



Universidad de Castilla-La Mancha

Escuela Superior de Informática

Departamento de Informática

Tesis Doctoral

MAON: Un Método de Análisis Orientado a la Necesidad

Volumen I: Memoria

Doctorando: Óscar Dieste Tubío

Directoras: Marcela Genero Bocco

Ana María Moreno Sánchez-Capuchino

Resumen

El análisis es una de las tareas más críticas de la Ingeniería de Requisitos, debido a la enorme importancia de los objetivos que persigue: (1) entender el problema a resolver; (2) desarrollar modelos conceptuales (MCs), los cuales representan el entendimiento logrado; y (3) definir las características de la solución, esto es, identificar los requisitos que serán satisfechos por el futuro sistema software.

Los MCs juegan un papel central durante el análisis, ya que permiten entender y representar la necesidad del usuario de un modo independiente de la implementación. No obstante, diversos autores han señalado que los MCs usados en la actualidad están orientados hacia aproximaciones de desarrollo específicas. Esta orientación tiene dos repercusiones: (1) los MCs poseen ligaduras computacionales, esto es, los MCs permiten incluir aspectos específicos de la implementación en los modelos del dominio y (2) los MCs prescriben el subsiguiente proceso de desarrollo, durante el cual dichos modelos son transformados, más o menos directamente, a modelos de diseño.

Los problemas anteriores implican que los MCs utilizados actualmente no son adecuados para realizar el análisis. En este trabajo de Tesis se propone una aproximación alternativa a la modelización conceptual. Esta alternativa, denominada “Método de Análisis Orientado a la Necesidad (MAON)”, se caracteriza por: (1) emplear formalismos de representación, denominados Modelos Conceptuales Genéricos (MCGs), que no presuponen ningún tipo de implementación; (2) definir un proceso de análisis detallado; y (3) derivar, a partir de los MCGs, el MC más adecuado (esto es, un MC utilizado actualmente, como los diagramas de flujo de datos, casos de uso, etc.) para proseguir con el desarrollo utilizando los métodos utilizados actualmente.

Abstract

Analysis is one of the most critical tasks in Requirements Engineering, due to the huge importance of its goals: (1) understand the problem to be solved; (2) develop conceptual models (CMs), which represent the problem understanding; and (3) define the features of the solution, that is, identify the requirements to be satisfied by the future software system.

CMs play a central role during analysis, as they make it possible to understand and represent the user need in an implementation-independent way. Nevertheless, several authors have argued that the CMs used nowadays are oriented to specific software development approaches. This orientation has two repercussions: (1) CMs have computational constraints, that is, CMs allows to include specific implementation characteristics in the domain models and (2) CMs prescribe the subsequent development process, in which the CMs are more or less directly transformed into design models.

The above-mentioned problems cause that the CMs used nowadays are not appropriate for performing analysis. In this PhD dissertation, an alternative approach to conceptual modelling is proposed. This approach, called “Need-Oriented Analysis Method”, is characterised by: (1) using representation formalisms, called Generic Conceptual Models (GCM), that do not presuppose any implementation paradigm; (2) defining a detailed analysis process; and (3) deriving, from the GCM, the best-suited CM (that is, a CM now used in SE, like data flow diagrams, use cases, etc.) to continue with development according to the methods used nowadays.

Índice

VOLUMEN I. MEMORIA

PARTE I

DESCRIPCIÓN DEL PROBLEMA DE INVESTIGACIÓN

1. Introducción	3
1.1 Área de investigación	3
1.2 Identificación del problema.....	5
1.3 Importancia del problema de investigación.....	7
1.4 Objetivo del trabajo	9
1.5 Aproximación a la solución.....	9
2. Estado de la Cuestión	13
2.1 Introducción	13
2.2 Características de las aproximaciones de desarrollo	14
2.2.1 Tipos de modelos.....	14
2.2.2 Modelos dominantes y complementarios.....	16
2.3 Modelos Conceptuales utilizados por las aproximaciones de desarrollo	18
2.3.1 Caracterización de las aproximaciones de desarrollo	18
2.3.2 Clasificación de modelos paradigmáticos.....	21
2.3.3 Relación entre los modelos paradigmáticos y las aproximaciones de desarrollo	23
2.3.4 Modelos excluidos.....	24
2.4 Criterios de valoración.....	25
2.5 Revisión de modelos conceptuales.....	28
2.5.1 Modelos orientados al procedimiento	28
2.5.1.1 Miniespecificación	28
2.5.1.1.1 Descripción	28
2.5.1.1.2 Métodos de análisis	28
2.5.1.1.3 Valoración	29
2.5.1.2 Tablas de decisión	29
2.5.1.2.1 Descripción	29

2.5.1.2.2 Métodos de análisis	30
2.5.1.2.3 Valoración	30
2.5.1.3 Árboles de decisión	31
2.5.1.4 Diccionario de datos.....	31
2.5.1.4.1 Descripción	31
2.5.1.4.2 Métodos de análisis	32
2.5.1.4.3 Valoración	33
2.5.1.5 Flujograma	33
2.5.1.5.1 Descripción	33
2.5.1.5.2 Métodos de análisis	34
2.5.1.5.3 Valoración	34
2.5.1.6 Lenguaje de diseño de programas.....	35
2.5.1.6.1 Descripción	35
2.5.1.6.2 Métodos de análisis	36
2.5.1.6.3 Valoración	36
2.5.2 Modelos orientados a la transformación	37
2.5.2.1 Diagrama de flujo de datos	37
2.5.2.1.1 Descripción	37
2.5.2.1.2 Métodos de análisis	39
2.5.2.1.3 Valoración	39
2.5.2.2 Diagrama de flujo de control	40
2.5.2.2.1 Descripción	40
2.5.2.2.2 Métodos de análisis	40
2.5.2.2.3 Valoración	41
2.5.2.3 Diagrama de flujo de datos para tiempo real	42
2.5.2.3.1 Descripción	42
2.5.2.4 Técnica de análisis y diseño estructurado	43
2.5.2.4.1 Descripción	43
2.5.2.4.2 Métodos de análisis	44
2.5.2.4.3 Valoración	44
2.5.2.5 Problem statement language/program statement analyzer (PSL/PSA).....	44
2.5.2.5.1 Descripción	44
2.5.2.5.2 Métodos de análisis	45
2.5.2.5.3 Valoración	45
2.5.2.6 Resumen	46
2.5.3 Modelos orientados a los objetos / datos	46
2.5.3.1 Modelos básicos.....	48
2.5.3.1.1 Modelo entidad – relación.....	49
2.5.3.1.1.1 Descripción	49
2.5.3.1.1.2 Métodos de análisis	50
2.5.3.1.1.3 Valoración	51
2.5.3.1.1.4 Otras consideraciones	52
2.5.3.1.2 Modelo funcional de datos	53
2.5.3.1.2.1 Descripción	53
2.5.3.1.2.2 Métodos de análisis	54
2.5.3.1.2.3 Valoración	54
2.5.3.1.2.4 Otras consideraciones	54
2.5.3.1.3 Modelo entidad – relación extendido	55
2.5.3.1.3.1 Descripción	55
2.5.3.1.3.2 Métodos de análisis	56
2.5.3.1.3.3 Valoración	56
2.5.3.1.3.4 Otras consideraciones	56
2.5.3.1.4 Resumen	57
2.5.3.2 Modelos avanzados	57
2.5.3.2.1 Modelos avanzados operativos	58
2.5.3.2.1.1 Descripción	58
2.5.3.2.1.2 Métodos de análisis	59
2.5.3.2.1.3 Valoración	59
2.5.3.2.2 Modelos avanzados declarativos.....	60
2.5.3.2.2.1 Descripción	60
2.5.3.2.2.2 Métodos de análisis	61

2.5.3.2.2.3 Valoración.....	61
2.5.3.2.3 Resumen.....	62
2.5.3.3 Modelos orientados a objetos.....	62
2.5.3.3.1 Diagrama de clases.....	63
2.5.3.3.1.1 Descripción.....	63
2.5.3.3.1.2 Métodos de análisis.....	63
2.5.3.3.1.3 Valoración.....	64
2.5.3.3.1.4 Otras consideraciones.....	65
2.5.3.3.2 Historia de la vida de las entidades.....	65
2.5.3.3.2.1 Descripción.....	65
2.5.3.3.2.2 Métodos de análisis.....	66
2.5.3.3.2.3 Valoración.....	66
2.5.3.3.3 Resumen.....	67
2.5.4 Modelos orientados a la dinámica.....	67
2.5.4.1 Máquinas de estado finito.....	67
2.5.4.1.1 Descripción.....	67
2.5.4.1.2 Métodos de análisis.....	68
2.5.4.1.3 Valoración.....	68
2.5.4.2 Diagramas de transición de estados.....	69
2.5.4.2.1 Descripción.....	69
2.5.4.2.2 Métodos de análisis.....	69
2.5.4.2.3 Valoración.....	70
2.5.4.3 Statecharts.....	70
2.5.4.3.1 Descripción.....	70
2.5.4.3.2 Métodos de análisis.....	71
2.5.4.3.3 Valoración.....	71
2.5.4.4 Resumen.....	72
2.5.5 Modelos orientados a la interacción.....	72
2.5.5.1 Casos de uso.....	72
2.5.5.1.1 Descripción.....	72
2.5.5.1.2 Métodos de análisis.....	73
2.5.5.1.3 Valoración.....	73
2.5.5.2 Escenarios.....	74
2.5.5.2.1 Descripción.....	74
2.5.5.2.2 Métodos de análisis.....	74
2.5.5.2.3 Valoración.....	74
2.5.5.3 Resumen.....	75
2.6 Resumen del estado de los conocimientos de la tarea de análisis.....	75
3. Planteamiento de la investigación.....	79
3.1 Definición.....	79
3.2 Hipótesis de trabajo.....	80
3.3 Descripción e la solución propuesta.....	82
3.3.1 Proceso de análisis.....	83
3.3.2 Modelo conceptual genérico.....	84
3.3.3 Identificación de la aproximación de desarrollo más adecuada.....	85
3.3.4 Derivación de los modelos conceptuales.....	86
3.3.5 Estructura de la exposición de la resolución.....	86

PARTE II

RESOLUCIÓN

4. Visión General de MAON.....	89
4.1 Bases cognitivas del proceso propuesto.....	90
4.2 Proceso de análisis propuesto.....	92
4.2.1 Análisis preliminar.....	94
4.2.2 Análisis exhaustivo.....	96

4.2.3	Identificación del modelo conceptual idóneo	97
4.2.4	Derivación del modelo conceptual seleccionado	97
5.	Análisis orientado al problema.....	99
5.1	Análisis preliminar.....	100
5.1.1	Descripción.....	100
5.1.1.1	Identificar los conceptos más destacados	102
5.1.1.2	Describir los conceptos identificados.....	102
5.1.1.3	Asociar los conceptos identificados	103
5.1.2	Técnicas utilizadas	104
5.1.3	Productos de entrada	104
5.1.4	Productos de salida	104
5.1.5	Excepciones	105
5.2	Análisis exhaustivo	105
5.2.1	Descripción.....	105
5.2.1.1	Distinguir conjuntos e individuos.....	106
5.2.1.2	Distinguir asociaciones estructurales.....	107
5.2.1.3	Describir propiedades	107
5.2.1.4	Identificar características dinámica	108
5.2.2	Técnicas utilizadas	108
5.2.3	Productos de entrada	108
5.2.4	Productos de salida	108
5.2.5	Excepciones	109
5.3	Elementos del MCG	110
5.3.1	Conceptos y asociaciones.....	110
5.3.2	Proposiciones	114
5.3.3	Conjuntos e individuos	115
5.3.4	Significado	116
5.3.4.1	Comprensión	116
5.3.4.2	Extensión.....	117
5.3.5	Conceptos especiales	118
5.3.5.1	Conceptos definidos mediante un predicado	118
5.3.5.2	Conceptos definidos mediante una función	119
5.3.5.3	Conceptos definidos mediante una transición	119
5.3.6	Asociaciones especiales	119
5.3.6.1	Asociaciones utilizadas con conjuntos.....	120
5.3.6.2	Asociaciones que denotan aspectos estructurales.....	120
5.3.6.3	Asociaciones que definen valores.....	121
5.3.7	Operadores.....	121
5.3.7.1	Identidad.....	121
5.3.7.2	Definición.....	122
5.3.7.3	Abstracción.....	122
5.4	Formalismos de representación del MCG	124
5.4.1	Mapa de conceptos: preliminar y exhaustivo	125
5.4.2	Mapa de conceptos: diagramación	125
5.4.3	Diccionario de identificación.....	131
5.4.4	Diccionario de descripción	131
5.4.4.1	Declaraciones.....	132
5.4.4.2	Definiciones.....	132
5.4.4.3	Proposiciones.....	133
5.4.5	Texto narrativo.....	134
5.5	Relación entre los formalismos de representación del MCG	134
5.5.1	Relación entre el mapa de conceptos y el diccionario de identificación.....	134
5.5.2	Relación entre el mapa de conceptos y el diccionario de descripción.....	137
5.5.2.1	Confección de la parte de declaraciones del diccionario de descripción	137
5.5.2.2	Confección de la parte de definiciones del diccionario de descripción.....	138
5.5.2.3	Casos especiales en la confección de la parte de definiciones del diccionario de descripción	140
5.5.2.4	Confección de la parte de proposiciones del diccionario de descripción	140
5.5.2.5	Efectos del operador de definición sobre el diccionario de descripción	141
5.5.2.6	Efectos del operador de abstracción sobre el diccionario de descripción.....	142
5.5.2.6.1	Declaraciones	142

5.5.2.6.2 Definiciones	144
5.5.2.6.3 Proposiciones	146
5.5.3 Relación entre el texto narrativo y los restantes formalismos de representación	149
6. Análisis orientado a la solución	153
6.1 Identificación del modelo conceptual idóneo	154
6.1.1 Descripción	154
6.1.1.1 Modelo canónico de requisitos	154
6.1.1.2 Identificación del modelo conceptual idóneo	155
6.1.2 Técnicas utilizadas	155
6.1.3 Productos de entrada	155
6.1.4 Productos de salida	156
6.1.5 Excepciones	156
6.2 Derivación del modelo conceptual seleccionado	156
6.2.1 Descripción de la tarea	157
6.2.2 Técnicas utilizadas	157
6.2.3 Productos de entrada	157
6.2.4 Productos de salida	158
6.2.5 Excepciones	158
6.3 Técnica de identificación del modelo conceptual idóneo	158
6.3.1 Aspectos preliminares	158
6.3.1.1 Estructura del diccionario de descripción	159
6.3.1.2 Modelo canónico	159
6.3.1.3 Estructuras de las etiquetas	160
6.3.2 Procedimiento de interpretación	161
6.3.2.1 Etiquetado de conjuntos, subconjuntos, predicados, funciones y transiciones	162
6.3.2.2 Etiquetado de asociaciones especiales	164
6.3.2.3 Etiquetado de conceptos y asociaciones de interpretación única	165
6.3.2.4 Etiquetado de conceptos o asociaciones de interpretación múltiple	173
6.3.2.5 Estudiar la sustitución de proposiciones	173
6.3.2.6 Definir proposiciones no interpretables	174
6.3.3 Selección del modelo conceptual idóneo	176
6.3.3.1 Estudio de la medida de adecuación	180
6.4 Derivación del modelo conceptual seleccionado	183
6.4.1 Aislamiento de proposiciones	184
6.4.2 Obtención de fragmentos	185
6.4.3 Unión de fragmentos	188
6.4.4 Refinamiento del modelo obtenido	188

PARTE III

VALIDACIÓN

7. Validación	193
7.1 Planteamiento de la validación	193
7.2 Resolución de los casos de prueba por el grupo de control	198
7.3 Validación de H1: selección del modelo más adecuado	198
7.3.1 Identificación de los modelos más adecuados por GP1, GP2 y GP3	198
7.3.2 Estudio del modelo más adecuado en los casos puros	200
7.3.3 Estudio del modelo idóneo en los casos mixtos	202
7.3.4 Tendenciosidad en la selección del modelo más adecuado	205
7.3.4.1 Estudio de las Preferencias de los sujetos	205
7.3.4.2 Estudio de la sensibilidad de MAON frente a la tendenciosidad	206
7.4 Validación de SH2: derivación de los modelos conceptuales de las aproximaciones tradicionales	211
7.5 Validación de SH3: versatilidad de MAON	211
8. Caso práctico	213
8.1 Los enunciados	214

8.1.1 Briefing	214
8.1.1.1 Descripción.....	214
8.1.1.2 Mapa de conceptos.....	215
8.1.1.3 Diccionario de descripción	216
8.1.1.4 Interpretación parcial.....	216
8.1.2 Planificación	217
8.1.2.1 Descripción.....	217
8.1.2.2 Mapa de conceptos.....	217
8.1.2.3 Diccionario de descripción	218
8.1.2.4 Interpretación parcial.....	218
8.1.3 Pases tipo.....	219
8.1.3.1 Descripción.....	219
8.1.3.2 Mapa de conceptos.....	219
8.1.3.3 Diccionario de descripción	220
8.1.3.4 Interpretación parcial.....	220
8.1.4 Planes.....	221
8.1.4.1 Descripción.....	221
8.1.4.2 Mapa de conceptos.....	222
8.1.4.3 Diccionario de descripción	223
8.1.4.4 Interpretación parcial.....	224
8.1.5 Alternativas.....	225
8.1.5.1 Descripción.....	225
8.1.5.2 Mapa de conceptos.....	225
8.1.5.3 Diccionario de descripción	225
8.1.5.4 Interpretación parcial.....	226
8.1.6 Tarifa base.....	227
8.1.6.1 Descripción.....	227
8.1.6.2 Mapa de conceptos.....	227
8.1.6.3 Diccionario de descripción	228
8.1.6.4 Interpretación parcial.....	228
8.1.7 Negociación.....	229
8.1.7.1 Descripción.....	229
8.1.7.2 Mapa de conceptos.....	229
8.1.7.3 Diccionario de descripción	230
8.1.7.4 Interpretación parcial.....	231
8.1.8 Modelo de Previsión.....	232
8.1.8.1 Descripción.....	232
8.1.8.2 Mapa de conceptos.....	232
8.1.8.3 Diccionario de descripción	233
8.1.8.4 Interpretación parcial.....	233
8.1.9 Campaña	234
8.1.9.1 Descripción.....	234
8.1.9.2 Mapa de conceptos.....	234
8.1.9.3 Diccionario de descripción	235
8.1.9.4 Interpretación parcial.....	235
8.1.10 Órdenes.....	236
8.1.10.1 Descripción.....	236
8.1.10.2 Mapa de conceptos.....	236
8.1.10.3 Diccionario de descripción	237
8.1.10.4 Interpretación parcial.....	238
8.1.11 Facturación.....	239
8.1.11.1 Descripción.....	239
8.1.11.2 Mapa de conceptos.....	239
8.1.11.3 Diccionario de descripción	240
8.1.11.4 Interpretación parcial.....	241
8.2 Aplicación de la técnica IMCI	242
8.3 Fragmentos.....	247
8.4 Modelo derivado	259

PARTE IV**CONCLUSIONES Y FUTURAS LINEAS**

9 Conclusiones	263
9.1 Principales contribuciones de la investigación.....	263
9.1.1 Modelo conceptual genérico	264
9.2.2 Método de pre-análisis: MAON.....	264
9.2 Análisis de la consecución de objetivos.....	267
9.3 Resultados de la investigación.....	269
9.4 Futuras líneas de investigación.....	
9.4.1 Definición de nuevas métricas de adecuación.....	272
9.4.2 Refinamiento del modelo conceptual obtenido con la técnica DMCS	272
9.4.3 Ampliación del alcance del problema de investigación	272
10 Bibliografía.....	275
Anexo A	287
A.1 Modelo canónico.....	287
A.2 Elementos y enlaces.....	288
A.3 Relación con los modelos conceptuales.....	290
A.4 Modificaciones realizadas al modelo canónico.....	291
A.4.1 Invocación de procesos	292
A.4.2 Invocación de procesos por un predicado	295
A.4.3 Pertenencia a conjuntos	296
A.4.4 Subconjuntos	298
A.4.5 Relaciones entre entidades y cardinalidad.....	298
A.4.6 Simplificación del link stimulus	299
A.4.7 Conceptos enlazados a proposiciones.....	300
A.4.8 Compounds	301
A.4.9 Asignación de valores.....	301
A.4.10 Link equivalence	301
Anexo B. Tablas IMCI.....	302
Anexo C. Tablas DMCS	329

Índice de Tablas

Tabla 2.1 Correspondencia entre los modelos seleccionados y las aproximaciones de desarrollo.....	23
Tabla 2.2 Valores de los criterios para los modelos orientados al procedimiento.....	37
Tabla 2.3 Valores de los criterios para los modelos orientados a la transformación	46
Tabla 2.4 Origen de los subgrupos de los modelos orientados a los objetos / datos.....	47
Tabla 2.5 Valores de los criterios para los modelos básicos	57
Tabla 2.6 Valores de los criterios para los modelos avanzados.....	62
Tabla 2.7 Valores de los criterios para los modelos orientados a los objetos	67
Tabla 2.8 Valores de los criterios par los modelos orientados a la dinámica	72
Tabla 2.9 Valores de los criterios para los modelos orientados a la interacción	75
Tabla 2.10 Resumen de la valoración de los criterios para los distintos modelos paradigmáticos.....	76
Tabla 5.1 Formato del Diccionario de Identificación	131
Tabla 5.2 Ejemplo de Diccionario de Identificación	131
Tabla 5.3 Formato de la parte de Declaraciones del Diccionario de Descripción	132
Tabla 5.4 Formato de la parte de Definiciones del Diccionario de Descripción.....	133
Tabla 5.5 Formato de la parte de Proposiciones del Diccionario de Descripción.....	134
Tabla 5.6 Confección de la parte de declaraciones del Diccionario de Descripción (parte 1).....	137
Tabla 5.6 Confección de la parte de declaraciones del Diccionario de Descripción (parte 2).....	138
Tabla 5.7 Confección de la parte de definiciones del Diccionario de Descripción.....	138
Tabla 5.8 Confección de la parte de Proposiciones del Diccionario de Descripción	141
Tabla 5.9 Equivalencias	148
Tabla 6.1 Etiquetado de conjuntos, subconjuntos y funciones	162
Tabla 6.2 Reglas de etiquetado de asociaciones especiales	165
Tabla 6.3 Combinaciones permitidas en el Modelo Canónico entre elementos y enlaces.....	168
Tabla 6.4 Combinaciones permitidas en el Modelo Canónico entre elementos, enlaces y proposiciones (parte 1).....	169
Tabla 6.5 Combinaciones permitidas en el Modelo Canónico entre elementos, enlaces y proposiciones (parte 2).....	170
Tabla 6.6 Ejemplo de tabla para el cálculo del Modelo Conceptual Idóneo	177
Tabla 6.7 Cálculo del Modelo Conceptual Idóneo para el ejemplo de la figura 6.11.....	179
Tabla 6.8 Cálculo del Método Idóneo para el ejemplo de figura 6.11.....	181
Tabla 6.9 Nuevo cálculo del Método Idóneo para el ejemplo de figura 6.11	183
Tabla 6.10 Fragmentos del diagrama de clases	187
Tabla 7.1 Casos de prueba	197
Tabla 7.2 Decisión del GC.....	198
Tabla 7.3 Resultados de los GP1, GP2 y GP3	199

Tabla 7.4 Resultados de los GP1, GP2 y GP3 para los casos puros	201
Tabla 7.5 Resultados de los GP1, GP2 y GP3 para los casos mixtos.....	203
Tabla 7.6 Resultados de GP2 considerando como acierto la identificación de todos los modelos adecuados	204
Tabla 7.7 Número de veces que cada sujeto ha seleccionado un mismo tipo de modelo	205
Tabla 7.8 Resultados del caso contaminado.....	208
Tabla 8.1 Diccionario de Descripción del BRIEFING	216
Tabla 8.2 Interpretación del Diccionario de Descripción del BRIEFING	216
Tabla 8.3 Diccionario de Descripción de la PLANIFICACIÓN	218
Tabla 8.4 Interpretación del Diccionario de Descripción de la PLANIFICACIÓN	218
Tabla 8.5 Diccionario de Descripción de los PASES TIPO.....	220
Tabla 8.6 Interpretación del Diccionario de Descripción de los PASES TIPO.....	220
Tabla 8.7 Diccionario de Descripción correspondiente a los PLANES	223
Tabla 8.8 Interpretación del Diccionario de Descripción correspondiente a los PLANES	224
Tabla 8.9 Diccionario de Descripción correspondiente a las ALTERNATIVAS	225
Tabla 8.10 Interpretación del Diccionario de Descripción correspondiente a las ALTERNATIVAS	226
Tabla 8.11 Diccionario de Descripción correspondiente a la TARIFA BASE.....	228
Tabla 8.12 Interpretación del Diccionario de Descripción correspondiente a la TARIFA BASE.....	228
Tabla 8.13 Diccionario de Descripción correspondiente a la NEGOCIACIÓN	230
Tabla 8.14 Interpretación del Diccionario de Descripción correspondiente a la NEGOCIACIÓN	231
Tabla 8.15 Diccionario de Descripción correspondiente al MODELO DE PREVISIÓN.....	233
Tabla 8.16 Interpretación del Diccionario de Descripción correspondiente al MODELO DE PREVISIÓN.....	233
Tabla 8.17 Diccionario de Descripción correspondiente a la CAMPAÑA	235
Tabla 8.18 Interpretación del Diccionario de Descripción correspondiente a la CAMPAÑA	235
Tabla 8.19 Diccionario de Descripción correspondiente a ORDENES	237
Tabla 8.20 Interpretación del Diccionario de Descripción correspondiente a ORDENES	238
Tabla 8.21 Diccionario de Descripción correspondiente a la FACTURACIÓN	240
Tabla 8.22 Interpretación del Diccionario de Descripción correspondiente a la FACTURACIÓN	241
Tabla 8.23 Aplicación de la técnica IMCI	242 – 246
Tabla 8.24 Fragmentos del Diagrama de Clases.....	247 – 258
Tabla B.1 Tabla IMCI para el link spec.....	303
Tabla B.2 Tabla IMCI para el link pof (parte 1/3).....	304
Tabla B.3 Tabla IMCI para el link pof (parte 2/3).....	305
Tabla B.4 Tabla IMCI para el link pof (parte 3/3).....	306
Tabla B.5 Tabla IMCI para el link subs.....	307
Tabla B.6 Tabla IMCI para el link rel (parte 1/2).....	308
Tabla B.7 Tabla IMCI para el link rel (parte 2/2).....	309
Tabla B.8 Tabla IMCI para el link activate (parte 1/2)	310
Tabla B.9 Tabla IMCI para el link activate (parte 2/2)	311
Tabla B.10 Tabla IMCI para el link operand (parte 1/3)	312
Tabla B.11 Tabla IMCI para el link operand (parte 2/3)	313
Tabla B.12 Tabla IMCI para el link operand (parte 3/3)	314
Tabla B.13 Tabla IMCI para el link stimulus (parte 1/3)	315
Tabla B.14 Tabla IMCI para el link stimulus (parte 2/3)	316
Tabla B.15 Tabla IMCI para el link stimulus (parte 3/3)	317
Tabla B.16 Tabla IMCI para el link response (parte 1/3).....	318
Tabla B.17 Tabla IMCI para el link response (parte 2/3).....	319
Tabla B.18 Tabla IMCI para el link response (parte 3/3).....	320
Tabla B.19 Tabla IMCI para el link sends (parte 1/3)	321
Tabla B.20 Tabla IMCI para el link sends (parte 2/3)	322
Tabla B.21 Tabla IMCI para el link sends (parte 3/3)	323
Tabla B.22 Tabla IMCI para el link receives (parte 1/3)	324
Tabla B.23 Tabla IMCI para el link receives (parte 2/3)	325
Tabla B.24 Tabla IMCI para el link receives (parte 3/3)	326
Tabla B.25 Tabla IMCI para el link hval.....	327
Tabla C.1 Tabla DMCS para el diagrama de clases (parte 1 de 2)	331
Tabla C.2 Tabla DMCS para el diagrama de clases (parte 2 de 2)	332
Tabla C.3 Tabla DMCS para el diagrama entidad-relación (parte 1 de 2)	333
Tabla C.4 Tabla DMCS para el diagrama entidad-relación (parte 2 de 2)	333
Tabla C.5 Tabla DMCS para el diagrama de flujo de datos (parte 1 de 5)	334

Tabla C.6 Tabla DMCS para el diagrama de flujo de datos (parte 2 de 5).....	335
Tabla C.7 Tabla DMCS para el diagrama de flujo de datos (parte 3 de 5).....	336
Tabla C.8 Tabla DMCS para el diagrama de flujo de datos (parte 4 de 5).....	337
Tabla C.9 Tabla DMCS para el diagrama de flujo de datos (parte 5 de 5).....	338
Tabla C.10 Tabla DMCS para los casos de uso	339
Tabla C.11 Tabla DMCS para el diagrama de transición de estados (parte 1 de 2)	340
Tabla C.12 Tabla DMCS para el diagrama de transición de estados (parte 2 de 2)	340
Tabla C.13 Tabla DMCS para el statechart (parte 1 de 2).....	340
Tabla C.14 Tabla DMCS para el statechart (parte 2 de 2).....	340

Índice de Figuras

Figura 1.1. Relación entre los términos Análisis ₁ y Análisis	6
Figura 1.2. Relación entre los términos Análisis ₁ , Análisis y pre-Análisis.....	9
Figura 1.1 Localización del pre-Análisis.....	11
Figura 1.2 Integración del pre-Análisis en las distintas aproximaciones de desarrollo	12
Figura 2.1 Modelos utilizados por el Análisis Estructurado y OMT.....	16
Figura 2.2 Modelos Dominantes	17
Figura 2.3 Clasificación de modelos paradigmáticos.....	20
Figura 2.4 Relación de los criterios elegidos con el desarrollo de software	27
Figura 2.5 Ejemplo de miniespecificación	29
Figura 2.6 Ejemplo de tabla de decisión	30
Figura 2.7 Ejemplo de árbol de decisión.....	31
Figura 2.8 Ejemplo de diccionario de datos	32
Figura 2.9 Símbolos convencionales utilizados en los flujogramas.....	34
Figura 2.10 Ejemplo de flujograma	35
Figura 2.11 Ejemplo de lenguaje de diseño de programas.....	36
Figura 2.12 Símbolos gráficos de un DFD	38
Figura 2.13 Ejemplo de diagrama de flujo de datos.....	39
Figura 2.14 Ejemplo de diagrama de flujo de control.....	41
Figura 2.15 Ejemplo de diagrama de flujo de datos para tiempo real	42
Figura 2.16 Ejemplo de la técnica de análisis y diseño estructurado	43
Figura 2.17 Ejemplo de PSL	45
Figura 2.18 Evolución e interacción entre distintos tipos de modelos [Loucopoulos et al., 1995].....	48
Figura 2.19 Ejemplo de modelo entidad-relación.....	51
Figura 2.20 Ejemplo de modelo funcional de datos.....	54
Figura 2.21 Jerarquía de especialización.....	55
Figura 2.22 Jerarquía de generalización.....	56
Figura 2.23 Descripción general del comportamiento en los modelos AO	58
Figura 2.24 Descripción general del comportamiento en los modelos AD	61
Figura 2.25 Ejemplo de diagrama de clases.....	64
Figura 2.26 Ejemplo de historia de la vida de las entidades.....	66
Figura 2.27 Ejemplo de máquina de estado finito.....	68
Figura 2.28 Ejemplo de Diagrama de Transición de Estados.....	69
Figura 2.29 Ejemplo de statechart	71
Figura 2.30 Ejemplo de un modelo de casos de uso.....	73
Figura 2.31 Ejemplo de escenario.....	74

Figura 3.1 Estructura del Método de Análisis Orientado a la Necesidad (MAON).....	83
Figura 4.1 Esquema del proceso de pre-Análisis propuesto.....	94
Figura 4.2 Secuencia de las tareas durante el Análisis Preliminar.....	95
Figura 4.3 Secuencia de las tareas del Análisis Exhaustivo.....	96
Figura 5.1 Secuencia de tareas del Análisis Preliminar.....	101
Figura 5.2 Secuencia de tareas durante el Análisis Exhaustivo.....	106
Figura 5.3 Representación gráfica, utilizando el Mapa de Conceptos, de las proposiciones “Cliente realiza Pedido” y “Cliente posee Cuenta Contable”.....	123
Figura 5.4 Resultado de Pedido de Cliente = ABS ([Cliente realiza Pedido]).....	124
Figura 5.5 Diagramación de una proposición.....	126
Figura 5.6 Diagramación de asociaciones entre un concepto y una proposición.....	126
Figura 5.7 Diagramación de dos proposiciones unidas por una asociación.....	127
Figura 5.8 Notación alternativa para la diagramación de proposiciones recursivas.....	127
Figura 5.9 Diagramación de un conjunto.....	128
Figura 5.10 Diagramación de un predicado.....	128
Figura 5.11 Diagramación de una función.....	129
Figura 5.12 Diagramación de una transición.....	129
Figura 5.13 Diagramación del operador EQ.....	130
Figura 5.14 Diagramación de una abstracción.....	130
Figura 5.15 Fragmento de un Mapa de Conceptos preliminar.....	135
Figura 5.16 Traslación de los conceptos desde el Mapa de Conceptos preliminar al Diccionario de Identificación.....	135
Figura 5.17 Traslación de los predicados desde el Mapa de Conceptos preliminar al Diccionario de Identificación.....	136
Figura 5.18 Estructura general de las funciones y predicados en el MCG.....	139
Figura 5.19 Predicado con operandos definidos sobre predicados construidos con asociaciones genéricas.....	140
Figura 5.20 Mapa de Conceptos exhaustivo que contiene una definición.....	141
Figura 5.21 Mapa de Conceptos exhaustivo una vez deshecha la definición.....	142
Figura 5.22 Abstracción que oculta declaraciones.....	143
Figura 5.23 Registro de las declaraciones en el Diccionario de Descripción.....	143
Figura 5.24 Abstracción que forma una proposición mediante la asociación pof.....	144
Figura 5.25 Derivación al Diccionario de Descripción.....	145
Figura 5.26 Abstracción que forma parte de un predicado.....	146
Figura 5.27 Abstracción que forma parte de una proposición genérica.....	147
Figura 5.28 Asociación entre un concepto y una abstracción.....	147
Figura 5.29 Otra asociación entre un concepto y una abstracción.....	148
Figura 5.30 Mezcla de niveles de abstracción.....	148
Figura 5.31 Mapa de Conceptos.....	149
Figura 6.1 Estructura del Diccionario de Descripción.....	159
Figura 6.2 Diccionario de Descripción que se utilizará a modo de ejemplo a lo largo de la presente sección.....	163
Figura 6.3 Primer etiquetado.....	163
Figura 6.4 Segundo etiquetado.....	164
Figura 6.5 Etiquetado de asociaciones especiales.....	165
Figura 6.6 Proposición recursiva por la izquierda y por la derecha.....	166
Figura 6.7 Descomposición de una proposición recursiva a izquierda y derecha.....	167
Figura 6.8 Interpretación de <concepto> Bel Entity [repl].....	171
Figura 6.9 Ejemplo de proposición construida recursivamente por la izquierda.....	172
Figura 6.10 Etiquetado de conceptos de interpretación única.....	172
Figura 6.11 Modelo Canónico de Requisitos.....	175
Figura 6.12 Fragmento de la tabla para el enlace spec.....	176
Figura 6.13 MCR en formato gráfico para el predicado Constraint:>= (Operand: Satespace: N ^o , Operand: Process: Sumatorio (Receives: Value: Aprobado, Receives: Statespace: N ^o _1)).....	178
Figura 6.14 Proposiciones derivables a un diagrama de clases en el ejemplo que se ha manejado en este capítulo (figura 6.11).....	184
Figura 6.15 Diagrama de clases obtenido por la unión de fragmentos.....	188
Figura 6.16 Diagrama de clases después del refinamiento.....	189
Figura 7.1 Planteamiento de la validación.....	194
Figura 7.2 Diagrama de Flujo de Datos obtenido mediante la aplicación de la técnica DMCS.....	209

Figura 7.3 Diagrama de Flujo de Datos / Tiempo Real obtenido tras la aplicación de la técnica DMCS para el Diagrama de Flujo de Datos / Tiempo Real.....	210
Figura 8.1 Mapa de Conceptos correspondiente al BRIEFING	215
Figura 8.2 Mapa de Conceptos correspondiente a la PLANIFICACIÓN	217
Figura 8.3 Mapa de Conceptos correspondiente a los PASES TIPO	219
Figura 8.4 Mapa de Conceptos correspondiente a los PLANES	222
Figura 8.5 Mapa de Conceptos correspondiente a las ALTERNATIVAS	225
Figura 8.6 Mapa de Conceptos correspondiente a la TARIFA BASE.....	227
Figura 8.7 Mapa de Conceptos correspondiente a la NEGOCIACIÓN	229
Figura 8.8 Mapa de Conceptos correspondiente al MODELO DE PREVISIÓN.....	232
Figura 8.9 Mapa de Conceptos correspondiente a la CAMPAÑA	234
Figura 8.10 Mapa de Conceptos correspondiente a ORDENES	236
Figura 8.11 Mapa de Conceptos correspondiente a la FACTURACIÓN	239
Figura 8.12 Diagrama de Clases confeccionado a partir de los fragmentos de la tabla 8.24	259
Figura A.1 Convenciones de diagramación	289
Figura A.2 Relación entre elementos / links y el modelo entidad-relación	291
Figura A.3 Estructura de invocación de un proceso por una entidad externa en un DFD.....	292
Figura A.4 Conceptos necesarios (del MCG) para derivar la estructura de invocación de un proceso por una entidad externa en un DFD	293
Figura A.5 Estructura de invocación de un proceso por un proceso en un DFD.....	294
Figura A.6 Invocación de un proceso (método) por una entidad (objeto).....	294
Figura A.7 Mecanismo de invocación de procesos utilizado en el presente trabajo de Tesis. Las llaves se han utilizado (arbitrariamente) para no utilizar un simbolismo concreto del modelo canónico, el cual sería incorrecto	294
Figura A.8 Estructura que define el predicado $4 > 2^{A^2}$	295
Figura A.9 Reescritura de la función 2^{A^2}	295
Figura A.10 Relación entre un individuo y un conjunto mediante el link spec	296
Figura A.11 Relación entre un individuo y un conjunto mediante el link pof.....	296
Figura A.12 Derivación del link spec a un diagrama de clases.....	297
Figura A.13 Derivación del link spec a un diagrama de secuencia	297
Figura A.14 Relación y cardinalidad en el lenguaje propuesto por Davis.....	298

Índice de Cuadros

Cuadro 5.1 Texto Narrativo derivado del Mapa de Conceptos mostrado en la Figura 5.31.....	150
Cuadro 5.2 Texto Narrativo mejorado para el ejemplo del cuadro 6.1	150
Cuadro C.1 Reglas de derivación para el diagrama de clases.....	332
Cuadro C.2 Reglas de derivación para el diagrama entidad-relación	333

Glosario de Acrónimos y Términos Especiales

Término o acrónimo	Significado
Análisis	Tarea de Análisis según el SWEBOK (Software Engineering Body of Knowledge)
Análisis ₁	Actividad de Análisis, previa a la actividad de Diseño, prescrita por las aproximaciones clásicas de desarrollo
Pre-Análisis	Tarea de Análisis previa a Analisis ₁ . MAON es una posible implementación de una tarea de pre-Análisis.
DMCS	Derivación del Modelo Conceptual Seleccionado.
IMCI	Identificación del Modelo Conceptual Idóneo
MCG	Modelo Conceptual Genérico
DI	Diccionario de Identificación
DD	Diccionario de Descripción
TN	Texto Narrativo
MC	Mapa de Conceptos
MAON	Modelo de Análisis Orientado a la Necesidad

PARTE I

**DESCRIPCIÓN
DEL PROBLEMA DE
INVESTIGACIÓN**

1. Introducción

1.1. ÁREA DE INVESTIGACIÓN

El proceso de desarrollo de software está compuesto por diversas actividades. Considerando únicamente las actividades técnicas, esto es, las actividades cuya esencia es producir software, se reconocen actualmente cinco actividades principales [SWEBOK, 2001]: Requisitos, Diseño, Implementación, Pruebas y Mantenimiento.

De las actividades anteriormente citadas, posee especial importancia la actividad de **Requisitos**, ya que su finalidad es identificar las necesidades que el futuro sistema software deberá satisfacer [Davis, 1993]. Como resultado palpable, a la finalización de la actividad de Requisitos se obtiene un documento denominado Especificación de Requisitos Software (ERS). La ERS es el documento donde se recogen los requisitos que el producto software a construir debe cumplir y, por lo tanto, la ERS es el documento a partir del cual los desarrolladores abordan las subsiguientes actividades del proceso de construcción de software.

La actividad de Requisitos se descompone en un conjunto de sub-actividades de menor nivel, denominadas tareas. Actualmente, parece existir un acuerdo en la comunidad de la Ingeniería del Software en que dichas tareas son [SWEBOK, 2001]: Educción, Análisis, Especificación o Documentación y Validación, además de una tarea no técnica como es la de Gestión de Requisitos. No obstante, es necesario indicar que las cuatro tareas anteriores pueden no ser universalmente reconocidas como integrantes de la actividad de Requisitos, debido a:

- Diferencias en la estructuración de la actividad de Requisitos. Por ejemplo, [Davis, 1993] reconoce únicamente dos tareas (Análisis y Especificación), mientras que [Loucopoulos et al., 1995] reconoce tres tareas (Educción, Modelización y Validación).

- Diferencias de denominación. Debido a que únicamente en fechas recientes los esfuerzos de la comunidad de Ingeniería del Software se han concentrado en la actividad de Requisitos [Siddiqi, 1994], los términos que hacen referencia a las tareas y productos que componen esta actividad se han definido y popularizado sin control. Así, por ejemplo, la tarea de Análisis según [Davis, 1993] engloba a las tareas de “Educción” y “Análisis” del SWEBOK [SWEBOK, 2001]. [Davis, 1993] y Faulk [Faulk, 1997] han estudiado la diversidad de variantes terminológicas existentes.

La investigación realizada en el presente trabajo se focaliza en la tarea conocida habitualmente como **Análisis**. El Análisis tiene como finalidad organizar la información recogida durante la tarea de Educción y detectar conflictos que pudieran existir en la misma, antes de proceder con las tareas de Especificación y Validación. La tarea de Análisis puede considerarse como un filtro previo a la creación de la ERS, que permite que muchos errores sean detectados y corregidos en los momentos tempranos de la actividad de Requisitos [Kotonya et al, 1998].

Existen en la literatura un conjunto –reducido– de técnicas para llevar a cabo la tarea de Análisis, como pueden ser las listas de comprobación (checklists) o las matrices de interacción [Kotonya et al., 1998]. Entre las técnicas de Análisis más utilizadas se consideran, asimismo, las técnicas de modelización.

Un modelo es una descripción usada para entender o visualizar una realidad que no puede ser directa o totalmente observada [Webster, 1994]. La finalidad de los modelos en la tarea de Análisis es organizar la información recogida durante la Educción. La organización se realiza mediante la adscripción de cada uno de los “cuantos” de información recogidos durante la Educción a un tipo de elemento del modelo utilizado. De esta forma, la masa inestructurada de información proveniente de la Educción recibe una estructuración y categorización, quizás provisional y contingente, pero que facilita sobremedida tanto la comprensión como la búsqueda de conflictos e inconsistencias. Debido a que los modelos permiten clasificar informaciones y, por tanto, concebir mejor el problema, se acostumbra a denominar a estos modelos **modelos conceptuales**.

Implícita a la organización y búsqueda de conflictos, subyace otra de las características de los modelos conceptuales: Estos modelos facilitan la comprensión de la información obtenida de la Educción. Es por ello que los modelos conceptuales son poderosas herramientas que facilitan el proceso de abstracción y entendimiento de la información obtenida en la tarea de Educción.

Nótese, no obstante, que debido a la enorme diversidad de terminología utilizada durante la actividad de Requisitos, no todo modelo de Análisis puede ser reconocido, universalmente, como modelo conceptual; y viceversa, no todo modelo conceptual puede reconocerse como modelo de Análisis. Originariamente, el término modelo conceptual aparece en el Estándar ISO de Modelización Conceptual [van Griethuysen, 1982], donde modelo conceptual hace referencia a la representación de un universo de discurso. Por universo de discurso puede entenderse la situación, hechos, objetos, etc. en los que se focaliza el estudio durante la Educción y que, por lo tanto, son relevantes para el desarrollo del futuro sistema software [Wieringa, 1995]. Sin embargo, el significado del término “modelo conceptual” ha evolucionado durante los 20 años transcurridos tras dicha definición inicial, siendo las acepciones más recientes:

- Descripción del universo de discurso, utilizando el lenguaje y forma de pensar de los expertos del dominio y de los usuarios [Beringer, 1995].
- Definición formal de aspectos del mundo físico y social que nos rodea, con el propósito de comprenderlo y poder comunicarlo [Loucopoulos et al., 1995].
- Ayuda para que los ingenieros de requisitos entiendan el dominio [Kaindl, 1999].

Actualmente, el significado de modelo conceptual en la Ingeniería del Software es el de representación del dominio del problema –otro término para denominar al universo de discurso- realizado con el propósito de comprender el problema y favorecer la comunicación entre desarrolladores y usuarios. Esta definición, que recoge y resume lo expuesto en párrafos anteriores, será la utilizada en el presente trabajo de Tesis.

Con bastante frecuencia, tras la finalización de la actividad de Requisitos, se obtienen un conjunto de modelos del sistema [Kotonya et al., 1998]. Los modelos del sistema se obtienen complementariamente a la ERS o, en algunos casos, en sustitución de la ERS. La obtención de modelos del sistema es habitual, por citar el caso más claro, cuando la actividad de Requisitos se realiza en el marco de una metodología como, por ejemplo, MÉTRICA [CSI, 2000], SSADM [Goodland et al., 1990] u OPEN [Graham et al., 1997].

Los modelos del sistema no son, o no deberían ser, iguales que los modelos conceptuales. De hecho, existe una gran diferencia entre los modelos conceptuales, tales como el Diagrama de Flujo de Datos (DFD), el diagrama Entidad-Relación (ER) o el Diagrama de Objetos (DO), y modelos del sistema, tales como las R-NET, Specification and Description Language (SDL) o las Redes de Petri. En los modelos conceptuales prima, o debería primar, favorecer el entendimiento del problema, mientras que en los modelos del sistema prima la descripción del comportamiento externo del producto software que se deberá desarrollar [Davis, 1993].

1.2. IDENTIFICACIÓN DEL PROBLEMA DE INVESTIGACIÓN

Las tareas en las que se subdivide la actividad de Requisitos son más una guía de cómo llevar a cabo la identificación de los requisitos software que un conjunto de procesos de obligado cumplimiento. En lugar de ejecutar dichas tareas en un proceso de desarrollo, lo más frecuente [Yourdon, 1992] es utilizar las actividades y tareas propuestas por lo que se conoce como Paradigmas, Aproximaciones u Orientaciones de Desarrollo.

