto de la fábrica.

El modelo ha servido para representar el proceso de estimación piezas de diferentes características fabricadas a partir de tornos automáticos multihusillo.

Hipótesis 2: El modelo propuesto puede representar el acceso al conocimiento contenido en otras aplicaciones

Los expertos de la fábrica, durante el proceso de estimación de precios y en la determinación de los procesos de fabricación implicados, realizan cálculos automatizados a través de hojas de cálculo y consultan bases de datos que contienen parámetros referidos a los materiales y las herramientas a emplear.

El conocimiento por tanto está representado en forma de bases de datos y hojas de cálculo, que requieren el acceso a través de una aplicación determinada. La utilidad no consiste en representar el mismo conocimiento que ya está representado en esas aplicaciones, en tanto en cuanto supondría la duplicación de información, con las desventajas de mantenimiento y consistencia de datos que conlleva. Lo que se representa es qué información es necesaria para el proceso de decisión y la forma de acceder a la misma.

En cuanto a qué información es necesaria para poder tomar decisiones, ésta se representa por medio de hechos. Estos hechos representan información que se deberá recuperar de la aplicación correspondiente. En cuanto a la forma de acceder a la información, se modela a través de una acción externa en la que se indica una interfaz que debe ser invocada, junto con un conjunto de parámetros que van a ser utilizados. En estos parámetros se pueden incluir hechos que contienen información que necesita la aplicación externa para poder proporcionar los resultados.

En el proceso de estimación se han representado 2 acciones externas: una representa la recuperación de datos de una base de datos, mientras que otra representa el acceso a una hoja de cálculo Excel. Se han representado 82 accesos a estas aplicaciones para obtener información relevante para el proceso, 41 a base de datos y 41 a hoja de cálculo. Las llamadas a bases de datos recuperan principalmente información relativa a materiales y parámetros del proceso, mientras que las llamadas a la hoja de cálculo envían parámetros del proceso y obtienen el resultado

de cálculos basados en el modelo detallado que complementa la experiencia del experto. Los parámetros están condicionados por la interfaz concreta que se vaya a utilizar.

Para la validación empírica, se han creado dos interfaces compatibles con el motor de solución implementado, para el acceso a bases de datos access y excel. En dicha validación se comprueba como a partir del modelado, el motor de solución puede acceder a los datos.

Hipótesis 3: El modelo propuesto puede representar el intercambio de información con el agente externo

Para la estimación del presupuesto de fabricación para una pieza determinada son necesarios un conjunto de datos que no pueden obtenerse de forma automática accediendo a otras aplicaciones. En la fábrica estudiada, las peticiones de oferta son procesadas por personas. Por tanto el proceso representado se orienta de forma que la información que no puede ser obtenida por el motor de solución, sea proporcionada por una persona.

Esta representación se realiza por medio de situaciones interactivas. Estas situaciones interactivas conllevan acciones externas, representadas por medio de páginas web en las que se recogen los hechos como elementos de un formulario, tal y como se ha descrito en el capítulo 7. Para el proceso de estimación se han representado cincuenta y tres situaciones interactivas, cincuenta y dos de ellas basadas en acciones externas en forma de páginas web y una a partir del resultado de una acción interna que recupera información en formato XML.

En la validación empírica se comprueba como la representación en forma de situaciones interactivas cumplen a nivel práctico con la función de enviar y recibir información de un agente externo.

8.3. Validación Empírica

En la validación empírica, además de refrendar a nivel práctico las tres primeras hipótesis, se van a validar principalmente dos hipótesis:

- **Hipótesis 4:** un proceso representado a través del modelo propuesto se puede reproducir a través de un motor de solución.
- **Hipótesis 5:** un proceso automatizado puede obtener resultados similares a los de un experto.

Para comprobar la **hipótesis 4**, a partir de la representación del conocimiento de los expertos de la fábrica, se ha realizado un conjunto de pruebas para determinar que el proceso puede ser reproducido por medio del motor de solución. Conjuntamente, para validar la **hipótesis 5** se ha comprobado que los resultados de la estimación son similares a los que proporcionaría un experto.

La validez de las dos hipótesis implica:

- Realizar el presupuesto para las máquinas establecidas y para el grupo de piezas determinado.
- Obtener resultados válidos y útiles para la fábrica.

La validación de estas hipótesis servirá además para la validación empírica de las tres primeras hipótesis relativas al modelo de representación propuesto.

Para determinar si los datos obtenidos eran válidos, se comparó el resultado obtenido por el proceso automatizado con la estimación realizada por el experto de la fábrica, de acuerdo con la **hipótesis 5**. Las estimaciones del experto en la fábrica han servido durante años con buenos resultados, por lo que la semejanza entre los resultados del proceso automático y los del experto demostraría la utilidad del modelo y su automatización para la fábrica. En la realización de las pruebas no fue posible acceder a datos históricos de fabricación, y tampoco era posible la comprobación del tiempo real de fabricación pues las configuraciones de la maquinaria en la empresa se realizan para periodos de tiempo muy largos.

La finalidad del proceso es establecer el precio por pieza a ofrecer al cliente. El precio se compone de la suma de un conjunto de costes asociados al proceso de fabricación, más un margen de beneficio. En el capítulo 7 se describen estos componentes y la forma de calcularlos a la vez que se explica como son modelados.

De los costes descritos, algunos no están disponibles para un acceso automático, por lo que deben ser introducidos por el agente externo que está siguiendo el proceso. Este es el caso de los costes de material, embalaje y transporte, así como los costes de mano de obra, pues pueden variar en función de factores internos de la fábrica (es necesario tener en cuenta que en los cálculos habituales, la fábrica incluye el margen de beneficio en el coste de la mano de obra). Estos costes no van a suponer diferenciación entre los resultados del proceso y los resultados del experto.

Otros cálculos como puede ser el peso del material se realizan a partir de fórmulas fijas, que son parte del modelo detallado que aplica el experto, por lo que tampoco van a suponer diferencia.

Donde se va a encontrar la diferencia real entre los resultados va a ser en el establecimiento del tiempo de fabricación y de preparación entre el experto y el proceso automatizado, pues es la parte donde influyen las decisiones que se toman durante el proceso de estimación.

8.3.1. Descripción de los casos de pruebas

Tras modelar el conocimiento del experto por medio de la herramienta de modelado, se instaló el motor de solución en un servidor accesible desde la fábrica, y se publicó el acceso al proceso automatizado.

Se llevaron a cabo dos tipos de prueba. En primer lugar se realizó una batería de pruebas en las que se accedía al proceso a través del servidor y se siguieron diversos caminos en función de las selecciones realizadas. El objetivo de estas pruebas es comprobar las hipótesis sobre el intercambio de información con el agente externo y con otras aplicaciones. Para ello se controlaron a través de los ficheros históricos de la base de hechos, los pasos que había dado cada sesión, los valores que se habían recibido del agente externo y los datos recibidos de las acciones internas.

Para la validación se han comparado las estimaciones del proceso automatizado con los resultados de nueve estimaciones que el experto consideró representativas para los procesos de estimación descritos en el capítulo anterior. Se han comparado los cálculos de tiempos pues es donde estriba la diferencia entre ambas estimaciones.

Los datos correspondientes a estas pruebas son los siguientes:

1. Casquillo (1). Tamaño de lote: 3500 unidades. Operación adicional de avellanado.
2. Bulón. Tamaño de lote 3500: unidades. Operación adicional de eliminación de pivotes.
3. Casquillo (2). Tamaño de lote: 5000 unidades. Operación adicional de avellanado.
4. Casquillo (3). Tamaño de lote: 10000 unidades. Operación adicional de avellanado.
5. Arandela (1). Tamaño de lote: 3500 unidades. Operación adicional de avellanado.
6. Casquillo (4). Tamaño de lote: 5000 unidades. Operación adicional de avellanado.
7. Eje. Tamaño de lote: 15000 unidades. Operación adicional de eliminación de pivotes.
8. Arandela (2). Tamaño de lote: 2000 unidades. Operación adicional de avellanado.
9. Separador. Tamaño de lote: 4000 unidades. Operación adicional de avellanado.

Las pruebas se llevaron a cabo con el experto de la fábrica accediendo al proceso automatizado a través del servidor web.

8.3.2. Resultados obtenidos

La **hipótesis 1** ha sido validada representando el conocimiento del experto en el dominio. La validez del resto de hipótesis planteadas viene a refrendar la validez de esta primera, pues confirmarán que los elementos representan los tipos de conocimiento que se establecieron en los objetivos y que la representación se puede automatizar obteniendo resultados aceptables.

Las **hipótesis 2 y 3** se refieren a la capacidad del modelo para representar el intercambio de información con otros sistemas (hipótesis 2) y con personas (hipótesis 3).

El intercambio de información con otras aplicaciones se ha representado por medio de acciones internas que están asociadas a situaciones automáticas. Como se ha descrito anteriormente, se ha representado el intercambio de datos con la aplicación Excel y Access, para lo cual se definieron sendos interfaces compatibles con el motor de solución. La representación del intercambio se representa, además de la definición de la acción interna en el modelo, con la definición de los parámetros necesarios para invocar la interfaz y obtener un resultado.

Por otra parte, el intercambio de información con personas se realiza a través de la interfaz web, y se materializa en el intercambio de información entre el agente externo y el motor de solución a lo largo del proceso. Este intercambio se representa por medio de situaciones interactivas y acciones externas en forma de páginas web o XML. En la aplicación del modelo a la fábrica de estudio, se definieron situaciones interactivas cuando era necesario obtener información acerca de la pieza o del proceso de fabricación que no podía ser calculada automáticamente, y para presentar resultados, tanto intermedios como finales.

Para comprobar que ambas representaciones son válidas, es decir, que representan la forma de pedir la información y recibirla, se ha realizado un conjunto de pruebas sobre la implementación realizada. En estas pruebas 25 usuarios accedieron al sistema con la intención de interactuar con el sistema y comprobar que los datos se intercambiaban correctamente y se recuperaban los valores correspondientes de las aplicaciones externas.