Existen dos aproximaciones básicas de desarrollo, conocidas bajo las denominaciones de Orientación Estructurada [DeMarco, 1979] [Palmer et al., 1984] [Yourdon, 1989] y Orientación a Objetos [Meyer, 1988] [Coad et al., 1990] [Rumbaugh et al., 1991] [Rumbaugh et al., 1998] [Fowler et al., 1999]. Además de estas dos aproximaciones principales para la construcción de sistemas software en general, existen otras más específicas, pero también utilizadas en la práctica, tales como la Aproximación de Tiempo Real [Hatley, 1984] [Ward et al., 1985] [Harel, 1981] o las Aproximaciones de Bases de Datos [Chen, 1976] [Martin, 1990] [Teorey et al., 1986] [Markowitz et al., 1992] [Brodie et al., 1982] [Gustafsson et al., 1982].

Debe notarse, no obstante, que dichas aproximaciones no son un todo integrado y coherente, sino en realidad, un nombre genérico que agrupa un variopinto conjunto de métodos, metodologías, procedimientos, etc. de muy diversa importancia teórica y práctica cada uno de ellos.

En la totalidad de las citadas aproximaciones, en lugar de existir una actividad de Requisitos a la que sigue la actividad de Diseño, se habla de una actividad de Análisis a la que sigue la actividad de Diseño. Para evitar confusiones, en el presente capítulo nos referiremos a dicha actividad como Análisis₁, para evitar entrar en confusión con la tarea de Análisis dentro de la actividad de Requisitos, tal y como se indica en la figura 1.1.

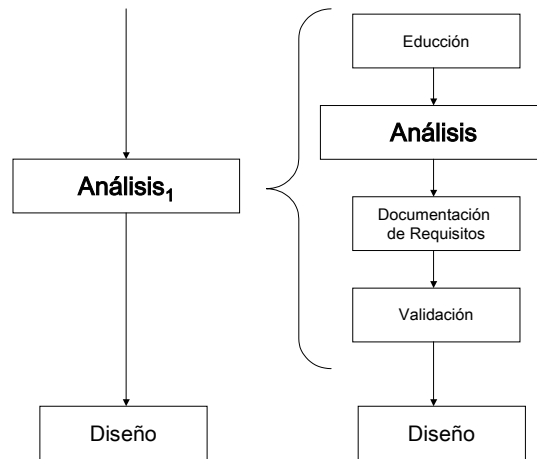


Figura 1.1. Relación entre los términos Análisis₁ y Análisis

La finalidad de la actividad de Análisis₁ es muy similar a la de la tarea de Análisis, ya que consiste en organizar y categorizar la información adquirida acerca del dominio del problema en un conjunto, dependiente de la aproximación concreta, de modelos conceptuales. No obstante, la diferencia fundamental entre la actividad de Análisis₁ y la tarea de Análisis reside en que los modelos obtenidos de Análisis₁, son utilizados como producto de entrada para la actividad de Diseño.

Como ya se ha indicado anteriormente, los modelos del sistema, obtenidos al finalizar la actividad de Requisitos, suponen una entrada válida para el Diseño, pero dichos modelos deben ser distintos de los modelos conceptuales debido a su diferente propósito (descripción del comportamiento del sistema los primeros, favorecer el entendimiento estos últimos).

Por ello, se puede afirmar que las aproximaciones más utilizadas en la actualidad para el desarrollo de software, tales como la Aproximación Estructurada o la Aproximación Orientada a Objetos **utilizan como modelos del sistema los mismos modelos conceptuales utilizados durante la actividad de Análisis₁**. Ello produce efectos perniciosos en el desarrollo de software, tal y como se indica inmediatamente a continuación. Es necesario indicar, no obstante, que el modo en que las Aproximaciones Estructurada y Orientada a Objetos contemplan el proceso de desarrollo de software no es privativo de ellas. Las restantes aproximaciones proponen un proceso de desarrollo que, en esencia, sigue el mismo esquema. Por ello, se puede afirmar que dichas aproximaciones poseen los mismos problemas que se señalarán para las Aproximaciones Estructurada y Orientada a Objetos.

El hecho de que una aproximación de desarrollo utilice como modelos del sistema los mismos modelos conceptuales utilizados durante el Análisis₁, produce dos efectos perniciosos para el proceso de desarrollo de software.

En primer lugar, **los modelos** (tanto conceptuales como del sistema, ya que al utilizar los mismos modelos tanto durante el Análisis₁ como durante el Diseño produce que ambos sean indistinguibles) **están fuertemente orientados a definir cómo se implementará el sistema software**, y no a favorecer la comprensión del dominio. Este efecto, o sesgo, surge debido a que no existen dos momentos diferenciados en el desarrollo, como son los Requisitos y el Diseño, fundiéndose ambos en uno sólo. De esta forma, los mismos modelos que se han utilizado para comprender el dominio son utilizados casi directamente, sin reelaboraciones dignas de consideración, para afrontar la definición de la estructura y comportamiento del futuro sistema software. Sin embargo es imposible que un mismo modelo sea adecuado para ambos momentos del desarrollo [McGuinness, 1992] [Høydalsvik et al., 1993], por lo que las Aproximaciones Estructurada y Orientada a Objetos potencian los modelos propios del Diseño, en detrimento de los modelos propios de los Requisitos. Este hecho, denominado **orientación a la solución de los modelos** [Dori, 1996], ha sido destacado, sobre todo, para la Aproximación Orientada a Objetos por diversos autores [Jalote, 1997] [Høydalsvik et al., 1993] [Northrop, 1997] [Bonfatti et al., 1994] [Daniels, 2002].

En segundo lugar, cada aproximación de desarrollo se caracteriza por la utilización de uno o varios modelos de un determinado tipo, que se conocen por el nombre de modelos dominantes [Yourdon, 1992]. Dichos modelos (tanto conceptuales como del sistema, ya que son indistinguibles en las aproximaciones tradicionales) **determinan prácticamente cómo se realizará todo el proceso de desarrollo posterior**.

Esto es, una vez que se ha utilizado un determinado modelo (dominante) durante el Análisis₁, las tareas posteriores (diseño, etc.) deberán realizarse utilizando un modelo compatible, esto es, un modelo propio de la misma aproximación de desarrollo y de la misma orientación que el modelo dominante. Por ejemplo, si se utiliza un DFD durante el Análisis Estructurado, la única forma de llevar a cabo el diseño será utilizando los modelos previstos por el Diseño Estructurado, esto es, los Diagramas de Estructuras [Yourdon et al, 1986]. La utilización de un modelo propio de otra aproximación de desarrollo como, por ejemplo, un diagrama de clases (propio del Diseño Orientado a Objetos), es imposible, ya que no existe forma de poner en común los conceptos representados en uno –DFD– y otro –Diagrama de clases–.

Esta imposibilidad viene motivada porque cada modelo actúa a modo de gafas que permiten observar y, por ende, comprender, únicamente una parte del dominio del problema. Dichas gafas destacan ciertas características, minimizando e incluso ocultando otras. Así, una vez que un dominio ha sido comprendido y modelado, es muy difícil recuperar cualquier aspecto que haya sido filtrado, en cualquier forma, por el modelo utilizado. La única forma posible de recuperar los aspectos perdidos es volver a “ver” el dominio utilizando un nuevo par de gafas, esto es, repetir el Análisis₁ utilizando otro modelo distinto. Esta situación ha sido señalada por diversos autores, sobre todo en el marco de la incompatibilidad de las Aproximaciones Estructurada y Orientada a Objetos [Coleman et al., 1994] [Champeaux et al., 1993] [Wieringa, 1991] [Juristo et al., 2000].

En conclusión, las Orientaciones Estructurada y de Objetos poseen dos limitaciones fundamentales:

1. Confunden las actividades de Requisitos y Diseño, por lo que los modelos utilizados en la actividad de Análisis₁ por dichas aproximaciones no son útiles para la comprensión del problema debido a su orientación a la solución.
2. Los modelos utilizados por las Orientaciones Estructurada y de Objetos restringen la libertad del desarrollador a la hora de definir la estructura y funcionalidad del futuro sistema software, ya que sólo permiten utilizar un número reducido de modelos (los modelos compatibles) en las actividades de Diseño e Implementación.

1.3. IMPORTANCIA DEL PROBLEMA DE INVESTIGACIÓN

Los inconvenientes indicados en el epígrafe anterior limitan fuertemente la capacidad de los desarrolladores para producir software. Uno de los factores más críticos en el desarrollo es comprender y representar adecuadamente los requisitos que el futuro sistema software debe satisfacer [Jackson, 1998]. Dicha comprensión y representación de los requisitos se obtiene tras un exhaustivo estudio del dominio del problema. Sin embargo, y debido a que las distintas aproximaciones de desarrollo introducen consideraciones de Diseño durante el Análisis₁, la comprensión y representación de los requisitos resulta mediatizada. Tal y como señala Davis, en las aproximaciones de desarrollo utilizadas en la actualidad la percepción del problema y el planteamiento de la solución se representan utilizando los mismos modelos. Ello produce que la percepción del problema deba adaptarse al planteamiento de la solución –y no al revés, como sería de esperar–, por lo que el éxito de la actividad de Requisitos –que va a depender primordialmente de una adecuada comprensión del problema– depende principalmente de la pericia del analista que la realiza [Davis, 1993] [Sutcliffe et al., 1992].

Por otra parte, la utilización de un determinado modelo durante el Análisis₁ condiciona a la utilización de una clase de modelos –modelos compatibles– en todas las actividades de desarrollo y, por lo tanto, implica una forma determinada de abordar la solución del problema [Henderson-Sellers et al., 1990] [Jalote, 1997]. Teniendo en cuenta que cada modelo está orientado a un espectro determinado de problemas [Davis, 1993], y que cada aproximación de desarrollo se define por los modelos que utiliza, cabe la posibilidad de que se detecte que la aproximación de desarrollo utilizada no sea adecuada **después de realizar la modelización**. Sin embargo, y debido a la orientación a la solución de las distintas aproximaciones de desarrollo, **es imposible cambiar de aproximación si no se vuelve a realizar de nuevo la actividad de Análisis₁ bajo la aproximación seleccionada**.

La casi nula existencia de métodos y técnicas de traducción entre aproximaciones de desarrollo agrava este problema. Existen pocos trabajos que aborden la traducción entre aproximaciones ([Meyer, 1988] [Ward et al., 1989] [Kuo, 1994] [George et al., 1996] son los más importantes). Dichos trabajos proponen una serie de guías de traducción basadas en procesos heurísticos, insuficientemente formalizados y no aplicables en todos los casos. Sin embargo, la independencia de la implementación que debe poseer la actividad de Requisitos [IEEE, 1998] [Davis, 1993] [Kotonya et al., 1998] [SWEBOK, 2001] [Pressman, 2001] debe conseguirse mediante métodos y técnicas formalizados, y no mediante procedimientos heurísticos, siempre dependientes del analista. Ello lleva a que dichos métodos y técnicas no puedan ser aprendidos más que con la experiencia y no puedan ser transmitidos en la academia a los

nuevos profesionales. Adicionalmente, y al no basarse en criterios contrastados y aceptados universalmente, no existe certidumbre acerca de su adecuación y fiabilidad.

A todo ello debe sumarse la presión temporal y económica que sufren la mayoría de los proyectos software realizados en la industria, la cual impide que se pueda cambiar de aproximación de desarrollo incluso si se comprueba que no es la adecuada. Evidentemente, es muy difícil justificar una vez iniciado un proyecto de desarrollo, la necesidad de cambiar de aproximación, máxime cuando este cambio no es transparente sino que supone tiempo y dinero. Debido a ello, y aunque no existen datos en la literatura, es muy probable que muchos sistemas se construyan utilizando una aproximación de desarrollo que no es la adecuada para abordar el problema. Este hecho, obviamente, repercute en la calidad de la solución software obtenida. Sin embargo, la calidad de productos que se consideren ingenieriles –como pretende el software– no puede ser dependiente del ingeniero, sino de los métodos y técnicas que utilizan para la creación de dichos productos [Baber, 1997] [Vinceti, 1990].

1.4. OBJETIVO DEL TRABAJO

El presente Trabajo de Tesis tiene como finalidad redefinir la actividad de Análisis₁, tal y como se contempla en las aproximaciones de desarrollo actualmente utilizadas (especialmente la Estructurada y la Orientada a Objetos), de tal forma que:

1. **Se realice siempre una clara** diferenciación entre los modelos conceptuales y los modelos del sistema independientemente de la aproximación de desarrollo utilizada.
2. **Se pueda identificar de forma objetiva la aproximación de desarrollo más adecuada** para un problema planteado por el usuario.

En otras palabras, el presente trabajo de investigación pretende introducir la tarea de Análisis, tal y como se considera actualmente en la actividad de Requisitos [SWEBOK, 2001], como una parte de la actividad de Análisis₁, tal y como se contempla en las distintas aproximaciones de desarrollo. Para evitar nuevamente problemas de denominación, la tarea de Análisis que se pretende llevar a cabo durante el Análisis₁ se denominará pre-Análisis, tal y como se observa en la figura 1.2.

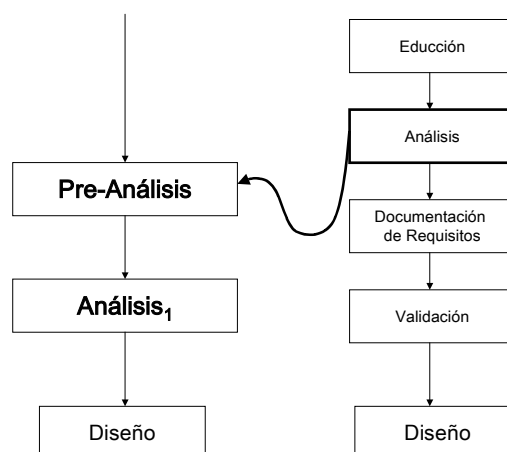


Figura 1.2. Relación entre los términos Análisis₁, Análisis y pre-Análisis

De esta forma, se pretende que cualquier aproximación de desarrollo, independientemente de qué modelos genere a la finalización del Análisis₁, **utilice siempre dos tipos de modelos: un conjunto de modelos conceptuales durante las fases tempranas de los Requisitos, con el objetivo de comprender el problema del usuario, y un conjunto de modelos –los propios de la aproximación de desarrollo– en las fases tardías de los Requisitos, con el objetivo de describir el comportamiento del futuro sistema software.**

Adicionalmente, es necesario considerar que cada aproximación de desarrollo prescribe la utilización de un conjunto determinado de modelos, así como un conjunto bien determinado de tareas para producir dichos modelos. Por ello, y con el propósito de no tener que considerar tantas alternativas de pre-Análisis como aproximaciones de desarrollo existentes, es preciso que tanto la **estructura de dicho pre-Análisis, como las técnicas a utilizar, sean flexibles y genéricas –para facilitar su introducción–, pero con la potencia suficiente para cumplir sus objetivos de facilitar la comprensión del dominio y detectar los conflictos e inconsistencias que puedan producirse.**

1.5. APROXIMACIÓN A LA SOLUCIÓN

Como se ha indicado anteriormente, debido a las diferencias entre aproximaciones de desarrollo en lo referido a las tareas que proponen, así como con respecto a los modelos que utilizan, el pre-Análisis que se propone introducir antes de llevar a cabo las tareas prescritas por cualquier aproximación de desarrollo concreta deberá poseer una gran generalidad o, lo que es lo mismo, un alto nivel de abstracción. Por alto nivel de abstracción se entiende que este pre-Análisis debería ser independiente de la tecnología subyacente, esto es, debe estar **orientado a la necesidad** planteada por el usuario, y debe permitir representar el problema planteado por el usuario en una serie de modelos conceptuales antes de comenzar a desarrollar una serie de modelos del sistema.

La solución propuesta consiste en anteponer el pre-Análisis a las tareas de Análisis₁ de cada aproximación concreta, tal y como se indica en la figura 1.3. Esta tarea de pre-Análisis constará de dos pasos procedimentales básicos:

1. Un primer paso de **Análisis Orientado al Problema**, cuyo objetivo será la definición de una serie de modelos conceptuales, denominados **Modelos Conceptuales Genéricos (MCG)**. Estos modelos son conceptuales, en la medida en que su función es favorecer la comprensión del dominio del problema. Sin embargo, son genéricos en la medida en que la clasificación que realizan de la información procedente de la Educación es muy lábil, por lo que cualquier modelo propuesto por cualquier aproximación de desarrollo será compatible con ellos, permitiendo así, una vez finalizado el pre-Análisis, proseguir con la actividad de Análisis₁ propuesta por cualquier aproximación de desarrollo.
2. Un segundo paso de **Análisis Orientado a la Solución**, cuyo objetivo consistirá en acoplar el MCG a cada aproximación de desarrollo concreta. Para ello, en el Análisis Orientado a la Solución se utilizará un **procedimiento de derivación** que permita, a partir de la información

recogida y clasificada en el Modelo Conceptual Genérico, obtener los modelos utilizados por cualquier aproximación de desarrollo. Los modelos producidos de esta forma no poseerán, en general, la misma calidad que se obtendría al aplicar desde el principio una aproximación de desarrollo concreta. Sin embargo, la mera posibilidad de derivar los modelos propios de cada aproximación es lo que permite anteponer la tarea de pre-Análisis, permitiendo completar en detalle dichos modelos durante el Análisis₁.

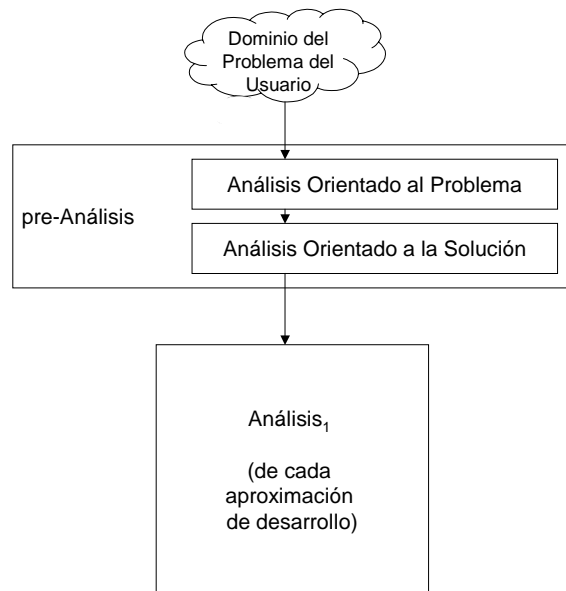


Figura 1.3. Localización del pre-Análisis

Debido a que, en general, va a ser posible generar un amplio rango de modelos al finalizar el pre-Análisis, se hace necesario identificar cuáles, de entre ellos, va a ser necesario obtener. Como ya se ha indicado anteriormente, las distintas aproximaciones de desarrollo utilizan conjuntos distintos de modelos conceptuales, por lo que una primera solución es generar únicamente aquellos modelos utilizados por la aproximación de desarrollo que se empleará durante el resto del proceso de desarrollo. Ahora bien; si la aproximación de desarrollo a emplear no está fijada de antemano, podría utilizarse una estrategia distinta, consistente en obtener los modelos conceptuales más adecuados para representar la necesidad del usuario y, en base a estos modelos, determinar la aproximación de desarrollo más prometedora. Para ello, se definirá dentro del Análisis Orientado a la Solución un **procedimiento de identificación**, que permita decidir qué modelo conceptual es el más adecuado para representar la necesidad del usuario.

El procedimiento de identificación, además de su función principal, posee otra utilidad. Como ya se ha indicado, cada modelo y, por ende, cada aproximación de desarrollo, es más útil en unos dominios que en otros [Davis, 1993]. Debido a que la utilidad de una aproximación viene determinada por la utilidad de sus modelos, poder determinar qué modelo será más completo equivale a determinar qué aproximación es más adecuada. Por ejemplo, supóngase que un determinado proyecto de desarrollo se abordase de dos formas distintas, bajo la Aproximación Estructurada y la Aproximación Orientada a Objetos. Si tras el Análisis₁, se obtuviese un DFD muy pobre, pero un Diagrama de Objetos muy rico, sería lógico pensar que la Aproximación Orientada a Objetos sería la más adecuada para acometer el desarrollo. Si ocurriera al

revés, esto es, se obtuviese un DFD muy rico, pero un Diagrama de Objetos muy deficiente, sería justo pensar que la aproximación más adecuada sería la Aproximación Estructurada.

El procedimiento de identificación permite evaluar qué modelo, y por tanto qué aproximación de desarrollo, es más adecuada sin llegar a desarrollar los modelos propios de cada aproximación. Ello implica que, en lugar de necesitar traducir modelos entre aproximaciones, **es suficiente con identificar la aproximación más adecuada desde el principio, y desapareciendo la necesidad de cualquier tipo de traducción posterior**, tal y como se ha establecido en los objetivos del trabajo de Tesis.

De esta forma, la figura 1.4 muestra un esquema más adecuado de cómo se integraría la tarea de pre-Análisis en las distintas aproximaciones de desarrollo.

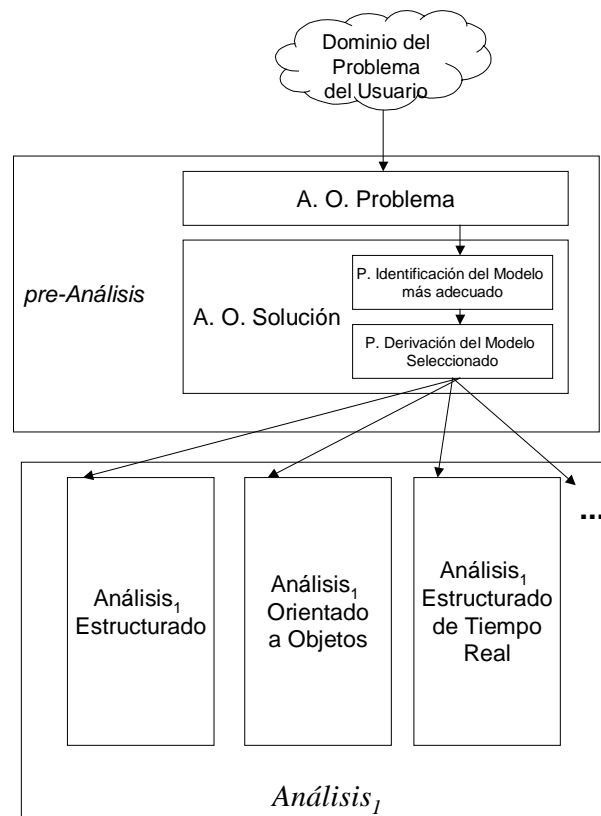


Figura 1.4. Integración del pre-Análisis en las distintas aproximaciones de desarrollo

En lugar de que los desarrolladores seleccionen la aproximación de desarrollo de un modo subjetivo, el procedimiento de identificación indicará qué aproximación de desarrollo es más adecuada dada la información registrada en el MCG. Posteriormente, mediante el procedimiento de derivación, será posible obtener una primera versión de los modelos conceptuales requeridos por la aproximación de desarrollo considerada más adecuada, con el propósito de iniciar la actividad de Análisis₁.

2. Estado de la Cuestión

2.1. INTRODUCCIÓN

En esta sección se estudiará cómo abordan las aproximaciones de desarrollo existentes la tarea de Análisis. Sin embargo, antes de proceder a dicho estudio, es necesario realizar ciertas consideraciones previas.

En primer lugar, es necesario precisar qué debe **entenderse por aproximación de desarrollo**. En efecto, no existe ningún método de desarrollo concreto que se conozca como Aproximación Estructurada, o Aproximación Orientada a Objetos. En su lugar, existen múltiples métodos, metodologías y propuestas con mayor o menor grado de formalización que pueden denominarse en su conjunto Aproximaciones Estructuradas o Aproximaciones Orientadas a Objetos. Es necesario precisar, por lo tanto, qué es una aproximación de desarrollo antes de proceder al estudio de cómo éstas afrontan la actividad de Análisis¹, y por ello la sección 2.2. estará dedicada a realizar dicha precisión.

La sección 2.2 desempeña un importante papel en el presente trabajo de Tesis debido a que establecerá que las distintas aproximaciones de desarrollo pueden caracterizarse por el tipo y número de modelos conceptuales que utilizan. La posibilidad de caracterizar una aproximación de desarrollo por sus modelos conceptuales es verdaderamente relevante, ya que una revisión de cómo aborda cada método, metodología, aproximación, etc. la tarea de Análisis supone un trabajo titánico debido al enorme número de métodos existentes. Se estima que existen varios centenares de métodos [Avison et al., 1995], pero no se conoce el número exacto de los mismos, ni existe un catálogo que indique los métodos concretos existentes. Esto, lógicamente, supone un inconveniente a la hora de realizar una revisión exhaustiva de la totalidad de los métodos de desarrollo.

En segundo lugar, se **identificará el conjunto de modelos conceptuales** utilizados por las Aproximaciones Estructurada, Orientada a Objetos, de Tiempo Real y de Bases de Datos. Ello se realizará en la sección 2.3.

En tercer lugar, se propondrán una serie de **criterios de evaluación** que se utilizarán para estudiar cómo se utiliza cada uno de los modelos durante el proceso de desarrollo. Dichos criterios se expondrán en la sección 2.4.

En cuarto lugar, se realizará un estudio de los distintos modelos conceptuales, basado en los criterios de evaluación definidos en la sección 2.4. Este estudio, que se realizará en la sección 2.5, y tendrá como objetivo verificar la existencia de los problemas indicados en la introducción del presente trabajo de Tesis, esto es: (1) Si los modelos están o no orientados a la solución, es decir, a definir cómo se implementará el futuro sistema software y (2) Si los modelos otorgan libertad al desarrollador para realizar el Diseño como considere más conveniente o si, por lo contrario, determinan una única forma de llevar a cabo el Diseño.

En quinto y último lugar, los resultados obtenidos, que suponen la aportación fundamental del presente capítulo de Estado de la Cuestión, se resumirán en la sección 2.6, **destacando las principales carencias de los modelos estudiados en cuanto al soporte que proporcionan para la realización de la actividad de Análisis en las distintas aproximaciones de desarrollo.**

2.2. CARACTERÍSTICAS DE LAS APROXIMACIONES DE DESARROLLO

Es posible entender el producto software como un conjunto de procesos, que operan sobre un conjunto de datos, modulados por los estímulos que el sistema software recibe del entorno donde está instalado.

Grosso modo, las aproximaciones de desarrollo existentes se caracterizan por centrar su atención en alguno de los aspectos indicados anteriormente –procesos, objetos o estímulos– [Davis, 1993]. Así, por ejemplo, las dos aproximaciones básicas de desarrollo, las aproximaciones Estructurada y Orientada a Objetos, obligan al analista a estudiar tanto las transformaciones que ocurren en el dominio del problema, como los objetos existentes en dicho dominio, aunque con diferente orientación. Adicionalmente a estas dos aproximaciones, deberán también contarse las Aproximaciones de Tiempo Real, las cuales focalizan su atención en los estímulos, así como las Aproximación de Bases de Datos, las cuales se centran en una variante de los objetos, que son los tipos de entidad estáticos que existen en el dominio del problema.

2.2.1. TIPOS DE MODELOS

Las procesos, objetos y estímulos, además de ser aspectos relevantes del dominio del problema, **coinciden con las tipologías más importantes de modelos conceptuales** [Davis, 1993]. No en vano, todos los métodos, metodologías, procedimientos, etc. pertenecientes a la Aproximación Estructurada se

¹ No se utiliza el término Análisis₁ debido a que no se volverá a hacer referencia directa a la tarea de Análisis dentro de la Especificación de Requisitos y, por ello, no hay posibilidad de confusión entre términos. En este y siguientes capítulos, se deberá entender, por lo tanto, que el término Análisis es equivalente a Análisis₁.

caracterizan por definir y utilizar modelos conceptuales orientados al proceso de datos, tales como el Diagrama de Flujo de Datos. De la misma forma, los métodos, metodologías, procedimientos, etc. pertenecientes a la Aproximación Orientada a Objetos utilizan modelos orientados a los objetos, tales como el Diagrama de Objetos; los pertenecientes a la Aproximación de Tiempo Real utilizan modelos orientados a la dinámica, tales como el Diagrama de Transición de Estados; y los pertenecientes a la Aproximación de Bases de Datos utilizan modelos orientados a los datos, como el Diagrama Entidad-Relación.

Como característica definitoria de una aproximación de desarrollo se puede considerar, entonces, el aspecto del dominio del problema en que focaliza su atención. Más concretamente, en lugar del aspecto concreto –procesos, objetos o estímulos–, **se puede afirmar que lo verdaderamente relevante son los modelos que permiten a cada aproximación filtrar la información del dominio del problema en función del aspecto que se considera más relevante.** Muestra palpable de este hecho es la coincidencia entre los modelos utilizados por métodos pertenecientes a una aproximación de desarrollo determinada. Por ejemplo, los modelos más relevantes de OMT [Rumbaugh et al., 1991] y UML [Larman, 1999] son similares (concretamente, los diagramas de transición de estados –o statecharts–, los diagramas de clases y los diagramas de secuencia).

No obstante, las distintas aproximaciones de desarrollo no utilizan un único modelo, sino que utilizan varios, de distintas características o, de forma más estricta, basados en distintas ontologías [Wand, 1996]. Los distintos modelos utilizados por una aproximación de desarrollo no son, sin embargo, privativos de ésta. Distintos métodos, propios de distintas aproximaciones, pueden llegar a utilizar los mismos modelos. Así, por ejemplo, dos métodos tan distintos como el Análisis Estructurado de Yourdon [Yourdon, 1989] y el Análisis Orientado a Objetos (OMT) de Rumbaugh [Rumbaugh et al., 1991], mostrados en la figura 2.1, utilizan cada uno de ellos 4 modelos, pero 2 de ellos son coincidentes. **La existencia de varios modelos complementarios permiten a cada aproximación organizar y clasificar una diversidad de informaciones del dominio del problema** [Beringer, 1996].

Un ejemplo clásico puede ser la dicotomía estático/dinámico del Análisis Estructurado de DeMarco [DeMarco, 1979]. Desde el punto de vista estático, DeMarco contempla el dominio del problema como compuesto, básicamente, de cosas (entidades) y asociaciones (relaciones), que explicita en un Diagrama Entidad-Relación. Desde el punto de vista dinámico, por el contrario, considera que el mundo se compone de procesos de transformación, explicitados mediante Diagramas de Flujo de Datos. La utilización de diversos modelos es lo que permite a las distintas aproximaciones de desarrollo complementar la visión del dominio que consideran más relevante (procesos de datos en el caso de las Aproximaciones Estructuradas, objetos en el caso de las Aproximaciones Orientadas a Objetos) expresando de esta forma una mayor riqueza de matices acerca del dominio del problema.

No obstante, **la utilización de varios modelos en una aproximación de desarrollo no significa que todos los modelos sean igual de relevantes.** En cada aproximación de desarrollo existe una relación jerárquica entre los modelos que utiliza, de tal forma que los modelos no coincidentes con la visión del dominio que propugna cada aproximación concreta (por ejemplo, el Diagrama Entidad-Relación en las Aproximaciones Estructuradas, o el Diagrama de Transición de Estados o *Statechart* en las Aproximaciones Orientadas a Objetos) se subordinan a los modelos que si son coincidentes con dicha visión del dominio (Diagrama de Flujo de Datos en las Aproximaciones Estructuradas, Diagrama de Clases en las

Aproximaciones Orientadas a Objetos). Ello permite afirmar que un determinado modelo, en una aproximación concreta, es dominante respecto a los otros. Este asunto se discute en la siguiente sección.

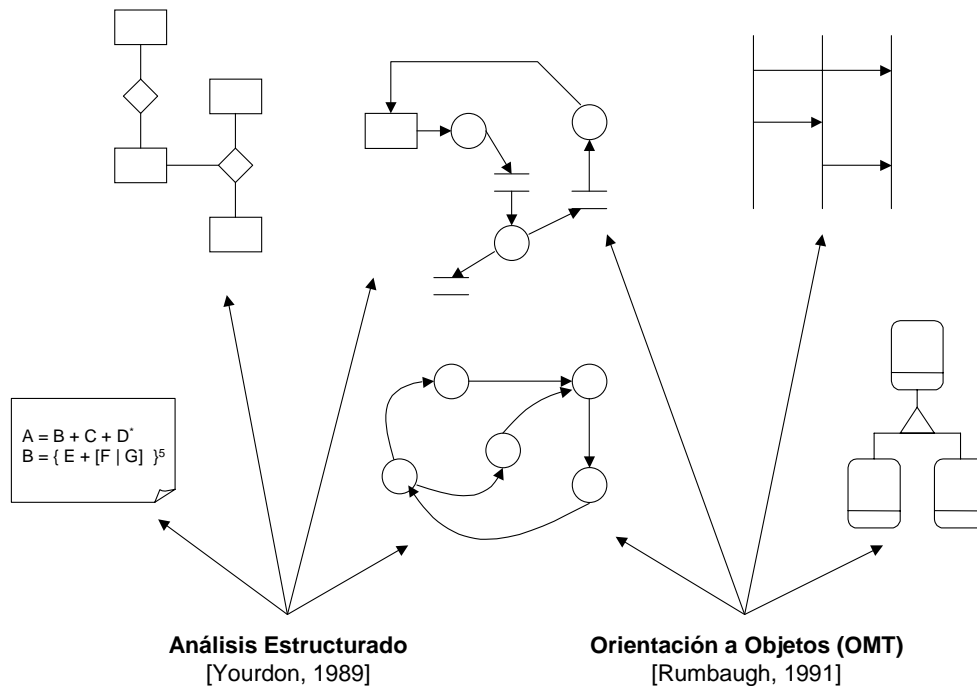


Figura 2.1. Modelos utilizados por el Análisis Estructurado y OMT

2.2.2. MODELOS DOMINANTES Y COMPLEMENTARIOS

Un problema común en todas las aproximaciones de desarrollo es la unificación de las informaciones organizadas y clasificadas en los distintos modelos para derivar un diseño posterior que contemple, en pie de igualdad, toda la información obtenida. Debido a dicho problema de integración, todas las aproximaciones poseen un **modelo dominante** [Yourdon, 1992], que guía todo el desarrollo posterior, tal y como puede observarse en la figura 2.2. En todas las aproximaciones de desarrollo, el modelo dominante es el modelo que se centra en aquellos aspectos del dominio del problema que la aproximación de desarrollo considera más relevantes; esto es, el modelo dominante es aquel que organiza la información del dominio del problema en concordancia con la visión del mundo propugnada por la aproximación de desarrollo.

Así, por ejemplo, el Diagrama de Clases es el modelo dominante de un desarrollo llevado a cabo con la Aproximación Orientada a Objetos; mientras que, en el caso de la Aproximación Estructurada, el modelo dominante es el Diagrama de Flujo de Datos o alguna de sus variantes.

No debe confundirse dominancia con unicidad. Ninguna aproximación utiliza un único modelo, sino varios. Como ya se ha indicado, cada uno de los modelos proporciona una determinada perspectiva del dominio del problema, pero que el modelo dominante actúa como catalizador de todo desarrollo posterior, mientras que los restantes modelos apoyan o matizan las decisiones tomadas a partir del modelo dominante.

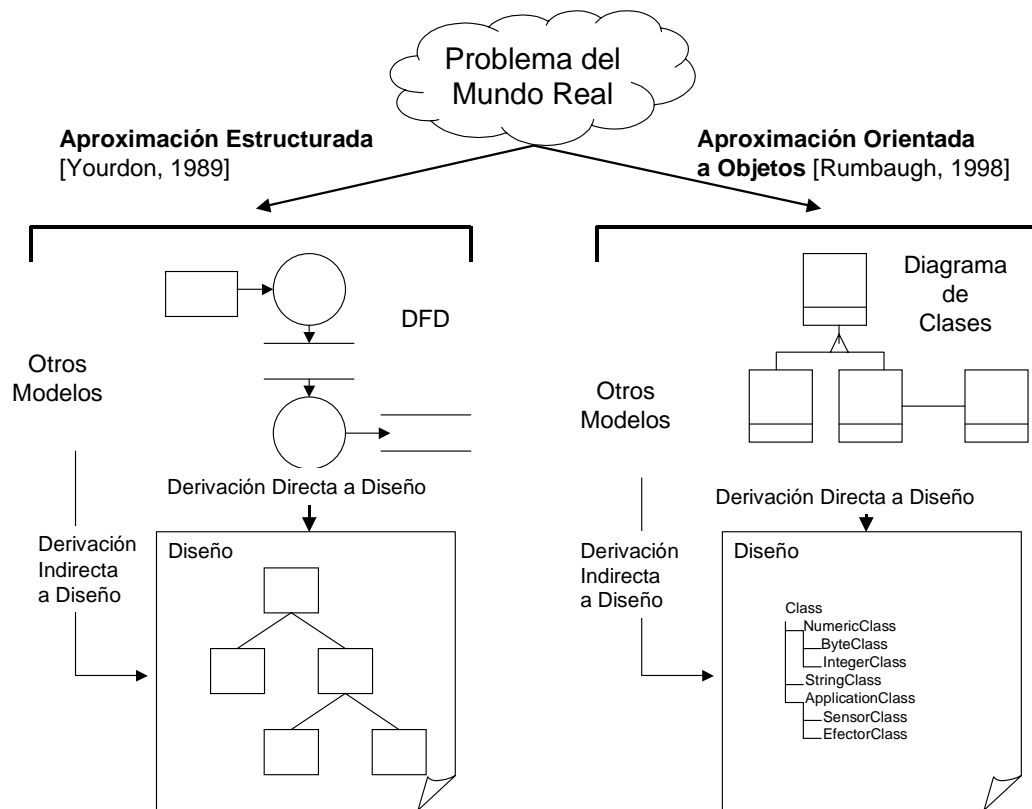


Figura 2.2. Modelos Dominantes

La existencia de un modelo dominante es la razón de que cada aproximación actúe, como se ha indicado en el capítulo 1, a modo de unas gafas especiales, que el ingeniero usa para observar el dominio y la realidad del usuario. Estas gafas destacan ciertos aspectos, amortiguan otros, y ocultan algunos. Una vez pasada la realidad por el filtro del modelo dominante, recuperar los aspectos perdidos y disminuidos es difícil. El modo más económico de recuperar los aspectos de la realidad perdidos en el filtrado producido por el modelo dominante es volver a analizar la realidad con otras gafas; es decir, repetir la operación con otro modelo distinto, probablemente perteneciente a una aproximación de desarrollo distinta.

Por tanto, el modelo dominante se utiliza para: (1) organizar y clasificar la información del dominio del problema y (2) tomar las principales decisiones del desarrollo. Dado que el modelo dominante mediatiza el desarrollo posterior, cada aproximación de desarrollo utilizará, necesariamente, modelos complementarios al modelo dominante, con el propósito de:

- Reflejar aspectos importantes del problema que el modelo dominante no representa adecuadamente. Por ejemplo, el diagrama Entidad-Relación se utiliza junto con el Diagrama de Flujos de Datos en las Aproximaciones Estructuradas [Yourdon, 1989] con el objetivo de describir la estructura de los datos del dominio del problema, aspecto que el Diagrama de Flujo de Datos solo permite describir parcialmente.

- Continuar el desarrollo a partir del modelo dominante; por ejemplo, los Diagramas de Estructuras para el Diseño [Yourdon et al., 1986], pueden derivarse fácilmente a partir del Diagrama de Flujo de Datos y permiten una transición sencilla entre el Análisis y el Diseño.

En consecuencia, la existencia de un modelo dominante es lo que permite caracterizar una aproximación de desarrollo. **Se considerará aproximación de desarrollo aquel conjunto de métodos, metodologías, procedimientos, etc. que posean un mismo modelo dominante** (o diversos modelos, prácticamente equivalentes, dominantes). Adicionalmente, cada aproximación de desarrollo poseerá un conjunto, variable, de modelos compatibles que apoyarán o permitirán matizar la información, así como las decisiones de diseño (nótese que las aproximaciones de desarrollo no distinguen modelos conceptuales de modelos del sistema) que se tomen a partir del modelo dominante.

2.3. MODELOS CONCEPTUALES UTILIZADOS POR LAS APROXIMACIONES DE DESARROLLO

2.3.1. CARACTERIZACIÓN DE LAS APROXIMACIONES DE DESARROLLO

Dado que, cada aproximación de desarrollo puede caracterizarse por el conjunto de modelos (dominantes y complementarios) que utiliza, tras una revisión exhaustiva de la literatura, se han identificado una serie de modelos que, en el presente trabajo de Tesis, se denominan **Modelos Paradigmáticos**. También se ha identificado una serie más larga de modelos que podrían denominarse **Modelos Secundarios**. Por modelo paradigmático debe entenderse aquel modelo que es paradigma de un tipo o clase de modelos. Esto es, un modelo paradigmático es aquel modelo similar a toda una clase de modelos que permiten representar la misma información. Estos últimos modelos forman lo que se ha denominado Modelos Secundarios.

La diferencia entre modelos paradigmáticos y secundarios es relevante a efectos simplificadores. Por ejemplo, existen multitud de variedades de diagramas de clases y objetos, conocidos bajo distintos nombres y denominaciones [Coad et al., 1990] [Rumbaugh et al., 1991] [Meyer, 1988] [Rumbaugh et al., 1998]; sin embargo, los conceptos que los diagramas de clases permiten representar son similares: clases y atributos, métodos, asociaciones, agregaciones, herencia, etc. Así, en lugar de identificar varios diagramas de clases distintos, y proceder a un estudio que, debido a la similitud entre los mismos, produciría los mismos resultados, se ha optado por considerar únicamente un representante de los diagramas de clases (el modelo paradigmático) que reúne las características de una pluralidad de modelos secundarios.

Obrando de la forma indicada, se han identificado los modelos indicados en la figura 2.3. Con el fin de sistematizar lo más posible el estudio de los mismos, se ha realizado una clasificación de dichos modelos atendiendo al aspecto del dominio del problema que cada modelo considera más relevante. Así, por ejemplo, un Diagrama de Flujo de Datos se considera clasificado en la clase de modelos Orientada a la Transformación. De la misma forma, un Diagrama de Clases está clasificado como Orientado a los Objetos/Datos.

Dicha clasificación representa, adicionalmente, una división del conjunto de los modelos en clases de equivalencia, matemáticamente hablando. Esto significa que cualquier otro modelo (los que se han denominado modelos secundarios) no contemplado en la figura 2.3 debe poder encuadrarse en la clasificación y, además, ser equivalente a alguno de los modelos paradigmáticos allí indicados. Por ejemplo, un modelo como el Diagrama de Flujo de Datos de Gane y Sarson [Gane et al., 1979] puede encuadrarse en la clase de modelos orientados a la transformación y, además, es similar al Diagrama de Flujo de Datos (de DeMarco [DeMarco, 1979]) allí recogido. Por lo tanto, el Diagrama de Flujo de Datos de Gane y Sarson no será estudiado y, en su lugar, se estudiará en Diagrama de Flujo de Datos de DeMarco como paradigma de dicho modelo, pero puede considerarse que el DFD de Gane y Sarson está contemplado en esta revisión de aproximación de desarrollo.

La clasificación mostrada en la figura 2.3 es, simplemente, un mecanismo de organización de una diversidad de modelos conceptuales propios de las cuatro aproximaciones de desarrollo (Estructurada, Orientada a Objetos, Bases de Datos y Tiempo Real). Por ello, pueden aparecer modelos dominantes en cualquier clase, y es posible que alguna clase (por ejemplo, los modelos orientados al procedimiento) no contenga ningún modelo dominante, sino únicamente modelos compatibles. Asimismo, es posible que un modelo perteneciente a cierta clasificación sea dominante respecto a alguna aproximación y compatible respecto a otra. El orden en que se muestran los distintos modelos en la figura 2.3 no posee, asimismo, relevancia alguna.

Nótese, no obstante, que la dicotomía dominante-compatible no es de interés, en sí misma, a la hora de estudiar los distintos modelos, sino a la hora de resumir los resultados del estudio de los mismos con el fin de verificar si las distintas aproximaciones de desarrollo poseen, o no, los problemas que aborda la presente investigación.

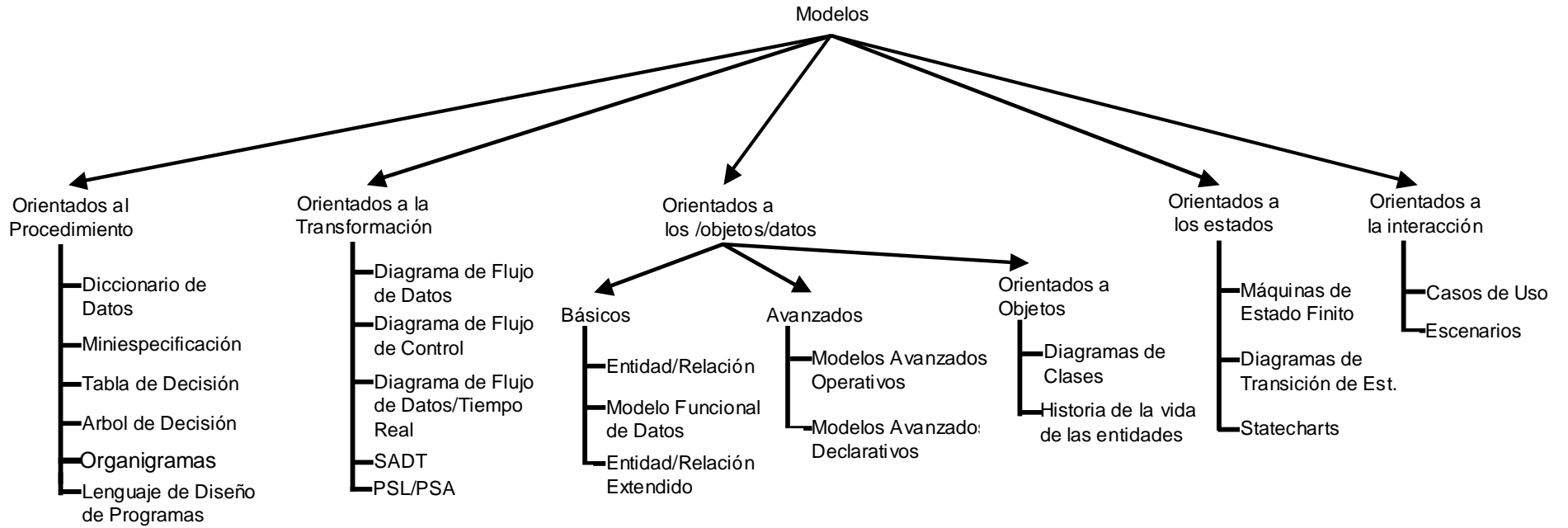


Figura 2.3. Clasificación de modelos paradigmáticos

2.3.2. CLASIFICACIÓN DE MODELOS PARADIGMÁTICOS

Los modelos conceptuales (paradigmáticos) que se han seleccionado para su estudio en la sección 2.5 se detallan a continuación, convenientemente organizados en clases o tipos:

1. Modelos Orientados al Procedimiento. Dentro de este grupo se han considerado cinco modelos distintos:

- Miniespecificación [DeMarco, 1979].
- Tabla de Decisión [Yourdon, 1989].
- Árbol de Decisión² [Yourdon, 1989].
- Diccionario de Datos [DeMarco, 1979].
- Flujogramas [Yourdon, 1989].
- Lenguaje de Diseño de Programas [Caine et al., 1975].

2. Modelos Orientados a la Transformación. Dentro de este grupo se pueden identificar cinco modelos distintos:

- Diagrama de Flujo de Datos [DeMarco, 1979].
- Diagrama de Flujo de Control [Hatley, 1984].
- Diagrama de Flujo de Datos para Tiempo Real [Ward, et al., 1985].
- Técnica de Análisis y Diseño Estructurado (SADT) [Ross, 1977].
- PSL/PSA [Teichroew et al., 1971].