Para realizar las pruebas, a lo largo de la puesta en funcionamiento del proceso automatizado se dio acceso a cada uno de los usuarios que participaron en la prueba y se dio libertad para completar el proceso hasta donde consideraran conveniente, debiendo utilizar al menos una vez la funcionalidad del motor de solución para regresar a una situación anterior. Además debían utilizar la funcionalidad terminar del motor de solución cuando decidieran dar por terminada su interacción. Los resultados de estas pruebas quedaron registrados en la base de hechos del



Figura 8.1: Situaciones visitadas por prueba.

motor de solución para, de esta forma, depurar posibles errores en el funcionamiento del motor de solución o posibles problemas en la representación.

La figura 8.1 muestra el número de situaciones que se produjeron en cada una de los 25 accesos de los usuarios de la fábrica. El número de situaciones es variable en función de los distintos caminos posibles que siguió cada uno de los usuarios y del momento en que decidiera dar por cerrado el proceso. En esta figura se aprecia que los procesos que se completaron rondaron entre 25 y 40 situaciones. El caso extremo de 44 situaciones enlazó dos procesos de estimación, dando por terminada la prueba hacia la mitad del segundo proceso.

De las situaciones que se han mostrado, es de interés conocer los tipos de situación que se produjeron, a la hora de comprobar si la interacción se ha producido con todos los tipos de situación posibles.

La figura 8.2 muestra el número de situaciones de cada tipo que se han producido durante cada proceso automático. Se puede observar como las sesiones que más lejos llegaron en el proceso son las que más situaciones automáticas tienen. Esto tiene sentido, pues cuando se llega al final del proceso, el motor de solución ya tiene suficiente información como para realizar por sí mismo las acciones necesarias para completar el proceso.

A partir de los datos de las figuras 8.1 y 8.2 se determina que el motor de solución reproduce el proceso definido y que se produce intercambio de información tanto con personas (a través del agente externo), como con otras aplicaciones a través de los interfaces definidos.

Se comprobó además si los resultados de la interacción eran los esperados. Estos datos esperados no se refieren a que se obtuvieran unos resultados correctos desde el punto de vista de la estimación, pues esto se comprobó después. Por datos esperados se hace referencia a valores que

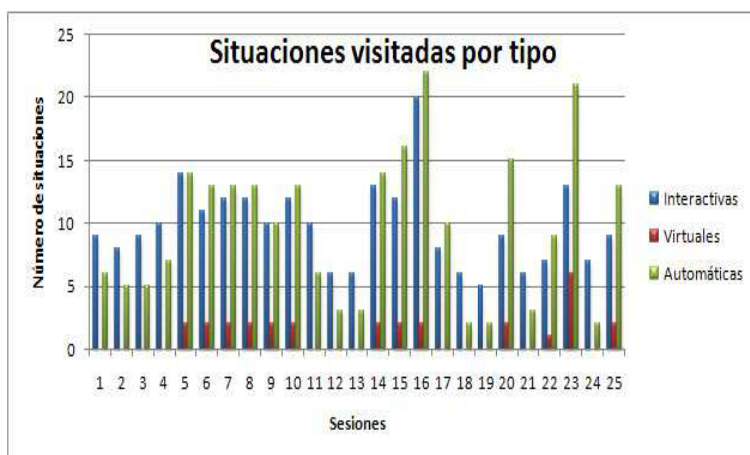


Figura 8.2: Número de situaciones visitadas por tipo de situación.



Figura 8.3: Resultados de la interacción realizada por cada usuario.

tuvieran coherencia con respecto a los valores que introdujo el usuario, por ejemplo, que todos los resultados que se mostraban tuvieran un valor o que no se produjera ningún error durante la interacción.

La figura 8.3 muestra si los resultados de las pruebas fueron los esperados o no, en función de la información intercambiada. De las 25 sesiones analizadas, 21 tuvieron un resultado correcto, mientras que 4 de ellas presentaron algún tipo de problema. Concretamente las sesiones afectadas tuvieron los siguientes problemas:

- La sesión 8 produjo un resultado inesperado al aceptar un valor no válido en una situación interactiva. Este error no es achacable al modelo, pues la solución consiste en añadir una nueva verificación asociada a la situación interactiva donde se recogieron los datos.

- La sesión 11 obtuvo unos resultados incorrectos del acceso a una hoja de cálculo para determinar la velocidad de corte. En este caso, el error tampoco era achacable al modelo, pues se habían representado correctamente los parámetros, sino que se trataba de un error al definir el hecho que contenía uno de los parámetros.
- Las sesiones 17 y 18, generaron sendos errores de conexión a la base de datos para realizar consultas en diferentes puntos. En este caso el problema tampoco residía en el modelo, sino a un problema en la configuración de las bases de datos en el momento de las pruebas.

Como consecuencia se desprende que un 84 % de las sesiones analizadas no presentaron ningún problema en la interacción. De las cuatro sesiones que tuvieron algún problema ninguna es achacable a la capacidad de representación del modelo, sino a la forma en que se ha empleado para representar: una sesión encontró que faltaba una verificación, otra un problema en la definición de un parámetro y las dos restantes fueron un problema de configuración en la base de datos.

Con estas pruebas, se ha comprobado que el modelo puede representar la comunicación con una persona que actúa como agente externo del proceso, y la comunicación con otras aplicaciones, intercambiando información en ambos casos. Además se ha comprobado que un motor de solución puede interpretar esta representación y ejecutarla. Con esto se demuestra la validez de las **hipótesis 2 y 3**.

A través de estas pruebas, el proceso representado ha podido ser reproducido a través del motor de solución, definiendo la estimación para tres posibles máquinas, más un proceso adicional para la actualización de las pinzas disponibles para cada máquina.

Los empleados de la fábrica a través de ordenadores conectados a internet, han podido conectar con un servidor donde se ha alojado el motor de solución propuesto y han podido reproducir el proceso.

Esto comprueba la **hipótesis 4** sobre la posibilidad de automatizar un proceso representado a partir del modelo propuesto.

Para la validación de la **hipótesis 5**, sobre la capacidad de conseguir resultados similares a los de la estimación del experto, se llevaron a cabo un conjunto de pruebas basadas en la experiencia del experto. La sección 8.3.1 describe los motivos de la comparación realizada.

La tabla 8.1 muestra los resultados de las estimaciones obtenidos a partir del proceso automatizado, comparados con las estimaciones realizadas por el experto. En dicha tabla la columna *T. Automático* indica el tiempo de producción por pieza estimado por el proceso automatizado, mientras que la columna *T. Experto* indica el tiempo de producción por pieza estimado por el

Tabla 8.1: Resultados de estimación para tornos.

<i>Pieza</i>	<i>Torno</i>	<i>Lote</i>	<i>T. Automático (horas)</i>	<i>T. Experto (horas)</i>	<i>Diferencia (%)</i>
Casquillo	Ruso	3500	0.0062	0.0064	-3.1 %
Bulón	Plitter	3000	0.0129	0.0135	-4.4 %
Casquillo	Ruso	5000	0.00585	0.0067	-12.7 %
Casquillo	Ruso	10000	0.0063	0.0059	+6.8 %
Arandela	Conomatic	15000	0.01058	0.0099	+6.9 %
Casquillo	Plitter	5000	0.0063	0.0059	+6.8 %
Eje	Plitter	15000	0.0096	0.0094	+2.1 %
Arandela	Ruso	2000	0.01397	0.01187	+17.6 %
Separador	Conomatic	4000	0.00732	0.00682	+7.3 %

experto.

Para calcular el tiempo unitario para cada una de las piezas se ha empleado la siguiente fórmula:

$$T_{total} = T_{prod1} + T_{prod2} + \frac{t_{prep1} + t_{prep2}}{Lote} \quad (8.1)$$

Donde:

- T_{total} es el tiempo total unitario por pieza.
- T_{prod1} es el tiempo de producción para la operación principal.
- T_{prod2} es el tiempo de producción para la operación adicional.
- T_{prep1} es el tiempo de preparación para la operación principal.
- T_{prep2} es el tiempo de preparación para las operaciones adicionales.
- $Lote$ es el tamaño del lote a fabricar.

Una vez determinados los tiempos, la comparación entre los tiempos de fabricación se realiza de acuerdo con la siguiente fórmula:

$$Diferencia(\%) = 100 \times \frac{t_{automatico} - t_{experto}}{t_{experto}} \quad (8.2)$$

Donde:

- Diferencia es el tanto por ciento de variación entre la estimación del proceso automatizado y al estimación del experto.
- $t_{automatico}$ es el tiempo estimado por el proceso automático.
- $t_{experto}$ es el tiempo estimado por el experto de la fábrica.

Los resultados detallados para cada uno de los ejemplos se muestran en la tabla 8.4. En esta tabla la columna $T_{preparacion}$ muestra el tiempo de preparación del torno (es de 6 horas en todos los casos), las columnas $T_{produccion}$ muestran el tiempo en horas estimado para producir una pieza sin incluir las operaciones adicionales, las columnas $T_{adicional}$ contienen la estimación del tiempo en horas de las operaciones adicionales para cada pieza, y, finalmente, las columnas $T_{prepadicional}$ indica la estimación del tiempo en horas para la preparación de la operación adicional. La palabra *Automático*, al igual que en la tabla anterior, indica que el tiempo corresponde a la estimación realizada por el proceso automatizado, mientras que la palabra *Experto* indica que el tiempo corresponde a la estimación del experto de la fábrica.

Analizados por el experto de la fábrica, los resultados obtenidos por el proceso automatizado son buenos. Según el criterio del experto la diferencia puede deberse a las condiciones específicas de la fábrica en el momento de la estimación, que no se consideraron para la estimación pues no se conocían con certeza los factores a considerar. De haberse conocido, el proceso permite a través de situaciones interactivas modificar los parámetros de la estimación para ajustarlo a las condiciones específicas, como pueden ser los tiempos de preparación, los tiempos de producción y las máquinas recomendadas.

Dejando a un lado los tiempos de preparación, pues se trata de valores fijos, las variaciones no son grandes salvo la estimación para la fabricación de la arandela, que se dispara a un 17.6%. Como se ha comentado, la diferencia puede estar en las condiciones concretas de la fábrica en el momento de realizar la estimación anterior.

Es necesario considerar que el precio final de la pieza se calcula teniendo en cuenta otros costes: el coste de material, los costes de tratamientos térmicos posteriores, los costes de portes y embalajes, y los costes de otros servicios. Estos costes no son calculados por el proceso automatizado en su versión actual, sino que son determinados por la persona a cargo de la estimación. Por lo tanto, la variación de la estimación que proporciona el proceso automatizado con respecto al precio establecido por el experto se debe a las diferencias en los tiempos de fabricación y preparación. Estos tiempos suponen, según los expertos, aproximadamente un 34 % del precio

total. Considerando esto, la diferencia media entre el precio final calculado por el proceso automatizado y el precio estimado por el experto es de aproximadamente un 2.5 %. Este margen ha sido considerado como bueno en la fábrica.

8.4. Resumen

En este capítulo se ha descrito la validación de las cinco hipótesis que se plantearon a partir de los objetivos de la investigación.

La primera de las hipótesis planteadas suponía que el modelo propuesto podía representar procesos de estimación de presupuestos basados en el conocimiento del experto. Esta hipótesis se ha validado representando el conocimiento del experto de una fábrica.

La segunda y tercera hipótesis planteaban que el modelo era capaz de representar el intercambio de información con personas y con otras aplicaciones. Esta hipótesis se ha validado representando diferentes casos de intercambio con personas y con aplicaciones y comprobado empíricamente que se puede realizar el intercambio de información representado.

La cuarta hipótesis planteaba que el proceso representado a través del modelo propuesto podía ser automatizado, es decir, que podía ser reproducido a través de un motor de solución. Esta hipótesis se ha validado con la implementación de un motor de solución y la puesta en marcha de un proceso automático en una fábrica real.

La quinta hipótesis planteaba que un proceso automatizado en base al modelo propuesto, obtendría resultados similares a los de un experto en estimación. Esta hipótesis ha sido validada a través de la puesta en marcha de un proceso automatizado en una fábrica.

Con estas hipótesis se ha comprobado la capacidad del modelo planteado al comienzo de la investigación para representar conocimiento relativo a procesos de estimación, su utilidad para la estimación de presupuestos de fabricación y su integración en la web.

Pieza	Lote	$T_{preparacion}$		$T_{produccion}$		Operación		$T_{adicional}$		$T_{preparadional}$	
		Experto	Automát.	Experto	Automát.	Adicional	Experto	Automát.	Experto	Automát.	
Casquillo	3500	6	0.0033	0.0033	0.0033	Avellanado	0.0011	0.001	1	0.5	
Bulón	3000	6	0.01	0.0089	0.0089	Pivotes	0.0015	0.002	0	0	
Casquillo	5000	6	0.004	0.0033	0.0033	Avellanado	0.0015	0.00125	0	0.5	
Casquillo	10000	6	0.004	0.0044	0.0044	Avellanado	0.00125	0.00125	0.5	0.5	
Arandela	15000	6	0.0085	0.0089	0.0089	Avellanado	0.001	0.00125	0	0.5	
Casquillo	5000	6	0.004	0.0044	0.0044	Avellanado	0.00125	0.00125	0.5	0.5	
Eje	15000	6	0.0077	0.0072	0.0072	Pivotes	0.0013	0.002	0	0	
Arandela	2000	6	0.0045	0.0056	0.0056	Avellanado	0.004	0.005	0.75	0.75	
Separador	4000	6	0.004	0.0047	0.0047	Avellanado	0.0012	0.001	0.5	0.5	

Tabla 8.2: Detalle de estimación (en horas).

Capítulo 9

Conclusions and Future Research

In order to satisfy the requirements for the european doctorate mention, this chapter includes the english translation of the conclusions of this phd thesis.

In this chapter final research conclusions are presented. Together with an overview of the main contributions of this thesis, a set of future trends are proposed in order to continue the research work. Finally, at the end of the chapter there is a list of contributions to congresses and journal papers related to this thesis.

9.1. Conclusions

At the beginning of this thesis, the problem of price estimation in the RFQ process was introduced. Such a problem lies on the necessity of a reliable estimation in a short period of time. A delay in quotation may cause a client to choose a faster offer from a competitor. On the other hand, a fast but poorly estimated response can lead to economic losses for the company if an estimated price is less than the cost of production or the loss of a client if the estimated price is too steep. Given the importance of the price estimation, this process is usually carried out by an expert of the factory. Such an expert has other important tasks to which he must attend, and, for this reason, the estimation can be delayed in time for days or weeks . The use of software systems provides a solution by means of price estimation applications. As mentioned in the “State of the Art” section, there are a number of applications that implements estimation models proposed in the literature. However, according to the mentioned studies, such systems are usually more focused on theoretical knowledge rather than encapsulating the expert’s practical knowledge (Niazi et al. 2006).

The representation of the knowledge of an expert, besides optimizing the time required for the estimation, leads a number of additional advantages:

- It makes possible the use of knowledge of the expert in order to approximate unknown parameters required for the application of detailed estimation models. Sometimes, the es-

timation models require detailed information about the product not known at the time of the request for quotation. An expert, based on previous experiences, can approximate such values.

- It allows the saving of the knowledge in the company. The represented knowledge remains in the organization although the expert leaves the company. If such knowledge is represented in a comprehensible way, it can be used for the instruction of new experts.
- It allows the knowledge sharing. The knowledge represented according a set of rules can be interchanged between companies.

As mentioned, the requirements of the new web technologies bring new challenges for the knowledge representation systems: it is necessary that the new models are integrated with such web technologies. However, not an integral model that covers the lacks identified have been found.

The main contribution of this thesis is a knowledge representation model as response to the identified issues about industrial engineering and artificial intelligence.

From the industrial engineering point of view, the RFQ process has been studied and the relevance of the price estimation based on incomplete information has been identified. Furthermore, due to the concrete characteristics of versatile manufacturing companies, the detailed estimation models are not usually appropriate for the initial development phases because of the detailed information they require. In this thesis several estimation models and techniques have been reviewed. One of the disadvantages of theoretical models is that they require more information than available when the quotation is requested. Parametrical models represent an exception, but such models do not take into account aspects about the product or the process that can affect the final price. The study of Niazi et al (2006) reviews the main estimation methodologies and describes its use in the actual industrial context. This study remarks the fact that actual expert systems for cost estimations are based on theoretical knowledge instead of experts' heuristic knowledge. Niazi et al. (2006) concludes that the use of knowledge based on the practical experience in expert systems for cost estimation is an open and interesting research area. In the "State of the art" section we have reviewed both knowledge-based estimation systems and conventional estimation systems applied to cost and price estimation in machining companies, and we have confirmed the affirmation of Niazi et al (2006) in the thesis domain. Most knowledge-based systems for cost and price estimation are oriented to theoretical models, and practical experience remains in the background supporting the theoretical model. In this regard, this thesis provides a knowledge representation model directed to capture the practical knowledge of an expert in the field of the

quotation process. The proposed model is based on the set of situations that can be identified by an expert in the RFQ process and the actions required in each situation.

From the artificial intelligence point of view, Grove (2000) and Duan et al (2005) point out the necessity of new representation models for adapting knowledge representation to new web technologies. These authors identified that new web-based expert systems are not based on specific web-oriented models, but they are based on ad-hoc developments using traditional expert system technologies. Such traditional technologies may be less effective in web environments (Duan et al, 2005).

Based on the problematic described above, this thesis provides a knowledge representation model for representing the knowledge of experts in the area of cost and price estimation in the quotation process. Although it is out of the scope of this thesis, the proposed model may also be applied to many other domains like help-desk or sales supporting.

In the validation conducted, it is observed that the proposed model allows representing the knowledge of experts according to the objectives pointed at the beginning of this research:

- Define the elements necessary for representing the price estimation process of an expert in the domain of machining parts.
- Represent detailed estimation models that compliment the expert knowledge.
- Integrate the model with new Web technologies.
- Automate the price estimation, allowing interaction with people and other applications.

The next contribution of this thesis is a solution engine based on the proposed model. Such an engine allows the execution of the represented processes in a web environment. The solution engine is neither restricted to the price estimation nor limited to the concrete problem of the case-of-study company.

A modelling tool has been developed in order to represent processes based on the proposed model in an easy way. This tool allows unifying in a single tool the expert system and the web system's point of view (Dokas and Alapetite, 2006).

Besides the objectives and hypotheses initially pointed out, we have included a formal definition of the knowledge representation model expressed in the form of an ontology based on the OWL-DL language, in order to describe the concepts, the relation between such concepts and the restrictions. It enables the representation of different processes by different experts in different places by means of a common web adapted language, according to the lines proposed by Duan et al (2005). It opens a range of future research lines that will be detailed in the next subsection.

9.2. Future Research

The presented model allows the representation of knowledge about the quotation process. Based on the proposed model, it is possible to define other manufacturing processes, making the process automation easier for the factories.

Furthermore, the use of the proposed knowledge representation model in other domains is other open research area, evaluating the advantages and capabilities of the model for each domain.

Based on the Duan et al (2005) proposals, the development of a modelling web tool for knowledge management may be possible. This tool will allow the on-line collaborative process definition based on the proposed model, allowing several experts to define and interchange process definitions. The development of a web site where experts in different domain can represent their knowledge, gives a number of possibilities in order to share the existing knowledge and increase it with the knowledge of other experts with common interests. Since the solution engine is integrated into a web server, the web site not only defines new processes and shares the knowledge but also it gives concrete solutions.