3. Modelos Orientados a los Objetos/Datos. Debido a la enorme diversidad de modelos pertenecientes a esta clase, es necesario realizar una subdivisión para no agrupar modelos excesivamente dispares. En concreto, se van a distinguir los siguientes subgrupos:

- **Modelos Básicos:** Este subgrupo contiene a los denominados Modelos Semánticos de Datos³. Los principales Modelos Semánticos de Datos son:
 - Modelo Entidad-Relación [Chen, 1976], paradigma de todos los modelos del tipo Entidad-Relación.
 - Modelo Funcional de Datos [Abrial, 1974], paradigma de los modelos Objeto-Papel.

² Determinados autores, como [Davis, 1993], no consideran las tablas y árboles de decisión como Modelos Conceptuales. Sin embargo, su utilización durante el Análisis en las Aproximaciones Estructuradas, así como en las Aproximaciones Orientadas a Objetos, justifica su inclusión.

³ Los modelos semánticos de datos nacen en el campo de las Bases de Datos, al amparo de la norma ANSI/X3/SPARC [ANSI, 1975]. Este tipo de modelos se utilizaron, inicialmente, para describir la estructura de los datos independientemente de su implementación en una base de datos. No obstante, los modelos semánticos de datos se utilizan en todos los métodos pertenecientes

- Modelo Entidad-Relación Extendido [Batini et al., 1986], el cual se distingue del Modelo Entidad-Relación por la introducción del concepto de generalización.
 - **Modelos Avanzados**, los cuales son una evolución de los modelos semánticos de datos. En este subgrupo se distinguirán dos modelos⁴ paradigmáticos:
 - Modelos Avanzados Operativos [Brodie et al., 1982].
 - Modelos Avanzados Declarativos [Gustafsson et al., 1982].
 - **Modelos Orientados a Objetos**. Este subgrupo contiene un conjunto de modelos que se caracterizan por no separar los conceptos y las operaciones, sino considerar que ambas, conjuntamente, forman un todo denominado objeto. Existen dos modelos paradigmáticos:
 - Diagramas de clases [Rumbaugh et al., 1998].
 - Historia de la Vida de las Entidades [Jackson, 1983]⁵.
- 4. Modelos Orientados a los Estados.** En este grupo se incluyen los siguientes modelos paradigmáticos⁶:
- Máquinas de estado Finito [Hatley, 1984].
 - Diagramas de Transición de Estados [Youdon, 1989].
 - Statecharts [Harel, 1981].
- 5. Modelos Orientados a la Interacción.** En este grupo se han identificado los modelos siguientes:
- Casos de Uso [Jacobson, 1992]
 - Escenarios [Larman, 1999].

a la Aproximación Estructurada desde [Palmer et al., 1984], así como en métodos *pseudo-Estructurados* más difíciles de clasificar como *Information Engineering* [Martín, 1990], y por ello son considerados en el presente trabajo de Tesis.

⁴ Los Modelos Avanzados Operativos y los Modelos Avanzados Declarativos son “modelos ideales”. Por modelo ideal debe entenderse como un modelo que engloba todas las características de los modelos reales sin coincidir plenamente con ninguno de ellos. Se ha optado por estudiar dichos modelos ideales debido a que es difícil encontrar un representante “real” de los Modelos Avanzados Operativos y de los Modelos Avanzados Declarativos que sea universalmente reconocido como paradigma. No obstante, se indicarán en el epígrafe correspondiente los modelos reales que engloban dichos modelos ideales.

⁵ La Historia de la Vida de las Entidades es difícil de clasificar, ya que posee características mixtas entre los modelos orientados a la dinámica y los modelos orientados a objetos. Se ha optado por clasificarla en este último grupo porque se estima que sus características de orientación a objetos prevalecen sobre sus características dinámicas.

⁶ Determinados autores, como [Davis, 1993], no consideran estos modelos como modelos conceptuales. Sin embargo, su utilización durante el Análisis en las Aproximaciones Estructurada, Orientada a Objetos y de Tiempo Real justifican su inclusión.

2.3.3. RELACIÓN ENTRE LOS MODELOS PARADIGMÁTICOS Y LAS APROXIMACIONES DE DESARROLLO

En la tabla 2.1 se explicita la relación entre los modelos identificados y las distintas aproximaciones de desarrollo. Aunque esta tabla no posee una utilidad directa en este punto, se adjunta para permitir al lector una fácil relación entre los modelos conceptuales identificados y las distintas aproximaciones de desarrollo. Se utiliza la letra D para indicar que un modelo es dominante dentro de una aproximación (o sea, es dominante en algún método, metodología, etc. perteneciente a dicha aproximación). La letra C se utilizará para indicar que un modelo es complementario dentro de una aproximación de desarrollo.

Nótese que, dado que cada aproximación de desarrollo se caracteriza por poseer una determinada visión del dominio del problema, las aproximaciones no suelen poseer un modelo dominante perteneciente a cada grupo de modelos. Así, por ejemplo, las Aproximaciones Estructuradas poseen modelos dominantes pertenecientes al grupo de modelos orientados al procedimiento, pero no modelos dominantes que pertenezcan a grupo de modelos orientados a los objetos/datos. Lo mismo es aplicable a las restantes aproximaciones de desarrollo.

Grupo	Subgrupo	Modelo Paradigmático	Aproximación Estructurada	Aproximación Orientada a Objetos	Aproximación de Bases de Datos	Aproximación de Tiempo Real
Orientados al Procedimiento		Miniespecificación	C			C
		Tabla de Decisión	C			C
		Árbol de Decisión	C			C
		Diccionario de Datos	C			C
		Flujograma	C			
		Lenguaje de Diseño de Programas	C			C
Orientados a la Transformación		Diagrama de Flujo de Datos	D	C		
		Diagrama de Flujo de Control	D			D
		Diagrama de Flujo de Datos / Tiempo Real	D			D
		SADT	D			
		PSL/PSA	D			
Orientados a los objetos / datos	Básicos	Modelo Entidad-Relación	C		D	C
		Modelo Entidad-Relación Extendido	C		D	C
		Modelo Funcional del Datos			D	
	Avanzados	Modelos Avanzados Operativos			D	
		Modelos Avanzados Declarativos			D	
	Orientado a Objetos	Diagrama de Clases			D	
Historia de la Vida de las Entidades		C		D		
Orientados a la Dinámica		Máquinas de Estado Finito				C
		Diagrama de Transición de Estados	C	C		D
		Statechart	C	C		D
Orientados a la Interacción		Casos de Uso		C		
		Escenarios		C		

Tabla 2.1. Correspondencia entre los modelos seleccionados y las aproximaciones de desarrollo

2.3.4. MODELOS EXCLUIDOS

En la clasificación anteriormente indicada no se han tenido en cuenta un conjunto de formalismos de representación existentes en la literatura. Dichos formalismos de representación poseen un aspecto similar al de los modelos conceptuales, pero sin embargo no son utilizados durante la actividad de Análisis por las distintas aproximaciones de desarrollo.

En este punto, y con el fin de justificar la exclusión realizada de ciertos modelos, es conveniente incidir en que la actividad de Análisis tiene como objetivo organizar la información obtenida de los clientes y, o, usuarios, con la finalidad de, durante la actividad de Diseño, determinar la estructura y funcionalidad de un sistema software que satisfaga dicha necesidad. Es por ello que, en el presente trabajo, sólo son interesantes aquellos modelos que cumplen dos condiciones: (1) puedan utilizarse durante la actividad de Análisis y (2) puedan servir de base para la realización de la actividad de Diseño, ya que, precisamente, en la continuidad entre tareas es donde surgen los problemas que poseen los modelos conceptuales. La mayoría de los modelos utilizados habitualmente en la Ingeniería del Software cumplen las dos condiciones indicadas anteriormente pero, no obstante, existen excepciones. Concretamente, los modelos que han sido excluidos son los siguientes:

- Modelos utilizados exclusivamente durante la actividad de Análisis, pero que no son posteriormente utilizados para llevar a cabo la actividad de Diseño. Esto es, modelos que cumplen (1) pero no (2):
 - o I* [Yu et al., 1994] [Yu, 1995].
 - o KAOS [Lansweerde et al., 1991] [Dardenne et al., 1993].
 - o Enterprise Modelling [F3, 1991] [Kirikova et al., 1994a] [Kirikova et al., 1994b].
- Modelos no utilizados durante la actividad Análisis, sino exclusivamente para confeccionar la Especificación de Requisitos del Software y, o, para llevar a cabo la actividad de Diseño. Esto es, modelos que cumplen (2) pero no (1):
 - o PAISley [Zave et al., 1981] [Zave, 1982] y similares.
 - o Z [Spivey, 1989], VDM [Jones, 1990], ampliaciones de métodos formales para especificación [Zave et al., 1996], etc.
 - o R-NET [Alford, 1977] [Alford, 1985] y similares (SDL [Rockstrom et al., 1982], RLP [Davis, 1978], etc.).
 - o Redes de Petri [Petri, 1962] [Peterson, 1977] y similares.
 - o Problem frames [Jackson, 1994] [Jackson, 2001], KSI [Jackson, 1995].
 - o Diagramas de secuencia o de interacción [Larman, 1999].

2.4. CRITERIOS DE VALORACIÓN

Para realizar el análisis de modelos indicados anteriormente, es necesario establecer un conjunto de criterios que permitan un estudio objetivo de los distintos modelos. Estos criterios se dividen en dos grupos ortogonales:

- **Criterios orientados al proceso**, los cuales permiten estudiar la utilización de los modelos dentro de las aproximaciones de desarrollo.
- **Criterios orientados al modelo**, los cuales permiten estudiar las cualidades de representación de los distintos modelos conceptuales, esto es, establecen la existencia, o no, de determinadas propiedades en los distintos modelos.

Los criterios orientados al proceso que se han escogido son los siguientes:

- **Procedimiento de Uso**. Este criterio determina si, en las aproximaciones de desarrollo que utilizan el modelo, existe una serie de pasos, definidos con el suficiente nivel de detalle, para guiar al analista en la confección del modelo.

Este criterio podrá tomar tres valores distintos: **Ninguna**, si no se establece una guía para realizar el análisis; **Parcial**, si existe un proceso genérico o un conjunto de reglas heurísticas y **Total**, si existe un procedimiento detallado para la confección del modelo.

- **Selección de Diseño**. Este criterio determina si, en las aproximaciones de desarrollo, existe algún procedimiento que permita decidir qué tipo de diseño es más apropiado utilizar o, si por el contrario, la aproximación prescribe, directamente, la utilización de un conjunto de modelos determinados (modelos compatibles) durante la actividad de Diseño.

Este criterio podrá tomar cuatro valores distintos: **No aconseja**, si las aproximaciones de desarrollo no definen qué diseño es más apropiado realizar; **Aconseja**, si dichas aproximaciones recomiendan o priorizan las distintas alternativas de diseño posibles; **Determina**, si se define qué diseño es más aconsejable realizar y **Prescribe**, si se determina unívocamente un tipo de diseño.

- **Derivación de Diseño**. Este criterio determina si, una vez identificado el tipo de diseño a utilizar, las aproximaciones de desarrollo disponen de algún tipo de procedimiento que permita derivar los modelos del tipo de diseño seleccionado.

Este criterio podrá tomar tres valores distintos: **Ninguno**, si las aproximaciones de desarrollo no establecen procedimiento alguno para derivar un diseño; **Parcial**, si existe un procedimiento genérico o un conjunto de reglas heurísticas y **Total**, si existe un proceso detallado para transformación de los modelos conceptuales en productos de diseño, sin que ello signifique, en cualquier caso, que dicha derivación es completamente automatizable.

Los criterios orientados al modelo que se han escogido son los siguientes:

- **Amplitud.** Este criterio determina si toda la información relevante del dominio del problema es representable por el modelo. Todos aquellos conceptos que no se pueden representar directamente en el modelo, acostumbran a ser plasmados mediante ciertos trucos permitidos por la notación de los distintos modelos. Estos trucos suponen, en la mayoría de los casos, restricciones en el diseño del futuro sistema.

Este criterio podrá tomar tres valores distintos: **Total**, cuando el modelo permita representar directamente todos los conceptos del dominio del problema y **Parcial**, cuando el modelo represente únicamente una parte del dominio del problema, o necesite forzar sus notaciones para representar algún aspecto de dicho dominio. Se utilizará el valor **Ninguno** cuando el modelo no explicita conceptos del dominio del problema, sino que soporta la utilización de algún otro modelo (por ejemplo, las miniespecificaciones, que soportan la descripción de los procesos de un Diagrama de Flujo de Datos).

- **Ligaduras Computacionales** Este criterio determina si el modelo contiene notaciones que permiten representar elementos que no se corresponden con ningún aspecto del dominio del problema. Estos conceptos, en la mayoría de los casos, son exponentes del paradigma computacional subyacente y determinan una forma única de implementar el futuro sistema. Este criterio podrá tomar dos valores distintos, que por simplicidad se denominarán **Sí** y **No**.

Los criterios indicados anteriormente se pueden poner en relación con las distintas tareas de desarrollo, principalmente referidas a la actividad de Requisitos, tal y como muestra la figura 2.4. Así, el criterio Procedimiento de Uso indica si las aproximaciones de desarrollo proponen algún tipo de procedimiento para llevar a cabo la modelización. El criterio Amplitud mide cuánto del dominio del problema es capaz de abarcar cada modelo, mientras que el criterio Ligaduras Computacionales mide cuánta información en el modelo no proviene, realmente, del dominio del problema.

Finalmente, los criterios Selección de Diseño y Derivación de Diseño determinan la ayuda que las distintas aproximaciones proporcionan al analista y diseñador a la hora de realizar la transición entre los Requisitos y el Diseño, en los términos de qué tipo de diseño utilizar y cómo conseguir, aún a modo provisional, un borrador del tipo de diseño seleccionado.

Por último, y con el objetivo de poder determinar las ventajas y carencias de los distintos modelos, es preciso establecer algún criterio de bondad en función de los criterios elegidos. Se considera que un **modelo es adecuado para realizar la actividad de Análisis** cuando *permita al ingeniero comprender el problema sin introducir ningún tipo de sesgo que obligue a adoptar*, durante la actividad de Diseño, una aproximación de desarrollo determinada. Adicionalmente, sería aconsejable que las capacidades expresivas del modelo fuesen lo más completas posibles, de tal forma que *cualquier tipo de información concerniente al dominio del problema pudiese ser representada*. Por último, sería deseable que, con el fin de evitar el juicio subjetivo del ingeniero, *el modelo indicase de alguna forma qué tipo de aproximación de desarrollo es más adecuada*, aunque ello no implique de ninguna forma que el modelo fije de antemano el tipo de diseño a utilizar.

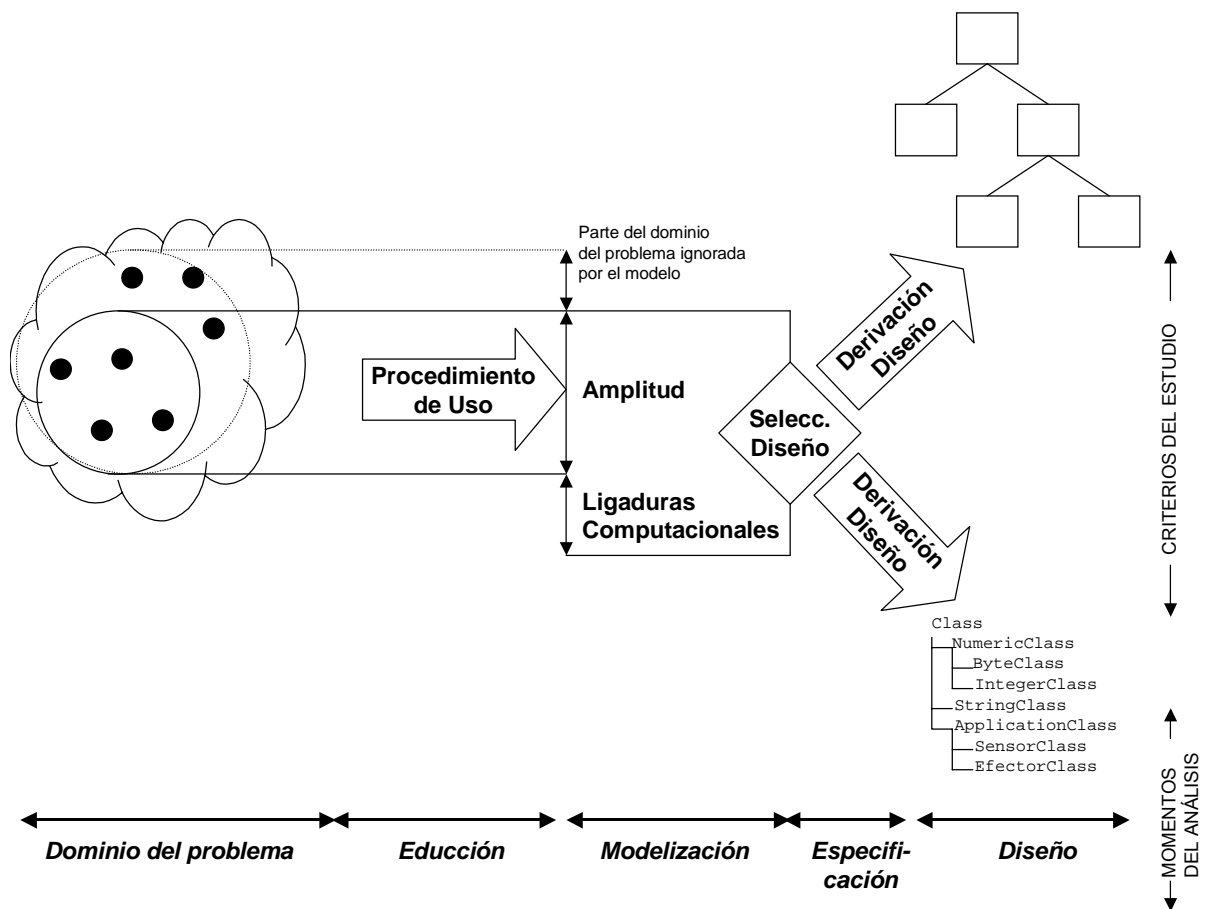


Figura 2.4. Relaci3n de los criterios elegidos con el desarrollo de software

Los criterios definidos anteriormente permiten establecer un patr3n de referencia que define cu3ndo un modelo es adecuado. Se considerar3 que un modelo es adecuado para realizar la actividad de An3lisis cuando su estudio proporcione los siguientes resultados en los criterios de valoraci3n:

- **Criterios Orientados al Proceso:**
 - o Procedimiento de Uso: **Total**
 - o Selecci3n de Diseño: **Determina**
 - o Derivaci3n de Diseño: **Total**
- **Criterios orientados al Modelo:**
 - o Amplitud: **Total**
 - o Ligaduras Computacionales: **No**

A continuación, y después de la identificación de modelos paradigmáticos y del establecimiento de los criterios de estudio de los distintos modelos, se realizará en la siguiente sección el estudio de los modelos en sí.

2.5 . REVISIÓN DE MODELOS CONCEPTUALES

En los siguientes epígrafes, se procederá al estudio de los distintos modelos conceptuales se han indicado en la clasificación de la sección 2.3. En su estudio se procederá, a menos que se indique lo contrario, de acuerdo con el siguiente esquema:

- Descripción de las características fundamentales del modelo.
- Enumeración de las aproximaciones/métodos que usan dicho modelo. Categorización de las aproximaciones/métodos en función de la utilización de dicho modelo.
- Valoración de los criterios para el modelo.

Al final del estudio de cada grupo de modelos, se confeccionará una tabla, que resumirá los resultados de la valoración de los criterios.

2.5.1. MODELOS ORIENTADOS AL PROCEDIMIENTO

Este conjunto de modelos se utilizan para registrar información concerniente a cómo se realizan determinadas operaciones en el dominio del problema, o para explicitar la estructura de la información existente en dicho dominio. Su finalidad es evitar la utilización de enunciados en lenguaje natural, debido al riesgo de ambigüedad intrínseco a este tipo de lenguaje [Eco, 1994]. Forman parte de este grupo la **Miniespecificación**, las **Tablas de Decisión**, los **Árboles de Decisión**, el **Diccionario de Datos**, los **Organigramas** y el **Lenguaje de Diseño de Programas**.

2.5.1.1. MINIESPECIFICACIÓN

2.5.1.1.1. Descripción

La miniespecificación aparece en [DeMarco, 1979] asociadas a los Diagramas de Flujo de Datos. La finalidad de la miniespecificación es describir la funcionalidad de un proceso no descomponible, o atómico, del Diagrama de Flujo de Datos. Dichos proceso atómico se corresponde con una transformación simple, o bien definida, que ocurre en el dominio del problema.

No existe una notación estándar para realizar una miniespecificación. Inicialmente, en [DeMarco, 1979], se propone que las miniespecificaciones se confeccionen utilizando el lenguaje natural. Actualmente, las miniespecificaciones tienden a realizarse utilizando formalismos como las tablas de decisión, árboles de decisión, flujogramas, Lenguaje de Diseño de Programas (LDP) o incluso gráficos [Yourdon, 1989]. Un ejemplo de miniespecificación se muestra en la figura 2.5.

2.5.1.1.2. Métodos de Análisis

Las miniespecificaciones se han usado únicamente en los métodos de Análisis Estructurado como modelo complementario de los Diagramas de Flujo de Datos [DeMarco, 1979] [Gane et al., 1979] [Palmer et al., 1984] [Orr, 1977] [Orr, 1981] [Yourdon, 1989]. También se utiliza en los DFD de Hatley [Hatley, 1984] [Hatley et al., 1987], y en el DFD/RT [Ward et al., 1985] [Ward, 1986].

Las facturas se calcularán sumando todos los pedidos realizados por un cliente en el último mes. Dichas facturas deberán almacenarse en el fichero de Facturas.

Los pedidos del cliente se encuentran en el fichero de Pedidos. No obstante, Es necesario tener en cuenta que el fichero de Pedidos no se borra cada mes, por lo que es necesario comprobar la fecha de cada pedido antes de sumar el importe al total de la factura.

Figura 2.5. Ejemplo de miniespecificación

2.5.1.1.3. Valoración

El desarrollo de miniespecificaciones **no está definido en ningún procedimiento**. Adicionalmente, **no existe ningún procedimiento que permita identificar qué tipo de diseño es más adecuado, ni que permita derivar dicho diseño**.

En lo referido a la capacidad de representación de la miniespecificación, al estar basada en el lenguaje natural, debería permitir representar todos los conceptos del dominio del problema. Sin embargo, esto no es cierto, ya que al usarse como apoyo de los Diagramas de Flujo de Datos, **su dominio del problema es el propio Diagrama de Flujo de Datos, y los conceptos que describe no son los existentes en el mundo real, sino los representados en dicho diagrama**.

La existencia de ligaduras computacionales va a depender del modo de utilización de las miniespecificaciones. Idealmente, las miniespecificaciones se utilizan para describir un proceso de transformación simple en el dominio del problema. Sin embargo, en la práctica, **se tiende a describir el mecanismo de implementación de dicha transformación en la computadora**, y no al modo en que se produce ésta en el dominio del problema.

2.5.1.2. TABLAS DE DECISIÓN

2.5.1.2.1. Descripción

Las tablas de decisión son modelos que no se han desarrollado dentro del campo de la Informática, aunque se ha usado extensivamente en el mismo. Su finalidad es describir las acciones que se deben

realizar cuando se dan una serie de condiciones en el dominio del problema. Las tablas de decisión están formadas por tres elementos constituyentes:

1. Una lista exhaustiva de condiciones. Las condiciones se especifican habitualmente utilizando lenguaje natural.
2. Una lista exhaustiva de acciones. Al igual que las condiciones, las acciones también acostumbran a especificarse utilizando lenguaje natural.
3. Una lista de combinaciones de acciones a las que se le asocia una o más acciones posibles.

Estos tres elementos se ordenan en una tabla, como la mostrada en la figura 2.6. La tabla permite relacionar directamente la combinación de condiciones y la acción que dispara.

Acciones en Caso de Incendio

Condiciones	Alarma Activada	Y	Y	Y	Y
	Puertas Bloqueadas	Y	Y	N	N
	Teléfono Averiado	Y	N	Y	N
Acciones	Avisar Bomberos		X		X
	Salir Ordenadamente			X	X
	Rezar lo que se Sepa	X	X		

Figura 2.6. Ejemplo de tabla de decisión

2.5 .1.2.2. Métodos de Análisis

Este formalismo de representación no es exclusivo de ningún método. Se utilizan preferentemente como alternativa al lenguaje natural en las miniespecificaciones [Yourdon, 1989].

2.5 .1.2.3. Valoración

No existe ningún procedimiento formalizado para desarrollar tablas de decisión. No obstante, en la práctica se utiliza habitualmente el Álgebra de Boole para hacer exhaustiva la combinación de condiciones, así como para simplificar las mismas [Deaño, 1999]. Ello favorece sobremanera la identificación de las acciones a realizar. Se puede afirmar, por lo tanto, **el que procedimiento de uso de las tablas de decisión es parcial.**

Por otra parte, este modelo es de alcance muy restringido. Por esta razón, y además por ser un modelo no específicamente orientado al análisis de sistemas software, **no permite especificar qué tipo de diseño es más adecuado ni permite derivar dicho diseño.**

En lo referente a las ligaduras computacionales, el modelo **no introduce conceptos** que no estén en el dominio del problema. Sin embargo, **su amplitud se restringe a los pares decisión-acción**, no permitiendo expresar ningún otro aspecto del dominio del problema.

2.5 .1.3. ÁRBOLES DE DECISIÓN

Los árboles de decisión son un modelo semánticamente equivalente a las tablas de decisión, y que se ha utilizado exactamente en los mismos contextos, especialmente en el marco de las Aproximaciones Estructuradas [Yourdon, 1989]. La única diferencia entre tablas de decisión y árboles de decisión se produce en la notación usada. Los árboles de decisión utilizan un formalismo gráfico con la misma notación que el flujograma, tal y como se muestra en la figura 2.7, aunque restringida en el número de símbolos que maneja. En concreto, los árboles de decisión emplean, únicamente:

1. Símbolos para expresar la evaluación de condiciones. Estos símbolos acostumbran diagramarse como rombos, siguiendo la norma DIN66001 [DIN, 1983].
2. Símbolos para expresar la realización de acciones. Estos símbolos, al igual que el caso anterior, respetan la norma DIN66001 y se diagraman como rectángulos.

Los problemas anteriormente indicados para las tablas de decisión son igualmente aplicables a los árboles de decisión.

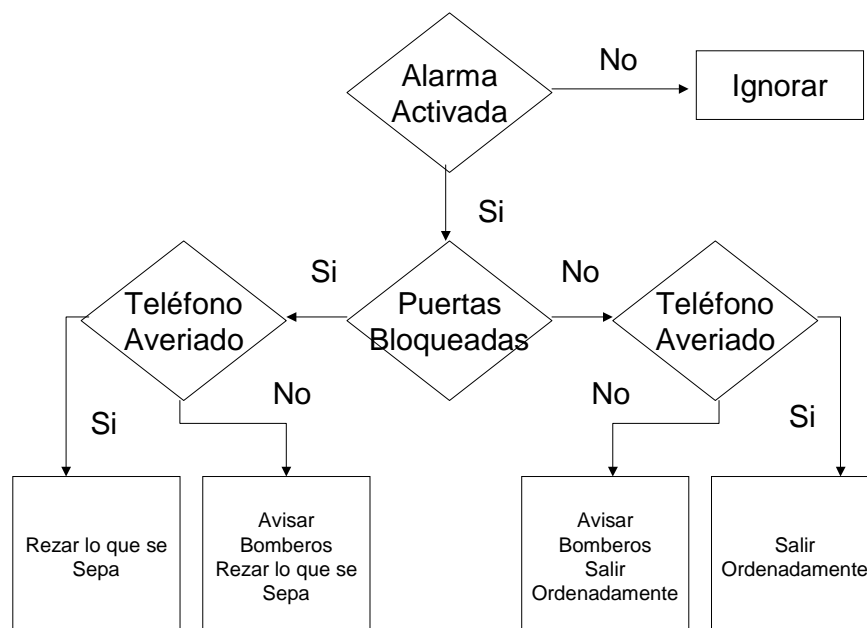


Figura 2.7. Ejemplo de árbol de decisión

2.5 .1.4. DICCIONARIO DE DATOS

2.5 .1.4.1. Descripción

El diccionario de datos es un modelo basado en las gramáticas regulares. Su finalidad es describir formalmente la estructura de los datos de entrada o salida a un proceso. Dicho proceso deberá estar recogido en un Diagrama de Flujo de Datos y, por lo tanto, representar una transformación en el dominio del problema.

Aunque existen variantes de notación, los diccionarios de datos siguen de forma más o menos fiel la notación utilizada habitualmente en la teoría de gramáticas regulares [Sudkamp, 1996]. Por ejemplo, Yourdon [Yourdon, 1989] propone las siguientes notaciones para la construcción de diccionarios de datos:

=	Igualdad
+	Concatenación
()	Opcionalidad
{ }	Iteración
[]	Denota la posibilidad de escoger entre varias alternativas
	Separa alternativas en []
@	Denota los <i>items</i> identificadores únicos o claves
* *	Comentario

Un ejemplo de diccionario de datos, realizado según la notación que propone Yourdon, se muestra en la figura 2.8.

Cliente = N° Cliente + Nombre Cliente + Dirección Cliente

Dirección Cliente = Calle + (Número +) Piso + Puerta + Código Postal

Pedido Cliente = [N° Cliente | Nombre Cliente] + {Productos}

Productos = N° Producto + Cantidad + Precio

Facturas = N° Cliente + { Producto + Precio } + Total Precio

Figura 2.8. Ejemplo de diccionario de datos

2.5 .1.4.2. Métodos de Análisis

Los diccionarios de datos se utilizan principalmente en el marco de la Aproximación Estructurada, con el objetivo de precisar los datos de entrada/salida de los procesos del Diagrama de Flujo de Datos

[DeMarco, 1979] [Gane et al., 1979] [Palmer et al., 1984] [Orr, 1977] [Orr, 1981] [Yourdon, 1989]. Por lo tanto, el diccionario de datos es un medio para precisar qué *ítems* de datos pueden, o no, usarse en la confección de una miniespecificación [Yourdon, 1989]. También se utiliza en los DFD de Hatley [Hatley, 1984] [Hatley et al., 1987], y en el DFD/RT [Ward et al., 1985] [Ward, 1986], pero únicamente para describir los flujos de datos, ya sean discretos o continuos, y nunca los flujos de control.

2.5 .1.4.3. Valoración

Los diccionarios de datos sufren de los mismos problemas que las miniespecificaciones. **No existe un proceso que guíe al analista para su desarrollo, no prescriben un determinado tipo de diseño ni pueden derivarse a dicho diseño.**

Desde el punto de vista de la amplitud y ligaduras computacionales, los diccionarios de datos se utilizan como apoyo de los DFD o DFD/RT y, por lo tanto, **su dominio del problema es el propio Diagrama de Flujo de Datos. Las capacidades expresivas del diccionario de datos están limitadas** ya que los conceptos que describe no son los existentes en el mundo real, sino los representados en dicho Diagrama.

Finalmente, y aunque a priori el diccionario de datos no posee ligaduras computacionales, su uso, ligado a la miniespecificación, **produce en su utilización práctica un sesgo hacia los aspectos de implementación** de los procesos del Diagrama de Flujo de Datos.

2.5 .1.5. FLUJOGRAMA

2.5 .1.5.1. Descripción

El flujograma es un modelo, al igual que las tablas de decisión, no específico del campo de la Informática. Su finalidad es describir los pasos concretos que se deben realizar para llevar a cabo algún tipo de procedimiento, tarea o proceso de transformación. Dichos pasos concretos se describen mediante la utilización de un conjunto de símbolos gráficos (rectángulos, rombos, etc.) que permiten representar operaciones concretas (transformaciones, decisiones, etc.).

Existe una gran cantidad de símbolos distintos. A continuación se indican los más frecuentemente utilizados en la práctica:

1. Inicio/fin del flujograma. Se utilizan para indicar dónde comienza y finaliza un procedimiento.
2. Conectores. Se utiliza para relacionar partes de un flujograma cuando éste tiene que confeccionarse utilizando distintas páginas debido a su tamaño.
3. Decisión. Se utiliza para indicar la existencia de una condición que rompe el flujo de proceso en dos caminos lógicos.
4. Proceso. Se utiliza para indicar la realización de una determinada tarea o acción.
5. Documento. Se utiliza para indicar la necesidad de generar un documento impreso.
6. Almacenamiento secuencial. Se utiliza para indicar el acceso a un medio de almacenamiento secuencial, tal como las cintas magnéticas.

7. Almacenamiento directo. Se utiliza para indicar el acceso a un medio de almacenamiento directo, tal como los discos magnéticos.
8. Presentación por pantalla. Se utiliza para indicar la salida de información en una pantalla.

Existe muchos más símbolos convencionales. La amplia utilización de los flujogramas en el pasado llevó a su estandarización bajo la norma DIN66001 [DIN, 1983], en la que se definen la totalidad de los mismos. La diagramación de cada uno de los símbolos indicados anteriormente se muestra en la figura 2.9.

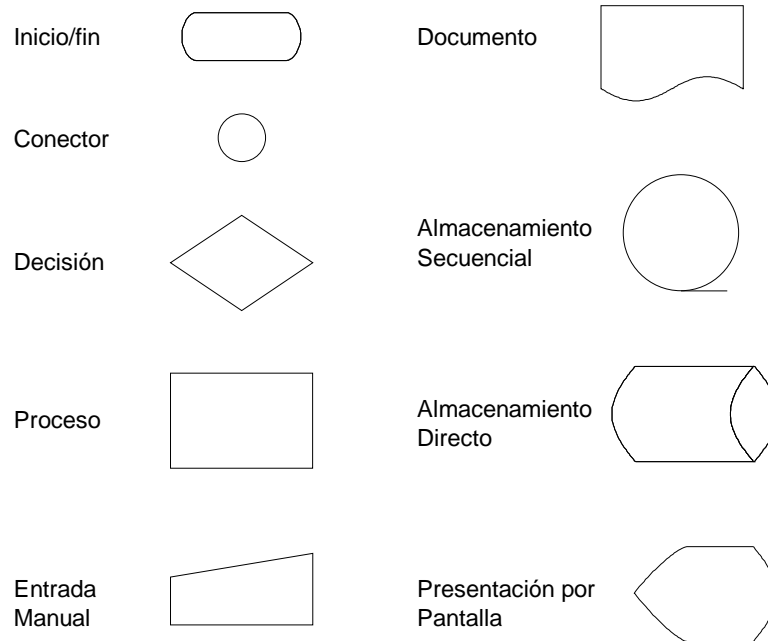


Figura 2.9. Símbolos convencionales utilizados en los flujogramas

Los flujogramas se utilizaron inicialmente en el campo de la informática para describir los flujos de trabajo en los que intervenían computadoras. Posteriormente, se utilizaron para describir la estructura interna de los programas que operaban en dichas computadoras. Actualmente, se utilizan, al igual que las tablas de decisión y árboles de decisión, como un mecanismo alternativo de escritura de miniespecificaciones. Un ejemplo de flujograma se muestra en la figura 2.10.

2.5.1.5.2. Métodos de Análisis

Los flujogramas se utilizan exclusivamente en el marco de la Aproximación Estructurada, como método alternativo de escritura de miniespecificaciones [Yourdon, 1989].

2.5.1.5.3. Valoración

Los flujogramas poseen problemas similares a las miniespecificaciones. Por una parte, **no existe ningún procedimiento para su desarrollo**. Sin embargo, el enorme detalle que proveen a la hora de

describir las operaciones o transformaciones prescriben, directamente, **la utilización de tipos de diseño con características algorítmicas**, preferentemente la utilización directa de algún tipo de lenguaje de programación. La obtención de dicho tipo de diseño es automático, ya que **los distintos símbolos del flujograma se corresponden, casi unívocamente, a los constructores de un lenguaje de programación** (no en vano, se utilizaron en su día para describir programas de computadora).

Desde el punto de vista de la amplitud y ligaduras computacionales, **las capacidades expresivas del flujograma están limitadas**, ya que únicamente pueden expresar los pasos concretos de los que se compone un proceso o función. **Las ligaduras computacionales del flujograma son bastante claras**, ya que aunque *a priori* un flujograma no tiene por qué introducir ligaduras computacionales en sus representaciones, en la práctica cualquier descripción se realiza utilizando los conceptos de un lenguaje de programación subyacente.

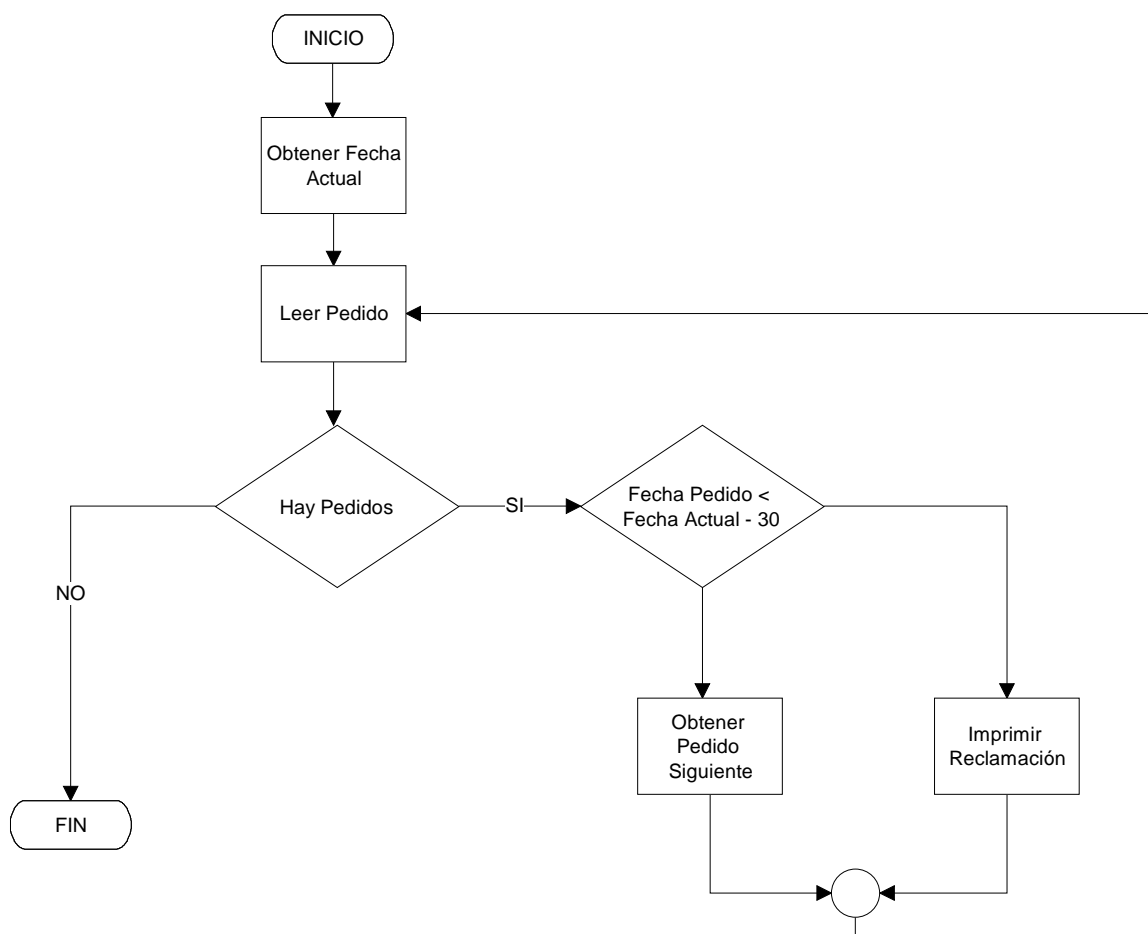


Figura 2.10. Ejemplo de flujograma

2.5.1.6. LENGUAJE DE DISEÑO DE PROGRAMAS

2.5.1.6.1. Descripción

El lenguaje de diseño de programas (LDP) es un modelo que utiliza los constructores básicos de cualquier lenguaje de programación estructurado para describir operaciones o transformaciones. Típicamente, se utilizan los siguientes constructores:

1. Constructores de selección, tales como IF-THEN o CASE
2. Constructores de repetición, tales como WHILE o REPEAT
3. Llamadas a procedimientos y funciones, utilizando palabras clave tales como CALL
4. Asignación de valores a variables, utilizando el símbolo "=", "!=" o incluso clausulas de cobol como ASSIGN.

Adicionalmente, también utiliza expresiones en lenguaje natural para definir las condiciones a evaluar y acciones a realizar, pero que no se desea describir de forma exhaustiva. La figura 2.11 muestra un ejemplo de LDP.

```

IF LEN(pin) <> 4
  THEN BEGIN
    DISPLAY ERROR
    EXIT
  END
ELSE BEGIN
  DISPLAY Menú_1
  ACCEPT Opción
  CASE Opción
    "1": CALL Retirada
    "2": CALL Movimientos
    "3": CALL Ingresos
    "4": EXIT
  END CASE
END
END IF

```

Figura 2.11. Ejemplo de lenguaje de diseño de programas

2.5.1.6.2. Métodos de Análisis

La implementación de LDP más usada en la actualidad es la de [Caine et al., 1975]. Aunque el LDP es un modelo utilizado durante de análisis, principalmente como mecanismo alternativo de escritura de miniespecificaciones [Yourdon, 1989], también se utiliza extensivamente como método de diseño de bajo nivel en muchos métodos, como por ejemplo el Diseño Estructurado [Yourdon et al., 1986] o metodologías, como por ejemplo SSADM [Goodland et al., 1990]. En parte, esta utilización como lenguaje de diseño se debe a la facilidad con que se puede pasar desde el Análisis hasta el Diseño utilizando LDP [Davis, 1988] [Davis, 1993].

2.5.1.6.3. Valoración

El LDP **no tiene un procedimiento asociado para realizar el análisis pero, al igual que el flujograma, determina completamente el tipo de diseño a utilizar.** De hecho, el LDP puede entenderse

directamente como un diseño de bajo nivel de tipo algorítmico. De esta forma, **la derivación del diseño es automática.**

Desde el punto de vista de la amplitud y ligaduras computacionales, **las capacidades expresivas del LDP están restringidas** a describir los pasos concretos de los que se compone un proceso o función. **Las ligaduras computacionales del LDP, al igual que las del flujograma, son bastante claras**, ya que cualquier descripción se realiza utilizando los conceptos de un lenguaje de programación subyacente, ya sea en la versión de [Caine et al., 1975] o en cualquier otra versión distinta.

2.5.1.7. RESUMEN

La valoración de los criterios para Modelos Orientados al Procedimiento se resume en la tabla 2.2.

Criterios	Mini-especificación	Tabla de Decisión	Árbol de Decisión	Diccionario de Datos	Flujograma	LDP
<i>Amplitud</i>	Ninguna	Parcial	Parcial	Ninguna	Parcial	Parcial
<i>Ligaduras Computacionales</i>	Sí	No	No	Sí	Sí	Sí
<i>Procedimiento de Uso</i>	Ninguno	Parcial	Parcial	Ninguno	Ninguno	Ninguno
<i>Selección de Diseño</i>	No Aconseja	No Aconseja	No Aconseja	No Aconseja	Prescribe	Prescribe
<i>Derivación de Diseño</i>	Ninguno	Ninguno	Ninguno	Ninguno	Total	Total

Tabla 2.2. Valores de los criterios para los modelos orientados al procedimiento

2.5.2. MODELOS ORIENTADOS A LA TRANSFORMACIÓN

Este conjunto de modelos están orientados a describir las transformaciones que sufren los datos usados en el dominio del problema. Las transformaciones se deben a la existencia de algún tipo de proceso, que habitualmente toma como entrada y genera como salida elementos lógicos (tales como datos), aunque es posible que tanto entrada como salida se refiera a elementos físicos (cosas).

El modelo más conocido de este grupo es el **Diagrama de Flujo de Datos**. También son importantes otros modelos, como el **Diagrama de Flujo de Control**, el **Diagrama de Flujo de Datos / Tiempo Real**, la **Técnica de Análisis y Diseño Estructurado** y **PSL/PSA**.

2.5.2.1. DIAGRAMA DE FLUJO DE DATOS

2.5.2.1.1. Descripción

Un diagrama de flujo de datos (DFD) es un modelo cuya finalidad es representar las transformaciones que ocurren en el dominio del problema. Este modelo, propuesto inicialmente por [DeMarco, 1979], se compone de cuatro elementos constituyentes:

1. Procesos. Representan cualquier acción que se desarrolle en el mundo real. Los procesos toman habitualmente datos como entrada y generan datos como salida, pero no existe ninguna razón para que un proceso reciba y genere exclusivamente datos. En principio, es posible especificar las transformaciones realizadas sobre elementos físicos. No obstante, la práctica habitual, teniendo en cuenta que el propósito del Análisis es construir, finalmente, un producto software, lleva a que habitualmente se consideren únicamente las transformaciones realizadas sobre los datos.
2. Flujos de Datos. Representan la entrada o salida de datos de un proceso.
3. Entidades Externas. Representan elementos que generan o consumen datos en el dominio de discurso. Se diferencian de los procesos en que la transformación que realizan está fuera del ámbito del Análisis.
4. Almacenes de Datos. Son reservorios temporales de los flujos de datos en su movimiento entre los distintos procesos.

Los cuatro elementos citados se representan mediante un conjunto de símbolos gráficos. Estos símbolos varían de método a método. Los símbolos originales propuestos por [DeMarco, 1979] se muestran en la figura 2.12.

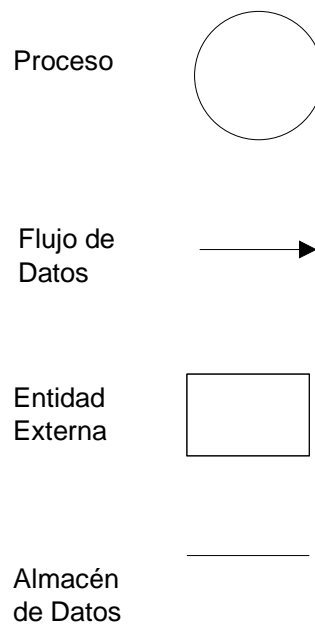


Figura 2.12. Símbolos gráficos de un DFD

Una de las particularidades del DFD es que los procesos son descomponibles, siguiendo el criterio de refinamiento sucesivo [Wirth, 1971]. Así, un DFD es sólo un diagrama en una secuencia de diagramas, desde un mayor nivel de abstracción a uno menor. El proceso de descomposición termina cuando los procesos de menor nivel son lo suficientemente sencillos como para no necesitar una descomposición ulterior. La figura 2.13 muestra un ejemplo de DFD.

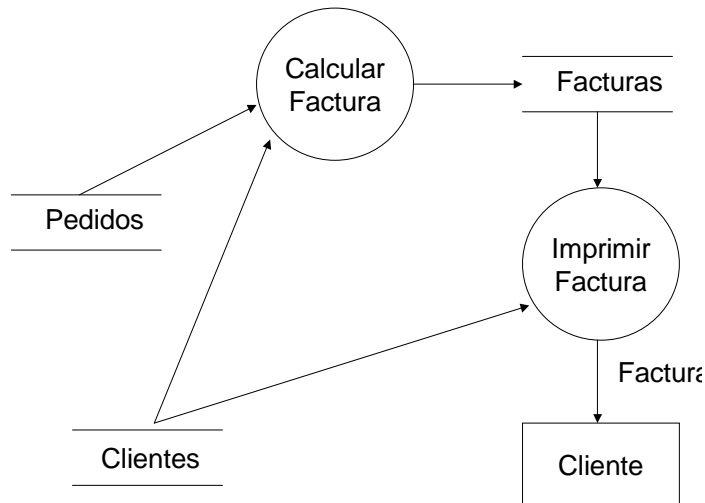


Figura 2.13. Ejemplo de diagrama de flujo de datos

2.5.2.1.2. Métodos de Análisis

El DFD se ha utilizado, sobre todo, en los métodos de Análisis Estructurado, que incluyen los trabajos de [DeMarco, 1979] [Palmer et al., 1984] y [Yourdon, 1989]. También ha sido utilizado en otros métodos como *Structured System Analysis* [Gane et al., 1979], *Structured Systems Development* [Orr, 1977] y *Structured Requirements Definition* [Orr, 1981]. Asimismo, ha sido utilizado en múltiples metodologías de desarrollo, como SSDAM [Goodland et al., 1990] o METRICA [CSI, 2000]. También es utilizado en métodos de desarrollo de tiempo real [Hatley, 1984] [Hatley et al., 1987].

En todos estos métodos, el modelo en sí se ha mantenido inalterado, aunque existen acusadas diferencias en términos de diagramación. En algunos casos, como en SSADM, se introduce información adicional en los diagramas, por ejemplo, la localización física de los procesos o el agente que efectúa dichos procesos [Goodland et al., 1990].

2.5.2.1.3. Valoración

La totalidad de los métodos que emplean DFD **establecen guías procedimentales** para definir el modelo. Por ejemplo, [DeMarco, 1979] propone modelar inicialmente el sistema físico para después redefinir el sistema teniendo en cuenta únicamente los procesos realizables mediante computadora. [Orr, 1981] propone un enfoque *bottom-up*, de identificación de procesos atómicos, para posteriormente crear la jerarquía de DFD. Una idea similar aparece en [Yourdon, 1989], donde se propone construir el DFD a partir de los acontecimientos (un concepto en cierta medida similar a un caso de uso) a los que debe responder el futuro sistema software.

Adicionalmente, el DFD se utiliza siempre habitualmente en conjunción con una técnica denominada Diseño Estructurado [Yourdon et al., 1986]. Existen, no obstante, metodologías como SSADM [Goodland et al., 1990] que, en lugar de utilizar la técnica de Diseño Estructurado, utilizan la Historia de la Vida de las

Entidades [Jackson, 1983]. Lo que ocurre en este caso es que la Historia de la Vida de las Entidades es el modelo dominante de SSADM.

La técnica de Diseño Estructurado permite trasladar los conceptos del DFD en una estructura jerárquica de módulos que representa un diseño arquitectónico. Por lo tanto, se puede derivar directamente un diseño desde un DFD, pero con la restricción de que dicho diseño se realice siempre mediante una descomposición modular jerárquica que transforme los datos mediante un movimiento de los mismos a través de la jerarquía de módulos.

La utilización de la técnica de Diseño Estructurado no permite, por ejemplo, evaluar si es necesario el empaquetamiento de datos y procesos conjuntamente, o si se puede realizar una descomposición distinta basada, por ejemplo, en la frecuencia de utilización de los procesos.

Por lo tanto, la utilización de un DFD prescribe siempre la utilización de un tipo de diseño. Adicionalmente, existe la posibilidad de derivar sistemáticamente dicho diseño.

Desde el punto de vista de la amplitud y ligaduras computacionales, las capacidades expresivas **están limitadas a los procesos de transformación**, no pudiendo expresar en el DFD aspectos del dominio del problema como reglas de negocio, o la organización y relaciones de los objetos del dominio. Dichos aspectos tienen que indicarse en otro tipo de formalismo, habitualmente utilizando documentos en lenguaje natural. A priori, el DFD no posee ligaduras computacionales, pero en la práctica un proceso del DFD siempre se corresponderá con un proceso en la computadora, y un almacén de datos con un fichero, debido a que **la proximidad entre los conceptos del DFD y de la computadora introducen inevitablemente un sesgo a favor de ésta última.**

2.5.2.2. DIAGRAMA DE FLUJO DE CONTROL

2.5.2.2.1. Descripción

El Diagrama de Flujo de Control (DFC) es una técnica basada en el DFD. En el DFC, se conservan únicamente los procesos del DFD, y se introduce un nuevo tipo de elemento, denominado Flujo de Control, habitualmente representado mediante una línea punteada. Un flujo de control conecta procesos en el DFC, pero su significado no es el de introducir o extraer datos del proceso, sino el de modificar su funcionamiento en función del estado actual del sistema. Un ejemplo de DFC se muestra en la figura 2.14.

2.5.2.2.2. Métodos de Análisis

Los DFC se han utilizado en el método de [Hatley, 1984] [Hatley et al., 1987] de análisis de aplicaciones de tiempo real. El DFC se utiliza conjuntamente con el DFD, y necesita adicionalmente de otras dos técnicas para representar el funcionamiento de la aplicación: Una Máquina de Estado Finito, equivalente a un Diagrama de Transición de Estados, y una Tabla de Activación de Procesos, equivalente a una Tabla de Decisión.

La Máquina de Estado Finito sirve para determinar el estado actual del sistema en función de las entradas que se han realizado. De esta forma, es posible representar el tiempo como una secuencia de estados discretos.

La Tabla de Activación de Procesos sirve para determinar, en función del estado actual del sistema, el efecto de los flujos de control sobre los procesos en los que incide.

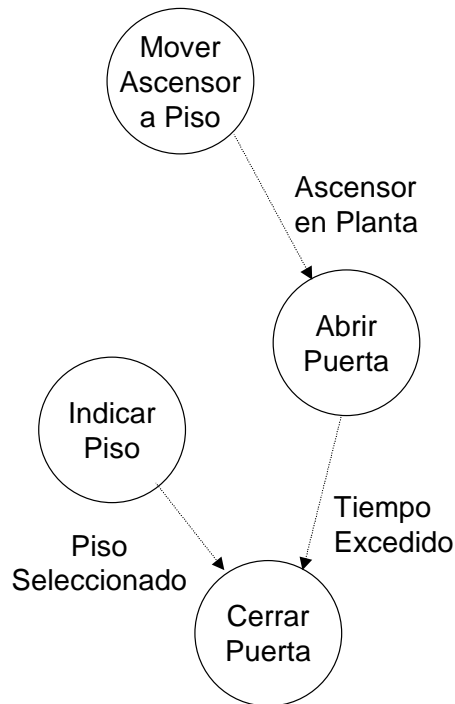


Figura 2.14. Ejemplo de diagrama de flujo de control

2.5.2.2.3. Valoración

Un DFC se construye después de haber desarrollado los DFD del sistema, por lo que **su procedimiento de construcción se sigue del procedimiento de construcción del DFD**, con un paso adicional que introduce los flujos de control y elabora la Máquina de Estado Finito y la Tabla de Activación de Estados.

Sin embargo, **no existe ningún procedimiento para transformar un DFC a un diseño de forma sistemática**. La complejidad que supone la introducción de estados es suficiente para explicar este hecho. Asimismo, **los DFC no prescriben un determinado diseño**, sino que existen varias formas de llegar a implementarlo.

Desde el punto de vista de la amplitud y ligaduras computacionales, es evidente que el flujo de control **representa un concepto no existente en el mundo real**. Un flujo de control permite representar las restricciones de funcionamiento de una máquina, pero no plasma las restricciones del dominio del problema. El concepto de estado, asimismo, es una adaptación al campo de la informática del concepto de tiempo, que es muy difícil de representar con sistemas discretos como las computadoras. La expresividad del DFC es, asimismo, muy limitada, **ciñéndose exclusivamente a representar relaciones de control**.

2.5.2.3. DIAGRAMA DE FLUJO DE DATOS PARA TIEMPO REAL

2.5.2.3.1. Descripción

El Diagrama de Flujo de Datos para Tiempo Real (DFD/TR) están también basado en el DFD pero, a diferencia del DFC, no modifica su formalismo sino que lo complementa con nuevos elementos. Estos elementos son:

1. Flujos de control. Distingue tres tipos de flujos: Señal, activación y desactivación. Al igual que en los DFC, estos flujos se denotan mediante líneas punteadas.
2. Flujos de datos continuos. Tienen el mismo funcionamiento y notación que los flujos de datos en un DFD, pero su naturaleza no es discreta. Representan, por ejemplo, una lectura continua realizada por un sensor, o datos de naturaleza continua enviados a un servomecanismo.
3. *Buffers*. Son almacenamientos similares a los almacenes de datos, pero su utilidad es representar almacenamiento temporal por cuestiones de tiempo de proceso. Esto es, representan los retardos que sufren los datos antes de ser atendidos por un proceso. Se denotan igual que un almacén de datos, pero con líneas punteadas.
4. Transformaciones de control. Representan transformaciones que, al igual que los datos, pueden sufrir los flujos de control. Este tipo de transformaciones permiten describir transformaciones complejas que, utilizando únicamente máquinas de transición de estados y tablas de activación de procesos, son difíciles de especificar. Se denotan igual que un proceso en un DFD, pero con líneas punteadas.

Un ejemplo de DFD/TR se muestra en la figura 15.

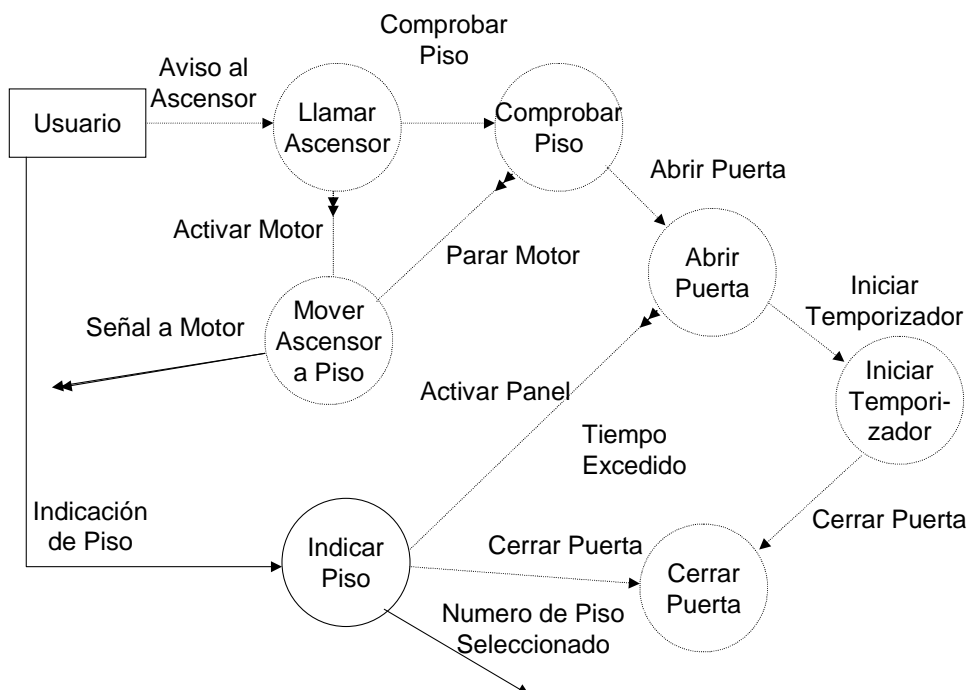


Figura 2.15. Ejemplo de diagrama de flujo de datos para tiempo real

2.5.2.4. TÉCNICA DE ANÁLISIS Y DISEÑO ESTRUCTURADO

2.5.2.4.1. Descripción

La Técnica de Análisis y Diseño Estructurado (TADE) es similar a los DFD. Su finalidad es representar procesos de transformación, considerados como una secuencia entrada-proceso-salida, siguiendo un esquema *top-down*. A diferencia del DFD, la TADE no posee almacenes de datos ni entidades externas, pero sí flujos de datos (representados mediante flechas) y procesos (representados mediante rectángulos). Adicionalmente, dispone de otros dos componentes:

1. Flujos de control. Representan restricciones al funcionamiento del proceso, pero no en el sentido de los DFC o DFD/TR. Los flujos de control especifican únicamente cómo tiene que realizarse la transformación. Se representan como un flujo incidente en la zona superior del rectángulo que denota el proceso.
2. Flujos de mecanismo. Definen con qué debe realizarse dicho proceso. Se representan como un flujo incidente en la zona inferior del rectángulo que denota el proceso.

Un ejemplo de TADE se muestra en la figura 2.16.

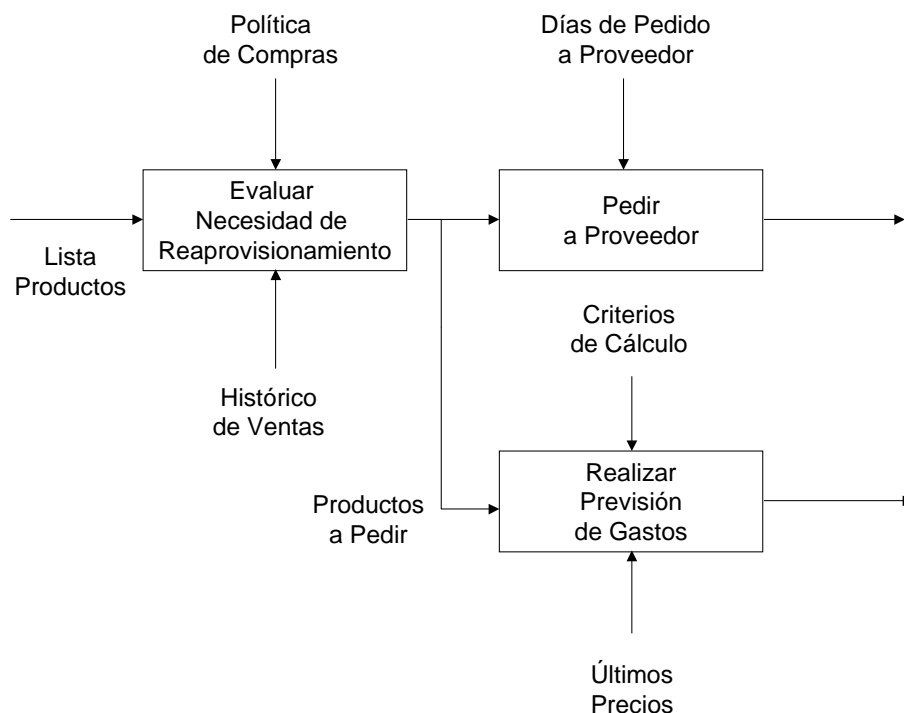


Figura 2.16. Ejemplo de la técnica de análisis y diseño estructurado

Al igual que el DFD, la TADE utiliza el concepto de refinamiento sucesivo. Por lo tanto, un modelo del dominio del problema realizado mediante TADE consta de un conjunto jerárquico de diagramas, donde los niveles inferiores representan un refinamiento de los niveles superiores. El proceso, al igual que en el DFD, termina cuando los procesos son lo suficientemente sencillos para poder ser comprendidos sin necesidad de posterior refinamiento.

2.5.2.4.2. Métodos de Análisis

La TADE se utiliza en el método de análisis de igual nombre (en inglés, SADT) [Ross, 1977] [Marca et al, 1988]. Este método de análisis no prescribe un procedimiento similar al propuesto por [DeMarco, 1979] para el desarrollo de la secuencia de modelos, sino que plantea la descripción del problema en una única fase cuyo fin último es la creación de la jerarquía de diagramas.

2.5.2.4.3. Valoración

No se ha definido **una secuencia estricta de pasos para la confección de un TADE**. El TADE es, únicamente, un formalismo de representación. Dada la naturalidad que se supone posee la visión del mundo como un conjunto de procesos, se deja a criterio del analista su realización. Por otra parte, **tampoco presupone un determinado diseño, ni posee un procedimiento bien definido para generarlo**.

En lo que respecta a la amplitud, la TADE **no puede expresar, al igual que en el caso del DFD, aspectos que se refieran a la globalidad del modelo**, como restricciones de tipo global (las habitualmente denominadas reglas de negocio). Sin embargo, este tipo de restricciones pueden simularse fácilmente usando los flujos de “control” y “mecanismo”, aun a costa de ser repetitivos, al contrario que en el DFD, donde su introducción es prácticamente imposible. **Tampoco puede expresar aspectos referidos a los objetos del dominio del problema**, o a la organización de los mismos.

Por último, la TADE **no introduce ningún elemento que se pueda considerar ajeno al dominio del problema**, ya que el concepto de flujo de “control” y “mecanismo” corresponden con elementos que, efectivamente, existen en el mundo real.

2.5.2.5. PROBLEM STATEMENT LANGUAGE/PROGRAM STATEMENT ANALIZER (PSL/PSA)

2.5.2.5.1. Descripción

PSL/PSA es un modelo que, a diferencia de los anteriores (los cuales son de naturaleza gráfica), utiliza sentencias de un lenguaje especial para recoger los procesos de transformación que ocurren en el dominio del problema. Se ha clasificado en este grupo debido a que la ontología subyacente es similar a las del DFD, DFC, DFD/RT o TADE; es decir, está basado en una visión del dominio como un conjunto de procesos que actúan sobre los datos realizando transformaciones en los mismos.

PSL/PSA es, en realidad, un conjunto de dos artefactos. PSL es un lenguaje de modelado, mientras que PSA es una herramienta automática de análisis de las sentencias escritas en PSL. PSL ha sufrido múltiples modificaciones a lo largo del tiempo, actualizando su capacidad de expresión. Sin embargo, los componentes más importantes del modelo están formados por las siguientes primitivas:

1. PROCESS. Define un proceso, al que se asigna un nombre.
2. GENERATES. Define los datos de salida del proceso.
3. RECEIVES. Define los datos de entrada al proceso.

4. SUBPARTS ARE. Define los procesos subordinados, es decir, los procesos en los que se descompone un proceso dado. Este descriptor indica que, por lo tanto, PSL sigue un criterio de refinamiento sucesivo a la hora de identificar los distintos procesos.
5. PART OF. Define de qué proceso es un refinamiento un proceso dado.

Adicionalmente, PSL introduce de forma explícita la noción de evento, que permite iniciar o terminar un proceso. La noción de evento es muy útil para describir la dinámica del dominio del problema, ligada a los procesos de transformación que ocurren en el mismo. PSL describe los eventos mediante las primitivas TRIGGERED BY y TERMINATION CAUSES. La figura 2.17 muestra un ejemplo de PSL/PSA.

PROCESS Calcular-Factura
DESCRIPTION Este proceso calcula el total a pagar por cada cliente en función de los pedidos realizados y genera una factura
GENERATES Factura
RECEIVES Clientes, Pedidos
PART OF: Gestionar-Pedidos
SUBPARTS ARE: Calcular-Total, Generar-Factura
DERIVES: Facturas
USING: Clientes, Pedidos

Figura 2.17. Ejemplo de PSL

2.5.2.5.2. Métodos de Análisis

PSL/PSA fue desarrollado por [Teichroew et al., 1971] [Teichroew et al., 1977] en el marco del proyecto ISDOS de la Universidad de Michigan. Debido a su antigüedad, y al hecho de que PSA es una herramienta automatizada, su naturaleza es textual, aunque no hay, en principio, ningún problema para desarrollar una notación gráfica para PSL similar, por ejemplo, a la del DFD en el ámbito del Análisis Estructurado. PSL/PSA ha sido usado en múltiples proyectos, y las dificultades encontradas durante los mismos ha propiciado su evolución desde su formulación inicial hasta su formulación actual [Sayani, 1990].

2.5.2.5.3. Valoración

No se ha definido ninguna secuencia de pasos para confeccionar un modelo PSL. PSL proporciona únicamente un formalismo de descripción, y su herramienta asociada, PSA, un mecanismo automatizado para realizar comprobaciones de sintaxis y extraer informes.

La ontología subyacente a PSL es totalmente similar a la del DFD, por lo que es muy sencillo derivar un diseño basado en una jerarquía modular paralela a los procesos definidos durante el análisis. Por lo tanto, se puede afirmar que PSL **prescribe un determinado formalismo de diseño. Sin embargo, PSL no posee, per se, un método para generar dicho diseño.**

PSL no introduce, como en los DFC o DFD/TR, conceptos no presentes en el dominio de discurso. Sin embargo **adolece, al igual que el DFD, de la capacidad para representar conceptos globales**, como aspectos temporales o reglas de negocio, así como los objetos y su organización en el dominio del problema.

2.5.2.6. RESUMEN

La valoración de los criterios para las técnicas orientadas al procedimiento se resume en la tabla 2.3.

Criterios	DFD	DFC	DFD/TR	TADE	PSL/PSA
<i>Amplitud</i>	Parcial	Parcial	Parcial	Parcial	Parcial
<i>Ligaduras Computacionales</i>	Sí	Sí	Sí	No	No
<i>Procedimiento de Uso</i>	Total	Total	Total	Ninguno	Parcial
<i>Selección de Diseño</i>	Prescribe	Ninguno	Ninguno	Ninguno	Prescribe
<i>Derivación de Diseño</i>	Total	Ninguno	Ninguno	Ninguno	Ninguno

Tabla 2.3. Valores de los criterios para los modelos orientados a la transformación

2.5.3. MODELOS ORIENTADOS A LOS OBJETOS/DATOS

Este grupo de modelos tiene como finalidad describir qué objetos existen en el dominio del problema, así como recoger las relaciones que se establecen ente dichos objetos. Muchos estudiosos únicamente reconocerán este grupo de modelos como los “verdaderos” modelos conceptuales. No obstante, en el presente trabajo de Tesis, este grupo de modelos se considerará en pie de igualdad con los restantes.

A diferencia de los restantes grupos, éste es el más numeroso y heterogéneo, debido al enorme numero de modelos existentes [Avison et al., 1995]. La heterogeneidad se manifiesta en el hecho de que, aunque dos modelos pertenecientes a este grupo describan exactamente los mismos aspectos del mundo real, pueden hacerlo atendiendo a diversos enfoques teóricos, a menudo dicotómicos. Ello se refleja, asimismo, en la clasificación establecida, ya que para abarcar a la totalidad de los modelos de este grupo ha sido necesario crear tres subgrupos distintos: Modelos Básicos, Avanzados y Orientados a Objetos.

Aun a costa de alargar en exceso la introducción, es necesario proporcionar una justificación adicional de la clasificación establecida. Para ello, es preciso discutir los enfoques teóricos que generan la clasificación establecida. Dichos enfoques teóricos se basan en las siguientes dicotomías:

1. Estructura vs. Funcionalidad: Los modelos orientados a los objetos/datos pueden describir:
 - a. Únicamente las entidades existentes en el dominio del problema, así como las relaciones que se establecen entre las mismas; esto es, describen la estructura de los datos.

- b. Adicionalmente a la estructura los modelos pueden describir aspectos de la utilización de los datos en el dominio del problema. Por ejemplo, una vez descritas las entidades “Alumno” y “Matrícula”, se podría describir la operación “Matriculación”.
2. Entidades vs Objetos: Los modelos orientados a los objetos/datos pueden describir:
- a. Entidades, esto es, elementos estáticos del dominio del problema.
 - b. Objetos, esto es, elementos dinámicos del dominio del problema.

Las dicotomías anteriores justifican la clasificación de modelos establecida, la cual se muestra en la tabla 2.4. Debido a que no tienen sentido aquellos modelos que describan objetos (elementos reactivos, por tanto), pero que no describan su comportamiento, dicha combinación se ha indicado como N/A (No Aplicable).

DICOTOMÍAS	Entidades	Objetos
Estructura	Modelos Básicos	N/A
Funcionalidad	Modelos Avanzados	Modelos Orientados a Objetos

Tabla 2.4. Origen de los subgrupos de los modelos orientados a los objetos/datos

Nótese que la existencia de dicotomías no implica que cada modelo deba estar clasificado, exactamente, acorde a cada una de ellas. Los diferentes enfoques teóricos surgen del análisis retrospectivo de los modelos que implementan dichos enfoques en la práctica. Adicionalmente, los distintos modelos no se desarrollan de forma aislada, sino que recogen conceptos existentes en modelos propuestos con anterioridad, refinados y ampliados en función de los objetivos de cada método particular. La figura 2.18, debida a [Loucopoulos et al., 1995] muestra la fecha de aparición de varios conceptos importantes para los modelos pertenecientes a este grupo. Es indudable que, tras la aparición de un determinado concepto de modelización, distintos modelos, de distinta orientación, incluirán dicho concepto en sus representaciones.

Por todo ello, es necesario indicar que existen modelos que poseen características pertenecientes a dos enfoques teóricos enfrentados. No obstante, la clasificación realizada recoge los modelos paradigmáticos, que son el **Modelo Entidad-Relación**, el **Modelo Funcional de Datos**, el **Modelo Entidad-Relación Extendido**, los **Modelos Avanzados Operativos y Declarativos**, el **Diagrama de Clases** y la **Historia de la Vida de las Entidades**.

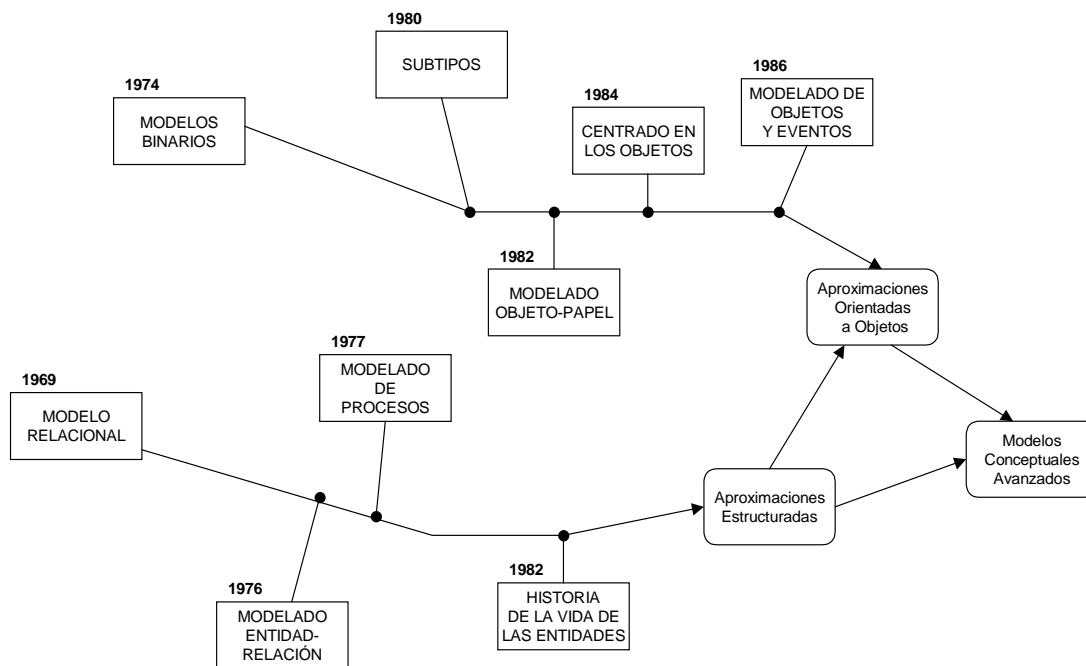


Figura 2.18. Evolución e interacción entre distintos tipos de modelos [Loucopoulos et al., 1995]

2.5.3.1. MODELOS BÁSICOS

Este subgrupo de modelos es también conocido como Modelos Semánticos. Su finalidad es describir las entidades existentes en el dominio del problema, así como las relaciones que se establecen entre las mismas.

Se pueden distinguir tres grandes tipos de Modelos Básicos. El primero, denominado Entidad-Relación (*Entity-Relationship*), usa un constructor explícito, la “relación”, para realizar asociaciones entre las entidades del modelo. Este tipo de modelos derivan del trabajo de [Bachman, 1969], y tienen como paradigma el modelo Entidad-Relación [Chen, 1976].

El segundo tipo, denominado Objeto-Papel (*Object-role*), elimina la diferencia entre relaciones y atributos y utiliza estos últimos para crear asociaciones entre las distintas entidades del modelo. Este tipo de modelos se deriva del trabajo de [Abrial, 1974], y un ejemplo paradigmático es el Modelo Funcional de Datos [Kerschberg et al., 1976].

El tercer tipo son los modelos conocidos como Extendidos. Este tipo de modelos utilizan el concepto de especialización para definir jerarquías de entidades que poseen características comunes. El modelo paradigmático es el modelo Entidad-Relación Extendido [Batini et al., 1986] [Teorey et al., 1986].

A continuación se describirán los modelos paradigmáticos indicados. Cuando sea necesario, en la descripción de cada uno de los modelos, se añadirá un epígrafe denominado “Otras consideraciones”. Dicho epígrafe se utilizará para describir, cuando sea aplicable: (1) los cambios ocurridos en los modelos paradigmáticos debido a la introducción de nuevos conceptos, (2) la influencia de los modelos

paradigmáticos en otros modelos y (3) los modelos que se desvían significativamente del modelo paradigmático.

2.5.3.1.1. Modelo Entidad-Relación

2.5.3.1.1.1. Descripción

El modelo Entidad-Relación (ER) fue propuesto inicialmente por [Chen, 1976]. Este modelo tiene como finalidad describir los tipos de entidad que existen en el dominio del problema, así como las relaciones que se establecen entre los mismos. Un tipo de entidad es la abstracción de las características comunes de un conjunto de objetos o elementos individuales, los cuales poseen interés para el desarrollo del futuro producto software [Chen, 1976]. Dichas características comunes se describen desde el punto de vista estático, esto es, no se tienen en cuenta los aspectos de comportamiento de los distintos elementos individuales, sino únicamente los elementos estructurales que permiten caracterizar un elemento como pertenecientes al tipo de entidad.

Para realizar la descripción de los objetos o elementos de un dominio, el modelo ER utiliza tres constructores básicos: las Entidades, Relaciones y Atributos:

1. Entidades. Representan los objetos o elementos, reales o abstractos, del dominio del problema. Ya que no es posible registrar, debido a la variabilidad inherente del mundo real, todas las posibles distinciones entre objetos existentes en el mismo, el modelo ER solo representa tipos de entidad. Un tipo de entidad es, formalmente, un conjunto formado por un número, variable en el tiempo, de entidades, descritas mediante una serie de características comunes, tal y como se ha comentado anteriormente. Dichas características comunes reciben el nombre de atributos. Aunque existe una gran variabilidad de notaciones, en el trabajo original de [Chen, 1976] las entidades se notan mediante rectángulos.

Las distintas entidades pertenecientes a un tipo deben poder distinguirse. Para esto, es necesario que cada entidad posea un identificador único o clave de la entidad. La clave de entidad permite, asimismo, distinguir dos tipos de entidades: las entidades regulares y las entidades débiles. Una entidad es débil cuando no es posible identificar una clave única, sino que es necesario poner dichas entidades en relación con otras para lograr una identificación unívoca. Las entidades regulares son las que si poseen un identificador único.

2. Relaciones. Son asociaciones entre entidades. Al igual que con los tipos de entidad, formalmente se debe hablar de tipos de relación, siendo una relación concreta una asociación entre dos o varias entidades concretas. No obstante, el término de uso habitual es simplemente el de "relación".

Formalmente, una relación es, en el sentido matemático del término, un subconjunto del producto cartesiano de dos tipos de entidad. Cada entidad desempeña un papel en la relación. Las relaciones se denotan por rombos que conectan a las entidades participantes mediante líneas. El modelo ER admite relaciones recursivas y relaciones múltiples. Existen dos tipos de

relaciones, al igual que en el caso de las entidades: regulares y débiles. Una relación débil es cualquier relación en la que participe una entidad débil, y regular en los restantes casos.

3. Atributos. Son cada una de las características comunes de un conjunto de entidades o un conjunto de relaciones del mismo tipo. Cada atributo toma un valor definido en un conjunto de valores o "dominio" (en el sentido matemático del término). Se representan habitualmente mediante círculos, donde aparece escrito el nombre del conjunto de valores, unidos al tipo de entidad o de relación al que pertenecen mediante flechas⁷ y, opcionalmente, el nombre del atributo próximo a dicha flecha. Asimismo, la habitualidad del diagrama y el hecho de no utilizar los conceptos formales durante su uso en la práctica, lleva a obviar la descripción del conjunto de valores y a colocar en el círculo directamente el nombre del atributo.

Los atributos pueden ser univaluados o multivaluados, y además pueden ser simples o compuestos. En el modelo original de [Chen, 1976], los atributos sólo podían ser univaluados y simples, siendo necesario un tipo de entidad y un tipo de relación, introducidos artificialmente, para representarlos. [Batini et al., 1992] introduce óvalos para diagramar los atributos multivaluados, mientras que los compuestos se describen mediante círculos unidos mediante líneas a otro círculo que da nombre al atributo.

El modelo ER contiene, asimismo, restricciones de integridad. Las restricciones de integridad aseguran que la información que contiene el modelo sea consistente con el dominio del problema. [Chen, 1976] distingue dos tipos de restricciones: Cardinalidad y dependencia.

La restricción de cardinalidad especifica el número máximo de veces que una entidad puede participar en una relación. [Chen, 1976] distingue tres tipos de cardinalidades: 1:1, 1:N y m:n, donde m y n significan "muchos". La restricción de dependencia se da entre una entidad débil y la entidad regular de la que depende. Esta restricción significa que no puede existir una entidad débil si no existe la entidad regular que la identifica.

Un ejemplo de modelo ER se muestra en la figura 2.19.

2.5.3.1.1.2. Métodos de Análisis

El modelo ER se ha usado extensivamente en el desarrollo de sistemas software, tanto de forma aislada, con el propósito de definir esquemas de bases de datos, o como un modelo más en el marco de diversos métodos de desarrollo.

El modelo ER se ha usado en los métodos estructurados [Palmer et al., 1984] [Yourdon, 1989] junto con el DFD para definir la base de datos asociada con el sistema software. Esta integración ha producido indudables ventajas, ya que la riqueza expresiva de los métodos estructurados aumenta al incorporar la visión de los datos. Sin embargo, también se han producido inconvenientes, debido a la necesidad de conciliar la visión de los datos (estática) proporcionada por el modelo ER, con la visión de los procesos

⁷ Este tipo de diagramación está tomado de [Saiedian, 1996]. En el artículo original de Chen no aparecen referencias a la diagramación de los atributos.

(dinámica), proporcionada por el Diagrama de Flujo de Datos. El modelo ER también es utilizado en conjunción con distintos métodos estructurados de tiempo real [Hatley, 1984] [Hatley et al., 1987] [Ward et al., 1985] [Ward, 1986], así como en diversas metodologías de desarrollo, tales como SSDAM [Goodland et al., 1990] o METRICA [CSI, 2000].

La integración del modelo ER en la Aproximación Estructurada ha sido, en general, fructífera [Yourdon, 1989]. No obstante, dicha integración no ha estado libre de problemas. El modelo ER puede describir aspectos del dominio del problema que el DFD no puede representar bien, o no puede representar en absoluto. En muchas ocasiones, dichos aspectos no complementaban el DFD, sino que lo contradecían. Al surgir una contradicción, y debido a que el DFD es el modelo dominante en la Aproximación Estructurada, la acción más corriente era desechar el modelo ER [Coad et al., 1990], obviando la información en él registrada.

El modelo ER también se ha usado en el marco de la Ingeniería de la Información [Martin, 1990]. En este caso, el modelo ER juega un papel distinto que en la Aproximación Estructurada. Bajo el punto de vista de que el esquema de datos tiene una mayor permanencia en el tiempo y describe de forma más acertada la estructura del dominio del problema que los procesos que intervienen en él, la Ingeniería de la Información prima la utilización inicial del modelo ER para modelar la totalidad de la información de una organización. Posteriormente, se delimita el marco de actuación para el desarrollo de un sistema software y se desarrolla un DFD de la misma forma que en la Aproximación Estructurada. Es claro que, en la Ingeniería de la Información, el modelo ER desempeña el papel de modelo dominante.

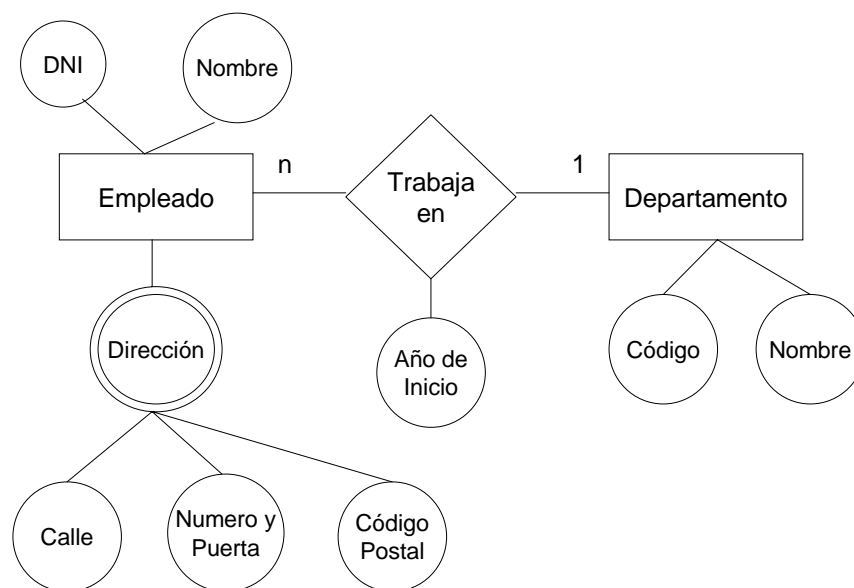


Figura 2.19. Ejemplo de modelo entidad-relación

2.5.3.1.1.3. Valoración

El modelo ER cumple, en buena medida, los criterios elegidos en el presente trabajo de Tesis. Aunque en el modelo original de Chen [Chen, 1976] no se especifica un procedimiento para la realización

del análisis, trabajos posteriores [Chen, 1983] **se establecieron guías procedimentales** para confeccionar un modelo ER a partir de una descripción del problema o del texto de una entrevista. Por otra parte, la utilización del modelo ER en el marco de cualquier método o aproximación **no implica la utilización de un determinado diseño** (en este caso, debido a que describe la estructura estática del dominio del problema, un determinado modelo lógico de bases de datos), ya que el modelo ER puede transformarse a cualquiera de ellos. Como contrapartida, en la literatura **no existe ningún procedimiento de selección del modelo lógico más adecuado**. Por último, **existen procedimientos bien definidos para derivar cualquiera de los modelos lógicos** (relacional, jerárquico o red).

Desde el punto de la amplitud y ligaduras computacionales, el modelo ER **no puede describir diversos aspectos** del dominio del problema. Estos aspectos son, principalmente, restricciones de integridad adicionales a las que incorpora, así como todos los aspectos dinámicos del dominio del problema. Sin embargo, todos los aspectos que si es capaz de describir son puramente conceptuales, **sin ligaduras computacionales de ningún tipo**.

2.5.3.1.1.4. Otras consideraciones

El modelo ER ha sufrido una considerable evolución. Las distintas aportaciones podrían agruparse desde dos puntos de vista: El de la modificación de los aspectos formales del modelo y el de la modificación del formalismo de diagramación.

Los aspectos formales han sufrido las siguientes extensiones:

- Refinamiento de la restricción de cardinalidad. [Chen, 1976], inicialmente, establece como posibles cardinalidades de una relación 1:1, 1:n y n:m. Posteriormente, se proponen cardinalidades del tipo (min, max), donde min es el número mínimo y max el número máximo de ocurrencias de una entidad en un tipo de relación. Se mantiene el criterio de que, si se especifica “n” como cardinalidad máxima, una entidad puede participar sin límite en un tipo de relación [Cezjdo et al., 1990] [Navathe et al., 1986] [Batini et al., 1986] [Batini et al., 1992].
- Introducción de la restricción de pertenencia. Debido al trabajo de [Batini et al., 1992], ya citado, se puede establecer cuándo es necesario que una entidad participe en un tipo de relación. Una participación mínima de 0 indica que la entidad no tiene por qué participar en un tipo de relación.
- Introducción de atributos compuestos y multivaluados. En el modelo ER inicial, para representar un atributo compuesto (o multivaluado) era necesario crear, artificialmente, un nuevo tipo de entidad, probablemente débil, y un nuevo tipo de relación.
- Introducción de *clustering*. Un tipo de entidad, en ocasiones, abstrae en exceso los elementos del dominio del problema. Así, por ejemplo, en un dominio empresarial se podría identificar el tipo de entidad “Trabajador”. Sin embargo, pueden existir muchos tipos de trabajadores distintos (Directores, obreros, etc.), que obedecen a distintas reglas y poseen diversos comportamientos. El *Clustering* [Feldman et al., 1986] es un mecanismo de abstracción que permite dividir un modelo ER en distintos niveles mediante la “descomposición” de un tipo de entidad de “mayor” nivel en varios tipos de entidad de “menor” nivel. Los tipos de relación pueden sufrir, igualmente,

de *clustering* [Jaeschke et al., 1993]. Actualmente, el *clustering* se puede evitar gracias al concepto de especialización.

En lo que respecta a los cambios en el formalismo de diagramación, es preciso mencionar la introducción de una notación gráfica para la inclusión de los atributos en el modelo ER. Aunque habitualmente se considera que en [Chen, 1976] se muestra la diagramación típica, que representa los atributos mediante circunferencias, dicha diagramación nunca se menciona. Dicha diagramación si aparece, por ejemplo, en [Chen, 1990]. También existen múltiples variaciones a la hora de representar los tipos de relación e incluso los tipos de entidad. Una revisión de distintos tipos de diagramación, aunque incompleta por definición, aparece en [Wertz, 1993]. Por último, mencionar que, en la práctica, para desarrollar una diagrama ER se utilizan las compilaciones realizadas a partir de la totalidad de las diagramaciones existentes, y no se prima un método en particular.

2.5.3.1.2. Modelo Funcional De Datos

2.5.3.1.2.1. Descripción

El Modelo Funcional de Datos (FD) fue propuesto por [Kerschberg et al., 1976], y refinado posteriormente por [Shipman, 1981]. El modelo FD posee características similares al modelo ER, anteriormente expuesto. De hecho, las capacidades de representación del modelo ER y el modelo FD son equivalentes.

No obstante, existen algunas diferencias significativas. La diferencia más acusada reside en cómo considera el modelo FD las relaciones. En el modelo ER anteriormente citado, las relaciones se modelaban mediante un constructor explícito, distinto de las entidades o atributos. En el modelo FD no existe un constructor para las relaciones, sino que se utilizan los atributos para relacionar las distintas entidades.

En el modelo FD los atributos se consideran funciones, cuyo dominio es un tipo de entidad y cuyo rango es otro tipo de entidad. Por ejemplo, supóngase que la entidad “persona” tiene un “nombre”. El modelo FD consideraría que “persona” y nombre” son entidades, y que “tiene” es una función de “Personas” a “nombres”.

Nótese que los conceptos de “entidad”, “relación” y “atributo” pierden su sentido al hablar del modelo FD. De hecho, el modelo FD no pertenece al tipo de modelos Entidad-Relación, sino a los Objeto-Papel. Las denominaciones correctas serían “objeto” (que engloba a “entidad” y “atributo”) y “papel” (que es un sinónimo no muy preciso de relación). Para el ejemplo anterior, “persona” y “nombre” serían objetos, mientras que “tiene” sería un papel.

El modelo FD posee un formalismo de diagramación introducido por [Shipman, 1981] y refinado por [Dayal et al., 1984]. La figura 2.20 muestra un ejemplo de modelo FD, donde se representa la misma información que en el modelo ER de la figura 2.19. Los rectángulos redondeados representan objetos, mientras que las flechas representan papeles.

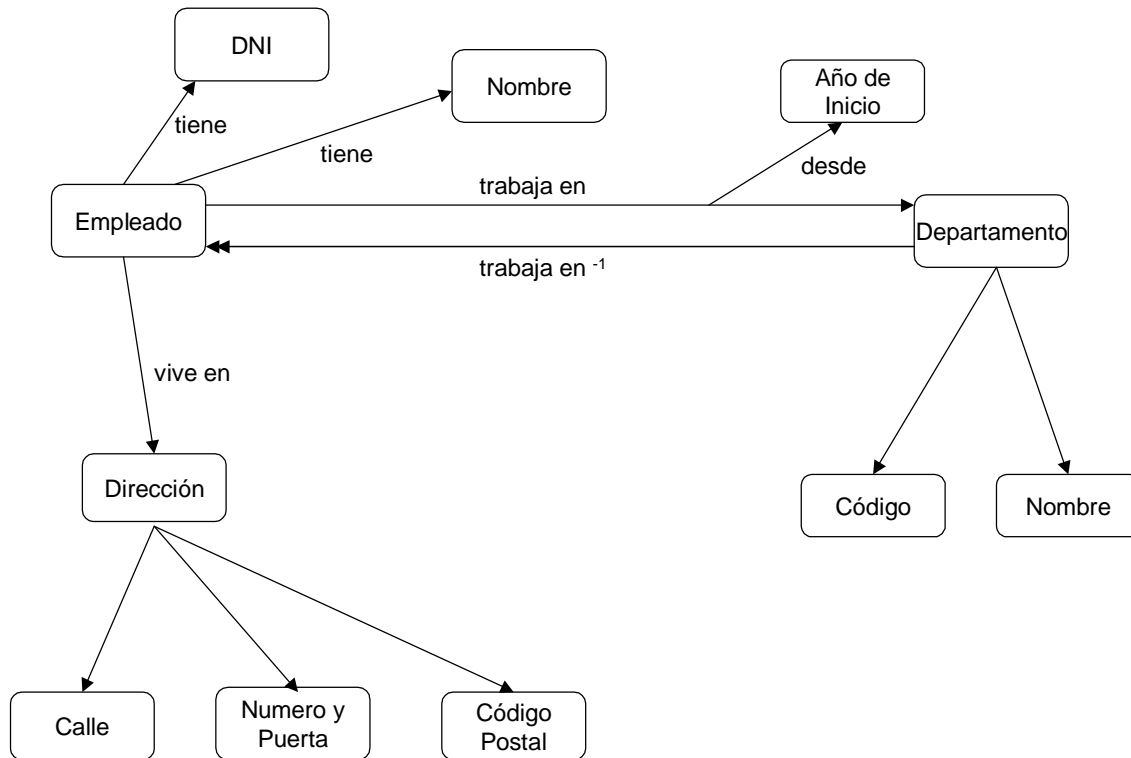


Figura 2.20. Ejemplo de modelo funcional de datos

2.5.3.1.2.2. Métodos de Análisis

El modelo FD únicamente se ha utilizado en el contexto de las Aproximaciones de Bases de Datos. El método más importante que ha utilizado el modelo FD es el lenguaje DAPLEX [Shipman, 1981].

2.5.3.1.2.3. Valoración

Debido a que los conceptos subyacentes del modelo FD son similares a los del modelo ER (a excepción de las funciones como sustitutos del concepto de relación), **las críticas realizadas al modelo ER son igualmente válidas para el modelo FD.**

2.5.3.1.2.4. Otras Consideraciones

El modelo FD, como modelo paradigmático de los modelos Objeto-Papel (*Object-Role*), ha tenido poca relevancia en el desarrollo de software. Su utilización ha estado restringida a los laboratorios de investigación.

Al igual que el modelo ER, el modelo FD ha evolucionado en el tiempo. Actualmente, el sucesor más relevante del modelo FD, y por lo tanto más conocido y usado, es NIAM [Nijssen et al., 1989]. NIAM posee una diagramación distinta de la propuesta por [Shipman, 1981] (circunferencias en lugar de rectángulos redondeados, explicitación de los papeles de cada objeto en la relación funcional que los une,

etc.). No obstante, NIAM, al igual que el modelo FD, tampoco posee gran relevancia en el desarrollo de software.