In the line of the development of the semantic web and social networks, it would be interesting to provide semantic annotation for the knowledge objects represented by means of the proposed model. Such annotations may be based either on a concrete ontology domain according to the process domain or on relevant information for the experts involved in the process. The semantic annotation of knowledge objects in a collaborative environment may ease the search for knowledge by experts, allowing to find processes related to a specific knowledge area in order either to reuse them or to collaborate in their improvement. By annotating not only the knowledge, but also the experts which define such knowledge, would ease the finding of other experts related with an expertise area in order to establish some kind of collaboration between them.

Semantic annotation of knowledge objects would allow the generation of new knowledge based on the metadata related to existing situations and actions in a defined process. For example, semantically similar situations for different processes could be automatically found when a new process is defined or new decisions could be automatically suggested based on the annotations. It would allow making suggestions about situations and decisions during the process definition.

Finally the trace module of the knowledge representation tool could be extended providing explanations about failures in the process execution and its consequences in other related situations based on the semantic annotation of the knowledge objects.

9.3. Related Publications

Based on the research results obtained, the next papers have been published in congresses and international journals:

- *A Review of Conventional and Knowledge Based Systems for Machining Price Quotation.* Ángel García-Crespo, Belén Ruiz-Mezcua, José Luis López-Cuadrado, Israel González-Carrasco. **Journal of Intelligent Manufacturing**. 2009. DOI: 10.1007/s10845-009-0335-1. **JCR 1.018 (2008)**.

This paper covers the reviews the knowledge-based systems that have been implemented in the area of machined part manufacturing for cost and price estimation. Recommendations as to how future knowledge-based systems for the estimation of this type of pricing should develop are also included. It describes the motivation and the background of this thesis.

- *Development of a Web Based Quotation Expert System for Machined Parts.* Lopez-Cuadrado, J. L.; García-Crespo, A.; Ruiz-Mezcua, B.; Gomez, J. M. and Gonzalez-Carrasco, I. **International Journal of Computer Applications in Technology**, Special Issue: Computer Applications in Research and Development of Complex Mechanical Systems. In press. This paper briefly describes the problem and presents the web-based expert system for drafting quotations based on a factory expert knowledge.

- *Arquitecturas para Gestión de Conversaciones B2B Basadas en Conocimiento.* Lopez-Cuadrado, J. L.; Gomez, J. M.; García-Crespo, A.; Ruiz-Mezcua, B.; González-Carrasco, I. and Alor-Hernandez, G. **Proceedings of the 7th IEEE Latin-American Conference in Software Engineering and Knowledge Engineering**, 2008.

In this paper, we present three architectures (based on the model presented in this thesis) that allow a knowledge-based management of B2B conversations. Each proposed architecture allows different kinds of interaction between business partners.

- *A Fully Web Oriented Expert System Tool.* Lopez-Cuadrado, J. L.; Gonzalez-Carrasco, I.; García-Crespo, Á. and Ruiz-Mezcua, B. **IADIS International Conference WWW/Internet**. Murcia, Spain., 2006.

This paper briefly describes the modeling tool developed based on the proposed model.

- *Applying an Expert System to Factory Budget Calculations.* López-Cuadrado, J. L.; González-Carrasco, I.; García-Crespo, A.; Ruiz-Mezcua, B.; Fernandez-Heijnen, Y. and Cantero-

Guisández, J. L.

10th **IEEE International Conference on Emerging Technologies and Factory Automation**. Catania, Italy, 2005.

This paper briefly describes the specific problem of a manufacturing company used for validating the proposed model.

Capítulo 10

Conclusiones y Futuras Líneas de Investigación

A fin de cumplir con los requisitos para la obtención de la mención europea del doctorado, este capítulo sobre las conclusiones fue escrito originalmente en inglés. A continuación se incluye la traducción del mismo al castellano.

En este capítulo se presentan las conclusiones finales de la investigación que se ha expuesto en la presente tesis doctoral. Junto con el repaso de las principales aportaciones de esta tesis, se propone un conjunto de futuras líneas de investigación que pueden ser continuación de la misma. Finalmente se lista el conjunto de publicaciones que se han realizado a partir de la investigación realizada.

10.1. Conclusiones

Al comienzo de la presente memoria se introdujo la problemática de la estimación de precios en el momento en que se recibe una solicitud de presupuesto. El problema radica en la necesidad de conseguir una estimación fiable en un tiempo de respuesta muy corto. Un tiempo de respuesta demasiado elevado puede hacer que el cliente se decida por otra oferta que haya recibido antes. Una respuesta rápida pero con una estimación por debajo del precio de coste conllevaría pérdidas económicas para la compañía, mientras que una estimación demasiado elevada decantaría al cliente hacia otra oferta. Dada la importancia y características de este proceso, la estimación suele ser llevada a cabo por un experto de la fábrica concreta. Dicho experto suele tener asignadas otras tareas de importancia, con lo cual en ocasiones el proceso se puede demorar en el tiempo. Las aplicaciones informáticas proponen una solución a este problema, por medio de software capaz de llevar a cabo estas estimaciones. Como se ha detallado en el capítulo relativo al estado del arte, se ha investigado en distintas aplicaciones que implementan modelos de estimación propuestos en la literatura. Sin embargo, de acuerdo con los estudios ya mencionados, se ha encontrado que la representación de la experiencia de los estimadores queda en un segundo plano, utilizando conocimientos de modelos teóricos (Niazi et al. 2006).

La representación de este conocimiento experto, además de optimizar el tiempo de estimación y la aplicación de otros métodos detallados, conlleva una serie de ventajas adicionales:

1. Posibilidad de aprovechar el conocimiento del experto para estimar parámetros desconocidos en la utilización de modelos detallados de estimación. En ocasiones los modelos de estimación empleados convencionalmente, necesitan información del producto que aún no se conoce en el momento de elaborar el presupuesto. Un experto puede aproximar estos valores en base a su conocimiento anterior.
2. Conservación del conocimiento en la compañía. El conocimiento representado permanece en la fábrica aunque el experto se vaya. Si este conocimiento está representado de una forma comprensible puede servir para mantenerlo en la compañía y utilizarlo posteriormente en la formación de nuevos expertos.
3. Compartición del conocimiento. El conocimiento representado de acuerdo con unas normas puede ser utilizado para ser intercambiado con otras organizaciones.

Como se ha señalado, las nuevas necesidades de la web han traído nuevos retos para la representación del conocimiento: es necesario que los nuevos modelos se integren con las nuevas tecnologías web. Sin embargo no se ha encontrado un modelo integral que de respuesta a las necesidades de representación del conocimiento experto identificadas.

En la presente tesis doctoral se ha aportado un modelo de representación del conocimiento, que surge como respuesta a las cuestiones que se han planteado, desde el ámbito de la ingeniería industrial y de la inteligencia artificial.

Desde el punto de vista de la ingeniería industrial se ha estudiado el proceso RFQ y se ha identificado la importancia de la estimación de precios para ofrecer al cliente a partir de datos escasos o incompletos. Además, las características concretas de las empresas de fabricación versátil hacen complicada la utilización de modelos detallados de estimación en las fases iniciales del desarrollo de un producto, como es el caso de la recepción de una petición de oferta. En la investigación llevada a cabo se han estudiado distintos modelos y técnicas de estimación que se utilizan para la estimación. Los modelos teóricos tienen la desventaja de ser demasiado detallados para utilizarlos en el momento de un pedido, excepción hecha de los modelos paramétricos, que tienen como desventaja que no tienen en cuenta aspectos del producto o del proceso que pueden ser relevantes para el precio. El estudio de Niazi et al. (2006) repasa las principales técnicas y la utilización que tienen en la industria actual. En este estudio se resalta que dentro de los sistemas expertos basados en conocimiento la tendencia es utilizar conocimiento teórico

frente a heurísticas del experto. Niazi et al. (2006) identifican explícitamente la utilización de la experiencia como un área de interés en la investigación en los métodos de investigación.

En el estudio del estado del arte realizado, se han analizado la investigación realizada tanto sistemas de estimación no basados en la inteligencia artificial como sistemas basados en conocimiento, aplicados a la estimación de costes y precios en compañías de fabricación, constatando la afirmación de Niazi et al. La mayoría de los sistemas basados en conocimiento se orientan a modelos teóricos y dejan la experiencia en un segundo plano, como apoyo al modelo teórico representado. En este sentido, la presente tesis doctoral proporciona un modelo de representación del conocimiento que permite capturar la experiencia del experto en el proceso de elaborar el presupuesto, en base a la identificación de las situaciones que se producen a lo largo del proceso de estimación y las distintas acciones a realizar en cada una de ellas.

Desde el punto de vista de la inteligencia artificial, los estudios de Grove (2000) y Duan et al. (2005) identifican la necesidad de modelos nuevos que se adapten a las nuevas tecnologías web. Los citados autores identifican que los nuevos sistemas expertos basados en web no utilizan un modelo propio sino que emplean los tradicionales sistemas adaptados, pudiendo no ser tan efectivas en entornos web (Duan et al, 2005).

Para responder a los dos aspectos que se han identificado, se ha aportado un modelo de representación del conocimiento para capturar la experiencia de un experto orientado a la estimación de presupuestos. Aunque no es objeto de demostración para la presente tesis doctoral, se puede deducir fácilmente que el modelo es aplicable a un amplio espectro de dominios como el soporte a la venta, help-desk y otros de problemas que puedan plantearse en forma de situaciones y acciones para adquirir información y tomar decisiones.

En la validación realizada se ha comprobado que el modelo propuesto permite la representación del conocimiento del experto de acuerdo con los objetivos planteados al comienzo de la investigación:

- Definir los elementos que permitan representar el proceso que sigue un experto para determinar un precio.
- Representar el conocimiento relativo a modelos detallados que puedan ser empleados por el experto para la determinación del precio.
- Permitir que el modelo se integre con la web.
- Permitir la automatización de la estimación de precios, interactuando con personas y con otras aplicaciones

La siguiente aportación de esta tesis doctoral es un motor de solución basado en el modelo propuesto capaz de automatizar los procesos de representación del conocimiento, en un entorno web. Se trata de un motor de solución genérico que no es exclusivo para la implementación del proceso de representación de la fábrica utilizado para la validación.