2.5.3.1.3. Modelo Entidad-Relación Extendido

2.5.3.1.3.1. Descripción

El modelo Entidad-Relación Extendido (ERE) surge debido a las deficiencias en la descripción del dominio del problema mostradas por el modelo ER. Dichas deficiencias impiden, por ejemplo, describir las taxonomías existentes en el dominio del problema.

La característica principal del modelo ERE es la introducción de mecanismos de generalización / especialización [Batini et al., 1986] [Teorey et al., 1986], que se agregan sobre el modelo ER básico. El concepto de especialización permite definir taxonomías en forma de jerarquías de especialización, donde un tipo de entidad (supertipo) se pone en relación en varios subtipos de entidad. Lógicamente, la introducción de mecanismos de generalización / especialización no altera los mecanismos subyacentes proporcionados por el modelo ER básico, por lo que el modelo ERE posee todas las características anteriormente descritas del modelo ER.

Existen dos restricciones que se pueden aplicar a las jerarquías de especialización. En primer lugar, la especialización puede ser total o parcial. Una especialización es total si toda entidad perteneciente al supertipo debe pertenecer, asimismo, a alguno de los subtipos. En el caso contrario, la especialización es parcial. En segundo lugar, los subtipos pueden ser disjuntos o solapados. Los subtipos son disjuntos si cada entidad pertenece, como mucho, a un único subtipo. En caso contrario, los subtipos están solapados. Al definir una jerarquía de especialización, los subtipos heredan la totalidad de los atributos del supertipo. Este concepto de herencia se deriva de la influencia ejercida por el paradigma de Orientación a Objetos [Hull et al., 1987].

Un ejemplo de jerarquía de especialización se presenta en la figura 2.21.

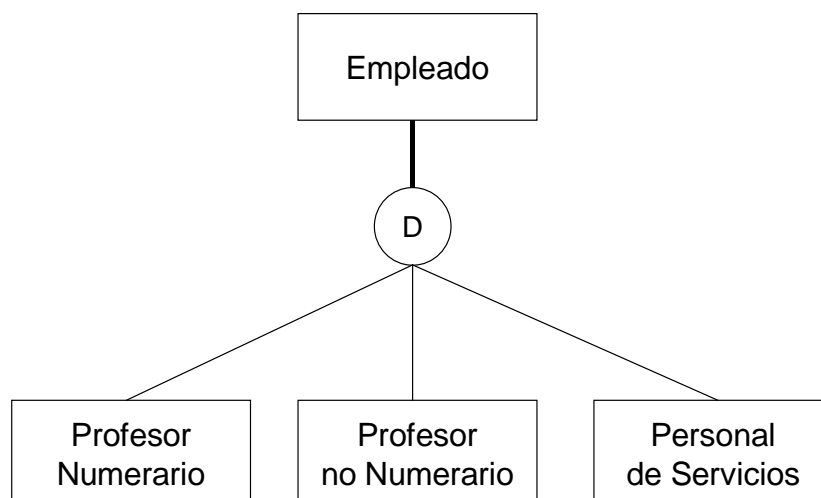


Figura 2.21. Jerarquía de especialización.

El concepto de generalización permite definir supertipos a partir de varios subtipos distintos. En este caso, la semántica es ligeramente distinta, ya que no se aplica el concepto de herencia. El supertipo puede contener atributos distintos de los existentes en los subtipos. De la misma forma, el concepto de subtipos disjuntos o solapados pierde sentido (los subtipos son siempre disjuntos), y únicamente conserva vigencia la restricción de generalización total o parcial.

Un ejemplo de jerarquía de generalización se presenta en la figura 2.22.

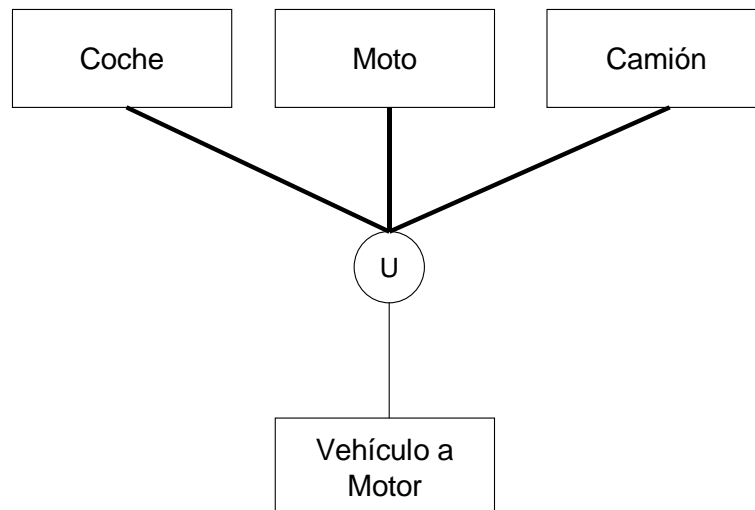


Figura 2.22. Jerarquía de generalización.

2.5.3.1.3.2. Métodos de Análisis

En la práctica, el modelo ER y el modelo ERE no se distinguen en su utilización; esto es, debido a que son prácticamente iguales, a excepción de los mecanismos de especialización / generalización, cualquier método donde se use el modelo ER puede usarse el modelo ERE de forma inmediata.

El modelo ERE también puede utilizarse de forma aislada. Una metodología de desarrollo detallada, basada en el modelo ERE, se puede encontrar en [Teorey et al., 1986].

2.5.3.1.3.3. Valoración

El modelo ERE posee exactamente las mismas características que el modelo ER del que se deriva. La introducción de mecanismos de especialización / generalización no hace más que favorecer la descripción de los elementos estructurales del dominio del problema, que el modelo ER podía igualmente describir, si bien con mayor dificultad.

2.5.3.1.3.4. Otras Consideraciones

Las características del modelo ERE expuestas no se refieren a un modelo concreto, sino que resumen las características más importantes de la multiplicidad de modelos existentes. Entre éstos, se pueden citar las siguientes variantes: el modelo *Extended Entity-Relationship* [Teorey et al., 1986], *Extended*

Entity-Relationship [Markowitz et al, 1992], *Extended Conceptual Entity-Relationship* [Czejdo et al., 1990], *Extended Entity-Relationship* [Hohenstein et al., 1988] y *Entity-Category-Relationship* [Navathe et al., 1986].

Por otra parte, es preciso indicar en este punto que la introducción de mecanismos de especialización / generalización no es privativo de los modelos Entidad-Relación. Los modelos Objeto-Papel (modelo FD [Kerschberg et al., 1976] [Shipman, 1981], NIAM [Nijssen et al., 1989]) también poseen capacidades de definición de taxonomías de especialización.

2.5.3.1.4. Resumen

La valoración de los criterios para los modelos básicos se resume en la tabla 2.5.

Criterios	ER	ERE	FD
<i>Amplitud</i>	Parcial	Parcial	Parcial
<i>Ligaduras Computacionales</i>	No	No	No
<i>Procedimiento de Uso</i>	Total	Total	Total
<i>Selección de Diseño</i>	Ninguno	Ninguno	Ninguno
<i>Derivación de Diseño</i>	Total	Total	Total

Tabla 2.5. Valores de los criterios para los modelos básicos

2.5.3.2. Modelos Avanzados

Los modelos avanzados también se denominan Modelos Conceptuales, debido a que siguen las normas dictadas en el estándar ISO de modelización conceptual [van Griethuysen, 1982]. No obstante, en el marco del presente trabajo de Tesis se denominan modelos avanzados para evitar confusiones.

Al igual que los modelos básicos, los modelos avanzados permiten describir la estructura y relaciones de los elementos u objetos del dominio del problema. Sin embargo, y a diferencia de los modelos básicos, los modelos avanzados permiten definir asimismo las leyes y reglas que gobiernan el comportamiento de dichos elementos u objetos en el dominio del problema.

En lo que respecta a la descripción de los elementos del dominio del problema, así como a las relaciones que se establecen entre los mismos, los modelos avanzados no poseen diferencias acusadas frente a los modelos básicos, anteriormente expuestos. La diferencia más relevante entre los modelos básicos y avanzados reside en la descripción del comportamiento de dichos elementos en el dominio del problema.

Existen tres tipos principales de modelos avanzados, las cuales se diferencian por la forma en que abordan la definición del comportamiento. Dos de estos tipos fueron indicados en el trabajo seminal de [Olivé, 1986], los cuales son **los modelos avanzados operativos y los modelos avanzados declarativos**. El tercer tipo de modelos avanzados son los modelos avanzados orientados a objetos, los cuales toman protagonismo con posterioridad al trabajo de Olivé, y por su relevancia actual se tratan en epígrafe posterior.

Se debe indicar que se habla de “modelos avanzados operativos” y “modelos avanzados declarativos”, en plural, y no de “modelo avanzado operativo” o “declarativo”, en singular. Ello se debe a que no puede identificarse un modelo paradigmático. La falta de un modelo paradigmático se debe a la baja repercusión de los modelos avanzados fuera del laboratorio. Es por ello que existen múltiples modelos, con muy similares características que únicamente se diferencian en los aspectos de notación, de los cuales ninguno es aceptado como “referente” del conjunto de modelos avanzados. La descripción que sigue se basa, por lo tanto, en un estudio de las características comunes de cada conjunto de modelos, sin pretender realizar una exégesis de cada uno de los mismos.

Al igual que ocurrió con los modelos básicos, cuando sea necesario, en la descripción de cada uno de los modelos, se añadirá un epígrafe denominado Otras Consideraciones.

2.5.3.2.1. Modelos Avanzados Operativos

2.5.3.2.1.1. Descripción

Los modelos avanzados operativos (AO), como ya se ha indicado, poseen capacidades de descripción de la estructura del dominio del problema similares a los modelos básicos.

En lo referido a la descripción del comportamiento, los modelos AO están muy influenciados por el concepto de transacción. Una transacción es un conjunto de operaciones que cambia el estado de la Base de Datos al producirse un evento predeterminado.

Las transacciones se definen, habitualmente, de forma procedimental. La definición procedimental implica la inclusión de estructuras de control, o lo que es lo mismo, la necesidad de describir los detalles de funcionamiento de la transacción. El detalle de funcionamiento implica determinar qué elementos del dominio del problema se ven afectados y qué acciones deben realizarse sobre los mismos. Para que se active la transacción asociada al evento, es necesario que se cumplan un conjunto de condiciones. En la práctica totalidad de los casos, los eventos y las condiciones se definen sobre la extensión actual del dominio del problema, esto es, sobre el conjunto de elementos u objetos del dominio de discurso.

Existen múltiples alternativas de notación para la descripción de los elementos del dominio de discurso, eventos, acciones y condiciones. De forma general, la especificación del comportamiento, en los modelos AO, tiene la forma general mostrada en la figura 2.23 [Rolland et al., 1992] [Loucopoulos et al., 1995].

```
WHEN <Evento>  
IF ALL <Condiciones>  
THEN <Transacción>  
ON <Elementos>
```

Figura 2.23. Descripción general del comportamiento en los modelos AO

Esta definición tiene un carácter global, ya que se define de forma separada a cualquier tipo de entidad del dominio del problema; esto es, la descripción de las posibles acciones se separa de la descripción de la estructura del dominio del problema.

La aproximación operativa tiene la ventaja de que el comportamiento se describe de forma muy compacta, pero asimismo posee dos inconvenientes [Olivé, 1986]:

- La definición de una operación (transacción) puede afectar a varios elementos (los indicados, en la figura 2.23, con el identificador ON). Asimismo, elemento puede verse afectado por varias operaciones distintas. La comprensión de qué operaciones pueden realizarse sobre un determinado elemento exige el estudio de todas las operaciones definidas.
- Como corolario de lo anterior, la introducción de nuevas reglas que gobiernan el dominio del problema, o la modificación de las existentes, exige habitualmente la modificación de varias transacciones.

2.5.3.2.1.2. Métodos de Análisis

Como se ha indicado anteriormente, los modelos AO apenas han tenido repercusión en la industria, dedicándose en su mayoría a la experimentación en el laboratorio. Ningún método de desarrollo relevante utiliza modelos AO.

No obstante, en el laboratorio existen multitud de métodos que utilizan modelos AO. Nótese que dichos métodos son laxos, en el sentido de que se configuran alrededor del modelo AO; esto es, el modelo AO forma el corazón del método. Métodos representativos son SHM+ [Brodie et al., 1982], REMORA [Rolland et al., 1982], TAXIS [Mylopoulos et al., 1980] y OBCM [Wand et al., 1989].

2.5.3.2.1.3. Valoración

Los métodos que utilizan los modelos AO **no proponen procedimientos bien definidos para construir el modelo**. En la mayoría de los casos el proceso de construcción es *ad-hoc*, dándose prioridad a las capacidades de representación del modelo sobre su utilización disciplinada.

Los modelos AO implican, necesariamente, **un determinado tipo de diseño de índole procedimental**, compatible con el concepto de transacción que utilizan. Este tipo de diseño puede implementarse sobre una base de datos que posea un lenguaje de manipulación de datos de tipo procedimental, o en programas de aplicación separados que actúen sobre la base de datos. En ambos casos, y aunque no existe un procedimiento de derivación explícito, los conceptos del modelo AO **pueden trasladarse, por proximidad, casi directamente al tipo de diseño utilizado**.

En lo referido a la amplitud y ligaduras computacionales, los modelos AO poseen una gran capacidad para describir todos los aspectos del dominio del problema. Los modelos AO permiten describir:

- Los elementos, así como las relaciones que establecen dichos elementos entre sí, en el dominio del problema.

- Las posibles acciones que ocurren en el dominio del problema, reflejadas por el concepto de transacción.
- Las distintas situaciones que se dan en el dominio del problema, reflejadas en los eventos.

No obstante, los modelos AO **representan con dificultad hechos** que afecten, de forma general, a un conjunto de datos, esto es, reglas de negocio, así como aspectos temporales.

Referido a las ligaduras computacionales, los modelos AO representan las acciones que ocurren en el dominio del problema **de una forma muy próxima a la implementación de dichas acciones** en una máquina subyacente, sin duda debido a la influencia del concepto de transacción en este tipo de modelos. La descripción de las acciones ocurre, por lo tanto, muy bajo nivel de detalle, que se puede considerar de diseño, lo cual es el origen de la facilidad de derivación de los modelos AO a un diseño determinado.

2.5.3.2.2. Modelos Avanzados Declarativos

2.5.3.2.2.1. Descripción

Los modelos avanzados declarativos (AD) poseen características similares a los modelos AO. NO obstante, también poseen marcadas diferencias. Estas diferencias se refieren principalmente al mecanismo por el que los modelos AD representan los aspectos dinámicos, de comportamiento, del dominio del problema.

Los modelos AD describen el comportamiento de los elementos u objetos del dominio del problema utilizando expresiones lógicas –reglas–. La ventaja de utilizar reglas reside en que es posible describir el comportamiento sin tener en cuenta la implementación de dicho comportamiento (al contrario de lo que ocurre en los modelos AO). No es necesario, por lo tanto, describir aspectos de control de las acciones, por lo que los modelos AD se abstraen de la máquina subyacente.

Un modelo AD está formado por tres componentes:

1. Hechos, que pueden ser elementales o derivables a partir de otros hechos elementales.
2. Reglas de derivación, que especifican qué es lo que debe suceder cuando se produce un hecho determinado.
3. Un conjunto de restricciones (condiciones) que determina qué hechos son válidos en un determinado momento.

Nótese la relación que se produce entre hechos y condiciones. Habitualmente, un hecho se da (o no) en el dominio sin referencia a ninguna condición. La mejor manera de entender la particular relación entre hechos y condiciones es considerar que los hechos proporcionan la misma capacidad expresiva que los eventos de los modelos AD.

Existen múltiples alternativas de notación para la descripción de los elementos del dominio de discurso, hechos, reglas y restricciones. De forma general, la especificación del comportamiento, en los modelos AD, tiene la forma general mostrada en la figura 2.24 [Rolland et al., 1992] [Loucopoulos et al., 1995].

IF <Hecho>
AND <Condiciones>
THEN <Regla de Derivación>

Figura 2.24. Descripción general del comportamiento en los modelos AD

La declaración de condiciones, hechos y reglas de derivación se realiza para cada entidad del dominio de discurso. En cierta medida, los modelos AD describen el dominio del problema de forma inversa a los modelos AO, coincidiendo los puntos fuertes de la aproximación declarativa con los débiles de la operativa. En concreto, en los modelos AD, la especificación del comportamiento está más atomizada y distribuida entre los diversos tipos de entidad que forman parte del modelo. Sin embargo, esta distribución hace más sencilla la comprensión de las operaciones especificadas en el modelo, así como la inclusión o modificación de hechos, condiciones y reglas.

Adicionalmente, es necesario indicar que los modelos AD, debido a su capacidad para expresar las acciones que ocurren en el dominio del problema mediante expresiones lógicas, permiten incluir con relativa facilidad aspectos temporales mediante la utilización de lógicas temporales. Así, por ejemplo, TELOS [Mylopoulos et al., 1990] permite la definición de hechos, condiciones y reglas que incluyen puntos o intervalos temporales.

2.5.3.2.2. Métodos de Análisis

Como se ha indicado anteriormente, los modelos AD, al igual que los modelos AO, apenas han tenido repercusión en la industria. Los métodos que utilizan modelos AD, al igual que en el caso de los modelos AO, se configuran alrededor del modelo AD. Los métodos más representativos son CIAM [Gustafsson et al., 82], ERAE [Dubois et al., 1989] y TELOS [Mylopoulos et al., 1990].

2.5.3.2.3. Valoración

Los métodos que utilizan los modelos AD **no proponen procedimientos bien definidos para construir el modelo**. En la mayoría de los casos el proceso de construcción es similar a lo que ocurre con los modelos AO, esto es, el proceso de construcción es *ad-hoc*.

Por el contrario, los modelos AD no predeterminan ningún tipo de diseño. Exceptuando el caso en que exista una base de datos que soporte la declaración de datos, hechos y reglas, en cuyo caso es lógico que el modelo se implemente directamente, los modelos AD **son independientes de cualquier tipo de**

diseño. No en vano, la diferencia entre un modelo AD y un método de especificación formal es prácticamente inexistente.

Desgraciadamente, no existe ningún procedimiento para identificar el tipo de diseño más adecuado, ni es posible derivar tal tipo de diseño.

En lo referido a la amplitud y ligaduras computacionales, los modelos AD poseen una capacidad total para describir los elementos, hechos y fenómenos existentes en el dominio del problema, incluido el tiempo, independientemente de la implementación que se utilice posteriormente.

2.5.3.2.3. Resumen

La valoración de los criterios para los modelos avanzados se resume en la tabla 2.6.

Criterios	Modelos Avanzados Operativos	Modelos Avanzados Declarativos
<i>Amplitud</i>	Parcial	Total
<i>Ligaduras Computacionales</i>	Si	No
<i>Procedimiento de Uso</i>	Ninguna	Ninguna
<i>Selección de Diseño</i>	Prescribe	No Aconseja
<i>Derivación de Diseño</i>	Total	Ninguno

Tabla 2.6. Valores de los criterios para los modelos avanzados

2.5.3.3. Modelos Orientados A Objetos

Es difícil exponer, desde un punto de vista sintético, qué significa el concepto de Orientación a Objetos. En palabras de [Rentsch, 1982]:

“Creo que la programación orientada a objetos será en los años 80 lo que fue la programación estructurada en los años 70. Todo el mundo estará a favor de ella. Todas las empresas de desarrollo afirmarán que sus productos soportan la orientación a objetos. Todos los directivos querrán utilizarla. Todos los programadores la practicarán (de forma distinta). Y nadie sabrá lo que es”.

Las palabras de Rentsch fueron premonitorias. Actualmente, todo modelo está orientado a objetos, independientemente de que se haya usado principalmente para expresar las transformaciones del dominio del problema (como los diagrama de flujo de datos), la dinámica (como los statecharts) o para describir la interacción entre agentes (como los casos de uso).

En el presente trabajo de Tesis, se hará una interpretación más restrictiva de la Orientación a Objetos gracias al concepto de modelo paradigmático. Se destacan, de esta forma, el **Diagrama de Clases** y la **Historia de la Vida de las Entidades**. Al igual que en los epígrafes anteriores, donde sea necesario se incluirá un epígrafe adicional denominado Otras consideraciones, en el que se indicarán determinados aspectos de los modelos sin romper la estructura de descripción utilizada en el presente Estado de la Cuestión.

2.5.3.3.1. Diagrama de Clases

2.5.3.3.1.1. Descripción

El diagrama de clases (DC) es un modelo cuya finalidad es representar las clases de objetos existentes en el dominio del problema, así como las relaciones que establecen entre las mismas.

Desde el punto de vista de la descripción del dominio del problema, el DC posee características muy similares al modelo ERE. De hecho, posee similares capacidades de representación, pudiendo describir clases (tipos de entidad en el modelo ERE), relaciones y jerarquías de especialización con herencia. El DC no puede describir generalizaciones, como el modelo ERE, pero en contrapartida permite utilizar relaciones de agregación.

Al igual que los modelos avanzados, el DC permite especificar (al menos, en principio) el comportamiento de los objetos pertenecientes al dominio del problema. En el DC, el comportamiento se especifica utilizando métodos, que se corresponden con los distintos servicios que cada clase (o cada objeto perteneciente a cada clase) puede ofrecer.

No obstante, y a diferencia de los modelos avanzados, los métodos del DC no acostumbran a describirse, sino únicamente a declarar su existencia. Ello implica que el DC no contiene la descripción de los servicios que proporciona cada clase u objeto. De hecho, actualmente los métodos del DC ni siquiera se indican durante la actividad de Análisis [Larman, 1999].

El formalismo de representación del DC es muy similar al de los modelos básicos, aunque existen grandes variaciones según los distintos autores y épocas. Actualmente, la notación más utilizada es UML [Rumbaugh et al., 1998]. Un ejemplo de DC, según la notación UML, se presenta en la figura 2.25.

2.5.3.3.1.2. Métodos de análisis

Existen una multitud de métodos de análisis que incorporan el DC, en concreto, todos los métodos orientados a objetos. Se pueden identificar cuatro grandes grupos de métodos:

1. Métodos orientados a las funciones. Este tipo de métodos se caracterizan por la realización, en una primera fase, de un análisis orientado a las funciones, utilizando habitualmente DFD's. Posteriormente, dichas funciones se utilizan (junto con otro tipo de información obtenida del dominio del problema) para derivar el DC. El método más relevante de este tipo de métodos es OMT [Rumbaugh et al., 1991].

2. Métodos orientados a los datos. Este tipo de métodos surgen debido a la influencia que los primeros trabajos en Análisis Orientado a Objetos sufrieron por parte de los modelos básicos y avanzados. Estos métodos se caracterizan porque los objetos se identifican, en una primera fase, de una forma similar que en el modelo ER. Asimismo, en esta primera fase, la información registrada en el DC es similar a la de un modelo ER. Posteriormente, el modelo inicial se enriquece al añadir aspectos de detalle, como los atributos o métodos. El método más relevante de este tipo es [Coad et al., 1990].
3. Métodos orientados a la interacción. Este tipo de métodos se caracterizan por identificar, en primer lugar, los servicios que el futuro sistema software debe ofrecer. Posteriormente, los servicios se utilizan para identificar los objetos y la interacción entre los mismos. El método más característico de este tipo es OOSE [Jacobson, 1992].
4. Métodos centrados en los objetos. Este tipo de métodos utilizan los conceptos de la Orientación a Objetos desde las primeras fases del desarrollo. Esta orientación persigue la independencia de la Orientación a Objetos de cualquier otra aproximación, en tanto en cuanto los conceptos implicados son lo suficientemente potentes para modelizar adecuadamente el dominio de discurso. Un ejemplo de este tipo de métodos es [Meyer, 1988].

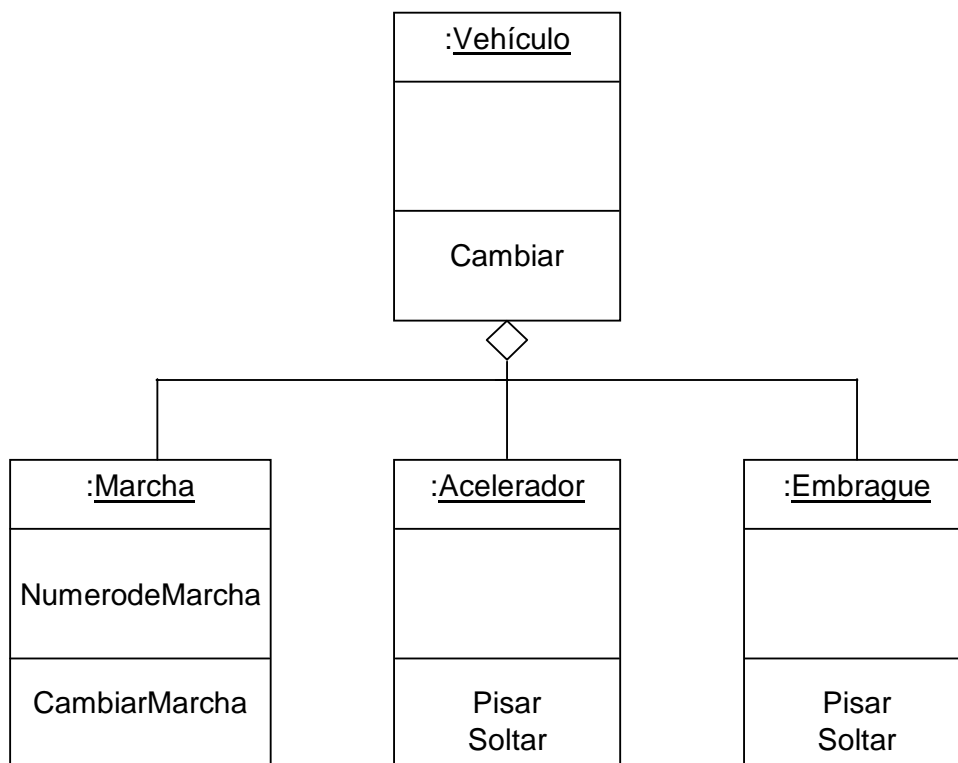


Figura 2.25. Ejemplo de diagrama de clases

2.5.3.3.1.3. Valoración

Existen **multitud de formas de derivar un diagrama de clases**, tal y como se refleja en la diversidad de métodos indicados anteriormente. Adicionalmente, **la utilización de un DC durante el**

análisis prescribe, necesariamente, la utilización de un Diseño Orientado a Objetos. La traslación del análisis al diseño es directa, ya que los conceptos y el formalismo de representación implicados en la fase de análisis es un subconjunto de los usados durante el diseño .

Desde el punto de vista de la amplitud y ligaduras computacionales, el DC es similar a un modelo ERE y, por lo tanto, **su capacidad de representación es parcial**. No obstante, y a diferencia del modelo ERE, el DC posee claras ligaduras computacionales, ya que **los conceptos que el DC puede representar son conceptos utilizados durante el diseño** de productos software.

2.5.3.3.1.4. Otras Consideraciones

Existen un conjunto de modelos avanzados, distintos de los modelos avanzados operativos o declarativos, que se pueden denominar Orientados a Objetos. Estos modelos se desarrollan como extensiones al modelo ER o ERE, incluyendo los conceptos característicos de la orientación a objetos: Identidad, clasificación, herencia y polimorfismo. Se pueden citar como integrantes de este grupo los modelos OOER (*Object Oriented Entity-Relationship*) [Navathe et al., 1988], BIER (*Behaviour-Integrated Entity-Relationship*) [Kappel et al., 1988] y OOERM (*Object-Oriented Entity-Relationship Model*) [Gorman et al., 1991].

Este conjunto de modelos poseen un formalismo de representación tanto de la estructura como del comportamiento del dominio del problema. La estructura, como en la generalidad de los modelos avanzados, apenas se diferencia de la introducida por los modelos ER o ERE. En lo que respecta a la descripción del comportamiento, es característico de este tipo de modelos el encapsulamiento de los datos y las operaciones de manipulación en una única unidad del modelo (el objeto). El nivel en que pueden especificarse los detalles de las operaciones es variable, aunque la mayoría de los modelos utilizan una especificación de tipo navegacional, habitualmente basada en adaptaciones del lenguaje SQL (*Structured Query Language*).

Asimismo, es posible considerar ciertos modelos desde varias perspectivas distintas. Por ejemplo, el modelo TELOS [Mylopoulos et al., 1990] puede considerarse como un modelo avanzado declarativo o un modelo orientado a objetos.

2.5.3.3.2. Historia de la Vida de las Entidades

2.5.3.3.2.1. Descripción

La Historia de la Vida de las Entidades (HVE) es un modelo cuyo propósito es describir las acciones a las que se ven sometidas los elementos del dominio del problema durante su tiempo de vida. La asociación de las entidades con las acciones que pueden sufrir es lo que caracteriza a la HVE como un modelo de la misma categoría que los orientados a objetos.

Por tiempo de vida debe entenderse el periodo de tiempo durante el cual un elemento determinado pertenece al dominio del problema. Así, por ejemplo, para una empresa, un cliente es interesante desde el momento en que dicho cliente manifiesta su intención de comerciar con la empresa. En el momento en que

dicho cliente deja de comprar, puede dejar de ser interesante. El periodo entre los dos citados momentos (el inicio de las compras, el fin de las compras) es el tiempo de vida del cliente.

La HVE utiliza dos conceptos básicos:

1. Entidades: Son elementos distinguibles del dominio del problema, similares a las entidades de un modelo ER.
2. Acciones: Son las distintas acciones que sufren las distintas entidades.

No existe una notación diferenciada para entidades y acciones, representándose ambas como rectángulos. Las acciones se indican en relación con las entidades que las sufren utilizando un árbol invertido, en cuya construcción se tienen en cuenta los aspectos temporales (la secuencia de acciones que sufre la entidad). En general, una acción ubicada a la izquierda ocurre antes que la situada a su derecha. Adicionalmente, es posible representar acciones optativas e iterativas, utilizando los símbolos "O" y "*", respectivamente. La raíz del árbol es siempre la entidad que sufre las acciones.

Un ejemplo de HVE se muestra en la figura 2.26.

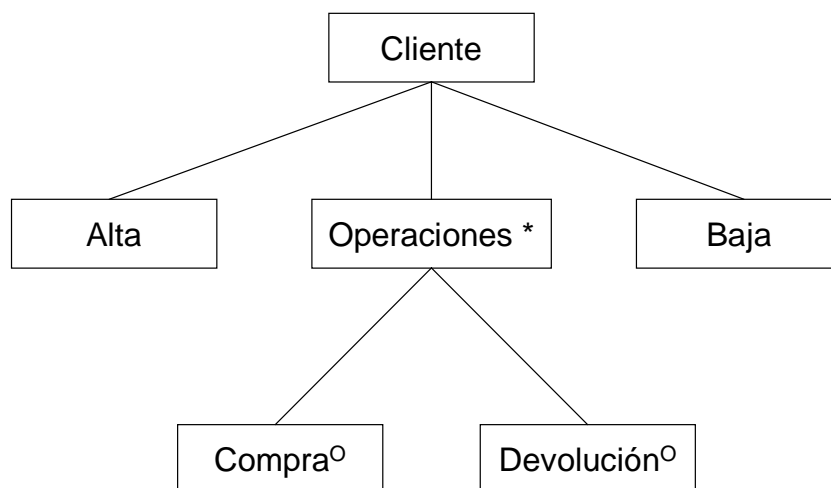


Figura 2.26. Ejemplo de historia de la vida de las entidades

2.5.3.3.2.2. Métodos de análisis

La HVE se ha utilizado principalmente en el método JSD [Jackson, 1983], así como en algunas metodologías estructuradas, tales como SSADM [Goodland et al., 1990].

2.5.3.3.2.3. Valoración

Tanto JSD [Jackson, 1983] como SSADM [Goodland et al., 1990] **proponen procedimientos para desarrollar** una HVE. Dichos procedimientos son distintos en lo que respecta a los pasos procedimentales, pero ambos cumplen con el propósito de generar un HVE.

La utilización de una HVE no determina ningún tipo de diseño, ya que puede derivarse igualmente a un diseño estructurado, orientado a objetos o al propio diseño JSD propuesto por [Jackson, 1983]. No obstante, aun que fácil, **no existe un procedimiento de derivación estándar**, aplicándose principalmente reglas heurísticas.

En lo referente a la amplitud y ligaduras computacionales, la HVE posee **limitadas capacidades expresivas**, pudiendo reflejar únicamente las entidades y acciones del dominio del problema. No obstante, el grado de descripción del detalle interno de entidades y acciones es mínimo. En contrapartida, la HVA **no posee ligaduras computacionales**, aunque su utilización en problemas sencillos pueda favorecer, en algunos casos, la realización de un diseño procedimental anticipado.

2.5.3.3.3. RESUMEN

La valoración de los criterios para los modelos orientados a objetos se resume en la tabla 2.7.

Criterios	Diagrama de Clases	Historia de la Vida de las Entidades
<i>Amplitud</i>	Parcial	Parcial
<i>Ligaduras Computacionales</i>	Si	No
<i>Procedimiento de Uso</i>	Total	Total
<i>Selección de Diseño</i>	Prescribe	No aconseja
<i>Derivación de Diseño</i>	Total	Parcial

Tabla 2.7. Valores de los criterios para los modelos orientados a objetos

2.5.4. MODELOS ORIENTADOS A LA DINÁMICA

Los modelos orientados a la dinámica se caracterizan por describir el comportamiento de los distintos elementos del dominio del problema en función del tiempo. Este tipo de modelos se utilizan, mayoritariamente, en la Aproximación de Tiempo Real, donde el software debe respetar estrictos límites temporales. Los modelos incluidos en este grupo son las **Máquinas de Estado Finito**, los **Diagramas de Transición de Estados** y los **Statecharts**.

2.5.4.1. MÁQUINAS DE ESTADO FINITO

2.5.4.1.1. Descripción

Las máquinas de estado finito (MEF) son un formalismo derivado de la Teoría de Automatas [Sudkamp, 1996]; más en concreto, de los autómatas regulares. Matemáticamente, los estados son un mecanismo que permite dotar a los autómatas de memoria, ya que el estado actual de un autómata dependerá de los estados anteriores y de la secuencia de entradas que han obtenido.

En el ámbito de la Ingeniería del Software, las MEF se utilizan con la finalidad de representar una situación que ocurre durante un tiempo indeterminado. El concepto de estado es útil para este propósito, ya que un estado puede considerarse como la representación de una situación, y la secuencia de estados como la secuencia de situaciones que se han dado en el dominio del problema.

Las MEF son un modelo de tipo tabular. En vertical, se enumeran exhaustivamente todas las entradas posibles, mientras que en horizontal se enumeran, también exhaustivamente, todos los estados por los que discurre el sistema. Los estados y entradas pueden intercambiarse de lugar. En las celdas apropiadas puede especificarse un estado del conjunto de los disponibles. La lectura que se hace de la tabla es la siguiente: Cuando ocurre una entrada en un estado determinado, se produce un cambio de estado, siendo el nuevo estado el indicado en la celda intersección de la entrada y del estado actual.

Un ejemplo de MEF se muestra en la figura 2.27.

Estado Entrada	Cajero en Espera	Aceptar PIN	Aceptar Transacción	Saldo	Reintegro	Imposición
Tarjeta Insertada	Aceptar PIN					
PIN Correcto		Aceptar Transacción				
PIN Erróneo		Cajero en Espera				
Petición Saldo			Saldo			
Petición Reintegro			Reintegro			
Petición Imposición			Imposición			
Transacción Correcta				Aceptar Transacción	Aceptar Transacción	Aceptar Transacción
Transacción Cancelada				Cajero en Espera	Cajero en Espera	Cajero en Espera
Salir			Cajero en Espera			

Figura 2.27. Ejemplo de máquina de estado finito

2.5.4.1.2. Métodos de Análisis

Las MEF se han usado sobre todo en el método de [Hatley, 1984] [Hatley et al., 1987].

2.5.4.1.3. Valoración

El método de Hatley [Hatley, 1984] [Hatley et al., 1987] **no define un procedimiento para**

confeccionar una MEF. Asimismo, la utilización de una MEF no determina un determinado tipo de diseño, ni existe ningún procedimiento de derivación a ningún tipo de diseño específico.

En lo que respecta a la amplitud y ligaduras computacionales, la MEF posee una **capacidad de representación muy limitada**, pudiendo únicamente describir las situaciones, desde un punto de vista genérico, que ocurren en el dominio del problema. En contrapartida, el concepto de estado **no posee ninguna ligadura computacional**.

2.5.4.2. DIAGRAMAS DE TRANSICIÓN DE ESTADOS

2.5.4.2.1. Descripción

Los diagramas de transición de estados (DTE) son un modelo también derivado de la Teoría de Automatas [Sudkamp, 1996]. Los DTE son expresivamente equivalentes a las MEF.

Los DTE utilizan un grafo, en lugar de una tabla, como mecanismo de representación. Los nodos corresponden a estados mientras que los arcos corresponden con transiciones entre los mismos, originadas al producirse una entrada. Un ejemplo de DTE se muestra en la figura 2.28.

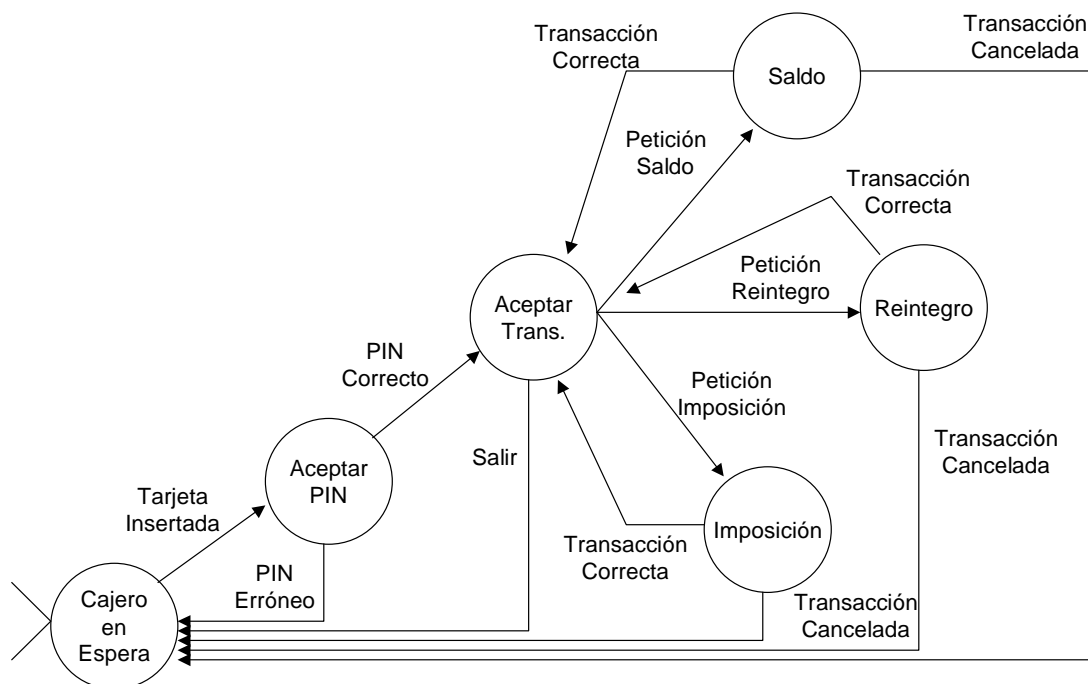


Figura 2.28. Ejemplo de Diagrama de Transición de Estados.

2.5.4.2.2. Métodos de Análisis

Los DTE se han utilizado en los en los métodos estructurados [Yourdon, 1989], así como en varias métodos de orientación a objetos para describir el comportamiento de los objetos en un dominio de discurso [Rumbaugh et al., 1991] [Booch, 1991] [Larman, 1999]. No obstante, es necesario indicar que los DTE

utilizados en los métodos orientados a objetos son más parecidos a los statecharts (descritos en el siguiente epígrafe) que a los DTE.

2.5.4.2.3. Valoración

El DTE es un modelo accesorio dentro de métodos estructurados y orientados a objetos. Por ello, **no se ha definido un procedimiento que apoye su construcción. Asimismo, no prescribe ningún tipo de diseño, y no existen procedimiento de derivación del DTE hacia ningún tipo de diseño determinado.**

En lo referente a la amplitud y ligaduras computacionales, el DTE posee las mismas características que las MEF, esto es, **su amplitud es parcial, y no posee ligaduras computacionales.**

2.5.4.3. STATECHARTS

2.5.4.3.1. Descripción

Los statecharts son una ampliación, en el sentido matemático del término, del formalismo de los grafos. En lugar de considerar únicamente dos conjuntos, nodos y arcos, y establecer una correspondencia entre dichos conjuntos, los hipergrafos permiten que un arco tenga como origen o destino varios nodos. Esta ampliación hace más difícil el tratamiento matemático de los statecharts, pero permite que la cantidad de situaciones que un hipergrafo puede representar sea netamente mayor [Harel, 1981] [Harel, 1988].

En la disciplina de Ingeniería del Software, se han utilizado los statecharts para mejorar las capacidades de expresión de los DTE (y, consecuentemente, de las MEF). Desde el punto de vista de la representación, los statecharts son equivalentes a DTE. Contiene, al igual que éstos, estados y transiciones entre estados. Adicionalmente, existen tres componentes de los statecharts que son de muy complicada expresión en los DTE. Estos son:

1. Profundidad: Los statecharts pueden agrupar en un macroestado varios estados atómicos. Esta capacidad es útil para modelar el comportamiento de sistemas en los que su funcionalidad básica dependa de una determinada situación en los que se encuentre el sistema o su entorno. Utilizando un ejemplo de [Davis, 1993], el funcionamiento de una lanzadera de misiles tierra-aire es diferente si está en un estado de "prueba" que en un estado de "funcionamiento" (por suerte).
2. Ortogonalidad: La palabra más adecuada sería funcionamiento concurrente. Es útil para modelar, de forma sencilla, situaciones en las que un sistema responda de una forma variada a estímulos en su entorno.
3. Comunicación por difusión: Es complementario a la ortogonalidad. Un evento producido en el entorno se comunica, de forma inmediata, a todos los estados del statechart.

La diagramación de un statechart es similar a la del DTE. Un ejemplo de statechart, mostrando las tres características enunciadas, se muestra en la figura 2.29.

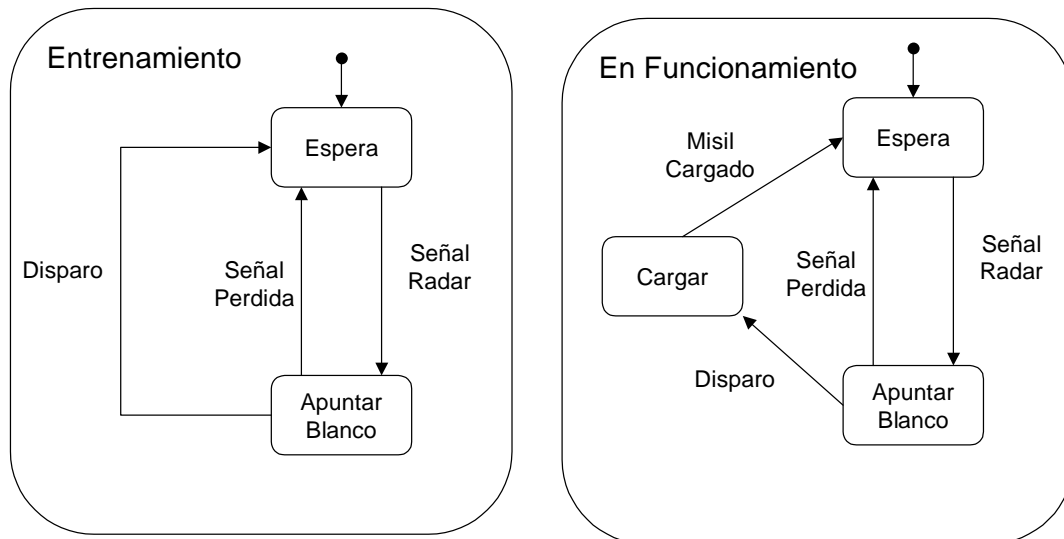


Figura 2.29. Ejemplo de statechart

Es necesario mencionar dos aspectos adicionales de los statecharts. En primer lugar, es posible establecer un predicado que califique cuando un evento genera una transición entre estados. Aunque el predicado podría ser cualquiera, la norma es definirlo en función del estado actual de la máquina. Así, se define un constructor básico, denominado $IN(\langle estado \rangle)$, que devuelve verdadero o falso en función de si la máquina se encuentra, o no, en dicho estado en el momento de la evaluación. Se pueden utilizar los operadores lógicos en los predicados, especificando conjunciones, disyunciones o negaciones del constructor IN .

En segundo lugar, los statecharts incorporan, por su propio formalismo, un mecanismo de abstracción. Al agrupar estados en macroestados, es posible, en un nivel de abstracción alto, estudiar únicamente las transiciones entre macroestados, para, al disminuir en nivel de abstracción, atender a los detalles de cada macroestado. Asimismo es posible, debido a la ortogonalidad, analizar cada macroestado separado de los demás.

2.5.4.3.2. Métodos de Análisis

Los statecharts se han utilizado, básicamente, en el método STATEMATE, desarrollado por i-Logix [i-Logix, 1987] [Harel et al., 1988]. Sin embargo, los statechart podrían utilizarse como sustitutos de los diagramas de transición de estados en todos sus campos de aplicación.

Un formalismo parecido a los statecharts se utiliza en la descripción de la dinámica de los objetos en los métodos Orientados a Objetos [Rumbaugh et al., 1991] [Booch, 1991] [Larman, 1999].

2.5.4.3.3. Valoración

En general, se pueden realizar **los mismos comentarios para el statechart que para el DTE y las MEF**. Aunque el statechart es el modelo dominante en el método STATEMATE [i-Logix, 1987], no se ha

definido un procedimiento detallado de desarrollo de statecharts, ya que se ha primado el aspecto notacional sobre el de construcción del modelo.

2.5.4.4. RESUMEN

La valoración de los criterios para modelos orientados a la dinámica se resume en la tabla 2.8.

Criterios	MEF	DTE	Statechart
<i>Amplitud</i>	Parcial	Parcial	Parcial
<i>Ligaduras Computacionales</i>	No	No	No
<i>Procedimiento de Uso</i>	Ninguna	Ninguna	Ninguna
<i>Selección de Diseño</i>	No Aconseja	No Aconseja	No Aconseja
<i>Derivación de Diseño</i>	Ninguno	Ninguno	Ninguno

Tabla 2.8. Valores de los criterios para los modelos orientados a la dinámica

2.5.5. MODELOS ORIENTADOS A LA INTERACCIÓN

Los modelos incluidos en este grupo tuvieron como finalidad inicial representar la secuencia de acciones que realizaba un determinado agente para resolver una tarea en el dominio del problema. Actualmente, estos modelos representan la interacción entre agentes existentes en el dominio del problema y el futuro sistema software a implementar. Los modelos incluidos en este grupo son los **Casos de Uso** y los **Escenarios**.