Finalmente se ha proporcionado una herramienta de modelado que permite la representación del conocimiento de un experto al tiempo que facilita el proceso de validación por medio de la funcionalidad de traza. Esta herramienta permite unificar la visión del sistema experto y sistema web en un único concepto (Dokas and Alapetite 2006).

Además de los objetivos e hipótesis inicialmente planteados, la presente tesis aporta la definición formal del modelo de representación del conocimiento expresada en forma de ontología en el lenguaje OWL-DL. Con ello se ofrece la posibilidad de compartir la representación de distintos procesos entre expertos de diferentes lugares, utilizando un lenguaje adaptado a la web, en la línea de las necesidades planteadas por Duan et al. (2005). Esto abre el abanico de futuras líneas de investigación, que se tratarán en la sección siguiente.

10.2. Futuras Líneas de Investigación

El modelo propuesto permite la representación del conocimiento del experto acerca del proceso de estimación de presupuestos. A partir de la representación propuesta es posible definir otro tipo de procesos relacionados con la industria, facilitando la automatización de procesos en las fábricas.

Así mismo, la utilización del modelo de representación del conocimiento propuesto en otros dominios supone otra línea de trabajo a continuar, evaluando la capacidad del modelo y las ventajas que puede aportar a cada dominio.

El modelo de representación del conocimiento ha sido definido a partir de una ontología para describir los diferentes conceptos, sus relaciones y restricciones. Esta representación ontológica abre un amplio abanico de posibilidades en las futuras líneas de investigación.

A partir de las líneas identificadas por Duan et al. (2005), sería posible desarrollar una herramienta de modelado web, que permita la definición online de procesos basados en el modelo de forma colaborativa, esto es, permitiendo la participación de distintos expertos en la definición de procesos, y el intercambio de distintas definiciones basadas en el modelo entre ellos. La creación de un portal de donde expertos de diferentes dominios puedan definir su conocimiento abre grandes posibilidades para compartir el conocimiento que ya se tiene y ampliarlo con el que pueden aportar otros expertos con intereses comunes. Como la ejecución de los procesos definidos tam-

bién se realiza a través de la web, no solo se trata de un portal para compartir el conocimiento sino que además aportaría soluciones.

En la línea del desarrollo de la web semántica y las redes sociales, resultaría interesante proporcionar etiquetado semántico de los objetos de conocimiento representados a través de la ontología. El etiquetado puede estar basado en el dominio concreto del proceso representado o en metainformación relevante para los objetivos de los expertos que participan. El etiquetado de los distintos elementos en un entorno de definición del conocimiento colaborativo, facilitaría la búsqueda de conocimiento por parte de los expertos, pudiendo localizar definiciones de procesos relacionados con su área de conocimiento para reutilizarlo o colaborar en su mejora. Etiquetando no solo el conocimiento, sino también los expertos que lo definen, facilitaría la búsqueda por parte de un experto de otros expertos relacionados con su dominio para colaborar.

El etiquetado de los objetos de representación abre también la posibilidad de investigar en la posibilidad de generar nuevo conocimiento a partir de la metainformación proporcionada para las distintas situaciones y acciones de un proceso. Por ejemplo, se podrían buscar automáticamente situaciones semánticamente similares entre diferentes procesos o determinar nuevas decisiones en función del conocimiento aportado por las etiquetas. Con ello se podrían sugerir decisiones y situaciones para un problema determinado.

Finalmente, también a partir del etiquetado, se puede extender el módulo de depuración de los procesos en la herramienta de modelado, proporcionando nuevas funcionalidades que expliquen las causas de un posible fallo en el proceso y las posibles interacciones del mismo con otras situaciones o acciones relacionadas.

10.3. Publicaciones Relacionadas

A partir de los resultados obtenidos en la presente investigación se han realizado las siguientes publicaciones:

1. *A Review of Conventional and Knowledge Based Systems for Machining Price Quotation*. Ángel García-Crespo, Belén Ruiz-Mezcua, José Luis López-Cuadrado, Israel González-Carrasco. **Journal of Intelligent Manufacturing**. 2009. DOI: 10.1007/s10845-009-0335-1. **JCR 1.018 (2008)**.

Este artículo revisa los sistemas basados en conocimiento implementados para la estimación de costes y precios en el área de fabricación de piezas mecanizadas. Se incluyen aspectos sobre cómo deben desarrollarse los futuros sistemas basados en conocimiento pa-

ra la estimación de este tipo de precio. Este artículo describe el problema de fondo y la motivación de esta tesis.

2. *Development of a Web Based Quotation Expert System for Machined Parts*. Lopez-Cuadrado, J. L.; García-Crespo, A.; Ruiz-Mezcua, B.; Gomez, J. M. and Gonzalez-Carrasco, I. **International Journal of Computer Applications in Technology**, Special Issue: Computer Applications in Research and Development of Complex Mechanical Systems. En impresión.

En esta revista se hace una descripción con mayor nivel de detalle del problema de la estimación en las compañías de fabricación versátil y se detalla la automatización del proceso, presentando los resultados obtenidos por el proceso automatizado en comparación con las estimaciones de un experto.

3. *Arquitecturas para Gestión de Conversaciones B2B Basadas en Conocimiento*. Lopez-Cuadrado, J. L.; Gomez, J. M.; García-Crespo, A.; Ruiz-Mezcua, B.; González-Carrasco, I. and Alor-Hernandez, G.

Proceedings of the 7th IEEE Latin-American Conference in Software Engineering and Knowledge Engineering, 2008.

En este congreso se presentan distintas alternativas sobre la automatización de un proceso de decisión a partir del modelo de representación propuesto y el motor de solución desarrollado para la validación de las hipótesis planteadas al comienzo de la investigación.

4. *A Fully Web Oriented Expert System Tool*. Lopez-Cuadrado, J. L.; Gonzalez-Carrasco, I.; García-Crespo, Á. and Ruiz-Mezcua, B. **IADIS International Conference WWW/Internet**. Murcia, Spain., 2006.

En este congreso se realizó una descripción de la herramienta de modelado desarrollada para la representación del conocimiento a partir del modelo propuesto en esta tesis.

5. *Applying an Expert System to Factory Budget Calculations*. López-Cuadrado, J. L.; González-Carrasco, I.; García-Crespo, A.; Ruiz-Mezcua, B.; Fernandez-Heijnen, Y. and Cantero-Guisández, J. L.

10th IEEE International Conference on Emerging Technologies and Factory Automation. Catania, Italy, 2005.

En este congreso se realizó una introducción a los conceptos principales del modelo de representación y se realizó una pequeña descripción del proceso automatizado en la fábrica donde se llevó a cabo el caso de estudio.

Bibliografía

- Abdalla, H. S. and Knight, J.: 1994, An expert system for concurrent product and process design of mechanical parts, *Proceedings of the Institution of Mechanical Engineers, Part B (Journal of Engineering Manufacture)* **208**, 167–172.
- Adams, J.: 2001, The feasibility of distributed web based expert systems, *IEEE International Conference on Systems, Man and Cybernetics*.
- Aderoba, A.: 1997, A generalised cost-estimation model for job shops, *International Journal of Production Economics* **53**, 257–263.
- Ahmadi, J. and Tirupati, D.: 1996, Oars: an order analysis and rescheduling system, *Production Planning and Control* **7**(1), 86–95.
- Alavi, M. and Leidner, D.: 2001, Knowledge management and knowledge management systems: Conceptual foundations and research issues, *MIS Quarterly* **25**(1), 107–136.
- Alexakos, C., Kalogeras, A., Likothanassis, S., Gialelis, J. and Koubias, S.: 2005, Workflow - coordinated integration of enterprise/ industrial systems based on a semantic service - oriented architecture, Vol. 2, pp. 7 pp.–.
- Amaro, G., Hendry, L. and Kingsman, B.: 1999, Competitive advantage, customisation and a new taxonomy for non make-to-stock companies, *International Journal of Operations & Production Management* **19**(4), 349–371.
- Andonoff, E., Bouzguenda, L. and Hanachi, C.: 2005, Specifying workflow web services using petri nets with objects and generating of their owl-s specifications, *E-Commerce and Web Technologies*, pp. 41–52.

- Arezoo, B., Ridgway, K. and Al-Ahmari, A. M.: 2000, Selection of cutting tools and conditions of machining operations using an expert system, *Computers in Industry* **42**, 43–58.
- Arroyo, S., Sicilia, M.-A. and Dodero, J.-M.: 2007, Choreography frameworks for business integration: Addressing heterogeneous semantics, *Computers in Industry* **58**(6), 487–503.
- Astrova, I., Korda, N. and Kalja, A.: 2007, Storing owl ontologies in sql relational databases, *International Journal of Electrical, Computer, and Systems Engineering* **1**(4), 242–247.
- Balachandra, R.: 2000, An expert system for new product development projects, *Industrial Management + Data Systems* **100**(7), 317–324.
- Beil, D. and Wein, L. M.: 2003, An inverse-optimization-based auction mechanism to support a multiattribute rfq process, *Management science* **49**(11), 1529 – 1545.
- Ben-Arieh, D.: 2000, Cost estimation system for machined parts, *International Journal of Production Research* **38**(17), 4481–4494.
- Ben-Arieh, D. and Li, Q.: 2003, Web-based cost estimation of machining rotational parts, *Production Planning and Control* **14**(8), 778–788.
- Bidanda, B., Kadidal, M. and Billo, R. E.: 1998, Development of an intelligent castability and cost estimation system, *International journal of production research* **36**(2), 547.
- Boothroyd, G. and Reynolds, C.: 1989, Approximate cost estimates for typical turned parts, *Journal of Manufacturing Systems* **8**(3), 185 – 193.
- Bouaziz, Z., Ben Younes, J. and Zghal, A.: 2006, Cost estimation system of dies manufacturing based on the complex machining features, *International Journal of Advanced Manufacturing Technology* **28**(3), 262–271.
- Brachman, R. J. and Levesque, H. J.: 2004, *Knowledge Representation and Reasoning*, Elsevier.
- Brewster, C. and O’Hara, K.: 2004, Knowledge representation with ontologies: the present and future, *IEEE Intelligent Systems* **19**(1), 72–81.
- Brinke, E. T., Lutters, E., Streppel, T. and Kals, H.: 2004, Cost estimation architecture for integrated cost control based on information management, *International Journal of Computer Integrated Manufacturing* **17**, 534–545.
- Buchanan, B. G. and Feigenbaum, E. A.: 1978, Dendral and meta-dendral: Their applications dimension, *Artificial Intelligence* **11**, 923–925.
- Calosso, T., Cantamessa, M., Vu, D. and Villa, A.: 2003, Production planning and order acceptance in business to business electronic commerce, *International Journal of Production Economics* **85**, 233–249.