2.5.5.1. CASOS DE USO

2.5.5.1.1. Descripción

Un caso de uso describe una forma particular de usar un sistema. El conjunto de todas los casos de uso se denomina Modelo de Casos de Uso, y especifica todas las interacciones posibles entre el sistema y su entorno. Un modelo de casos de uso posee 3 componentes:

1. Actores: Son elementos del dominio del problema que proporcionan información o disparan eventos en el futuro sistema software. Un actor no tiene por qué ser un usuario, sino que mediante este componente puede describir cualquier tipo de dispositivo que suministre información al sistema software como, por ejemplo, un sensor, un panel de control, etc.
2. Entidades Externas: Son dispositivos, almacenes de datos o sistemas software externos al dominio del problema. Se diferencian de los actores en que no pueden iniciar una interacción.
3. Casos de uso: Representa una funcionalidad, habitualmente compleja, que el futuro sistema software debe ofrecer a su entorno.

Un modelo de casos de uso se construye a partir de una colección de casos de uso. Es posible que, entre todas las funciones posibles que debe realizar el sistema, existan algunas que sean casos particulares, ampliaciones o partes de otra función determinada. Para esto, en el modelo de casos de uso se permite la utilización de asociaciones (como USA o EXTIENDE), que precisan la relación existente entre casos de uso.

La diagramación asociada a los casos de uso es muy sencilla, como puede apreciarse en el ejemplo que se muestra en la figura 2.30. Un óvalo representa el caso de uso, mientras que una figura, parecida a un monigote, representa a los actores y, por último, un rectángulo representa a las entidades externas.

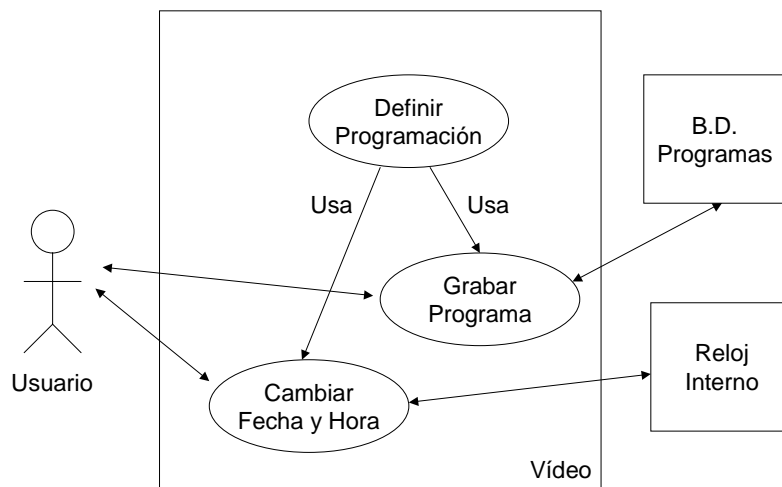


Figura 2.30. Ejemplo de un modelo de casos de uso

2.5.5.1.2. Métodos de Análisis

Los casos de uso, desde su introducción en OOSE [Jacobson, 1992], se utilizan ampliamente en los métodos orientados a objetos, principalmente utilizando la notación UML [Rumbaugh et al., 1998] [Larman, 1999].

2.5.5.1.3. Valoración

Los casos de uso son un formalismo de representación que muchos desarrolladores consideran “natural”. Sin embargo, se deben realizar dos apreciaciones. La primera apreciación se refiere a su formalismo de construcción. Un caso de uso representa una interacción, en cierta medida simple, entre el sistema y su entorno. La identificación de esta interacción **no se lleva a cabo mediante un procedimiento riguroso**, sino que su existencia se supone dada y explicitada en algún tipo de documento (por ejemplo, un documento de requisitos) [Larman, 1999]. Adicionalmente, **no existe un método definido y explícito para generar un diseño a partir de un caso de uso, ni dicho modelo determina o aconseja la utilización de un determinado tipo de diseño.**

La segunda apreciación se refiere a su capacidad de representación. Los casos de uso **únicamente representan interacciones simples** entre el sistema, agentes y entidades externas. Los casos de uso, por lo tanto, no pueden describir aspectos de tipo estructural, detalles de las funciones o restricciones generales que puedan existir en el dominio del problema. Como contrapartida, los casos de uso **no poseen ningún tipo de ligadura computacional**.

2.5.5.2. ESCENARIOS

2.5.5.2.1. Descripción

Un escenario representa una interacción estereotipada entre un agente y el futuro sistema software. Una interacción estereotipada es, simplemente, la secuencia de acciones que se deben llevar siempre a cabo para obtener realizar una función bien determinada con el futuro sistema software, sin atender a los errores o excepciones que puedan surgir.

Los escenarios no poseen un formalismo de representación único. Las dos notaciones mas utilizadas son la textual, que se muestra en la figura 2.31, y la una notación similar a la de los DTE.

Escenario para cambiar la fecha y la hora de un vídeo

1. El usuario pulsa el botón de Fijar Fecha y Hora.
2. El vídeo muestra la fecha actual de forma intermitente.
3. El usuario introduce la fecha actual con 8 dígitos (DDMMAAAA).
4. El vídeo muestra la hora actual de forma intermitente.
5. El usuario introduce la hora actual con 6 dígitos (HHMMSS).
6. El vídeo muestra la fecha actual de forma intermitente.
7. El usuario pulsa el botón Volver.

Figura 2.31. Ejemplo de escenario

2.5.5.2.2. Métodos de Análisis

Los escenarios se utilizan tanto independientemente como en el marco de los métodos orientados a objetos [Larman, 1999]. En este último caso, se considera que un escenario especifica el detalle de un caso de uso.

2.5.5.2.3. Valoración

Los escenarios se construyen “siguiendo el hilo” de las posibles interacciones con el sistema, aunque **no existe un formalismo definido para su desarrollo. Al igual que los casos de uso, no determinan ningún tipo de diseño ni disponen de un procedimiento formalizado para derivar un diseño.**

Desde el punto de vista de la amplitud y ligaduras computacionales, los escenarios **poseen una capacidad de descripción parecida a los casos de uso**, con la salvedad de que detallan con mayor

profundidad la interacción entre el agente y el futuro sistema software. Al igual que los casos de uso, los escenarios **no poseen ligaduras computacionales**.

2.5.5.3. RESUMEN

La valoración de los criterios para los modelos orientados a la interacción se muestran en la tabla 2.9.

Criterios	Casos de Uso	Escenarios
<i>Amplitud</i>	Parcial	Parcial
<i>Ligaduras Computacionales</i>	No	No
<i>Procedimiento de Uso</i>	Ninguna	Ninguna
<i>Selección de Diseño</i>	No Aconseja	No Aconseja
<i>Derivación de Diseño</i>	Ninguno	Ninguno

Tabla 2.9. Valores de los criterios para los modelos orientados a la interacción

2.6. RESUMEN DEL ESTADO DE LOS CONOCIMIENTOS DE LA TAREA DE ANÁLISIS.

El estudio realizado en la sección 2.5 tiene como objetivo determinar en qué grado los modelos conceptuales paradigmáticos son adecuados para realizar la tarea de Análisis. Se ha considerado que un modelo es adecuado cuando permite al ingeniero comprender el problema a resolver sin introducir sesgos hacia una u otra aproximación de desarrollo. Adicionalmente, sería aconsejable que las capacidades expresivas del modelo fuesen lo suficientemente amplias como para representar cualquier tipo de información del dominio del problema, y que el mismo modelo proporcionase consejo acerca de qué aproximación de desarrollo es más ventajosa.

En las secciones anteriores se ha presentado el estudio realizado sobre una diversidad de modelos conceptuales. Las tablas parciales presentadas al final de cada sección se resumen en la tabla 2.10, donde se muestra la valoración de los criterios para los distintos modelos conceptuales. De la tabla 2.10, puede destacarse lo siguiente:

- La capacidad de representación de los modelos conceptuales es, **en la mayoría de los casos, parcial**. La única excepción son los modelos avanzados declarativos, los cuales poseen una amplitud total.
- Los modelos más utilizados en el desarrollo de software, tales como el Diagrama de Flujo de Datos o el Diagrama de Clases, poseen **ligaduras computacionales y prescriben un determinado tipo de diseño**. Ello se ve agravado por el hecho de que dichos modelos son modelos dominantes.

- Solo poseen procedimientos de uso los modelos más utilizados en el desarrollo de software. Los modelos minoritarios se construyen de forma ad-hoc.
- Ningún modelo **determina** un tipo de diseño. En el mejor de los casos, no aconsejan el tipo de diseño a utilizar; en el peor, lo prescriben.
- Solo existen procedimientos de derivación de diseño para unos pocos modelos conceptuales. La información recogida en los modelos debe plasmarse en un diseño de forma heurística en la mayoría de los casos.

Grupo	Subgrupo	Modelo Paradigmático	Amplitud	Ligaduras Computacionales	Procedimiento de Uso	Selección de Diseño	Derivación de Diseño
Orientados al Procedimiento		Miniespecificación	Ninguna	Sí	Ninguno	No Aconseja	Ninguno
		Tabla de Decisión	Parcial	No	Parcial	No Aconseja	Ninguno
		Árbol de Decisión	Parcial	No	Parcial	No Aconseja	Ninguno
		Diccionario de Datos	Ninguna	Sí	Ninguno	No Aconseja	Ninguno
		Fujograma	Parcial	Sí	Ninguno	Prescribe	Total
		Lenguaje de Diseño de Programas	Parcial	Sí	Ninguno	Prescribe	Total
Orientados a la Transformación		Diagrama de Flujo de Datos	Parcial	Sí	Total	Prescribe	Total
		Diagrama de Flujo de Control	Parcial	Sí	Total	Ninguno	Ninguno
		Diagrama de Flujo de Datos / Tiempo Real	Parcial	Sí	Total	Ninguno	Ninguno
		SADT	Parcial	No	Ninguno	Ninguno	Ninguno
		PSL/PSA	Parcial	No	Parcial	Prescribe	Ninguno
Orientados a los datos / objetos	Básicos	Modelo Entidad-Relación	Parcial	No	Total	Ninguno	Total
		Modelo Entidad-Relación Extendido	Parcial	No	Total	Ninguno	Total
		Modelo Funcional del Datos	Parcial	No	Total	Ninguno	Total
	Avanzados	Modelos Avanzados Operativos	Parcial	Sí	Ninguna	Prescribe	Total
		Modelos Avanzados Declarativos	Total	No	Ninguna	No aconseja	Ninguno
	Orientado a Objetos	Diagrama de Clases	Parcial	Si	Total	Prescribe	Total
	Historia de la Vida de las Entidades	Parcial	No	Total	No Aconseja	Parcial	
Orientados a la Dinámica		Máquinas de Estado Finito	Parcial	No	Ninguna	No Aconseja	Ninguno
		Diagrama de Transición de Estados	Parcial	No	Ninguna	No Aconseja	Ninguno
		Statechart	Parcial	No	Ninguna	No Aconseja	Ninguno
Orientados a la Interacción		Casos de Uso	Parcial	No	Ninguna	No Aconseja	Ninguno
		Escenarios	Parcial	No	Ninguna	No Aconseja	Ninguno

Tabla 2.10. Resumen de la valoración de los criterios para los distintos modelos paradigmáticos

En resumen, **el estudio presentado muestra que no existe ningún modelo conceptual orientado al problema que permita describir la totalidad de aspectos del dominio del problema independientemente de la aproximación de desarrollo utilizada, así como identificar el tipo de diseño más adecuado y derivar dicho diseño utilizando procedimientos formalizados.**

Por lo tanto, puede inferirse, adicionalmente, que **ninguna aproximación de desarrollo permite realizar de forma efectiva la tarea de Análisis.** Esto es, debido a que las distintas aproximaciones de desarrollo se caracterizan por los modelos que utilizan, y dado que ningún modelo entre todos los estudiados posee todas las características que sería deseables durante el Análisis, cualquier aproximaciones de desarrollo hereda las carencias de los modelos que utiliza, realizando, por lo tanto, de forma no apropiada la actividad de Análisis.

3. Planteamiento de la Investigación

3.1. DEFINICIÓN DEL PROBLEMA DE INVESTIGACIÓN

Como se ha indicado ya, la presente investigación se enmarca en la actividad de Análisis de las distintas aproximaciones de desarrollo. Para la realización de la actividad de Análisis, acostumbran a utilizarse modelos conceptuales. La utilización de modelos conceptuales es propugnada, sobre todo, por las Aproximaciones Estructurada y Orientada a Objetos.

Como ya se ha mostrado en el Estado de la Cuestión, los modelos conceptuales utilizados por las Aproximaciones Estructurada y Orientada a Objetos poseen varias deficiencias. Estas deficiencias pueden resumirse en:

- Los modelos conceptuales **están fuertemente orientados a definir cómo se implementará el sistema software**, y no a favorecer la comprensión del dominio. Este hecho, denominado **orientación a la solución** [Dori, 1996], ha sido destacado, sobre todo, para la Aproximación Orientada a Objetos por diversos autores [Jalote, 1997] [Høydalsvik et al., 1993] [Northrop, 1997] [Bonfatti et al., 1994].
- **La utilización de un modelo conceptual durante el Análisis determina prácticamente todo el desarrollo posterior.** Esta deficiencia viene motivada porque cada modelo filtra el dominio del problema del usuario, permitiendo registrar únicamente un subconjunto de información. Esta situación ha sido señalada por diversos autores, sobre todo en el marco de la incompatibilidad

de las Aproximaciones Estructurada y Orientada a Objetos [Coleman et al., 1994] [Champeaux et al., 1993] [Wieringa, 1991] [Juristo et al., 2000].

Es necesario indicar, no obstante, que las limitaciones anteriormente mencionadas no son privativas de las Aproximaciones Estructurada y Orientada a Objetos. Éstos ocurren igualmente en otras aproximaciones de desarrollo, tales como la Aproximación de Tiempo Real o la Aproximación de Bases de Datos

El presente Proyecto de Tesis tiene como finalidad redefinir la actividad de Análisis, tal y como se contempla en las aproximaciones de desarrollo actualmente utilizadas, de tal forma que:

1. **Se realice siempre una clara** diferenciación entre los modelos conceptuales y los modelos del sistema en cualquier aproximación de desarrollo.
2. **Exista siempre la posibilidad de cambiar la aproximación de desarrollo** utilizada en un proyecto si se comprueba que ésta es inadecuada.

La redefinición de la actividad de Análisis se llevará a cabo mediante la inserción, de forma previa a la utilización de cualquier actividad de desarrollo, de una tarea de pre-Análisis. La utilización del pre-Análisis permitirá conseguir los objetivos mencionados anteriormente.

3.2. HIPÓTESIS DE TRABAJO

El estudio de los modelos conceptuales, realizado en el capítulo 2, ha demostrado que no existe ningún modelo conceptual que permita describir la totalidad de aspectos del dominio del problema del usuario, identificar el tipo de diseño más adecuado y derivar dicho diseño utilizando procedimientos formalizados.

La inexistencia de un modelo con tales características supone que las distintas aproximaciones de desarrollo son deficientes a la hora de realizar la actividad de Análisis. La razón de dicha deficiencia reside en que cada aproximación de desarrollo se fundamenta en la utilización de un conjunto de modelos, los cuales proporcionan su potencia, pero también sus carencias, a las aproximaciones de desarrollo que los utilizan. Por ello, la inexistencia de un modelo conceptual realmente orientado al problema del usuario, esto es, un modelo que posea amplias capacidades de representación de información, junto con independencia respecto a cualquier tipo de diseño, impide la correcta realización de la actividad de Análisis.

El objetivo del presente trabajo de Tesis consistirá en la definición de un método de análisis para modelización conceptual poliparadigma, o más brevemente, un método de **pre-Análisis.** El pre-Análisis se convierte así en una tarea previa a la aplicación de cualquier aproximación de desarrollo, durante la cual será posible realizar un modelo completo del dominio del problema del usuario, esto es, un modelo que contenga toda la información relevante para llevar a cabo el desarrollo del futuro sistema software. Dicho modelo estará libre, adicionalmente, de cualquier ligadura de diseño y permitirá, por lo

tanto, derivar los modelos utilizados en cualquier aproximación de desarrollo o, más concretamente, los modelos propios de la aproximación de desarrollo que se considere más adecuada, siendo esta adecuación definida formalmente y, por lo tanto, independiente del ingeniero que utilice el método de pre-Análisis.

Para el desarrollo del método de pre-Análisis, se ha definido una Hipótesis General (HG) descompuesta en tres hipótesis de trabajo. La Hipótesis General dice:

HG: Es viable obtener definir un método de pre-análisis **capaz de determinar la aproximación de desarrollo más adecuada** para la necesidad del usuario que el sistemas software pretende atender, y **permite continuar el desarrollo con las aproximaciones tradicionales:** Estructurada, Orientada a Objetos y de Tiempo Real.

Esta hipótesis, demasiado compleja para ser validada en conjunto, se divide en 3 sub-hipótesis que permiten su validación:

SH1: **El método de pre-Análisis es capaz de determinar qué aproximación de desarrollo (estructurada, orientada a objetos, de tiempo real) es la más adecuada para proseguir el proceso de desarrollo.**

El método de pre-Análisis debe poseer algún tipo de procedimiento, técnica o métrica que permita identificar, a partir de la información recogida en los modelos conceptuales, qué aproximación de desarrollo es la más adecuada para continuar con las subsiguientes tareas de desarrollo. Dicho procedimiento, técnica o métrica es necesaria para evitar que factores ajenos al propio problema del usuario influyan en la decisión del ingeniero.

SH2: **El método de pre-Análisis permite derivar los modelos utilizados por las aproximaciones de desarrollo existentes en la actualidad (estructurada, orientada a objetos y de tiempo real).**

El método de pre-Análisis deberá permitir derivar, a partir del conjunto de modelos conceptuales que utiliza, los modelos del sistema prescritos por la aproximación de desarrollo más adecuada. Ello producirá una gran economía en el proceso de desarrollo, al ser posible obtener directamente los modelos exigidos por la aproximación identificada como más adecuada sin necesidad de reanalizar el problema planteado por el usuario, inconveniente que sufren las distintas aproximaciones de desarrollo en la actualidad.

SH3: **El método de pre-Análisis tiene la capacidad de representación suficiente para ser utilizado en las circunstancias en las que serían utilizables las aproximaciones estructuradas, orientadas a objetos o de tiempo real.**

El método de pre-Análisis se utilizará previamente a la aplicación de las aproximaciones de desarrollo estructuradas, orientadas a objetos y de tiempo real. Ello implica que el pre-Análisis debe poseer la suficiente flexibilidad como para adaptarse a un amplio espectro de problemas y dominios.

La flexibilidad en una aproximación de desarrollo viene marcada por la capacidad de representación de los modelos conceptuales que utiliza. Si los modelos conceptuales poseen una amplia capacidad de representación, podrán registrar información de una

variada gama de dominios y problemas. A medida que la capacidad de representación de los modelos disminuye, la aplicabilidad de los mismos se restringe a una serie de dominios particulares o a una clase específica de problemas.

El pre-Análisis deberá, por lo tanto, utilizar modelos conceptuales con capacidades de representación muy amplias, de tal forma que sea posible registrar en dichos modelos información referida a una gran cantidad de dominios y problemas distintos. Los dominios y problemas planteados por los usuarios en los que será posible aplicar el método de pre-Análisis deberán ser, al menos, todos aquellos en los que sería aplicable cualquiera de las aproximaciones de desarrollo citadas anteriormente.

Se podría, exigir que el método de pre-Análisis fuese aplicable en todas las circunstancias, pero dicha exigencia sería gratuita, ya que una validación exhaustiva de la hipótesis, así formulada, sería imposible. No obstante, aún sin poseer ninguna pretensión de totalidad, las aproximaciones estructurada, orientada a objetos y de tiempo real configuran la práctica totalidad de las aproximaciones de desarrollo existentes en la actualidad.

En su conjunto, las tres sub-hipótesis planteadas, en caso de verificarse, permitirán afirmar que el método de pre-Análisis posee grandes ventajas frente a los restantes métodos de Análisis existentes. Las mejoras que introducirá el método de pre-Análisis vendrán dadas, fundamentalmente, por:

- (1) su capacidad de representar la información de cualquier tipo de problema independientemente de cualquier aproximación de desarrollo,
- (2) su capacidad de identificar de modo objetivo la aproximación de desarrollo más adecuada, y
- (3) su capacidad de generar los modelos exigidos por dicha aproximación de desarrollo. Ello permitirá al ingeniero seleccionar objetivamente qué método utilizar en las tareas posteriores de desarrollo e iniciar éstas de forma inmediata.

3.3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

La tarea de pre-Análisis propuesta parte de la idea de que, ante cualquier problema planteado por el usuario, es posible utilizar un proceso de resolución en dos fases: Una primera fase de estudio o **Análisis Orientado al Problema** y una segunda fase de síntesis de una solución, o **Análisis Orientado a la Solución**. Durante el Análisis Orientado al Problema, los objetivos del analista se circunscriben a la comprensión del planteado por el usuario, en el dominio donde se manifieste. Una vez comprendido adecuadamente dicho problema, es posible, mediante un Análisis Orientado a la Solución, establecer una solución adecuada para el mismo. La diferenciación de dos fases en el proceso de resolución de un problema determina el esqueleto de tarea de pre-Análisis propuesta. Más detalladamente, la aproximación de pre-Análisis posee cuatro características fundamentales:

1. Define un proceso formalizado para llevar a cabo la tarea de pre-Análisis.

2. Utiliza un conjunto de formalismos de representación que se han denominado, conjuntamente, Modelo Conceptual Genérico (MCG). La propiedad más importante del MCG es su orientación hacia la necesidad del usuario, lo cual posibilita la utilización de dichos modelos con el propósito exclusivo de comprensión.
3. Permite identificar qué aproximación de desarrollo de software es más adecuada para resolver un del usuario determinado.
4. Es posible derivar, a partir del MCG, los modelos (por ejemplo, diagramas de flujo de datos, diagramas de casos de uso, etc.) utilizados por la aproximación de desarrollo identificada como más adecuada para resolver un problema determinado.

3.3.1. PROCESO DE PRE-ANÁLISIS

El método de pre-Análisis propuesto, denominado **Método de Análisis Orientado la Necesidad** (MAON) refleja en su propia estructura las dos fases de resolución de un problema que se han indicado anteriormente (una fase de estudio del problema y una fase de solución del mismo), tal y como muestra la figura 3.1.

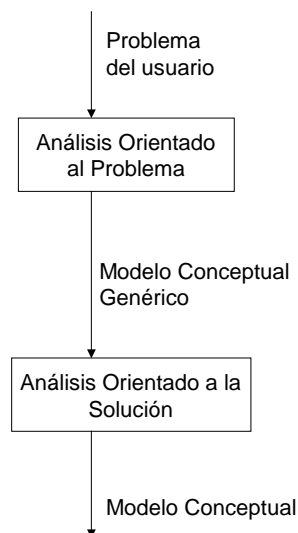


Figura 3.1. Estructura del Método de Análisis Orientado a la Necesidad (MAON)

La primera fase, denominada Análisis Orientado al Problema, **tiene como objetivo comprender la tarea a resolver y sus modos de solución**, y finaliza una vez desarrollado el MCG, el cual representa la información adquirida del dominio del problema del usuario. Dicho MCG es el producto de entrada para la segunda fase, denominada Análisis Orientado a la Solución, la cual **tiene como objetivo identificar qué aproximación de desarrollo es más adecuada** para llevar a cabo el desarrollo del sistema que resolverá el problema del usuario bajo estudio, así como derivar los modelos conceptuales utilizados por dicha aproximación.

Ambas fases anteriormente mencionadas se dividen a su vez en pasos, y dichos pasos en tareas. En su nivel más detallado, el proceso propuesto posee un nivel de descripción adecuado para guiar las acciones que debe realizar a un analista durante la realización de la tarea de pre-Análisis.

3.3.2. MODELO CONCEPTUAL GENÉRICO

El Modelo Conceptual Genérico (MCG) es un conjunto de formalismos de representación **orientados al problema del usuario**. Por orientados al problema del usuario, debe entenderse que:

1. Dichos formalismos no poseen ningún tipo de ligadura computacional, esto es, no introducen ningún aspecto de diseño en sus representaciones.
2. Dichos formalismos poseen una amplia capacidad de representación, de tal modo que es posible registrar en dichos formalismos cualesquiera informaciones que pueden obtenerse en el dominio del problema, no privilegiando ni amortiguando ninguna en especial.

El MCG está constituido por formalismos de modelización distintos, aunque con capacidades de representación similares, esto es, cualquier tipo de información puede registrarse en cualquier formalismo de representación. Los formalismos del MCG son los siguientes:

- **Mapas de Conceptos:** son estructuras de representación de información, consolidados o no, pertenecientes a un determinado dominio. Los mapas de conceptos se derivan, con considerables modificaciones, de los Mapas Conceptuales, inspirados por los trabajos de Ausubel en Teoría del Aprendizaje y Psicología, y más tarde formalizados por Novak [Novak et al., 1988].

Los mapas de conceptos permiten registrar cualquier tipo de información existente en el dominio del problema. Ello se consigue mediante una categorización muy débil de la información. Hasta tal punto son suaves que los mapas de conceptos sólo poseen dos constructores: los conceptos y las asociaciones.

Los conceptos permiten registrar cualquier hecho, elemento, objeto, fenómeno, etc. que existe u ocurre en el dominio del problema del usuario. Esto es, a diferencia de lo que ocurre habitualmente en la Ingeniería del Software, donde “concepto” se utiliza con el significado de “dato” en muchas ocasiones, en los Mapas de Conceptos los “conceptos” hacen referencia tanto a “conceptos estáticos”, como son los datos, reglas o hechos, como a los “conceptos dinámicos”, como los procesos, eventos, etc.

Las asociaciones permiten registrar las interacciones que ocurren entre los distintos elementos u objetos del dominio del problema del usuario. Una asociación, en el Mapa de Conceptos, es simplemente algo que se afirma de un concepto.

Los mapas de conceptos se utilizan durante distintos momentos de la fase de Análisis Orientado al Problema **con el nivel de refinamiento exigido por cada paso del proceso propuesto.**

- **Diccionarios:** son esquemas de representación de información en forma tabular. Los diccionarios poseen un conjunto de columnas predefinidas, las cuales definen qué tipos de información se deben registrar en ellas. Se pueden distinguir dos tipos de diccionarios principales:
 - **Diccionario de Identificación:** Este Diccionario se utiliza durante un primer momento del Análisis Orientado al Problema. El Diccionario de Identificación, también denominado **Glosario**, permite registrar los conceptos más relevantes del dominio con miras a un refinamiento posterior.
 - **Diccionario de Descripción:** Su finalidad es registrar información refinada acerca del dominio del problema del usuario, con un nivel de formalidad mucho mayor que el Diccionario de Identificación. El Diccionario de Descripción es especialmente relevante en MAON, ya que es el formalismo utilizado durante la fase de Análisis Orientado a Solución, para identificar el modelo conceptual idóneo y derivar el modelo conceptual seleccionado para la necesidad planteada por el usuario.
- **Texto Narrativo:** El Texto Narrativo es un formalismo de representación que utiliza el lenguaje natural. Se utiliza para transcribir la información recogida tanto en el Mapa de Conceptos como en los Diccionarios de Identificación y Descripción.

El Texto Narrativo tiene como finalidad validar la información recogida en el MCG con clientes y usuarios. Evidentemente, poder obtener una descripción textual de la información recogida en formalismos de representación complejos es importante para lograr una adecuada comprensión de dicha información por los clientes y usuarios.

3.3.3. IDENTIFICACIÓN DE LA APROXIMACIÓN DE DESARROLLO MÁS ADECUADA

Una vez finalizada la fase de Análisis Orientado al Problema, se habrán obtenido los distintos formalismos de representación que conforman el MCG. A partir de la información recogida en el MCG, se aplica una técnica que se ha denominado **Técnica de Identificación del Modelo Conceptual Idóneo** (Técnica IMCI), de la cual se obtiene como salida el modelo o modelos (por ejemplo, un diagrama de flujo de datos, o un diagrama de clases) que mejor pueden registrar la información recogida en el MCG.

Para poder utilizar la Técnica IMCI, el formalismo del MCG más importante es el Diccionario de Descripción. Sobre el Diccionario de Descripción se aplica un procedimiento, denominado **Procedimiento de Interpretación**, el cual es gran parte algorítmico; esto es, su aplicación no depende del analista que lo utiliza.

Dicho procedimiento, que forma parte de la Técnica IMCI, permite asignar, a cada modelo utilizado por las aproximaciones de desarrollo (Estructurada, Orientada a Objetos, Tiempo Real, etc.) un valor numérico que representa cuánto de adecuado es cada modelo para representar la información recogida en el MCG. El modelo idóneo será, naturalmente, aquel cuya capacidad de representación sea mayor. Del modelo idóneo se deriva la aproximación de desarrollo idónea. La aproximación de desarrollo idónea será aquella que posea como modelo dominante el modelo idóneo identificado.

3.3.4. DERIVACIÓN DE LOS MODELOS CONCEPTUALES

Una vez identificado el modelo idóneo, así como la aproximación de desarrollo más adecuada, solo resta derivar, a partir del MCG, dicho modelo idóneo.

La derivación del modelo idóneo se realiza utilizando la **Técnica de Derivación del Modelo Conceptual Seleccionado** (Técnica DMCS). Esta técnica toma como producto de entrada el Diccionario de Descripción, sobre el que se han realizado una serie de transformaciones en el paso anterior, Identificación de la Aproximación de Desarrollo más Adecuada.

A partir del Diccionario de Descripción, y utilizado un **Procedimiento de Derivación** totalmente algorítmico, se obtiene el modelo idóneo. Este modelo idóneo es el punto final de MAON, y representa el producto de entrada para comenzar con el proceso de desarrollo tal y como se prescribe en la aproximación de desarrollo más adecuada.

3.3.5. ESTRUCTURA DE LA EXPOSICIÓN DE LA RESOLUCIÓN

Los siguientes capítulos del presente trabajo de Tesis estarán dedicados a describir, en detalle, la solución propuesta. En el capítulo 4, se realizará una introducción general de MAON, el cual servirá de base para la exposición, más detallada, que se realizará en los siguientes capítulos.

El capítulo 5 estará dedicado a la primera fase de MAON, el Análisis Orientado al Problema. En este capítulo se describirán en detalle los pasos que componen este proceso, así como las tareas en que se descomponen cada uno de estos pasos. Así mismo se describirán detalladamente los formalismos de representación que forman parte del MCG y las relaciones que existen entre los mismos.

El capítulo 6 se centrará en la segunda fase de MAON, el Análisis Orientado a la Solución. En este capítulo se detallarán los pasos de Identificación del Modelo Conceptual Idóneo y Derivación del Modelo Conceptual Seleccionado. Se prestará especial atención en este capítulo a los procedimientos de identificación y selección, debido a que son especialmente relevantes en la utilización práctica de MAON.

El capítulo 7 estará dedicado a realizar la validación de las hipótesis expuestas en el presente capítulo de Planteamiento de la Investigación.

Finalmente, el capítulo 8 tendrá como objetivo presentar la solución de un caso práctico, realizado en un proyecto de desarrollo real, el cual demostrará la aplicabilidad de MAON en la práctica.

PARTE II

RESOLUCIÓN

4. Visión General de MAON

Como se ha indicado en el Capítulo 3, el objetivo del presente trabajo de Tesis es definir un modelo conceptual orientado a la necesidad que permita describir la totalidad de aspectos del dominio de un problema planteado por el usuario, así como identificar el tipo de diseño más adecuado y derivar dicho diseño utilizando procedimientos formalizados. Dicho modelo, así como los procedimientos asociados, configurarán una tarea, denominada pre-Análisis, que se realizará previamente a la actividad de Análisis, tal y como se considera en las distintas aproximaciones de desarrollo.

El objetivo anteriormente mencionado se ha logrado mediante la definición de un método de pre-Análisis que consta de cuatro componentes principales:

- Un proceso detallado para la realización de la tarea de pre-Análisis, denominado **Método de Análisis Orientado a la Necesidad** (MAON). MAON define un conjunto de pasos que prescriben exhaustivamente las acciones a realizar por el ingeniero durante la tarea de pre-Análisis, así como las técnicas a emplear y los productos a generar en cada paso del proceso.
- Un conjunto de formalismos de representación, denominados en su totalidad **Modelo Conceptual Genérico** (MCG). El MCG está formado por cuatro formalismos, o modelos, distintos:
 - Mapa de Conceptos
 - Diccionarios
 - Texto Narrativo
- Una **técnica de identificación del Modelo Conceptual Idóneo** (Técnica IMCI). La Técnica IMCI permite identificar, a partir de la información recogida en el MCG, qué modelo conceptual

es más adecuado, y por ello qué aproximación de desarrollo debe utilizarse para proseguir con el desarrollo del futuro producto software.

- Una **técnica de derivación del modelo conceptual seleccionado** (Técnica DMCS). La Técnica DMCS permite derivar, a partir del MCG, el modelo conceptual identificado como más adecuado para realizar el desarrollo del futuro sistema software.

En el presente capítulo se presenta una visión general de la solución propuesta, focalizándose en el proceso. Dicho proceso es altamente relevante ya que determina la utilización del MCG y la aplicación de las Técnicas IMCI y DMCS. No obstante, antes de proceder a su exposición, es conveniente enunciar algunos hechos y evidencias que fundan teóricamente MAON. A ello se dedicará la sección 4.1. Una vez detallados los fundamentos teóricos de MAON, la sección 4.2 se dedicará a describir los pasos que conforman el proceso propuesto.

4.1. BASES COGNITIVAS DEL PROCESO PROPUESTO

La actividad de Análisis en el desarrollo de software es un tipo especial de proceso de solución de problemas [Sutcliffe et al., 1992]. Ello implica que cualquier proceso de Análisis debe adaptarse a las estrategias cognitivas humanas, las cuales operan a un nivel inconsciente durante la realización de cualquier proceso de solución de problemas.

Existe una abundante literatura acerca de la resolución de problemas. En el ámbito de desarrollo de software, existe igualmente un buen número de estudios acerca del tema, pero lamentablemente centrados en actividades como la programación o el diseño [Jeffries et al., 1981] [Adelson et al., 1985] [Guindon et al., 1987] [Visser, 1987] [Adelson et al., 1988] [Rist, 1991] [Davies, 1991]. La lista de trabajos que tratan la tarea de Análisis [Vitalari et al., 1983] [Fickas et al., 1988] [Guindon et al., 1988] [Sutcliffe et al., 1992] es mucho más reducida. No obstante lo reducido de los estudios existentes, se han identificado varias características relevantes de la actividad de Análisis:

- Característica 1. **La actividad de Análisis consta de dos pasos bien diferenciados** [Sutcliffe et al., 1992]: un primer paso consistente en un acercamiento y estructuración iniciales del dominio del problema planteado, y un segundo paso en el que se realiza una estructuración más detallada que permite razonar sobre dicho dominio.
- Característica 2. **La efectividad del Análisis reside, en gran medida, en la formación de modelos mentales del dominio del problema planteado**, los cuales se plasman físicamente en modelos conceptuales [Gentner et al, 1983] [Adelson et al., 1985] [Sutcliffe et al., 1992].

La formación de modelos mentales durante el análisis de problemas también es estudiada por la Psicología de la Educación. Ello no es extraño, en la medida

en que analizar un dominio es similar a adquirir pericia en una determinada materia mediante autoestudio, sin la guía que proporciona por un profesor.

Entre los resultados más importantes en la Psicología de la Educación, en lo tocante a la formación de modelos mentales, debe destacarse lo que se conoce como "*Aprendizaje Significativo*". El aprendizaje significativo consiste en promover la asimilación de nuevos conceptos por parte de un alumno basándose en los conceptos previamente existentes en su estructura cognitiva [Novak et al., 1988]. Del aprendizaje significativo nació una corriente que se denomina Aprendizaje por Descubrimiento. Esta corriente, de típica aplicación en las ciencias naturales, implica que el alumno debe aprender las materias no mediante su estudio teórico y su asimilación directa, sino mediante un proceso de descubrimiento caracterizado por la adquisición inicial de conceptos y su organización e integración posteriores [Ontoria et al., 1996].

El aprendizaje por descubrimiento posee muchas de las facetas de la actividad de Análisis. De hecho, ambos implican: (1) una investigación de un dominio de la realidad bien definido; (2) una asimilación de conceptos inestructurados de dicho dominio y (3) dar forma y organizar la diversidad de conceptos adquiridos en un todo coherente.

Una de las herramientas más poderosas utilizadas por los métodos de enseñanza por descubrimiento son los mapas de conceptos. Los mapas de conceptos son modelos de tipo conceptual que permiten registrar aspectos de un dominio del mundo real y efectuar asociaciones arbitrarias entre ellos. Dichos modelos han proporcionado buenos resultados en la práctica para describir los elementos u asociaciones existentes en dominios muy variados [Novak et al., 1988].

La característica más importante de los mapas de conceptos es la baja categorización que realizan de los elementos y asociaciones existentes en un dominio. Al contrario que los modelos conceptuales utilizados en el desarrollo de software, los cuales tienden a filtrar la información obtenida [Kop et al., 1998], los mapas conceptuales no realizan ninguna suposición sobre el dominio, permitiendo registrar información de modo genérico.

Característica 3. **La actividad de Análisis se caracteriza por la utilización de múltiples estrategias distintas de resolución de problemas**, por lo que la adherencia a un método prescriptivo penaliza la efectividad del Análisis [Sutcliffe et al., 1992].

Las evidencias indicadas anteriormente parecen lo suficientemente sólidas como para ser tenidas en cuenta durante la definición de un proceso de pre-Análisis. Por ello, el proceso definido para MAON se ha confeccionado de tal forma que:

- Distingue dos pasos fundamentales durante el pre-Análisis, tal y como indica la característica 1. En términos de MAON, dichos pasos se han denominado:
 - **“Análisis Preliminar”**, cuyo objetivo es proporcionar al ingeniero el vocabulario básico del dominio del problema del usuario. De forma paralela a la obtención de vocabulario, el ingeniero realiza una estructuración de los distintos elementos de dicho dominio.
 - **“Análisis Exhaustivo”**, cuyo objetivo es comprender con precisión el dominio del problema del usuario.
- Los modelos conceptuales que serán utilizados durante los pasos de Análisis Preliminar y Análisis Exhaustivo, deberán ser aquellos que permitan el Aprendizaje por Descubrimiento, tal y como indica la característica 2. Este tipo de modelos conceptuales, entre los que destaca el **mapa de conceptos**, permiten una fácil traslación hacia y desde los modelos mentales. Los mapas de conceptos han sido utilizados con éxito en la educación, así como en otros campos científicos e industriales [Novak et al., 1988], por lo que, previsiblemente, también serán útiles en el desarrollo de software.
- **El Análisis Preliminar y el Análisis Exhaustivo no se realizarán de forma secuencial, sino oportunista**, tal y como indica la característica 3. La variedad de procedimientos de resolución de problemas aplicables durante la actividad de Análisis no permiten, en general, una aplicación sistemática de cualquier tipo de proceso, por lo que MAON deberá permitir que el ingeniero escoja la estrategia de resolución que considere más apropiada.

El proceso propuesto para MAON, el cual posee las características indicadas en la presente sección, se describe a continuación.

4.2. PROCESO DE PRE-ANÁLISIS PROPUESTO

Debido a que el pre-Análisis se configura como una tarea previa a la actividad de Análisis, el proceso propuesto debe contemplar y guiar una diversidad de acciones:

- La adquisición de información del dominio del problema del usuario.
- El registro y consolidación de la información utilizando los distintos formalismos de representación que configuran el MCG.
- El refinamiento de la información obtenida, de manera que el dominio del problema del usuario esté fielmente representado en el MCG.
- La comprobación de que la información obtenida es correcta y completa.
- La identificación de los modelos conceptuales que mejor pueden registrar y representar la información obtenida y registrada en el MCG.

- La identificación de los métodos de desarrollo que utilizan dichos modelos conceptuales.
- La transformación del MCG al modelo conceptual seleccionado para proseguir el desarrollo de software.

El proceso de pre-Análisis define, como puede inferirse a partir de las acciones anteriormente indicadas, **una actividad de Análisis completa**, previa a la actividad de Análisis prescrita por los distintos métodos de desarrollo. Ésta es la razón principal del nombre de “pre-Análisis” que se ha asignado a la tarea que propone la presente investigación. Obviamente, y con la finalidad de evitar duplicaciones inútiles, en el proceso propuesto no se definen técnicas y procedimientos que suplanten a técnicas y procedimientos existentes de probada valía. En concreto, el proceso propuesto no contempla:

- Técnicas específicas de educación. Actualmente, existe un gran número de técnicas de educación, como las entrevistas, escenarios, análisis de protocolos, *Brainstorming*, etc. Así pues, en el proceso propuesto no se definen técnicas específicas de educación debido a que las técnicas anteriormente indicadas, así como otras técnicas existentes, pueden utilizarse directamente en la educación que alimenta la tarea de pre-Análisis.
- Técnicas específicas de validación. Al igual que ocurre con las técnicas de educación, existen diversas técnicas de validación como el prototipado o las listas de comprobación. Dichas técnicas, así como otras técnicas existentes, son igualmente compatibles con la tarea de pre-Análisis.

El proceso de pre-Análisis propuesto, denominado **MAON**, como ya se ha indicado anteriormente, está formado por dos fases principales: Una primera fase denominada Análisis Orientado al Problema y una segunda fase denominada Análisis Orientado a la Solución. El propósito de la fase de Análisis Orientado al Problema es **comprender el problema planteado por el usuario y sus modos de solución**, de acorde con las características descritas anteriormente. Esta primera fase finaliza una vez desarrollado el MCG, el cual representa la información adquirida acerca del dominio del problema del usuario. Por el contrario, la fase de Análisis Orientado a la Solución, tiene como objetivo **identificar qué aproximación de desarrollo es más adecuada** para resolver el problema del usuario bajo estudio, así como **derivar los modelos conceptuales** utilizados por la aproximación seleccionada. Ambas fases se dividen en cuatro pasos principales, tal y como muestra la figura 4.1.

Cada uno de los pasos de la figura 4.1 contiene varias tareas, que en su conjunto definen la totalidad de acciones a realizar durante la tarea de pre-Análisis. Un estudio en detalle de los pasos y tareas de MAON se postergará a los capítulos 5 y 6 del presente trabajo de Tesis, dedicándose la presente sección únicamente a introducir el proceso propuesto.

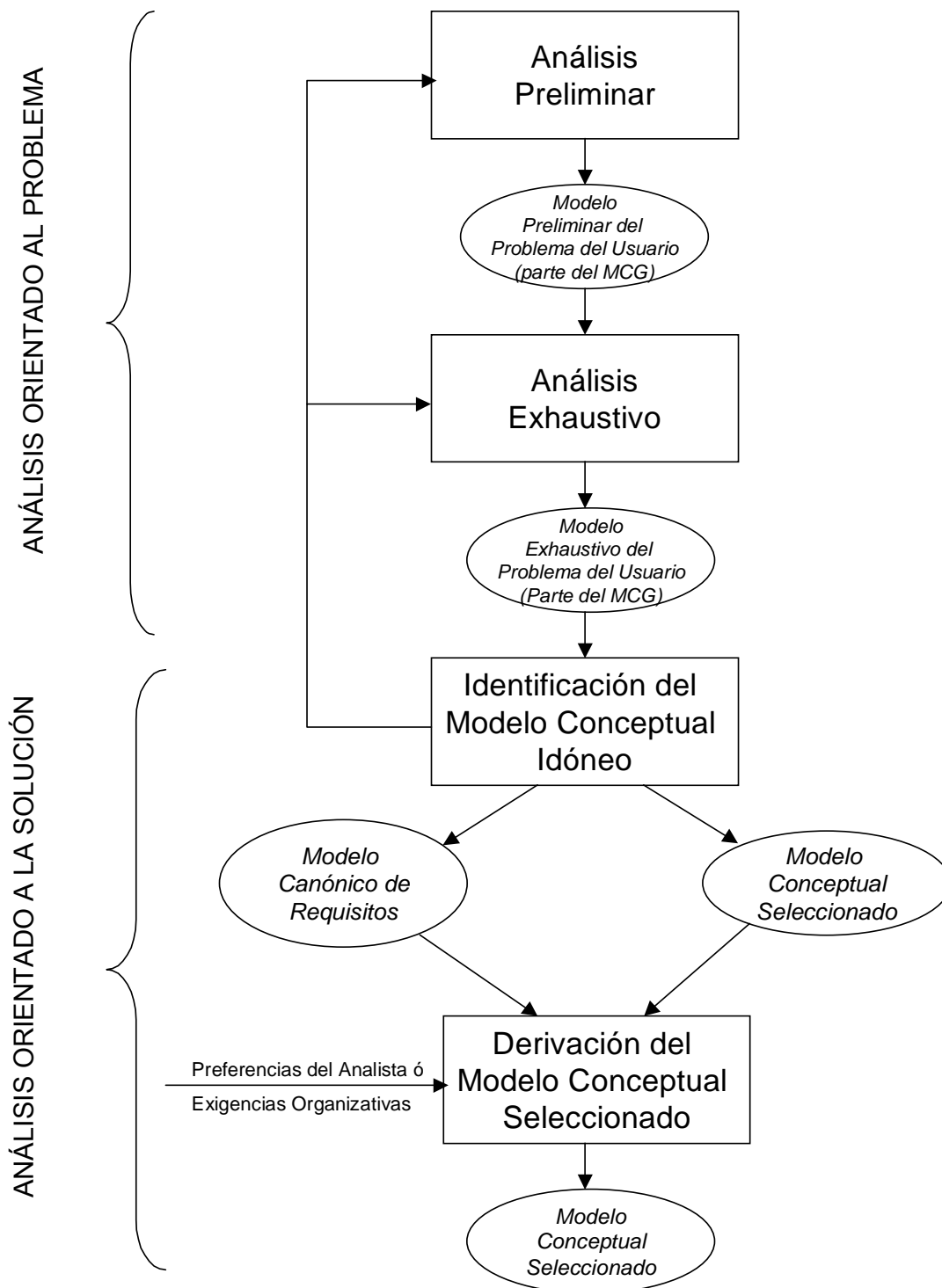


Figura 4.1. Esquema del proceso de pre-Análisis propuesto

4.2.1. ANÁLISIS PRELIMINAR

El Análisis Preliminar es el primer paso del proceso propuesto. La finalidad de este paso es realizar un estudio de aquellos aspectos más relevantes del dominio del problema del usuario con el objetivo de crear el **Modelo Preliminar del Problema del Usuario**.

Durante el Análisis Preliminar, el ingeniero adquiere información acerca del dominio del problema planteado por el usuario. Esta información consta de: (1) los elementos más importantes del dominio, (2) la descripción de cada uno de dichos elementos y (3) las asociaciones más evidentes que ligan los distintos elementos entre sí. Los distintos elementos y las asociaciones que establecen entre sí poseen, durante este paso, una naturaleza principalmente estática, aunque nada impide que aparezcan, igualmente, elementos y asociaciones dinámicas. Para identificar la información indicada anteriormente, el Análisis Preliminar se divide en tres tareas, mostradas en la figura 4.2.