- Chan, D.: 2003, Expert system for product manufacturability and cost evaluation, *Materials and Manufacturing Processes* **18**, 313–322.
- Chan, D. S. K. and Lewis, W. P.: 2000, The integration of manufacturing and cost information into the engineering design process, *International journal of production research* **38**(17), 4413–.
- Chan, F.: 2005, Application of a hybrid case-based reasoning approach in electroplating industry, *Expert Systems with Applications* **29**, 121–130.
- Chen, K.-Z., X.-A., F. and Zhang, B.-B.: 2003, Development of computer-aided quotation system for manufacturing enterprises using axiomatic design, *International Journal of Production Research* **41**(1), 171–191.
- Chen, M. and Sairamesh, J.: 2006, A knowledge model-driven recommender system for business transformation, *IEEE International Conference on Services Computing (SCC'06)*, pp. 77–84.
- Chen, T.-Y.: 2008, Knowledge sharing in virtual enterprises via an ontology-based access control approach, *Computers in Industry* **59**(5), p502 – 519.
- Chougule, R. G. and Ravi, B.: 2006, Casting cost estimation in an integrated product and process design environment, *International Journal of Computer Integrated Manufacturing* **19**, 676–688.
- Collan, M. and Liu, S.: 2003, Fuzzy logic and intelligent agents: towards the next step of capital budgeting decision support, *Industrial Management + Data Systems* **103**(6), 410–422.
- Cooper, R. and Kaplan, R.: 1988, Measure costs right: Make the right decisions., *Harvard Business Review* **66**(5), 96–103.
- Cunningham, A. and Smart, R.: 1993, Computer-aided parts estimation, *AI Magazine* **14**, 39–49.
- Dellas, N., Koutsoloukas, E., Lioudakis, G., Kaklamani, D. and Venieris, I.: 2007, A performance comparison of ontolog reasoning and rule engines, *Mobile and Wireless Communications Summit, 2007. 16th IST* pp. 1–5.
- Dewhurst, P. and Boothroyd, G.: 1988, Early cost estimating in product design, *Journal of Manufacturing Systems* **7**(3), 183 – 191.
- Dietze, S., Gugliotta, A. and Domingue, J.: 2008, Situation-driven processes for semantic web services, *32nd Annual IEEE International Computer Software and Applications, 2008. COMPSAC 08.*, pp. 987 – 992.
- Dokas, I. M.: 2005, Developing web sites for web based expert systems: A web engineering approach, *Proceedings of the Information Technologies in Environmental Engineering (ITEE2005)*, pp. 202–217.

- Dokas, I. M. and Alapetite, A.: 2006, A development process meta-model for web-based expert systems: the web engineering point of view, *Technical report*, Riso National Laboratory. Denmark.
- Doney, L. D.: 1971, Machine job shop prepares quotations by computer, *Industrial Engineering* **3**, 47–51.
- Dong, B., Qi, G., Gu, X. and Wei, X.: 2008, Web service-oriented manufacturing resource applications for networked product development, *Advanced Engineering Informatics* **22**(3), 282–295.
- Downs, K. A. and Trappey, A. J. C.: 1992, A sequential block approach to the design of a cost estimating system, *Computers and Industrial Engineering* **23**, 423–426.
- Duan, Y., Edwards, J. S. and Xu, M. X.: 2005, Web-based expert systems: benefits and challenges, *Information & Management* **42**, Issue 6, 799–811.
- Duda, R., Hart, P. E., Nilsson, N. J., Reboh, R., Slocum, J. and Sutherland, G.: 1978, Development of the prospector consultation system for mineral exploitation, *Technical report*, Stanford Research Institute, Menlo Park, CA.
- Duverlie, P. and Castelain, J. M.: 1999, Cost estimation during design step: parametric method versus case based reasoning method, *International Journal of Advanced Manufacturing Technology* **15**(12), 895–906.
- Efstathiou, J., Calinescu, A. and Blackburn, G.: 2002, A web-based expert system to assess the complexity of manufacturing organizations, *Robotics and Computer-Integrated Manufacturing* **18**(3), 305–311.
- Eppinette, M., Inman, R. A. and Pick, R. A.: 1997, Expert systems and the implementation of quality customer service, *Industrial Management + Data Systems* **97**(1), 63–68.
- Er, A. and Dias, R.: 2000, A rule-based expert system approach to process selection for cast components, *Knowledge-Based Systems* **13**(4), 225–234.
- Feigenbaum, E. A.: 1982, Knowledge engineering in the 1980s, *Technical report*, Dept. of Computer Science, Stanford University, Stanford, CA.
- Fensel, D.: 2000, *Ontologies: a silver bullet for knowledge management and electronic commerce*, Springer.
- Fensel, D., Erdmann, M. and Studer, R.: 1997, Ontology groups: Semanticall enriched subnets of the www, *Proceedings of the 1st International Workshop Intelligent Information Integration during the 21st German Annual Conference on Artificial Intelligence*, pp. 9–12.
- Fernandez Heijnen, Y.: 2004, *Desarrollo de un sistema experto para la realización de presupuestos de piezas mecanizadas en la empresa APRIM*, PFC. Universidad Carlos III de Madrid.

- Filipowska, A., Kaczmarek, M., Starzecka, M., Stolarski, P. and Walczak, A.: 2008, Semantic enterprise description for the needs of business process automation, *32nd Annual IEEE International Computer Software and Applications, 2008. COMPSAC '08.*, pp. 987 – 992.
- Forgy, C.: 1982, Rete: A fast algorithm for the many pattern / many object patten match poroblem, *Artificial Intelligence* **19**, 17–37.
- Fridman-Noy, N. and Hafner, C. D.: 1997, The state of the art in ontology design, *AI Magazine* **18**(3), 53–74.
- Giarranto, J. and Riley, G.: 1998, *Expert Systems: principles and programming*, 3rd edn, PWS Publishing Company.
- Gómez, A., Juristo, N., Montes, C. and Pazos, J.: 1997, *Ingenieria del Conocimiento*, Centro de Estudios Ramón Areces.
- Gomez-Perez, A., Fernandez-Lopez, M. and Corcho, O.: 2004, *Ontological engineering : with examples from the areas of knowledge management, e-commerce and the semantic web*, Springer.
- González, A. and Dankel, D.: 1993, *The Engineering of Knowledge Based Systems*, Prentice Hall.
- Grove, R.: 2000, Internet-based expert systems, *Expert Systems* **17**(3), 129–135.
- Gruber, T.: 1993, A translation approach to portable ontology specifications., *Knowledge Acquisition* **5**(2), 199–220.
- Guimaraes, T., Yoon, Y. and O’Neal, Q.: 1995, Success factors for manufacturing expert system development, *Computers & Industrial Engineering* **28**, 545–559.
- Helo, P. T.: 2006, Product configuration analysis with design structure matrix, *Industrial Management + Data Systems* **106**(7), 997–1011.
- Hicks, C., McGovern, T. and Earl, C.: 2000, Supply chain management: a strategic issue in engineer to order manufacturing, *International Journal of Production Economics* **65**(2), 179–190.
- Horrocks, I., Boley, P. F. P.-S. H., Tabet, S., Grosf, B. and Dean, M.: 2004, Swrl: A semantic web rule language combining owl and ruleml., W3C member submission. <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>
- Huang, G. Q. and Mak, K. L.: 2001a, Issues in the development and implementation of web applications for product design and manufacture, *International Journal of Computer Integrated Manufacturing* **14**(1), 125–135.
- Huang, G. Q. and Mak, K. L.: 2001b, Web-integrated manufacturing: recent developments and emerging issues, *International Journal of Computer Integrated Manufacturing* **14**(1), 3–13.

- Hunter, R., Perez, J., Marquez, J. and Hernandez, J.: 2007, Modeling the integration between technological product specifications and inspection process, *Journal of Materials Processing Technology* **191**(1-3), p34 – 38.
- Hvam, L. and Have, U.: 1998, Re-engineering the specification process, *Business Process Management* **4**(1), 25–43.
- Hvam, L., Malis, M., Hansen, B. and Riis, J.: 2004, Reengineering of the quotation process: application of knowledge based systems, *Business Porocess Management Journal* **10**(2), 200–213.
- Hvam, L., Pape, S. and Nielsen, M. K.: 2006, Improving the quotation process with product configuration, *Computers in Industry* **57**(7), 607–621.
- Jahna-Shahi, H., Shayan, E. and Masood, S. H.: 2001, Multivalued fuzzy sets in cost/time estimation of flat plate processing, *International Journal of Advanced Manufacturing Technology* **17**(10), 751–759.
- Jeong, D., Choi, M., Jeon, Y.-S., Han2, Y.-H., Jeong, Y.-S. and Han, S.-K.: 2006, A novel memory-oriented owl storage system, *Frontiers of High Performance Computing and Networking ISPA 2006 Workshops*, pp. 542–549.
- Jiang, B. C. and Hsu, C.: 2003, Development of a fuzzy decision model for manufacturability evaluation, *Journal of Intelligent Manufacturing* **14**, 169–181.
- Jiao, J. and Helander, M. G.: 2006, Development of an electronic configure-to-order platform for customized product development, *Computers in Industry* **57**(3), 231–244.
- Jung, J., Choi, I. and Song, M.: 2007, An integration architecture for knowledge management systems and business process management systems, *Computers in Industry* **58**(1), 21–34.
- Jung, J.-Y.: 2002, Manufacturing cost estimation for machined parts based on manufacturing features, *Journal of Intelligent Manufacturing* **13**, 227–238.
- Kadidal, M. and Bidanda, B.: 1993, A castability expert system, *Computers & Industrial Engineering* **25**, 99–102.
- Kamsu Foguem, B., Coudert, T., Béler, C. and Geneste, L.: 2008, Knowledge formalization in experience feedback processes: An ontology-based approach, *Computers in Industry* **59**(7), 694–710.
- Kennedy, K. and Shao, D.: 1989, The design, implementation and evaluation of a diagnostic expert system for the work order procedures for small manufacturing companies, *Computers & Industrial Engineering* **17**, 95–100.
- Kim, W., song, Y. U. and Hong, J. S.: 2005, Web enabled expert system using hyperlink-based inference, *Expert Systems with Applications* **28**, 79–91.