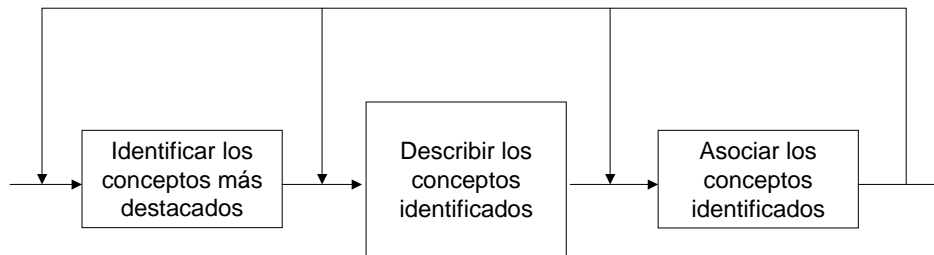


Figura 4.2. Secuencia de las tareas durante el Análisis Preliminar

El producto de salida del Análisis preliminar es el Modelo Preliminar del Problema del Usuario. Este modelo está formado por tres formalismos de representación del MCG. En concreto, los formalismos utilizados son los siguientes:

- Mapa de Conceptos
- Diccionario de Identificación
- Texto Narrativo

Una vez finalizado el Análisis Preliminar, el ingeniero habrá logrado identificar los aspectos más relevantes del dominio del problema del usuario. A modo de analogía, el resultado del Análisis Preliminar es similar a la primera fase de la composición de un puzzle. Cuando todas las piezas del puzzle están sueltas, dispersas, no es factible abordar el montaje del puzzle como un todo. En su lugar, deben seleccionarse las piezas más sencillas de ubicar, que serán probablemente aquellas que pertenecen a los bordes o a los motivos más destacados. Una vez ubicadas estas piezas en el marco, o en determinadas zonas del puzzle, se puede iniciar la composición, más complicada, del resto de las piezas.

La información recogida durante el Análisis Preliminar es similar a las primeras piezas identificadas del puzzle. Esta información conforma una especie de mapa o esquema del dominio, el cual permite iniciar, posteriormente, un estudio más detallado del mismo. Dicho estudio se realiza durante el paso de Análisis Exhaustivo.

4.2.2. ANÁLISIS EXHAUSTIVO

El Análisis Exhaustivo es el segundo paso del proceso de pre-Análisis propuesto. Este paso tiene como finalidad profundizar en el estudio del problema del usuario, hasta conseguir una descripción detallada del mismo. Dicha descripción detallada configura el **Modelo Exhaustivo del Problema del Usuario**.

Para obtener una descripción detallada del dominio del problema, el ingeniero debe realizar cuatro actividades básicas: (1) clasificar los elementos en grupos; (2) clasificar los elementos de forma jerárquica; (3) identificar las propiedades que posee, o las restricciones que debe cumplir cada elemento y (4) identificar cómo se comporta o evoluciona el dominio mediante la identificación de los aspectos dinámicos del problema del usuario. Dichas actividades configuran las tareas de la actividad de Análisis Exhaustivo, las cuales se muestran en la figura 4.3.

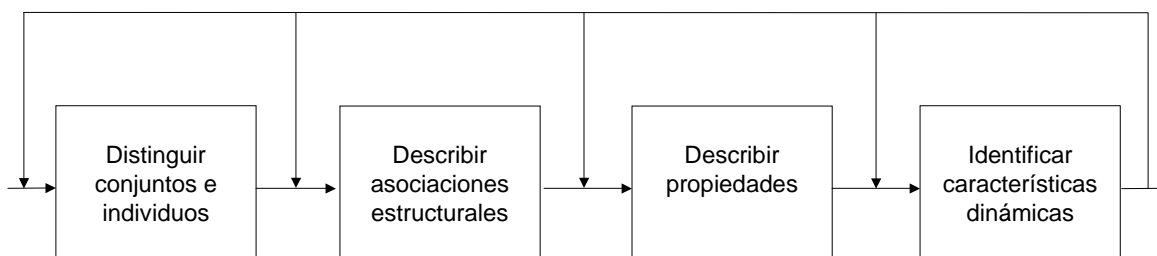


Figura 4.3. Secuencia de las tareas del Análisis Exhaustivo

Una vez finalizado el Análisis Exhaustivo, se obtiene el Modelo Exhaustivo del Problema del Usuario. Este modelo está formado por los siguientes formalismos de representación del MCG:

- Mapa de Conceptos
- Diccionario de Descripción

Una vez confeccionado el Modelo Exhaustivo del Problema del Usuario se da por terminado el análisis propiamente dicho. El resultado del Análisis Exhaustivo es equivalente a la terminación de un puzzle, continuando con la analogía iniciada anteriormente. El Análisis Preliminar proporciona los conceptos más importantes, similares a las piezas más destacadas del puzzle, a partir de los cuales es posible comprender y registrar información más refinada y detallada. Esta nueva información proporciona una visión completa del problema bajo estudio y de sus modos de solución, de igual forma que el motivo del puzzle aparece cada vez más visible a medida que el puzzle se completa.

El paso siguiente estará plenamente orientado a la solución y consistirá en la identificación del modelo más adecuado para proseguir con el subsiguiente desarrollo del producto software.

4.2.3. IDENTIFICACIÓN DEL MODELO CONCEPTUAL IDÓNEO

El paso de Identificación del Modelo Conceptual Idóneo tiene como finalidad seleccionar el modelo (propio de las distintas aproximaciones de desarrollo, tales como la Aproximación Estructurada, o la Aproximación Orientada a Objetos), más adecuado para registrar y representar la información del MCG.

Para realizar tal identificación, en el presente sub-paso se utiliza la **Técnica de Identificación del Modelo Conceptual Idóneo** (Técnica IMCI). Esta técnica toma como producto de entrada el Diccionario de Descripción, el cual, como ya se ha indicado, es parte del Modelo Exhaustivo del Problema del Usuario.

La Técnica IMCI se basa en un **procedimiento de interpretación**. El procedimiento de interpretación pone en relación la información registrada en el Diccionario de Descripción con un **catálogo de constructores de modelos**. Dicho catálogo, que se ha denominado **Modelo Canónico**, está inspirado en el trabajo de [Davis et al., 1997], aunque ha sido profundamente modificado por el autor del presente trabajo de Tesis, tal y como se indica en el anexo A. Una de las características más importantes del procedimiento de interpretación es su formulación algorítmica. El procedimiento de interpretación propuesto en este Trabajo es totalmente automatizable, aunque las múltiples combinaciones que se pueden producir entre los distintos elementos constructores de modelo limitan las posibilidades de interpretación automática.

Al poner en relación el Diccionario de Descripción con el Modelo Canónico, lo que se realiza es una correlación entre el Diccionario de Descripción y los constructores de los modelos utilizados por las distintas aproximaciones de desarrollo (Estructurada, Orientada a Objetos, etc.). Esta correlación permite asignar, a cada tipo de información registrado en el Diccionario de Descripción, un determinado constructor de un modelo (por ejemplo, mediante un proceso o una clase).

Una vez que se han identificado los constructores asociados a los distintos tipos de información recogidos en el Diccionario de Descripción, es posible identificar el Modelo Conceptual Idóneo mediante el uso de una métrica definida en este trabajo y que se ha denominado **Métrica de Adecuación** para modelos. La Métrica de Adecuación proporciona un valor numérico de adecuación para cada modelo (diagrama de clases, diagrama de flujo de datos, etc.). El modelo que posea el valor de la métrica más alto resulta ser el modelo idóneo.

De la misma forma que existe una Métrica de Adecuación para modelos, también se ha definido una **Métrica de Adecuación para métodos**. El resultado de la Métrica de Adecuación de métodos es similar al de la Métrica de Adecuación de modelos, con la diferencia de que indica el método, y no el modelo, idóneo.

Como subproducto del paso de Identificación del Modelo Conceptual Idóneo, se obtiene el **Modelo Canónico de Requisitos**. Este Modelo Canónico de Requisitos se obtiene por el etiquetado de cada uno de los tipos de información del Diccionario de Descripción con los constructores propios de los modelos utilizados por las distintas aproximaciones de desarrollo.

4.2.4. DERIVACIÓN DEL MODELO CONCEPTUAL SELECCIONADO

El paso de Derivación del Modelo Conceptual Seleccionado tiene como finalidad obtener el modelo conceptual que se ha identificado como idóneo.

No obstante, en muchas ocasiones las exigencias organizativas, o las propias preferencias del ingeniero, pueden no ser compatibles con el modelo conceptual identificado como idóneo. En dicho caso, el presente paso de MAON no tiene como finalidad derivar el Modelo Conceptual Idóneo, sino el seleccionado por el ingeniero. A ello debe su nombre el presente paso de MAON.

La derivación del modelo idóneo se realiza utilizando la **Técnica de Derivación del Modelo Conceptual Seleccionado** (Técnica DMCS). Esta técnica toma como producto de entrada el Modelo Canónico de Requisitos. Dicho modelo que, como ya se ha indicado, se deriva del Diccionario de Descripción, se utiliza como producto de entrada para el **procedimiento de derivación**. El procedimiento de derivación está formado por un algoritmo, por lo tanto completamente independiente del ingeniero que lo utiliza, que permite obtener el Modelo Conceptual Idóneo a partir del Modelo Canónico de Requisitos.

Para obtener el Modelo Conceptual Idóneo, el procedimiento de derivación utiliza una serie de tablas que permiten obtener fragmentos del Modelo Conceptual Idóneo a partir del Modelo Canónico de Requisitos. La integración de todos los fragmentos da como resultado el modelo conceptual deseado.

Nótese, no obstante, que el procedimiento de derivación no tiene por qué utilizarse únicamente para obtener el Modelo Conceptual Idóneo. En el caso de que el ingeniero haya preferido, por cualquier razón, obtener un modelo conceptual distinto, siempre puede aplicar el procedimiento de derivación para obtener dicho modelo.

Esta característica del procedimiento de derivación posee una utilidad adicional. De la misma forma que existe, en el paso de Identificación del Modelo Conceptual Idóneo, una Métrica de Adecuación de métodos, también es posible utilizar el procedimiento de derivación para obtener no sólo un modelo, **sino también un conjunto de modelos. De hecho, es posible obtener todos los modelos prescritos por una aproximación de desarrollo determinada**. Así, por ejemplo, a partir de un único Modelo Canónico de Requisitos es posible obtener todos los modelos utilizados por las Aproximaciones Estructuradas simultáneamente (un diagrama de flujo de datos, un diagrama entidad-relación, un Diccionario de datos y una miniespecificación). Ello también puede realizarse para las Aproximaciones Orientadas a Objetos, Aproximaciones de Tiempo Real y Aproximaciones de Bases de Datos.

5. Análisis Orientado al Problema

El presente capítulo tiene como finalidad profundizar en la fase de Análisis Orientado al Problema del Usuario de MAON. Como se ha indicado en el Capítulo 4, esta fase tiene como objetivo comprender el problema planteado por un usuario, y finaliza una vez desarrollado el MCG, el cual representa la información adquirida del dominio del problema. Para desarrollar el MCG, MAON contempla dos etapas procedimentales: el **Análisis Preliminar** y el **Análisis Exhaustivo**, que se detallan en las secciones 5.1 y 5.2. La descripción se ajustará al siguiente esquema:

- Descripción, en la que se indican las tareas que se deben realizar para completar cada etapa.
- Técnicas utilizadas, donde se describen aquellas técnicas, recetas o procedimientos objetivos (esto es, no sometidos a interpretación) que facilitan la consecución de los objetivos de cada etapa.
- Productos de entrada, en la que se describen los productos de pasos anteriores que se utilizan en cada etapa.
- Productos de salida, donde se describen los productos generados en cada etapa.
- Excepciones, en la que se describen las posibles alteraciones del curso normal de cada etapa, en especial las condiciones de retorno a una etapa previa.

Las etapas del Análisis Preliminar y el Análisis Exhaustivo, conjuntamente, configuran el núcleo del Método de Análisis Orientado a la Necesidad (MAON), en la medida en que, tras su finalización, se habrá

obtenido el Modelo Exhaustivo del Problema del Usuario. La obtención de este modelo es el hito necesario para la aplicación de las Técnicas de Identificación del Modelo Conceptual Idóneo (IMCI) y Derivación del Modelo Conceptual Seleccionado (DMCS).

Debido a lo anterior, para realizar una exposición completa de los pasos de Análisis Preliminar y Análisis Exhaustivo, es necesario detallar también de manera pormenorizada los distintos formalismos de representación que configuran el MCG. Dicha descripción se realizará en las secciones 5.3, 5.4 y 5.5. La sección 5.3 se dedicará a presentar los conceptos comunes a todos los formalismos de representación del MCG. La sección 5.4 abordará concretamente cada uno de los formalismos que conforman el MCG por separado y, finalmente, la sección 5.5 se dedicará a exponer las relaciones existentes entre los distintos formalismos de representación.

5.1. ANÁLISIS PRELIMINAR

El Análisis Preliminar es el primer paso de MAON. La finalidad del Análisis Preliminar es identificar y registrar la información más sobresaliente del dominio del problema. El registro y representación de dicha información conformará un primer modelo del problema, denominado **Modelo preliminar del problema**.

5.1.1. DESCRIPCIÓN

Durante la realización del Análisis Preliminar, el ingeniero se enfrenta al problema en estado puro, sin ningún tipo de segmentación u organización previa; esto es, el problema a resolver es, en este momento, un todo informe e indiferenciado, al que el ingeniero deberá dar un orden y organización lógicas.

El paso del Análisis Preliminar es, en cierta medida, como la confección de un mapa de una región desconocida. En el terreno, el cartógrafo no conoce más ríos, valles o mesetas que las que tiene directamente ante sus ojos. Para hacerse una idea de conjunto, el cartógrafo deberá acudir a aquellos accidentes que están siempre visibles, esto es, las cotas altas del terreno. Allí fijará sus referencias geodésicas para, más tarde, ir cartografiando el terreno poco a poco.

El ingeniero, durante el Análisis Preliminar, actúa como el cartógrafo. Para hacerse una idea general del dominio del problema, el ingeniero fija sus "referencias geodésicas" en las cotas altas del dominio del problema, esto es, en aquellos aspectos o informaciones sobresalientes, que destacan por su importancia frente a otras que, por su menor relevancia, quedan ocultas. La identificación de los aspectos salientes permite al ingeniero alcanzar un primer entendimiento del problema, probablemente provisional y seguramente superficial, pero que le faculta para seguir avanzado y desbrozando el terreno.

La actuación del ingeniero durante el Análisis Preliminar, por lo tanto, consiste en un ciclo iterativo de identificación-descripción-organización de información. Más concretamente, el Análisis Preliminar consta de tres tareas procedimentales, ya referenciadas en el Capítulo 4, y mostrados en la figura 5.1.

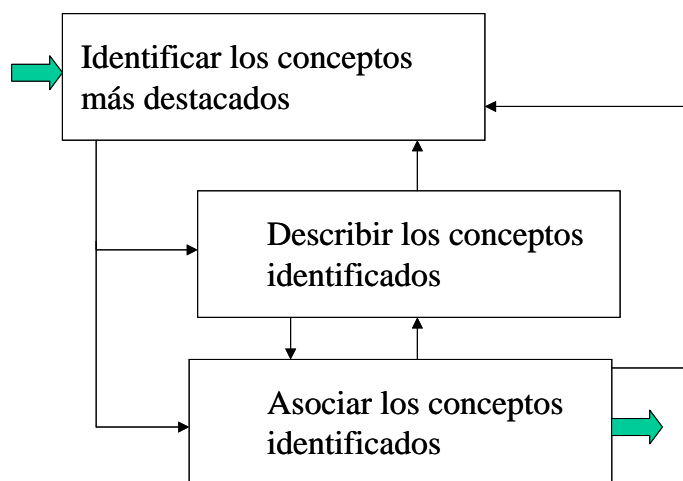


Figura 5.1. Secuencia de tareas del Análisis Preliminar

Las tres tareas mostradas en la figura 5.1 no se realizan de forma secuencial, sino oportunista; esto es, el ingeniero no sigue un esquema predefinido de actuación sino que mantiene activas las tres tareas durante todo el Análisis Preliminar (esto se ha indicado en la figura mediante el solapamiento de las cajas que representa cada tarea). En ciertas ocasiones, parecerá que está identificando, describiendo u organizando, de forma exclusiva, pero ello se deberá, únicamente, a que el ingeniero percibe que identificar, describir u organizar es la acción más útil en ese momento. No obstante, si el ingeniero cree que puede obtener algún resultado de utilidad abandonando una tarea y lanzándose a la realización de otra, lo hará de forma inmediata. El comportamiento oportunista durante el Análisis Preliminar no refleja, en absoluto, una falta de disciplina, sino una estrategia que proporciona buenos resultados a un bajo coste. Adicionalmente, la relativa baja cantidad de información manejada durante el Análisis Preliminar hacen posible retener en la memoria grandes partes de ésta, favoreciendo el comportamiento oportunista.

Una vez que el ingeniero ha logrado una visión general del problema, es decir, los conceptos más relevantes han sido identificados, descritos y estructurados, el Análisis Preliminar puede darse por finalizado y proceder al siguiente paso de MAON, esto es, al Análisis Exhaustivo.

No obstante, es difícil precisar un cuándo y un cómo en la finalización del Análisis Preliminar. Existen varias posibilidades para determinar cuándo se ha finalizado la identificación de los conceptos sobresalientes. Por ejemplo, se podría utilizar una heurística como el amortiguamiento en la identificación de nuevos conceptos, esto es, la frecuencia de aparición de nuevos conceptos disminuye. Sin embargo, y retomando la analogía del cartógrafo, la identificación de referencias geodésicas puede no tener fin. Una vez marcados los picos de 3.000 metros, se puede seguir con los de 2.000, 1.000, 500 metros, y así sucesivamente. La única regla posible que determina la finalización del Análisis Preliminar es la calidad del mapa que el ingeniero ha obtenido. Dicho de otra forma, el ingeniero debe hacerse la pregunta: *Cuándo hablo con los clientes y usuarios, ¿existe alguna situación en la que no comprendo nada de lo que me dicen?* Si la respuesta es afirmativa, el ingeniero todavía no ha marcado las referencias geodésicas adecuadas, esto es, faltan conceptos relevantes por identificar o asociar. Si la respuesta es negativa, el ingeniero posee un buen mapa del problema, y está preparado para proceder con la siguiente etapa del MAON.

En los epígrafes siguientes, se comentan con mayor detalle las tres tareas indicadas en la figura 5.1.

5.1.1.1. IDENTIFICAR LOS CONCEPTOS MÁS DESTACADOS.

La primera tarea del Análisis Preliminar consiste en la identificación de las referencias geodésicas, esto es, los conceptos más sobresalientes del problema del usuario.

Por concepto debe entenderse cualquier “regularidad que se percibe en los hechos u objetos, o registro de hechos y objetos, y que se designa mediante una etiqueta” [Novak et al, 1988]. Un concepto es, para el ingeniero, un hecho, elemento, situación, objeto, etc. que posee una naturaleza propia y distinta de otros hechos, elementos, situaciones, etc. en el dominio del problema.

La identificación de conceptos es algo similar a la tarea de denotar a las cosas. De la misma forma que un científico da nombres a los elementos o fenómenos que ocurren en su dominio de estudio, con el propósito de poder refinarlos y distinguirlos de otros elementos o fenómenos, el ingeniero, durante esta primera actividad, da nombres a ciertas regularidades que percibe en el dominio del problema.

Por ejemplo, si el ingeniero se enfrenta a un problema consistente en desarrollar un sistema de vigilancia para un edificio, es posible que aparezcan en primer lugar conceptos como “alarma”, “intruso” o “policía”. Estos conceptos son los más relevantes del problema, ya que, de alguna forma, son los conceptos que permiten un primer nivel de entendimiento y estructuración. Dicho de otra forma, el objetivo del sistema de vigilancia es “Activar la alarma, para avisar a la policía, cuando se detecta la presencia de un intruso”.

5.1.1.2. DESCRIBIR LOS CONCEPTOS IDENTIFICADOS

Identificar un concepto no implica, necesariamente, que dicho concepto se comprenda adecuadamente, esto es, nombrar un hecho u objeto no implica que se comprenda. Por ejemplo, y retomando el ejemplo anterior, ¿qué se debe considerar como “intruso”? ¿Cualquier persona que entre en el recinto vigilado?, o ¿cualquier persona que entra sin autorización?. Está claro que identificar un concepto no significa que éste concepto se comprenda adecuadamente.

Tanto la disciplina de la semiótica [Eco, 1999], como la etnografía [Velasco et al., 1997] y la psicología [Anderson, 2000] investigan, bajo distintos prismas, la formación de conceptos. En todas las citadas disciplinas, se reconoce que el acto de nombrar (definir la existencia del concepto) no equivale a comprender (conocer las propiedades del concepto). Siguiendo a Eco, existe una gran diferencia entre el Tipo Cognitivo (TC), que corresponde con lo que se ha venido denominando “concepto”, y el Contenido Molar (CM), que aproximadamente se corresponde con el entendimiento o la comprensión de un concepto.

Por ejemplo, existe un TC del ratón doméstico (*mus muris*). Este TC lo posee el autor de la Tesis, sus Directores y, probablemente, mucha más gente. Sin embargo, y a diferencia del TC de ratón, que todo el mundo posee de igual forma, el CM del ratón es muy variable. La persona que no haya visto nunca un ratón (por ejemplo, en las ciudades hay pocos ratones), tenderá a asociar el CM del ratón con el CM de la

rata (en las ciudades hay muchas más ratas), cosa que no ocurrirá en una persona del campo. Y viceversa, un madrileño asociará el TC “pirulí” con “Torre España”, cosa que no ocurre en un nativo de Santander.

El CM es, por lo tanto, el entendimiento particular que cada persona posee de un TC determinado y, por lo tanto, no es compartido. Dicho de otra forma, el TC es un fenómeno cultural, mientras que el CM es un fenómeno personal.

La diferencia entre TC y CM es muy relevante en el Análisis, debido a que cada dominio se corresponde, en la mayoría de las ocasiones, con una sub-cultura. Por sub-cultura se debe entender el sistema de organización y valores de un conjunto de personas, en el sentido de la Etnografía [Velasco et al., 1997]. Cuando un ingeniero intenta estudiar un problema en un determinado dominio, no solo deberá identificar los conceptos (esto es, los TC), sino identificar el significado que dichos conceptos evoca en las personas del dominio (esto es, identificar el CM para dicha sub-cultura).

La identificación del significado de un concepto (el CM) solo es posible mediante la descripción de los conceptos identificados (TC) de tal forma que las personas del dominio (clientes y usuarios) estén de acuerdo en que dicha descripción es fidedigna. En definitiva, el procedimiento de indentificar un concepto equivale a establecer su TC y describirlo a establecer su CM.

Para realizar la descripción de los conceptos, el ingeniero utiliza, durante esta actividad, uno de los modelos del MCG, en concreto, el Diccionario de Identificación (DI). El **DI permite registrar cada concepto identificado** así como las asociaciones que posee con otros conceptos. El **Texto Narrativo (TN)**, otro de los formalismos de representación del MCG, **se utiliza para presentar la descripción de los conceptos a usuarios y clientes**, de tal forma que puedan afirmar estar de acuerdo con la descripción o mostrar su discrepancia.

5.1.1.3. ASOCIAR LOS CONCEPTOS IDENTIFICADOS

Los conceptos nunca aparecen de forma aislada, sino que establecen asociaciones entre sí, que contribuyen a su significado. Ello es, precisamente, el mecanismo que utiliza el ingeniero al confeccionar el DI.

Sin embargo, en muchas ocasiones, los conceptos únicamente establecen asociaciones con un número determinado de otros conceptos, esto es, los conceptos son como islas en un archipiélago. Las islas del mismo archipiélago están muy juntas, pero a cientos, o miles, de kilómetros de otras islas pertenecientes a distintos archipiélagos. En un dominio, los conceptos poseen una estructura de archipiélago, que queda reflejado por una red neuronal. Los conceptos acostumbran a establecer asociaciones con muchos otros conceptos, las cuales no son evidentes durante las primeras fases del Análisis Preliminar.

La presente actividad de organización tiene como propósito profundizar en la identificación de las asociaciones que se establecen entre distintos conceptos, de tal modo que sea posible construir un segundo nivel de descripción, más rico que el identificado en la actividad anterior. Para el registro del significado, ampliado, de los conceptos, se utilizan tanto el DI como el Mapa de Conceptos preliminar. Adicionalmente, el TN sigue siendo el mecanismo de comunicación con clientes y usuarios.

5.1.2. TÉCNICAS UTILIZADAS

MAON no pretende proponer ninguna técnica nueva para la realización del paso de Análisis Preliminar. Dado que el Análisis Preliminar consiste básicamente en identificar y recolectar información, durante esta etapa se utilizarán técnicas de educación, como por ejemplo la entrevista o la observación de tareas.

5.1.3. PRODUCTOS DE ENTRADA

El único producto de entrada para el Análisis Preliminar es la información procedente de la educación de conocimientos (notas del ingeniero, transcripciones de entrevistas, conclusiones de un brainstorming, etc.).

5.1.4. PRODUCTOS DE SALIDA

El producto de salida del Análisis Preliminar es el Modelo preliminar del problema. Este modelo está compuesto por dos formalismos de representación del MCG, que serán descritos en detalle en el epígrafe 5.3:

- **Diccionario de Identificación o Glosario.** Es un formalismo de representación de tipo tabular que permite registrar los conceptos identificados así como la descripción de los mismos.
- **Mapa de Conceptos.** Es un formalismo de representación gráfico que permite organizar y estructurar los conceptos. En concreto, durante el paso del Análisis Preliminar se utiliza una versión del Mapa de Conceptos denominada Mapa de Conceptos preliminar. El Mapa de Conceptos Preliminar utiliza menos recursos expresivos de los que permite un Mapa de Conceptos en toda su generalidad, aunque en el fondo Mapa de Conceptos y Mapa de Conceptos preliminar son el mismo formalismo de representación.

Ambos formalismos de representación son semánticamente equivalentes. La razón de utilizar dos formalismos de representación con igual capacidad expresiva reside en el hecho de que la información se percibe y maneja de distinta forma dependiendo de su formato de presentación [Vessey, 1991]. La información tabular es más fácil de consultar y revisar, mientras que la representación gráfica (en forma de grafo) facilita el razonamiento y las inferencias.

El Texto Narrativo no es un producto de salida del Análisis Preliminar, ya que la información que contiene aparece igualmente reflejada en el Diccionario de Identificación y el Mapa de Conceptos preliminar. La utilidad del Texto Narrativo durante el Análisis Preliminar es, simplemente, la de ayudar al ingeniero a validar con clientes y usuarios el entendimiento que posee de los distintos conceptos identificados.

5.1.5. EXCEPCIONES

No existen excepciones en este paso, excepto el hecho de que el ingeniero se declare incompetente para resolver el problema, lo cual no parece habitual que ocurra.

5.2. ANÁLISIS EXHAUSTIVO

El Análisis Exhaustivo es la segunda etapa del MAON. La finalidad del Análisis Exhaustivo es profundizar en el estudio del problema, hasta que toda la información relevante acerca del mismo haya sido identificada y estructurada adecuadamente. Al finalizar el Análisis Exhaustivo se obtiene el Modelo Exhaustivo del Problema, el cual supone el fin del análisis, propiamente dicho, y el comienzo de otras etapas de MAON destinados a identificar y derivar el modelo conceptual más adecuado para proseguir con el futuro desarrollo del producto software.

5.2.1. DESCRIPCIÓN

La etapa de Análisis Exhaustivo representa la culminación, por parte del ingeniero, del estudio del problema del usuario. Mientras en el Análisis Preliminar únicamente se identificaban y estructuraban los conceptos más sobresalientes, la presente tarea debe tender a la exhaustividad, intentando lograr que todos los conceptos relevantes del problema del usuario sean identificados, descritos y estructurados adecuadamente.

La analogía del cartógrafo es útil para ilustrar la finalidad del Análisis Exhaustivo. Una vez que las principales referencias geodésicas han sido marcadas sobre el terreno, el cartógrafo puede situarse en cualquier lugar de la región desconocida, siempre y cuando tenga dos referencias geodésicas visibles y proceder a estudiar con detalle el lugar. Del mismo modo, el ingeniero aprovecha las referencias proporcionadas por los conceptos identificados durante el Análisis Preliminar, de tal forma que puede profundizar en el estudio del problema del usuario sin perderse gracias a la información de referencia que proporcionan dichos conceptos.

El Análisis Exhaustivo está formado por cuatro tareas procedimentales, las cuales acostumbran a llevarse a cabo, al igual que en el Análisis Preliminar, de forma oportunista. No obstante, es necesario indicar que el comportamiento oportunista en el Análisis Exhaustivo es menor que en el Análisis Preliminar. Ello es debido a que, en el Análisis Exhaustivo, ya no se está explorando un dominio desconocido, por lo que es posible un grado de sistematización mayor durante esta fase. Dicho comportamiento sistemático se recoge en la secuencia de tareas propuesta para el Análisis Exhaustivo, la cual se muestra en la figura 5.2.

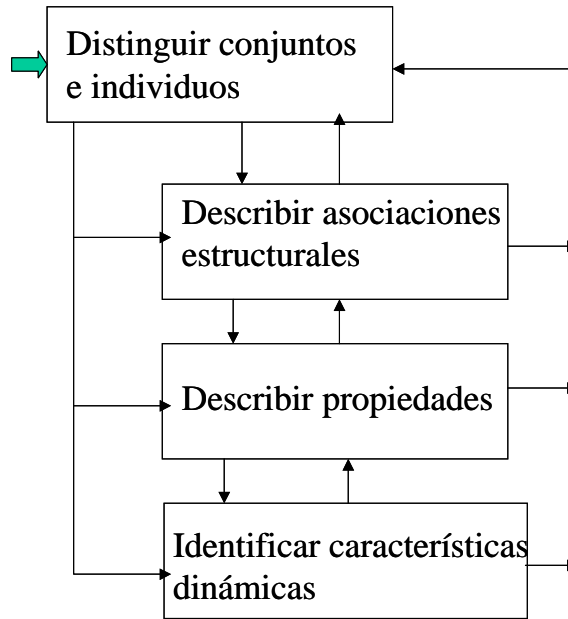


Figura 5.2. Secuencia de tareas durante el Análisis Exhaustivo

Los siguientes epígrafes describen, en detalle, las tareas mostradas en la figura 5.2.

5.2.1.1. DISTINGUIR CONJUNTOS E INDIVIDUOS

Los conceptos sobresalientes, identificados durante el Análisis Preliminar, pueden hacer referencia tanto a elementos individuales del dominio como a tipos, que contienen o hacen referencia a una pluralidad de elementos. Asimismo, estos elementos pueden organizarse en jerarquías complejas, poseer atributos, poseer un comportamiento bien definido y, en resumen, poseer una estructura y dinámica complejas.

Las acciones realizadas durante el Análisis Preliminar, son tendentes a conseguir únicamente una visión de alto nivel, esto es, una especie de plano esquemático del dominio del problema del usuario, que guíe al ingeniero en un estudio detallado posterior. Este estudio detallado se lleva a cabo durante el Análisis Exhaustivo, donde es necesario conseguir una mayor precisión y formalidad en la información registrada.

El primer paso en el refinamiento y formalización de la información adquirida se realiza durante la presente tarea del Análisis Exhaustivo. Esta tarea tiene como finalidad distinguir los conceptos que hacen referencia a una generalidad de elementos (esto es, los conjuntos), de los conceptos que hacen referencia a una singularidad concreta (esto es, los individuos). Esta distinción conjunto-individuo será esencial en las tareas siguientes del Análisis Exhaustivo, ya que permitirá refinar y precisar la estructura de los distintos elementos del dominio del problema del usuario, así como identificar claramente los procesos dinámicos en los que se ven involucrados.

La distinción conjunto-individuo se realiza utilizando el Mapa de Conceptos (MC). Sin embargo, y a diferencia del Análisis Preliminar, el MC generado en el paso de Análisis Exhaustivo se denomina Mapa de Conceptos Exhaustivo (MC Exhaustivo), ya que, por una parte, registrará la información del dominio con el

mayor nivel de refinamiento posible y, por otra, para realizar tal registro se utilizarán, con toda probabilidad, la totalidad de constructores y operadores del MC.

5.2.1.2. DISTINGUIR ASOCIACIONES ESTRUCTURALES

Una vez que se han identificado los conjuntos e individuos, es conveniente precisar las asociaciones del Modelo Preliminar del Problema del Usuario. La segunda tarea de refinamiento de la información durante el Análisis Exhaustivo consiste en identificar las relaciones estructurales que existen entre los distintos elementos del dominio. En MAON, dichas relaciones se registran mediante asociaciones especiales, las cuales pueden ser de tres tipos:

- Especialización (*spec*).
- Agregación (*pof*).
- Relación (*rel*).

Al igual que en la tarea anterior, el formalismo para registrar las asociaciones de especialización, agregación y relación es el MC Exhaustivo.

La identificación de las asociaciones notables de especialización, agregación y relación, además de proporcionar un mayor grado de refinamiento a la información obtenida del dominio, es esencial para una efectiva aplicación de las técnicas utilizadas durante la fase de Análisis Orientado a la Solución.

5.2.1.3. DESCRIBIR PROPIEDADES

En muchas ocasiones, un elemento del dominio se caracteriza por poseer ciertas propiedades. Por ejemplo, un coche se caracteriza por poseer ruedas, en número de 4. Igualmente, una persona mayor de edad se caracteriza por tener 18 o más años. Una tercera tarea de refinamiento del modelo preliminar consistirá, por lo tanto, en determinar qué propiedades poseen, o que restricciones deben cumplir, los distintos elementos del dominio.

La identificación de propiedades se realiza una vez identificadas las relaciones estructurales que poseen los distintos elementos del dominio, lo cual se ha llevado a cabo en la tarea anterior. En la presente tarea, se debe refinar el modelo para incluir todas las propiedades y restricciones que afecten a conjuntos e individuos, así como las asignaciones de valores a atributos. Las propiedades se registran utilizando la asociación especial concepto-atributo (*attr*), mientras que las asignaciones de valores a atributos se registran utilizando la asociación especial *val*. El registro de propiedades o restricciones más complejas se lleva a cabo utilizando predicados y funciones.

Tanto las restricciones como la asignaciones de valores a atributos se registran utilizando el MC exhaustivo.

5.2.1.4. IDENTIFICAR CARACTERÍSTICAS DINÁMICAS

La última tarea del Análisis Exhaustivo consiste en identificar y precisar todos los aspectos dinámicos del dominio. Cada dominio posee una estructura y unas reglas de evolución particulares, las cuales deberán ser registradas en el MCG en la medida en que éstas características sean importantes para comprender y resolver el problema bajo estudio.

Las características dinámicas del dominio se representan, en general, utilizando predicados y funciones. Los predicados y funciones proveen, en general, la suficiente capacidad expresiva para definir la dinámica de cualquier sistema, hecho que se ve refrendado por su utilización en una gran diversidad de modelos conceptuales y modelos de especificación utilizados en la actualidad.

Todas las características dinámicas del dominio se registran utilizando el MC Exhaustivo.

5.2.2. TÉCNICAS UTILIZADAS

MAON no pretende proponer ninguna técnica específica para la realización del paso de Análisis Exhaustivo. No obstante, y dado que el Análisis Exhaustivo es, principalmente, un paso de refinamiento y formalización de la información obtenida del dominio del problema del usuario, va a ser necesario la utilización de técnicas de validación. Dichas técnicas de validación, como los escenarios o el prototipado, tienen como objetivo asegurar que la información recogida es correcta y completa.

5.2.3. PRODUCTOS DE ENTRADA

El producto de entrada del Análisis Exhaustivo es el Modelo Preliminar del Problema del Usuario. Como se ha indicado ya, este modelo está compuesto por dos formalismos de representación:

- Diccionario de Identificación.
- Mapa de Conceptos preliminar.

5.2.4. PRODUCTOS DE SALIDA

El producto de salida del Análisis Exhaustivo se denomina Modelo Exhaustivo del Problema del Usuario. El Modelo Exhaustivo del Problema del Usuario está compuesto por dos formalismos de representación, que serán detallados en el epígrafe 5.3:

- **Diccionario de Descripción.** Es un formalismo de representación de tipo tabular que permite registrar todos los conceptos y asociaciones presentes en un problema. En este sentido, el Diccionario de Descripción es similar al Diccionario de Identificación utilizado durante el Análisis Preliminar. Sin embargo, el Diccionario de Descripción posee dos características importantes que no posee el Diccionario de Identificación:

- Permite registrar información más refinada. Así, por ejemplo, el Diccionario de Descripción registra la diferencia entre clases e individuos, jerarquías de especialización y agregación, etc.
 - Es el modelo utilizado como entrada para la Técnica de Identificación del Modelo Conceptual Idóneo, así como la Técnica de Derivación del Modelo Conceptual Seleccionado. Ambas técnicas se utilizan durante la fase de Análisis Orientado a la Solución
- **Mapa de Conceptos Exhaustivo.** Es un formalismo de representación gráfico que permite organizar y estructurar los conceptos más relevantes. Al igual que en el caso del Diccionario de Descripción, el Mapa de Conceptos Exhaustivo permite registrar información más refinada del dominio del problema que el Mapa de Conceptos Preliminar.

Ambos formalismos de representación son, al igual que en el Análisis Preliminar, semánticamente equivalentes. De hecho, en el capítulo 6, se describe como obtener el Diccionario de Descripción a partir del Mapa de Conceptos exhaustivo. La razón de utilizar dos formalismos de representación es la indicada para el Análisis Preliminar, esto es, la facilidad de manejo de la información dependiendo del formato en que se presenta.

5.2.5. EXCEPCIONES

Siguiendo la analogía del terreno inexplorado, es posible que el cartógrafo cometa errores. Por ejemplo, una vez fijadas las referencias geodésicas, puede ocurrir que, al entrar en un valle, el cartógrafo no tenga a la vista las referencias que le sirven para ubicar los distintos accidentes en el espacio. Ello es debido, simplemente, a un error de apreciación. El cartógrafo creía que las referencias geodésicas eran las suficientes para triangular todos los accidentes. Sin embargo, pueden quedar zonas de sombra, para las que es necesario fijar referencias especiales antes de comenzar a realizar el plano de la zona.

Durante el Análisis Exhaustivo puede ocurrir un efecto similar. El ingeniero ha dado por finalizado el Análisis Preliminar cuando posee la confianza de comprender, de modo provisional pero parcial, la totalidad de los conceptos más importantes del dominio del problema del usuario. Sin embargo, al igual que el cartógrafo, puede localizar zonas de sombra, en las que los conceptos adquiridos durante el Análisis Superficial no le resultan suficientes para orientarse.

En este caso, el ingeniero deberá abandonar el Análisis Exhaustivo y volver al Análisis Superficial, si bien se circunscribirá, únicamente, a la zona donde su conocimiento se ha demostrado insuficiente.

5.3. ELEMENTOS DEL MCG

El MCG está formado por tres formalismos básicos de representación (Mapa de Conceptos –MC–, Diccionario de Identificación –DI–, y Diccionario de Descripción –DD–), además de un mecanismo adicional de comunicación, denominado Texto Narrativo.

El conjunto de formalismos de representación anteriormente indicados representan los mismos aspectos del dominio, aunque ello no significa que los modelos sean iguales, ya que cada uno de ellos destaca o da visibilidad a un subconjunto de conceptos en detrimento de otros. Sin embargo, el hecho de que todos los modelos estén basados en los mismos principios hace que sea posible exponer los elementos comunes de forma unificada para, posteriormente, destacar los aspectos particulares.

Nótese, no obstante, que la existencia de una diversidad de elementos en el MCG no implica que todos deban utilizarse (aunque *se puedan*) en todos los casos. En muchas ocasiones, es preferible confeccionar modelos utilizando los menos elementos posibles. En el capítulo 8, donde se presenta un ejemplo de utilización de MAON, se indican, asimismo, ciertas guías prácticas en lo que se refiere al uso práctico del MCG.

En los siguientes epígrafes se presentan los elementos que forman el MCG.

5.3.1. CONCEPTOS Y ASOCIACIONES

Los formalismos de representación propuestos en este trabajo utilizan únicamente dos categorías de descripción del dominio: los conceptos y las asociaciones. Estas categorías se definen como sigue:

- **Concepto:** Etiqueta que permite referenciar cualquier regularidad percibida, esto es, a cualquier elemento separable o distinguible del dominio. Un concepto es un nombre convencional que se asigna a un hecho, objeto, fenómeno, etc. que existe o tiene lugar en un determinado dominio, es relevante para un problema determinado, y que permite nombrar (referenciar) a dicho hecho, objeto, fenómeno, etc.
- **Asociación:** Etiqueta que representa cualquier conexión que pueda establecerse entre dos conceptos relevantes del dominio del problema planteado por el usuario. Las asociaciones permiten conectar conceptos, de tal forma que es posible expresar afirmaciones sobre el dominio.

Derivado de las definiciones anteriores, un concepto debe entenderse como cualquier hecho, objeto, fenómeno, etc. que despierte la atención de un ingeniero, en la medida en que el ingeniero cree que dicho hecho, objeto, fenómeno, etc. es relevante para la comprensión o solución del problema del usuario bajo estudio. Por ejemplo, en una empresa donde se desee implantar un sistema de facturación, conceptos importantes podrían ser:

- Cliente
- Factura

- Pedido
- Mercancía, etc.

Los conceptos indicados anteriormente son típicos de un modelo de negocio, y por lo tanto de naturaleza estática (aptos para ser representados mediante un diagrama de clases o un diagrama ER, por ejemplo). No obstante, los conceptos no poseen únicamente naturaleza estática, ya que pueden hacer referencia a aspectos dinámicos. Por ejemplo, para el mismo dominio anterior, esto es, la empresa donde debe implantarse el sistema de facturación, se podrían identificar los siguientes conceptos:

- Facturar (ó facturación). Nótese que los conceptos se expresan de forma más natural como sustantivos, que como verbos o adjetivos.
- Enviar (ó envío)
- Devolver (ó devolución), etc.

“Facturar”, “Enviar” o “Devolver” son conceptos dinámicos porque hacen referencia a una operación, proceso o acción en el dominio del problema del usuario. Por el contrario, “Factura” o “Pedido” son conceptos estáticos porque hacen referencia a elementos del dominio que no cambian en el tiempo a menos que sufran algún tipo de acción.

La posibilidad de que la categoría de conceptos permita expresar una diversidad de aspectos tanto estáticos como dinámicos es altamente relevante y, sobre todo, representa una ruptura frente a los modelos conceptuales utilizados en la actualidad. Los modelos conceptuales utilizados habitualmente en el desarrollo de software poseen el inconveniente de que realizan una categorización excesivamente temprana de todos los aspectos del dominio [Kop et al., 1998], distinguiendo, por ejemplo, entre procesos, objetos, relaciones, etc. No obstante, esta categorización es a todas luces subjetiva, ya que un mismo concepto se puede interpretar de diversas formas, dependiendo del contexto particular en el que se encuentre el ingeniero.

A modo de ejemplo, supóngase un sistema para una biblioteca. En la biblioteca hay libros y autores. Los libros parecen una entidad de primer orden ya que, al fin y al cabo, forman el fondo de la biblioteca. Sin embargo, ¿qué son los autores? En un contexto determinado, los autores pueden ser simplemente un atributo de los libros (por ejemplo, cuando no es necesario realizar ningún tipo de operación con ellos) pero, en otros casos, los autores deben considerarse una entidad de primer orden y surge, en consecuencia, una relación (LIBRO-ESCRITO POR-AUTOR).

El ejemplo anterior ilustra la importancia de que los conceptos puedan expresar una diversidad de aspectos del mundo real. Esta capacidad expresiva evita que el ingeniero deba decidir, durante las fases tempranas del análisis, cómo interpretar la realidad que se presenta ante sus ojos y, por lo tanto, pueda diferir la toma de decisiones a momentos posteriores del análisis, donde el problema del usuario está mejor comprendido. Como los distintos formalismos del MCG no realizan ningún tipo de categorización previa y, de hecho, no distinguen los conceptos estáticos de los dinámicos, permiten al ingeniero centrarse en el problema del usuario antes de tomar decisiones de representación de los conceptos.

En otro orden de cosas, una asociación, al contrario que un concepto, no hace referencia a ningún hecho, objeto, fenómeno, etc., sino a las posibles conexiones, las cuales pueden ser muy diversas, existentes entre dichos hechos, objetos, fenómenos, etc. Por ejemplo, supóngase el mismo dominio anterior; entre los conceptos “Cliente”, “Factura”, “Pedido” y “Mercancía” se podrían identificar las siguientes asociaciones (las asociaciones se indican entre paréntesis):

- Cliente (realiza) Pedidos
- Factura (se refiere a) Pedidos
- Cliente (paga) Facturas
- Pedido (referencia) Mercancía, etc.

Se puede observar que la diferencia entre concepto y asociación es arbitraria, ya que depende del problema particular bajo estudio, esto es, depende del objetivo del análisis. En un mismo dominio, e independientemente de éste, al estudiar dos problemas distintos, los conceptos de un problema pueden convertirse en asociaciones de otro, y viceversa. A modo de ejemplo, supóngase las dos situaciones siguientes, las cuales explotan los ejemplos anteriores:

1. Una empresa desea realizar un sistema de facturación. Podría identificar, en línea con lo expuesto hasta el momento, los conceptos siguientes:

- Pedido
- Facturas

Y la asociación, señalada con paréntesis:

- Factura (se realiza a partir de) Pedidos

2. La misma empresa, una vez realizado el sistema de facturación deseado, intenta idear un procedimiento eficiente para hacer uso del mismo. Los conceptos identificados, en este caso, podrían ser:

- Operador
- Facturación

Y la asociación, señalada con paréntesis:

- Operador (inicia) Facturación

Donde es claro que “facturación” es el concepto que hace referencia a “Factura (se realiza a partir de) Pedidos” o, más estrictamente, a la asociación “se realiza...”.

El ejemplo anterior puede parecer un poco forzado. Ello se debe a que en dominios como el empresarial abundan los conceptos estáticos o, más estrictamente, a nivel superficial estos dominios están muy formalizados y estandarizados desde el punto de vista de objetos (documentos), y no de hechos o

fenómenos. Hasta ahora, como en lo que sigue, se han utilizado ejemplos extraídos de dichos dominios, con el propósito de claridad proporcionado por la habitualidad, pero ello no agota las posibilidades de los formalismos de representación. Supóngase un dominio completamente distinto, como, por ejemplo, una compañía ferroviaria. El ejemplo anterior podría reescribirse como sigue:

1. La compañía ferroviaria desea automatizar sus pasos a nivel. En dicho dominio, se podrían describir los conceptos:

- Tren
- Barrera

Y la asociación, señalada con paréntesis:

- La Barrera (debe bajarse cuando pase) un Tren

2. La misma compañía ferroviaria, una vez automatizados sus pasos a nivel, puede desear realizar un mantenimiento de los mismos. Los conceptos en este caso podrían ser:

- Movimiento de la barrera
- Barrera

Y la asociación (señalada con paréntesis):

- El movimiento de la barrera (afecta a largo plazo al funcionamiento) de la barrera

El “movimiento de la barrera” hace referencia, evidentemente, al hecho de que la barrera “debe bajarse cuando pase...” un tren.

Cabe realizar varias consideraciones a este segundo ejemplo:

- Se ha indicado que los conceptos son etiquetas, y no nombres, para evitar contradicciones con las ciencias lingüísticas. Así, por ejemplo, “movimiento” es un nombre, pero el concepto puesto como ejemplo ha sido “movimiento de la barrera”. La regularidad o elemento distinguible, en este caso un fenómeno, es que la barrera se mueve.
- En líneas generales, no siempre va a existir un nombre para cada hecho, objeto, fenómeno, etc. de interés, aunque siempre se puede crear uno nuevo. Por ejemplo, la asociación “afecta a largo plazo al funcionamiento” también podría ser considerada como un concepto, que se podría denominar “degradación”. En los campos técnicos o científicos es muy común definir denominaciones para los hechos, fenómenos, objetos, etc. de interés.
- Para expresar asociaciones lingüísticamente correctas, se requiere el uso de artículos (determinados o indeterminados) y proposiciones. El uso de estos elementos sincategoremáticos no altera los conceptos entendidos como regularidades: “el tren”, o “un tren”, hacen referencia a la misma regularidad percibida.
- Por último, puede observarse que las asociaciones son conceptos complejos (el paso del tren, el desgaste a largo plazo la barrera por el movimiento).