- Kingsman, B. G. and de Souza, A. A.: 1997, A knowledge-based decision support system for cost estimation and pricing decisions in versatile manufacturing companies, *International Journal of Production Economics* **53**, 119–139.
- Kingsman, B., Hendry, L., Mercer, A. and de Souza, A.: 1996, Responding to customer enquiries in make-to-order companies: problems and solutions, *International Journal of Production Economics* **46-47**, 219–231.
- Ko, K. H., Pochiraju, K. and Manoochehri, S.: 2007, An embedded system for knowledge -based cost evaluation of molded parts, *Knowledge-Based Systems* **20**, 291–299.
- Koonce, D., Judd, R., Sormaz, D. and Masel, D. T.: 2003, A hierarchical cost estimation tool, *Computers in Industry* **50**(3), 293–302.
- Koumoutsos, G. and Thramboulidis, K.: 2008, Towards a knowledge-base for building complex, proactive and service-oriented e-negotiation systems, *2008 International MCETECH Conference on e-Technologies*, pp. 178–189.
- Kumar, R. and Midha, P. S.: 2004, An objective approach for identifying the strategic components of a pdm system, *Industrial Management + Data Systems* **104**(1), 56–67.
- Lan, C. H.: 2002, Optimal control of a multistage machining operation on a computer numerically controlled machine, *International Journal of Advanced Manufacturing Technology* **20**(11), 807–811.
- Lan, H. and Ding, Y.: 2007, Price quotation methodology for stereolithography parts based on stl model, *Computer and Industrial Engineering* **52**(2), 241–256.
- Layer, A., TEN Brinke, E., Van Houten, F., Kals, H. and Haasis, S.: 2002, Recent and future trends in cost estimation, *International Journal of Computer Integrated Manufacturing* **15**, 499–510.
- Li, S.: 2005, A web-enabled hybrid approach to strategic marketing planning: Group delphi+a web-based expert system, *Expert Systems with Applications* **29**, 393–400.
- Liao, S.-H.: 2005, Expert system methodologies and applications - a decade review from 1995 to 2004, *Expert Systems with Applications* **28**, 93–103.
- Lin, Z., Hamalainen, M. and Winston, A.: 2002, *Expert Systems. The technology of knowledge management and decision making for de 21st century*, number Knowledge-Based Appr, 4 edn, Academic Press, pp. 1155–1172.
- Liu, Y. and Nie, G.: 2007, Implementation of supply chains coordination using semantic web service composition, *IEEE International Conference on Service-Oriented Computing and Applications, 2007. SOCA '07.*, pp. 249 – 254.

- Locascio, A.: 2000, Manufacturing cost modeling for product design, *International Journal of Flexible Manufacturing Systems* **12**, 207–217.
- Lopez-Cuadrado, J. L., García-Crespo, A., Ruiz-Mezcua, B., Gomez, J. M. and Gonzalez-Carrasco, I.: To appear, Development of a web based quotation expert system for machined parts, *International Journal of Computer Applications in Technology* **Special Issue: Computer Applications in Research and Development of Complex Mechanical Systems**.
- López-Cuadrado, J. L., González-Carrasco, I., García-Crespo, A., Ruiz-Mezcua, B., Fernandez-Heijnen, Y. and Cantero-Guisández, J. L.: 2005, Applying an expert system to factory budget calculations, *10th IEEE International Conference on Emerging Technologies and Factory Automation. Catania, Italy*.
- Maropoulos, P. G. and Baker, R. P.: 2000, Integration of tool selection with design: Part 1. feature creation and selection of operations and tools, *Journal of Materials Processing Technology*, **107**(1-3), 127–134.
- Maropoulos, P. G., Baker, R. P. and Paramor, K. Y. G.: 2000, Integration of tool selection with design: Part 2: Aggregate machining time estimation, *Journal of Materials Processing Technology*, **107**(1-3), 135–142.
- Maropoulos, P. G., Bramall, D. G. and McKay, K. R.: 2003, Assessing the manufacturability of early product designs using aggregate process models, *Proceedings of the Institution of Mechanical Engineers – Part B – Engineering Manufacture* **217**(9), 1203.
- Masmoudi, F., Bouaziz, Z. and Hachicha, W.: 2007, Computer-aided cost estimation of weld operations, *The International Journal of Advanced Manufacturing Technology* **33**(3-4), 262–271.
- McCarthy, J. and Hayes, P. J.: 1969, Some philosophical problems from the standpoint of artificial intelligence, in B. Meltzer and D. Michie (eds), *Machine Intelligence 4*, Edinburgh University Press, pp. 463–502.
- McDermott, J. and Bachant, J.: 1984, R1 revisited: Four years in the trenches, *AI Magazine* **3**, 21–32.
- Meditskos, G. and Bassiliades, N.: 2008, A rule-based object-oriented owl reasoner, *Knowledge and Data Engineering, IEEE Transactions on* **20**(3), 397–410.
- Metaxiotis, K. and Psarras, J.: 2003, Expert systems in business: applications and future directions for the operations researcher, *Industrial Management + Data Systems* **103**(5), 361–368.
- Metaxiotis, K. S., Askounis, D. and Psarras, J.: 2002, Expert systems in production planning and scheduling: a state-of-the-art survey, *Journal of Intelligent Manufacturing* **13**, 253–260.

- Metaxiotis, K. S., Psarras, J. E. and Askounis, D. T.: 2002, Genesys: an expert system for production scheduling, *Industrial Management + Data Systems* **102**(6), 309–317.
- Mohamed, A. and Celik, T.: 1998, An integrated knowledge-based system for alternative design and materials selection and cost estimating, *Expert Systems with Applications* **14**, 329–339.
- Mohamed, A. and Celik, T.: 2002, Knowledge based-system for alternative design, cost estimating and scheduling, *Knowledge-Based Systems* **15**(3), 177–188.
- Montagut, F. and Molva, R.: 2005, Enabling pervasive execution of workflows, *2005 International Conference on Collaborative Computing: Networking, Applications and Worksharing* p. 10 pp.
- Montagut, F., Molva, R. and Tecumseh Golega, S.: 2008, The pervasive workflow: a decentralized workflow system supporting long-running transactions, *IEEE Transactions on Systems, Man, and Cybernetics-Part C (Applications and Reviews)* **38**(3), p319 – 333.
- Motik, B., Horrocks, I. and Sattler, U.: 2007, Bridging the gap between owl and relational databases, *WWW '07: Proceedings of the 16th international conference on World Wide Web*, ACM, New York, NY, USA, pp. 807–816.
- Muntslag, D. R.: 1994, Profit and risk evaluation in customer driven engineering and manufacturing, *International Journal of Production Economics* **36**, 97–107.
- Newell, A. and Simon, H. A.: 1972, *Human Problem Solving*, Prentice-Hall.
- Niazi, A., Dai, J., Balabani, S. and Seneviratne, L.: 2006, Product cost estimation: Technique classification and methodology review, *Journal of Manufacturing Science & Engineering* **128**(2), 563–575.
- Nikolopoulos, K. and Assimakopoulos, V.: 2003, Theta intelligent forecasting information system, *Industrial Management + Data Systems* **103**(9), 711–726.
- Noor, N. M., Papamichail, K. and Warboys, B.: 2006, An integrated web-based decision support system for tendering processes, *Join International Conference on Computing and Decision Making in Civil and Building Engineering*, pp. 238–247.
- Nurminen, J. K., Karonen, O. and Hatonen, K.: 2003, What makes expert systems survive over 10 years - empirical evaluation of several engineering applications., *Expert Systems with Applications* **43**, 199–211.
- Ou-Yang, C. and Lin, T. S.: 1997, Developing an integrated framework for feature-based early manufacturing cost estimation, *International Journal of Advanced Manufacturing Technology* **13**(9), 618–629.
- Pan, J. Z.: 2007, A flexible ontology reasoning architecture for the semantic web, *IEEE Transactions on Knowledge and Data Engineering* **19**(2), 246–260.

- Pandit, A. and Zhu, Y.: 2007, An ontology-based approach to support decision-making for the design of eto (engineer-to-order) products, *Automation in Construction* **16**(6), 759–770.
- Park, J. and Seo, K.: 2004, Incorporating life-cycle cost into early product development, *Proceedings of the Institution of Mechanical Engineers, Part B (Journal of Engineering Manufacture)* **218**, 1059–1066.
- Park, K. and Kim, S.: 1998, Artificial intelligence approaches to determination of cnc machining parameters in manufacturing: a review, *Artificial Intelligence in Engineering* **12**, 127–134.
- Parkinson, E. L., Hailey, M. L., Lo, C. F., Whitehead, B. A., Shi, G. Z. and Garrison, G. W.: 1994, Integration architecture of expert systems, neural networks, hypertext, and multimedia can provide competitive opportunities for industrial applications, *Computers & Industrial Engineering* **27**, 269–272.
- Pedrinaci, C., Domingue, J., Brelage, C., van Lessen, T., Karastoyanova, D. and Leymann, F.: 2008, Semantic business process management: Scaling up the management of business processes, *IEEE International Conference on Semantic Computing*, pp. 546 – 553.
- Pham, D. and Pham, P.: 1999, Artificial intelligence in engineering, *International Journal of Machine Tools & Manufacture* **39**, 937–949.
- Poernomo, I. and Umarov, T.: 2008, Business process development in semantically-enriched environment, *Fifth International Conference on Information Technology: New Generations, 2008. ITNG 2008*, pp. 57–62.
- Qiqin, C., Shan, J., Weidong, L. and Junmin, D.: 1996, Application of ie to the plan of production line in factory, *Computers & Industrial Engineering* **31**(3), 561–564.
- Rao, M. P. and Miller, D. M.: 2004, Expert systems applications for productivity analysis, *Industrial Management + Data Systems* **104**(9), 776–785.
- Rehg, J. A.: 2005, *Computer-integrated manufacturing*, 3rd edn, Prentice Hall.
- Ruffo, M., Tuck, C. and Hague, R.: 2006, Cost estimation for rapid manufacturing - laser sintering production for low to medium volumes, *Proceedings of the Institution of Mechanical Engineers, Part B (Journal of Engineering Manufacture)* **220**, 1417–1427.
- Seo, K.-K., Park, Jang and Wallace, D.: 2002, Prediction of the life cycle cost using statistical and artificial neural network methods in conceptual product design, *International Journal of Computer Integrated Manufacturing* **15**, 541–554.
- Sharma, R. and Gao, J. X.: 2002, A progressive design and manufacturing evaluation system incorporating step ap224, *Computers in Industry* **47**(2), 155–167.

- Shehab, E. and Abdalla, H.: 2002a, An intelligent knowledge-based system for product cost modelling, *International Journal of Advanced Manufacturing Technology* **19**(1), 49–65.
- Shehab, E. M. and Abdalla, H. S.: 2001, Manufacturing cost modelling for concurrent product development, *Robotics and Computer-Integrated Manufacturing* **17**(4), 341–353.
- Shehab, E. M. and Abdalla, H. S.: 2002b, A design to cost system for innovative product development, *Proceedings of the Institution of Mechanical Engineers, Part B (Journal of Engineering Manufacture)* **216**, 999–1019.
- Shehab, E. M. and Abdalla, H. S.: 2006, A cost-effective knowledge-based reasoning system for design for automation, *Proceedings of the Institution of Mechanical Engineers, Part B (Journal of Engineering Manufacture)* **220**, 729–743.
- Shen, D., Yu, G., Kou, Y., Nie, T. and Zhao, Z.: 2007, Resolving heterogeneity of web-service composition in network manufacturing based on ontology, *International Journal of Computer Integrated Manufacturing* **20**(2), 222–233.
- Shortliffe, E. H.: 1976, *Computer-Based Medical Consultations: MYCIN.*, American Elsevier.
- Silva, C., Roque, L. and Almeida, A.: 2006, Mapp - a web-based decision support system for the mould industry, *Decision Support Systems* **42**(2), 999–1014.
- Siong, L. B., Imao, T., Yoshida, H., Goto, K., Koh, S. L., Lim, D., Chin, L. and Gan, S. C.: 1992, Integrated modular fixture design, pricing and inventory control expert system, *International Journal of Production Research* **30**(9), 2019–2044.
- Slater, P. J.: 1999, Pconfig: a web-based configuration tool for configure-to-order products, *Knowledge-Based Systems* **12**(5), 223–230.
- Stockton, D. and Wang, Q.: 2004, Developing cost models by advanced modelling technology, *Proceedings of the Institution of Mechanical Engineers, Part B (Journal of Engineering Manufacture)* **218**, 213–224.
- Studer, R., Benjamings, V. R. and Fensel, D.: 1998, Knowledge engineering: Principles and methods, *Data and Knowledge Engineering* **25**(1-2), 161–167.
- Studer, R., Eriksson, H., Gennari, J. H., Tu, S. W., Fensel, D. and Musen, M.: 1996, Ontologies and the configuration of problem-solving methods, in B. R. Gaines and M. A. Munsen (eds), *Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*.
- Taiber, J. G.: 1996, Optimization of process sequences considering prismatic workpieces, *Advances in Engineering Software*, **25**(1), 41–50.

- Tang, D. B., Zheng, L. and Li, Z. Z.: 2001, An intelligent feature-based design for stamping system, *International Journal of Advanced Manufacturing Technology* **18**(3), 193–200.
- Tang, D., Eversheim, W., Schuh, G. and Chin, K. .-.: 2003, Concurrent metal stamping part and die development system, *Proceedings of the Institution of Mechanical Engineers, Part B (Journal of Engineering Manufacture)* **217**, 805–825.
- Thomson, A. and Willoughby, I.: 2004, A web-based expert system for advising on herbicide use in great britain, *Computers and Electronics in Agriculture* **42**(1), 43–49.
- Tran, V. X. and Tsuji, H.: 2007, Owl-t: An ontology-based task template language for modeling business processes, *Fifth International Conference on Software Engineering Research, Management and Applications*, pp. 101–108.
- Trinkunas, J. and Vasilecas, O.: 2007, A graph oriented model for ontology transformation into conceptual data model, *Information Technology And Control* **36**(1A), 126–132.
- Tu, Y. L., Xie, S. Q. and Fung, R. Y.: 2007, Product development cost estimation in mass customization, *IEEE Transactions on Engineering Management* **54**(1), 29–40.
- Veeramani, D. and Joshi, P.: 1997, Methodologies for rapid and effective response to requests for quotation (rfqs), *IIE Transactions* **29**(10), 825–838.
- Venkatadri, U., Srinivasan, A., Montreuil, B. and Saraswat, A.: 2006, Optimization-based decision support for order promising in supply chain networks, *International Journal of Production Economics* **103**, 117–130.
- Verstichel, S., Strobbe, M., Simoens, P., Turck, F. D., Dhoedt, B. and Demeester, P.: 2008, Distributed reasoning for context-aware services through design of an owl meta-model, *Fourth International Conference on Autonomic and Autonomous Systems* **0**, 70–75.
- Vidal, J., Lama, M., Bugarin, A. and Barro, S.: 2003, Workflow-based information system for furniture budgeting, *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*.
- Vitvar, T., Zaremba, M. and Mocan, A.: 2008, Execution model for heterogeneous web services, *SERVICES '08. IEEE Congress on Services - Part I, 2008.*, pp. 219 – 222.
- Vysniauskas, E. and Nemuraite, L.: 2006, Transforming ontology representation from owl to relational database, *INFORMATION TECHNOLOGY AND CONTROL* **35**(3A), 333–343.
- Wang, H., Zhou, X. H. and Ruan, X. .-.: 2003, Research on injection mould intelligent cost estimation system and key technologies, *International Journal of Advanced Manufacturing Technology* **21**(3), 215–222.

- Wang, S., Shen, W. and Hao, Q.: 2006, An agent-based web service workflow model for inter-enterprise collaboration, *Expert Systems with Applications* **31**(4), 787–799.
- Wei, Y. and Egbelu, P. J.: 2000, A framework for estimating manufacturing cost from geometric design data, *International Journal of Computer Integrated Manufacturing* **13**, 50–63.
- Winston, P.: 1992, *Artificial Intelligence*, number Book, Whole; Book, W, Addison Wesley Publishing.
- Wu, Z., Eadon, G., Das, S., Chong, E., Kolovski, V., Annamalai, M. and Srinivasan, J.: 2008, Implementing an inference engine for rdfs/owl constructs and user-defined rules in oracle, *IEEE 24th International Conference on Data Engineering, 2008. ICDE 2008*. pp. 1239–1248.
- Xie, S. Q.: 2006, A decision support system for rapid one-of-a-kind product development, *International Journal of Advanced Manufacturing Technology* **28**, 643–652.
- Xie, S. Q., Tu, P. L., Aitchison, D., Dunlop, R. and Zhou, Z. D.: 2001, A www-based integrated product development platform for sheet metal parts intelligent concurrent design and manufacturing, *International Journal of Production Research* **39**(17), 3829–3852.
- Xie, S. Q., Tu, Y. L., Liu, J. Q. and Zhou, Z. D.: 2001, Integrated and concurrent approach for compound sheet metal cutting and punching, *International Journal of Production Research* **39**(6), 1095–1112.
- Xie, S., Tu, Y., Fung, R. and Zhou, Z.: 2003, Rapid one-of-a-kind product development via the internet: a literature review of the state-of-the-art and a proposed platform, *International Journal of Production Research* **41**(18), 4257–98.
- Ye, Y., Yang, D., Jiang, Z. and Tong, L.: 2006, A knowledge- and workflow-based system for supporting order fulfillment process in the build-to-order supply chains, *The Semantic Web - ASWC 2006. First Asian Semantic Web Conference.*, pp. 711–724.
- Yu, C.-C.: 2004, A web-based consumer-oriented intelligent decision support system for personalized e-services, *ICEC '04: Proceedings of the 6th international conference on Electronic commerce*, ACM Press, pp. 429–437.
- Zadeh, L. A.: 1965, Fuzzy sets, *Information and Control* **8**, 338–353.
- Zhang, Y. F., Fuh, J. Y. and Chan, W. T.: 1996, Feature-based cost estimation for packaging products using neural networks, *Computers in Industry* **32**(1), 95–113.
- Zhao, W. and Liu, J.: 2008a, Owl/swrl representation methodology for express-driven product information model part i. implementation methodology, *Computers in Industry* **59**(6), p580 – 589.
- Zhao, W. and Liu, J.: 2008b, Owl/swrl representation methodology for express-driven product information model part ii. practice, *Computers in Industry* **59**(6), p590 – 600.

Zhou, J., Ma, L., Liu, Q., Zhang, L., Yu, Y. and Pan, Y.: 2006, Minerva: A scalable owl ontology storage and inference system, *The Semantic Web (ASWC 2006)*, pp. 429–443.