Abundando en lo anterior, la posibilidad de que las asociaciones se transformen en conceptos, y viceversa, puede ser explotada durante el análisis. Por ejemplo, el ingeniero podría identificar la siguiente proposición durante el estudio de un problema:

- Alumno (se matricula) en Asignaturas

La asociación “se matricula” esconde claramente el concepto “matrícula”. Este hecho podría transformar la proposición anterior en dos proposiciones:

- Alumno (realiza) Matrícula
- Matrícula (contiene) Asignaturas

5.3.2. PROPOSICIONES

Hasta el momento, la exposición se ha centrado en aclarar las categorías concepto y asociación. En este punto, es conveniente introducir la categoría derivada proposición:

- **Proposición:** Menor unidad dotada de significado en los formalismos de representación (MC, DI, DD). Una proposición es el conjunto formado por dos conceptos y una asociación.

Una proposición es, por lo tanto, cada afirmación (frase u oración) creada al definir una asociación entre conceptos. Por ejemplo, las siguientes afirmaciones, ya utilizadas anteriormente, se definen como proposiciones en los formalismos de representación (MC, DI, DD):

- Cliente (realiza) Pedido
- Cliente (paga) Facturas
- Pedido (referencia) Mercancía, etc.

En una proposición, se conoce como **concepto principal** al concepto del cual se afirma o predica algo. El concepto que predica algo de otro concepto se conoce como **concepto subordinado**. Nótese que un concepto puede ser principal y subordinado al mismo tiempo si forma parte de varias proposiciones. Por ejemplo, en las proposiciones anteriores:

- Cliente (realiza) Pedido
- Pedido (referencia) Mercancía.

Cliente es concepto principal de la primera proposición. Pedido es concepto subordinado de la primera proposición y principal de la segunda proposición.

Una proposición también puede estar formada como un concepto asociado a una proposición, o

incluso dos proposiciones unidas por una asociación; esto es, las proposiciones se definen de forma recursiva. Por ejemplo, lo que sigue se define como una proposición, utilizando la notación [] para delimitar la proposición y facilitar así su lectura:

- Impago (se produce cuando) [Cliente (no paga) facturas]
- [Alumno (realiza) Matrícula] (durante) el mes de septiembre
- [Alumno (realiza) Matrícula] (cuando) [Alumno (aprueba) curso anterior]

Las proposiciones así construidas se denominarán **proposiciones recursivas**.

5.3.3. CONJUNTOS E INDIVIDUOS

Los conceptos, como categorías de descripción, basan su fortaleza en su falta de precisión. Un concepto es cualquier regularidad percibida, por lo que puede referirse a hechos u objetos reales (mesa, avión), ficticios (ángel, Vulcaniano), concretos (vaso), abstractos (bondad), materiales, inmateriales, etc.

A efectos de representar conocimientos, ciertas precisiones son necesarias. Una de dichas precisiones hace referencia a la genericidad de los conceptos, esto es, a si los conceptos hacen referencia a clases o individuos. Por ejemplo, sea la proposición:

- Mesa (tiene) 4 patas

La proposición anterior puede entenderse como “todas las mesas tienen 4 patas” ó, si nos estamos refiriendo a una mesa concreta, “esta mesa tiene 4 patas”. Evidentemente, ambas interpretaciones son distintas y, en un dominio determinado, no tienen por que ser ambas ciertas al mismo tiempo.

En la lengua habitual se usan deícticos (esta, esa, aquella...) para señalar objetos particulares en un momento espacial (y temporal, utilizando adverbios) determinado. Ello permite distinguir las afirmaciones particulares de las afirmaciones generales, a las que estamos más acostumbrados, ya que son las más comunes en el discurso.

En los formalismos de representación no se dispone de ninguna categoría que sustituya a deíctivos y adverbios, por lo que **es necesario precisar si un concepto determinado referencia a una clase o a un individuo**. Dicha diferenciación se realizará utilizando los símbolos { }, esto es:

- Mesa referencia a una mesa particular.
- {Mesa} referencia al conjunto o clase de las mesas.

Cuando en el MCG nos refiramos a un concepto que denota un conjunto o clase, será aconsejable, además de utilizar los símbolos { }, utilizar la forma plural de la etiqueta, la cual sugieren de forma inmediata una pluralidad de elementos u objetos. Así, en lugar de {Mesa}, es preferibles utilizar {Mesas}.

5.3.4. SIGNIFICADO

La proposición es la menor unidad de expresión de los formalismos de representación (MC, DI, DD), debido a que permite **definir el significado de un concepto**.

Existen dos formas básicas de expresar el significado de cualquier concepto: por extensión, enunciando los hechos, objetos, fenómenos, etc. subsumidos por un concepto determinado, o por comprensión, proporcionando una definición que englobe a todos los hechos, objetos, fenómenos, etc. subsumidos por el concepto [Díez et al., 1997]. En MAON se utilizan ambas formas, aunque la más usada y preferible, por su brevedad, es la de comprensión.

5.3.4.1. COMPRENSIÓN

En los formalismos de representación (MC, DI, DD), el significado de un concepto se define como el conjunto de asociaciones que un concepto forma con otros conceptos (o proposiciones, en su caso). Esta forma de definir el significado es claramente de comprensión, y posee dos ventajas:

- No es necesario fijar el significado de un concepto una vez que éste se ha adquirido. Por lo general, en los estadios tempranos del análisis, no es posible llegar a comprender el dominio del problema del usuario de una forma completa. Al no ser necesaria una definición para cada concepto adquirido, es posible acelerar la actividad de análisis.
- El significado de un concepto evoluciona a medida que se van identificando asociaciones entre dicho concepto y otro(s) concepto(s) o proposición(es). De esta forma, el significado de un concepto se actualiza automáticamente a medida que progresa la actividad de análisis.

El significado de un concepto en el MCG es determinado, por lo tanto, por las asociaciones que establece con otros conceptos, debido a que cada asociación describe algo con relación a un concepto. Por ejemplo, las proposiciones indicadas anteriormente:

- Cliente paga Facturas
- Cliente realiza Pedidos

describen el concepto de Cliente, de la forma siguiente:

- Cliente es algo que:
 - o paga Facturas
 - o realiza Pedidos

El significado (por comprensión) es, probablemente, uno de los puntos más delicados del MCG, aunque en la práctica se vuelve trivial. A modo de ejemplo, supóngase la siguiente fórmula de lógica de

predicados:

$$\text{hombre}(X) \rightarrow \text{mortal}(X)$$

La lectura que se podría hacer de esta fórmula es “si X es un hombre, entonces X es mortal”. Dicha lectura viene dada por dos factores:

- La habitualidad. Esta fórmula es la primera que se introduce en todo curso de lógica de predicados.
- Las etiquetas. La fórmula simplemente es comprensible porque se utilizan los términos “hombre” y “mortal”, en la medida en que el lector posee los conceptos de “hombre” y “mortal”. Si la fórmula se escribiera $H(X) \rightarrow M(X)$, no sería tan fácilmente comprensible.

En el MCG, ocurre lo mismo que en la fórmula anterior. Si se posee el concepto de “Cliente”, es absurdo afirmar que su significado viene dado porque “realiza Pedidos” y “paga Facturas”. Dichas asociaciones son conocidas por todo aquel que conozca qué es un “Cliente”, y la propia invocación del concepto de cliente puede despertar en quien lo posee una gran conjunto de definiciones que sería impensable registrar en cualquier formalismo de representación.

Ahora bien; si el concepto de “Cliente” se etiquetase como “C”, ¿a qué hecho, objeto, fenómeno, etc. referencia? Imposible saberlo.

¿Y si se sabe que “C”:

- realiza Pedidos y
- paga Facturas?

Entonces, se podría afirmar que “C” es un “Cliente”.

Para dominios donde el ingeniero no posea una gran habitualidad, la definición de significado aportada anteriormente es suficiente. En MAON, los formalismos de representación registran todo el conocimiento necesario para **describir** y, en consecuencia, **identificar**, el concepto de “Cliente”, y por ello determinan el significado del concepto de cliente.

Nótese, adicionalmente, que la definición por comprensión en el MCG es aplicable a cualquier concepto, mientras que la definición por extensión solo es válida para conjuntos de individuos. Esta es otra razón para preferir la definición por comprensión siempre que sea posible.

5.3.4.2. EXTENSIÓN

Una definición por extensión exige la descripción de todos los elementos pertenecientes a un

conjunto. En los formalismos de representación, ello se consigue mediante una asociación especial de manipulación de conjuntos, que por simplicidad se denomina “bel” (belongs). Dicha asociación especial se en un epígrafe posterior.

5.3.5. CONCEPTOS ESPECIALES

Con cierta frecuencia, en los problemas reales aparecen conceptos que no es posible definir por extensión o comprensión. Dichos conceptos, como por ejemplo, el concepto de “raíz cuadrada”, se expresan mejor utilizando los mecanismos que provee la lógica de predicados, esto es, los predicados y funciones. En el MCG, dichos conceptos se denominan **conceptos especiales**. Dichos conceptos se describen en los siguientes epígrafes.

5.3.5.1. CONCEPTOS DEFINIDOS MEDIANTE UN PREDICADO

En algunas ocasiones, un objeto, hecho, situación, etc. no puede expresarse como un concepto, por la simple razón de que no existen nombres para todos los objetos, hechos o situaciones, o porque dicho nombre no es conocido por el ingeniero. Por ejemplo, imagínese un paciente en una consulta médica. Tras el examen, el médico puede observar que:

Paciente (tiene) fiebre

Paciente (tiene) manchas rojizas

Con ambos síntomas, el médico no puede establecer si, por ejemplo, el paciente ha contraído una infección o una intoxicación alimentaria, por ejemplo. Sin embargo, ambos síntomas son importantes y, para posteriores pruebas, deben ser tenidos en cuenta conjuntamente. Esto es, de algún modo, ambos forman una unidad indisoluble.

Para expresar ambos síntomas, es necesario utilizar la conjunción lógica AND, esto es, es necesario afirmar que:

[Paciente (tiene) fiebre] AND [Paciente (tiene) manchas rojizas]

y ambas proposiciones, conjuntamente, deben considerarse como un único concepto.

El MCG permite **definir cualquier tipo de predicado** que de cuenta de cualquier hecho o situación del dominio del problema del usuario. Para ello, es posible utilizar los operadores lógicos usuales, los operadores relacionales (>, <, etc.) y, en general, cualquier expresión que produzca un valor de verdad, ya

sea en lógica tradicional, difusa, temporal o modal¹.

5.3.5.2. CONCEPTOS DEFINIDOS MEDIANTE UNA FUNCIÓN

Al igual que en el caso de los conceptos definidos por un predicado, con frecuencia es necesario definir un concepto que resulta de la aplicación de algún tipo de proceso, transformación o función sobre un hecho, objeto o situación. Supóngase, por ejemplo, el concepto televisión. Sobre este concepto (nótese que denota un objeto) es posible realizar una operación como, por ejemplo, “encender”.

El MCG permite **definir cualquier tipo de función** que dé cuenta de cualquier hecho o situación del dominio del problema del usuario. No es necesario especificar el detalle interno (esto es, del funcionamiento) de la función.

5.3.5.3. CONCEPTOS DEFINIDOS MEDIANTE UNA TRANSICIÓN

Con mucha frecuencia, en el mundo real surgen relaciones entre hechos u objetos que hacen referencia a aspectos de organización temporal. Por ejemplo, “el sol sale después de la noche”, y, viceversa, “el sol se pone antes de la noche”.

La relación entre salir o ponerse el sol, y la noche es temporal (uno ocurre después o antes que el otro). Para establecer relaciones temporales, los predicados, utilizando lógicas temporales, serían suficientes. Sin embargo, las lógicas temporales son complejas y, además, son poco usadas en la Ingeniería del Software.

En su lugar, es preferible introducir en el MCG, para representar este tipo de relaciones temporales entre hechos u objetos –conceptos–, un concepto especial que se denominará **“transición”**, en coherencia con las transiciones de los modelos orientados a estados, las cuales sirven para representar, aproximadamente, el mismo tipo de relaciones temporales. Nótese que, no obstante, las transiciones en el MCG no establecen necesariamente un orden estricto, o relación de causalidad, entre hechos u objetos, sino algún tipo de relación u ordenación temporal, la cual no tiene por qué estar completamente definida en el MCG.

5.3.6. ASOCIACIONES ESPECIALES

En el MCG se utilizan varios tipos de asociaciones predefinidas, que se han denominado **asociaciones especiales**. Dichos tipos de asociaciones son muy importantes en la construcción del MCG, ya que aparecen con mucha frecuencia en la mayoría de los problemas reales. Dichas asociaciones especiales se describen en los siguientes epígrafes.

¹ En MAON, la lógica utilizada no afecta al modelo, ya que los operadores lógicos y su semántica no son tratados de ninguna forma.

5.3.6.1. ASOCIACIONES UTILIZADAS CON CONJUNTOS

La distinción clase-individuo implica que pueden existir dos semánticas distintas de las asociaciones utilizadas con conjuntos:

- La asociación refleja alguna propiedad o enuncia algún hecho predicable de todos los miembros del conjunto. Por ejemplo:

{Hombres} (son) mortales

Ser mortal es una propiedad de cada una de las personas pertenecientes al conjunto {Hombres}, esto es, cada persona es mortal.

- La asociación refleja alguna propiedad o enuncia algún hecho predicable del conjunto, pero no de cada uno de sus miembros. Por ejemplo:

{Hombres} (es) finito

La propiedad del finitud es del conjunto {Hombres}, no de cada uno de los hombres.

En líneas generales, al realizar el análisis, las propiedades o hechos predicados sobre un conjunto se refieren en la inmensa mayoría de los casos a propiedades o hechos que se pueden predicar de cada uno de los miembros del conjunto, y no del conjunto como un todo. En este sentido deben entenderse las asociaciones utilizadas con conjuntos en el MCG.

5.3.6.2. ASOCIACIONES QUE DENOTAN ASPECTOS ESTRUCTURALES

En el MCG se han definido una serie de asociaciones que denotan ciertas relaciones taxonómicas o partonómicas entre conceptos, esto es, relaciones estructurales. Aunque no son estrictamente necesarias, el gran número de veces que ocurren en el mundo real, lo cual se refleja igualmente en el número de modelos conceptuales que utilizan este tipo de relaciones, hace preferible definir las de antemano. Adicionalmente, dicha definición facilita la aplicación de las técnicas IMCI y DMCS.

Dichas asociaciones especiales son las siguientes:

- **Spec** (de *Specialisation*). Esta asociación denota una relación de generalización/especialización entre dos conceptos.
- **Pof** (de *part-of*). Esta asociación denota una relación de agregación entre dos conceptos.
- **Rel** (de *relation*). Esta asociación denota una relación, de tipo general, entre dos conceptos..
- **Bel** (de *belongs*). Esta asociación denota la pertenencia de un concepto (que denota un individuo) a otro concepto (que denota un conjunto).

- **Subs** (de *subset*). Esta asociación denota la relación de inclusión de un concepto (que denota un conjunto) en otro concepto (que denota otro conjunto).
- **Attr** (de *attribute*). Esta asociación denota una relación entre dos conceptos, donde el concepto subordinado es una propiedad del concepto principal.

5.3.6.3. ASOCIACIONES QUE DEFINEN VALORES

En el caso de que un objeto, hecho, situación, etc. posea propiedades, definidas mediante la asociación *attr*, es muy frecuente que sea necesario especificar algún valor para dicha propiedad. La definición de un valor para una propiedad se realiza mediante la asociación **val** (de *value*).

5.3.7. OPERADORES

Los operadores son un mecanismo que permite la manipulación de proposiciones, con el fin de hacer más simple el MCG. Se han definido tres operadores, que se exponen en los siguientes epígrafes.

5.3.7.1. IDENTIDAD

En el MCG, dos conceptos que posean la misma etiqueta se consideran, salvo evidencia en contrario, como potencialmente distintos. Por ejemplo, supóngase que en el Modelo Exhaustivo del Problema del Usuario aparezcan dos conceptos “cliente”². En principio, es necesario considerar que ambos conceptos denota a clientes distintos, con la posibilidad de que en algún caso, el cliente al que denotan ambos conceptos sea el mismo. Ambos conceptos pueden participar en distintas asociaciones y poseer, por lo tanto, distinto significado.

La operación de identidad, notada **EQ** (de *equal*) permite simplificar el MCG al transformar dos o más conceptos $\kappa_1, \kappa_2, \dots, \kappa_n$ en un único concepto κ . El concepto κ asume todas las asociaciones de $\kappa_1, \kappa_2, \dots, \kappa_n$. Por ejemplo, supónganse las siguientes proposiciones:

- Cigarrillos son malos para la Salud.
- Fumar produce Cáncer.

Entonces, si se realizase la operación de identidad $\text{Tabaco} = \text{EQ}(\text{Cigarrillos}, \text{Fumar})$, el MCG se transformaría a:

- Tabaco son malos para la Salud.
- Tabaco produce Cáncer.

² Ello es muy fácil que ocurra si, por ejemplo, dos analistas estudian partes del problema independientemente y sin comunicación entre ellos.

5.3.7.2. DEFINICIÓN

El operador de definición permite agrupar un conjunto de conceptos y, o, proposiciones, y referirse a ellos como si de un concepto se tratase. El operador de definición permite simplificar el MCG, introduciendo profundidad al mismo, esto es, permite representar conceptos y proposiciones a distintos niveles de detalle.

Un ejemplo ayudara a entender mejor su funcionamiento. Supóngase la proposición:

- Pedido (se recibe) por Fax.

La proposición anterior hace referencia a un pedido que se recibe por fax. En algunos contextos, podría ser interesante disponer del concepto³ de Pedido-Fax, que hace referencia precisamente a la proposición anterior. En el MCG, ello puede conseguirse con la utilización del operador de definición, que se nota **DEF** (de *definition*). De esta forma, el ejemplo anterior podría escribirse como:

- Pedido-Fax = DEF(Pedido, [Pedido (se recibe por) Fax])

En la proposición anterior, nótese que el concepto “pedido” se destaca dentro del operador DEF. Este concepto se denomina “concepto de enlace”, ya que permite que se utilice el concepto “Pedido-fax” como si de un pedido se tratase. Por ejemplo, dada la proposición:

- Cliente (realiza) Pedido-Fax

Gracias a la definición y al concepto enlace, es posible derivar automáticamente que:

- Cliente (realiza) pedido
- Pedido (se recibe por) fax

El operador de definición es un operador sin pérdida de significado, al igual que le ocurre al operador EQ. Se entiende que un operador no pierde significado si, una vez aplicado, es posible recuperar el MCG original sin derivar proposiciones potencialmente incorrectas. Un operador con pérdida de significado similar al de definición es el operador de abstracción, que se expone a continuación.

5.3.7.3. ABSTRACCIÓN

El operador de abstracción permite agrupar un conjunto de conceptos y, o, proposiciones, y referirse

³ M. Jackson, en diversos artículos (ver, por ejemplo, The meaning of requirements, Annals of Software Engineering, Balzer Science Publishers, Vol. 3, 1997) cita dos mecanismos de descripción del entorno (básicamente, lo que en este trabajo de Tesis se denomina “problema”): La Designación y la Definición. El significado de ambos mecanismos en Jackson es similar al de concepto y asociación, así como al de definición, respectivamente, del presente trabajo.

a ellos como si de un concepto se tratase, al igual que el operador de definición. Su propósito es, igualmente, el mismo; esto es, simplificar el MCG, introduciendo profundidad al mismo. Sin embargo, el operador de abstracción es un operador con pérdida de significado. Un ejemplo ayudara a entender mejor su funcionamiento. Supóngase que la proposición:

- Cliente (realiza) Pedidos

puede abstraerse bajo el concepto de:

- Pedidos de Cliente

que es (más o menos, siempre en función del contexto) equivalente a la proposición [Cliente realiza Pedido].

Dentro del MCG, el operador de abstracción se nota mediante **ABS** (de *abstraction*). Esto es, el ejemplo anterior podría escribirse como:

- Pedidos de Cliente = ABS([Cliente realiza Pedidos])

Es destacable el hecho de que el operador ABS, en determinadas ocasiones, puede inducir a pérdidas de significado y significados incorrectos.

Por pérdida de significado debe entenderse que, al aplicar el operador de abstracción, el significado de un concepto se reduce necesariamente (de ahí el nombre de abstracción). Por ejemplo, supónganse las siguientes proposiciones, cuya representación, por motivos de claridad, se expone de forma gráfica, utilizando la notación del Mapa de Conceptos, en la figura 5.3:

- o Cliente realiza Pedidos
- o Cliente posee Cuenta Contable

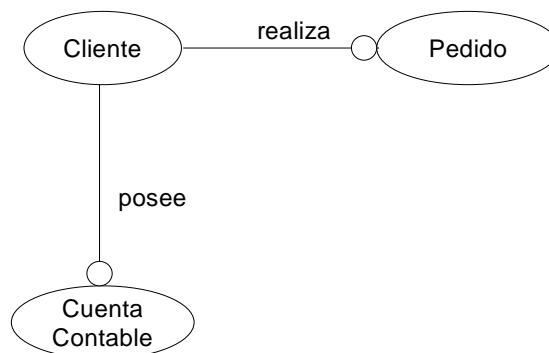


Figura 5.3. Representación gráfica, utilizando el Mapa de Conceptos, de las proposiciones “Cliente realiza Pedido” y “Cliente posee Cuenta Contable”

Si se realiza la operación de abstracción Pedido de Cliente = ABS ([Cliente realiza Pedidos]), se sustituye la proposición “Cliente realiza Pedidos” por el concepto “Pedidos de Cliente”. Ello posee un efecto sobre las asociaciones que poseen los conceptos los involucrados con la proposición abstraída. La operación de abstracción hace migrar las asociaciones hacia el concepto definido por abstracción. Esto es, en formato gráfico, la operación de abstracción produce el efecto mostrado en la figura 5.4.

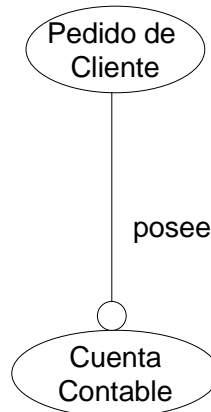


Figura 5.4. Resultado de Pedido de Cliente = ABS ([Cliente realiza Pedido])

El efecto causado por la operación de abstracción ha producido lo siguiente:

- Pérdida de significado. Ya no aparece el concepto de “Cliente”. Ya no se puede afirmar que “Cliente realiza Pedidos” y “Cliente posee Cuenta Contable”.
- Significado incorrecto. “posee Cuenta Contable” era parte del significado de cliente. Ahora han pasado a ser significado de “Pedidos de Cliente”. ¿Es ello correcto? Probablemente, no.

5.4. FORMALISMOS DE REPRESENTACIÓN DEL MCG

El MCG está formado, como ya se ha indicado anteriormente, por tres formalismos de representación, junto con un mecanismo adicional de comunicación con los clientes y, o, usuarios. Éstos formalismos son los siguientes:

- Mapa de Conceptos, de tipo gráfico. Existe en dos versiones: El **Mapa de Conceptos Preliminar** y el **Mapa de Conceptos Exhaustivo**.
- **Diccionario de Identificación o Glosario**, de tipo tabular.
- **Diccionario de Descripción**, de tipo tabular.

El mecanismo auxiliar de comunicación con clientes y, o, usuarios se conoce como **Texto Narrativo**. Como su propio nombre indica, el Texto Narrativo es un mecanismo de representación de tipo textual, que puede obtenerse de forma semiautomática a partir del Mapa de Conceptos o, alternativamente,

del Diccionario de Identificación. Técnicamente, también es posible obtenerlo del Diccionario de Descripción, aunque la complejidad de esto último es mayor. El Texto Narrativo se utiliza como mecanismo de comunicación debido a que no implica ningún tipo de aprendizaje para los clientes y, o, usuarios. No obstante, no es el único mecanismo de comunicación cliente-ingeniero; simplemente, es el medio de comunicación más sencillo de utilizar.

En los epígrafes siguientes se describen en profundidad los citados formalismos de representación.

5.4.1. MAPA DE CONCEPTOS: PRELIMINAR Y EXHAUSTIVO

El Mapa de Conceptos es un formalismo de representación de tipo gráfico, el cual permite representar proposiciones, destacando los conceptos, asociaciones y proposiciones que las componen.

Existen dos modalidades del Mapa de Conceptos: el **Mapa de Conceptos Preliminar** y el **Mapa de Conceptos Exhaustivo**. Ambos se diferencian, a nivel superficial, por la actividad del proceso de MAON en la que se utilizan, pero la diferencia fundamental consiste en los conceptos y asociaciones mayormente utilizadas en cada uno de ellos.

En el **Mapa de Conceptos Preliminar**, se utilizan principalmente conceptos y asociaciones genéricos, esto es, conceptos y asociaciones no predefinidas del MCG.

Por el contrario, en el **Mapa de Conceptos Exhaustivo** se utilizan conceptos y asociaciones genéricos, así como los conceptos y asociaciones especiales.

La razón de que existan ambos tipos de Mapas de Conceptos es coherente con las actividades del proceso de MAON, así como con la realidad del proceso de análisis. En efecto, el proceso de análisis no puede entenderse como un proceso de ganancia progresiva de conocimientos, sino como un proceso fuertemente marcado por una discontinuidad, que es la que se produce entre el Análisis Preliminar y el Análisis Exhaustivo. Durante la primera de estas tareas, los conocimientos son escasos, probablemente incorrectos y probablemente inconexos. Este tipo de conocimientos no exige en absoluto ninguna formalización, y por ello se registran en el Mapa de Conceptos preliminar. Por el contrario, durante la segunda de estas tareas los conocimientos están más claros y estructurados, y son más adecuados para ser formalizados. Por ello, estos conceptos se registran en el Mapa de Conceptos exhaustivo, donde las asociaciones estructurales y de asignación de valores, así como los predicados y funciones, permiten formalizar adecuadamente dichos conocimientos de mayor calidad.

No obstante la diferencia entre el Mapa de Conceptos preliminar y el Mapa de Conceptos exhaustivo, ambos poseen las mismas normas de diagramación, las cuales se exponen en el epígrafe siguiente.

5.4.2. MAPA DE CONCEPTOS: DIAGRAMACIÓN

Las normas de diagramación del Mapa de Conceptos, independientemente de si se trata de un Mapa de Conceptos preliminar o un Mapa de Conceptos exhaustivo, son similares. Estas normas pueden resumirse en:

- Los conceptos se diagraman como óvalos, con la etiqueta en su interior.
- Las asociaciones se diagraman como líneas que unen dos conceptos. A cada línea se le añade la etiqueta correspondiente.
- Cuando una asociación relaciona un concepto y una proposición, o dos proposiciones, se diagraman de la misma forma que una proposición simple, pero rodeando con un cuadro la(s) proposición(es) que cumplen la función de un concepto.

La figura 5.5 representa el caso de diagramación más sencillo, referido únicamente a dos conceptos unidos por una asociación. El símbolo \circ se utiliza para denotar el concepto secundario, esto es, el concepto del que se afirma algo.

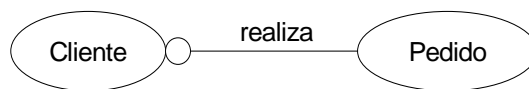


Figura 5.5. Diagramación de una proposición

La diagramación de asociaciones entre un concepto y una proposición se realizaría como indica la figura 5.6.

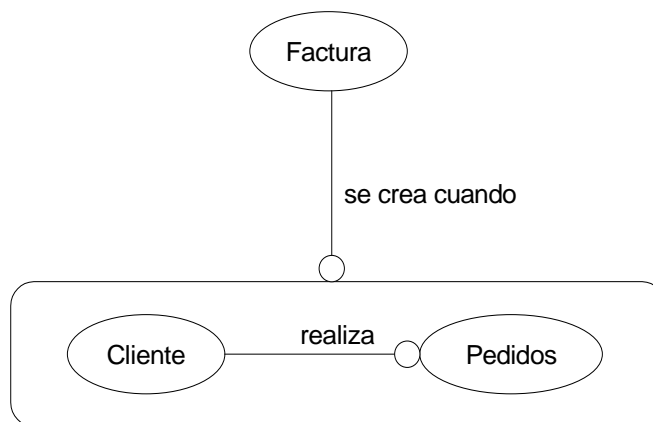


Figura 5.6. Diagramación de asociaciones entre un concepto y una proposición

Y la figura 5.7 muestra el caso de dos proposiciones unidas por una asociación.

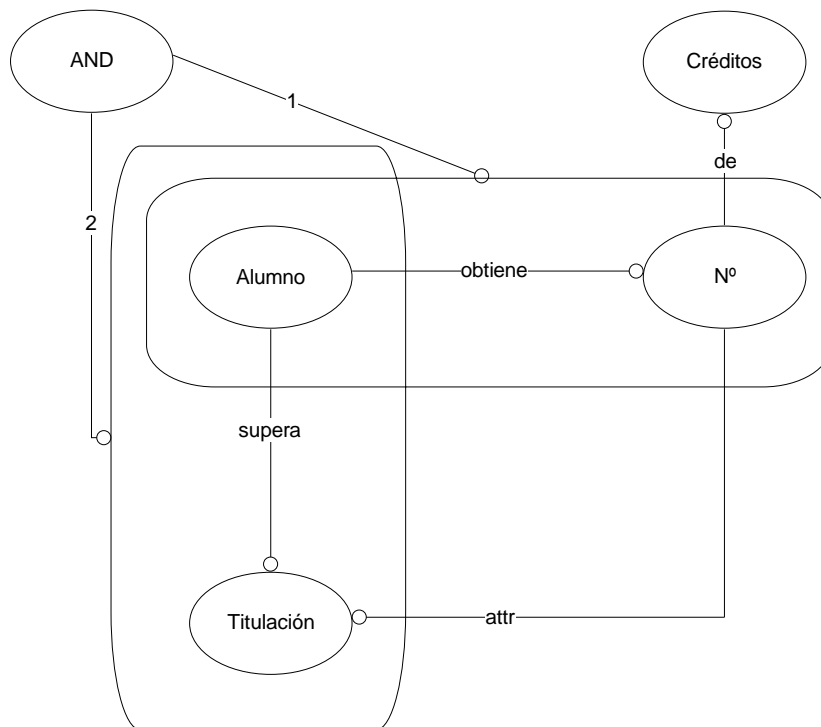


Figura 5.7. Diagramación de dos proposiciones unidas por una asociación

En algunos casos, sobre todo cuando el Mapa de Conceptos es complejo, la notación de la figura 5.7 introduce una “sobrecarga visual”, esto es, hace difícil entender qué se está expresando. Por ello, existe una notación alternativa para la diagramación de proposiciones recursivas, y que consiste en sustituir el óvalo por el símbolo \otimes , tal como muestra la figura 5.8.

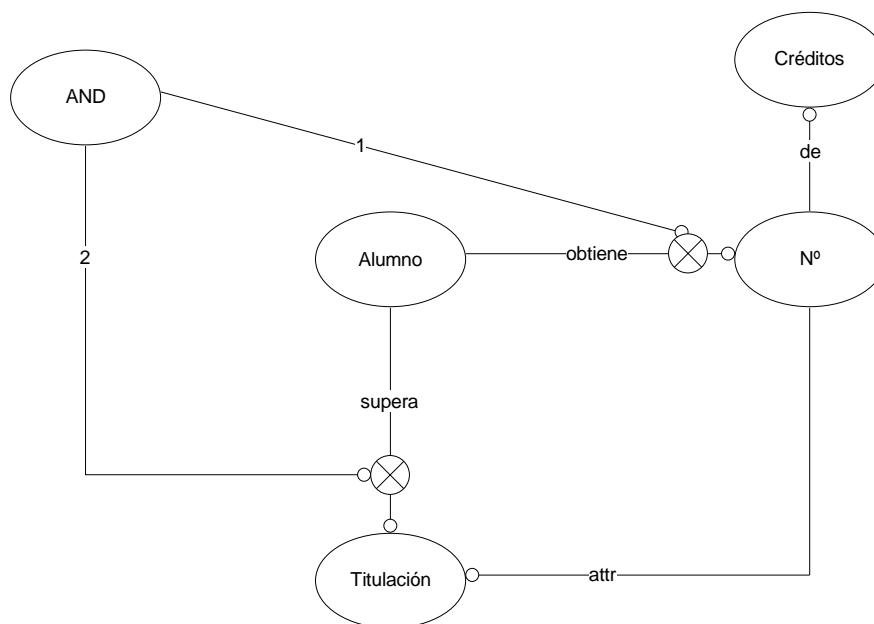


Figura 5.8. Notación alternativa para la diagramación de proposiciones recursivas

Se pueden indicar cuatro precisiones a las normas anteriores, las cuales se refieren a cuando un concepto se refiere a un conjunto, un predicado, una función o una transición:

- Si un concepto se refiere a un conjunto, la etiqueta de dicho concepto se escribe entre { }. Un ejemplo se muestra en la figura 5.9.

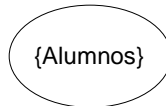


Figura 5.9. Diagramación de un conjunto

- Si un concepto se refiere a un predicado, su etiqueta se antecede con el símbolo **P:**. Como un concepto predicado no posee realmente asociaciones con otros conceptos, sino que dichos conceptos sirven como **operandos** del predicado, la relación entre el operador (predicado) y operando (concepto) se denota mediante una asociación etiquetada con un número que une operador y operando. El número que etiqueta la asociación indica, asimismo, el orden de los operandos. Un ejemplo se muestra en la figura 5.10.

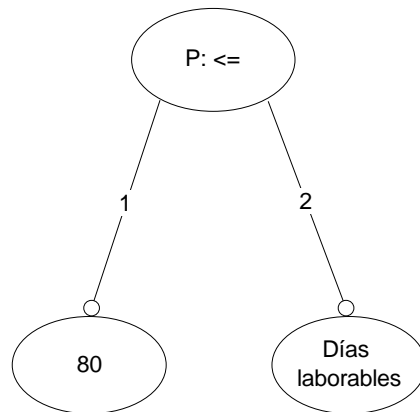


Figura 5.10. Diagramación de un predicado

En caso de que un concepto pueda interpretarse como un predicado sin ambigüedad, se puede prescindir del símbolo **P:**.

- Si un concepto se refiere a una función, su etiqueta se antecede con el símbolo **F:**. El concepto función no posee, igual que en el caso anterior, asociaciones con otros conceptos, sino que dichos conceptos sirven como **operandos** de la función. Por ello, el esquema de diagramación es el mismo que el indicado para el predicado, tal y como muestra la figura 5.11.

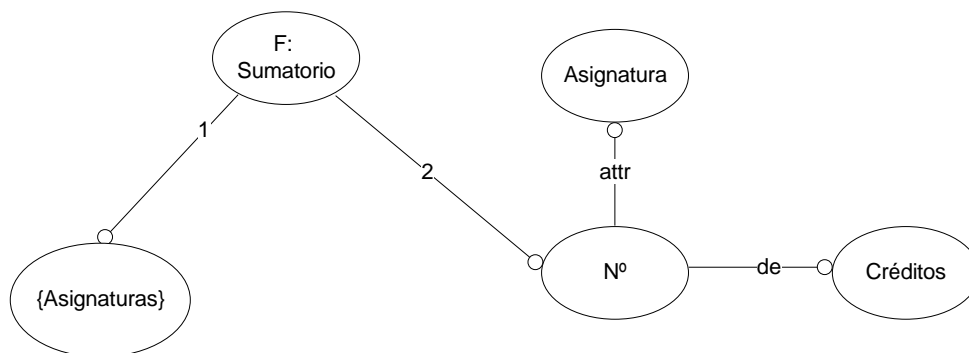


Figura 5.11. Diagramación de una función

En caso de que un concepto pueda interpretarse como una función sin ambigüedad, se puede prescindir del símbolo F:

- Si un concepto se refiere a una transición, su etiqueta se antecede con el símbolo **T**:. El concepto transición, al igual que en el caso de las funciones y predicados, establece asociaciones con otros conceptos que sirven como operandos, esto es, los hechos u objetos que están relacionados temporalmente. El esquema de diagramación es el mismo que el indicado para el predicado y la función, tal y como se muestra en la figura 5.12.

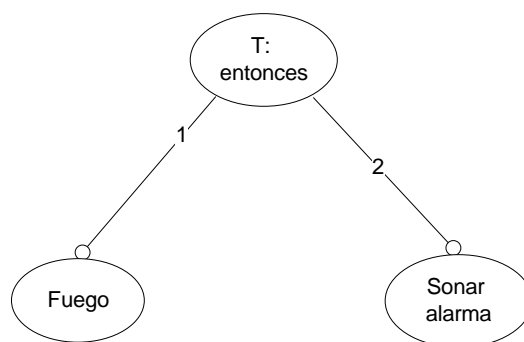


Figura 5.12. Diagramación de una transición

Ya, para finalizar, únicamente resta indicar las normas de diagramación de los operadores EQ, DEF y ABS:

- El operador EQ se diagrama exactamente igual que una asociación, aunque sin utilizar el símbolo O, tal y como muestra la figura 5.13.

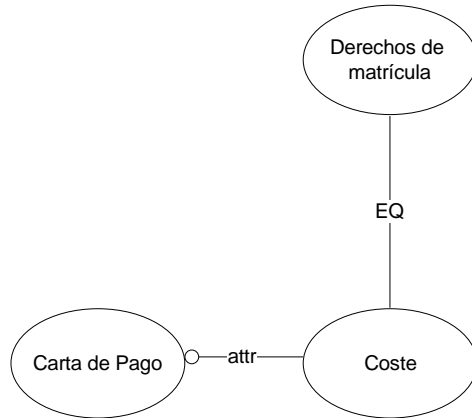


Figura 5.13. Diagramación del operador EQ

- Tanto el operador DEF como el operador ABS se diagraman como un concepto, con la única salvedad de que éste aparece sombreado. No existe en el Mapa de Conceptos otra forma de indicar los conceptos o proposiciones contenidos dentro de la definición o abstracción que utilizar un Mapa de Conceptos suplementario para cada definición u abstracción. Un ejemplo se muestra en la figura 5.14.

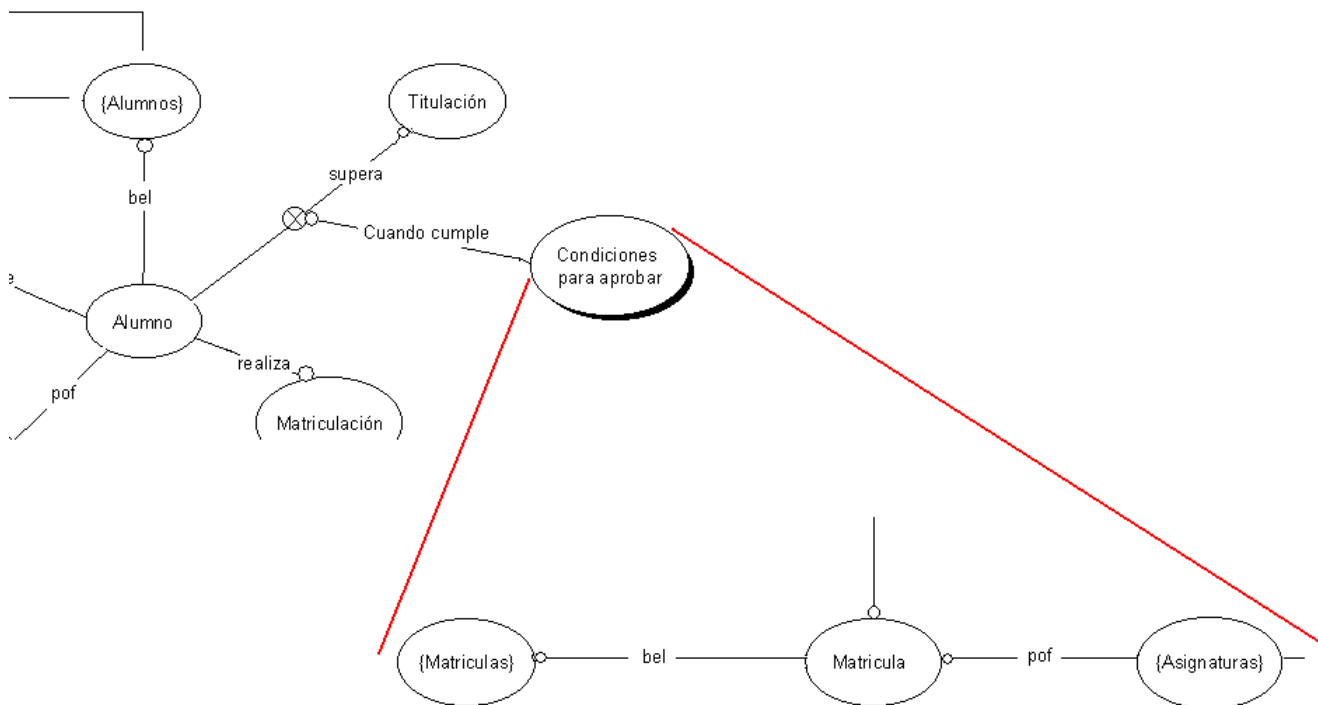


Figura 5.14. Diagramación de una abstracción. Una definición se diagrama de igual modo

5.4.3. DICCIONARIO DE IDENTIFICACIÓN

El Diccionario de Identificación (DI) es un formalismo de representación de tipo tabular, que permite registrar contenidos muy simples. El formato del DI se presenta en la Tabla 5.1.

Concepto	Descripción

Tabla 5.1. Formato del Diccionario de Identificación

La primera columna de la Tabla 5.1 (Concepto) sirve para registrar cada conceptos del que se predica algo. Por ejemplo, dada la proposición:

Alumno (curso) Asignaturas

El concepto “alumno” se registraría en la columna “Concepto”.

La segunda columna de la Tabla 5.1 (Descripción) sirve para registrar lo que se predica de un concepto. Dado el mismo ejemplo anterior, “curso asignaturas” se registraría en la columna “Descripción”. De esta forma, el DI (solo para la proposición anteriormente indicada) sería de la forma indicada en la Tabla 5.2.

Concepto	Descripción
Alumno	Cursa asignaturas

Tabla 5.2. Ejemplo de Diccionario de Identificación

El DI también se denomina, en el MCG, con el nombre de **glosario** debido a que presenta un concepto junto con todos los predicables del mismo (lo que coincide con la definición de significado en el MCG). Por ello, el DI es una herramienta que no resulta extraña a ningún ingeniero, puesto que es universalmente utilizada durante las fases tempranas del análisis.

5.4.4. DICCIONARIO DE DESCRIPCIÓN

El Diccionario de Descripción (DD) es, al igual que el DI, un formalismo de representación de tipo tabular. Sin embargo, las similitud del DD con el DI se restringen a lo antedicho, ya que la información se registra de una forma completamente diferente. El DD se compone de tres partes diferenciadas:

1. Declaraciones.
2. Definiciones.
3. Proposiciones.

En lo que sigue se describen cada una de las partes constituyentes, anteriormente mencionadas.

5.4.4.1. DECLARACIONES

Como se ha mencionado anteriormente, los conceptos en el MCG pueden referirse a un individuo particular (un coche, una persona, etc.) o a un conjunto de individuos que se aprecian similares. Esta distinción entre individuo y conjunto, clase o tipo aporta un grado adicional de formalización al problema del usuario bajo estudio. En el DD, dicha distinción entre conjuntos e individuos se realiza en la parte de Declaraciones. La parte de declaraciones del DD posee el formato indicado en la Tabla 5.3.

Declaraciones			
Conjuntos:	(1)		
	...		
Subconjuntos:	(2)	subs	(3)

Individuos:	(4)	bel	(5)

Tabla 5.3. Formato de la parte de Declaraciones del Diccionario de Descripción

La información registrada en cada una de las celdas de la Tabla 5.3. es la siguiente:

- Los conjuntos (1).
- Si se hubieran definido subconjuntos, se registrarían en (2). Todo subconjunto debe estar incluido en un conjunto declarado anteriormente en (1). Dicho conjunto se indica en (3).
- Los individuos se registran en (4). Un individuo debe pertenecer a un conjunto o subconjunto. El conjunto o subconjunto se registra en (5).

5.4.4.2. DEFINICIONES

Como ya se ha indicado anteriormente, existen dos tipos de conceptos en el MCG: los conceptos genéricos y los conceptos especiales. Lo mismo ocurre con las asociaciones, las cuales pueden ser, igualmente, genéricas o especiales. En el DD, ambos tipos de conceptos y asociaciones se registran separadamente.

La parte de Definiciones del DD está destinada a registrar los conceptos y asociaciones especiales⁴, esto es, los conceptos y asociaciones que hacen referencia a:

- Aspectos estructurales (asociaciones attr, spec, rel y pof)
- Procedurales (asociación val)
- Predicados, funciones y transiciones.

Desde el punto de vista del formato, la parte de declaraciones del DD es trivial. Dicho formato se indica en la Tabla 5.4.

Definiciones	
Índice	Definición

Tabla 5.4. Formato de la parte de Definiciones del Diccionario de Descripción

Como puede observarse, el formato de las definiciones del Diccionario de Descripción es libre, debido a que la variedad de las mismas (sobre todo debido a la posibilidad de predicados, funciones y transiciones) hace excesivamente complicado (y de poca utilidad, lo cual es más importante) la definición de un formato predeterminado.

La única restricción de formato que se ha introducido es la inclusión de un número de índice que sirve para identificar unívocamente las distintas definiciones del modelo, ya que de esta forma es posible referirse a una proposición por su número. Se utiliza una “D” antepuesta al número de índice para identificar una definición del modelo.

5.4.4.3. PROPOSICIONES

Todas las proposiciones definidas mediante conceptos y asociaciones genéricas se registran en la parte de proposiciones del DD. El formato de esta parte del DD se muestra en la Tabla 5.5. Cada una de las columnas de la Tabla 5.5 sirve para registrar un componente de una proposición, de la forma siguiente:

- Índice: Sirve para numerar las unívocamente las proposiciones. Se utiliza una “P” antepuesta al número de índice para identificar cualquier proposición del modelo.
- Concepto-1: Permite registrar el concepto principal.
- Asociación: Permite registrar la asociación entre dos conceptos de una proposición.

⁴ Desde un punto de vista estricto, cualquiera de los conceptos o asociaciones indicadas anteriormente podrían registrarse en la última parte del DD (proposiciones, descrita en el siguiente epígrafe). Sin embargo, a diferencia de los conceptos y asociaciones genéricas, los conceptos y asociaciones especiales poseen habitualmente una interpretación bien definida. Ello será relevante a la hora de aplicar la Técnica de Identificación del Modelo Conceptual Idóneo, en el capítulo 7.

- Concepto-2: Permite registrar el concepto subordinado.

Proposiciones			
Índice	Concepto – 1	Asociación	Concepto – 2

Tabla 5.5. Formato de la parte de Proposiciones del Diccionario de Descripción

5.4.5. TEXTO NARRATIVO

El Texto Narrativo (TN) es el formalismo de representación más sencillo de todos los expuestos, debido a que utiliza texto en lenguaje natural, de una forma semi-libre. La semi-libertad se refiere a que no es posible utilizar toda la riqueza del lenguaje natural, sino únicamente las construcciones que permite el MC o, alternativamente, el DI o DD.

5.5. RELACIÓN ENTRE LOS FORMALISMOS DE REPRESENTACIÓN DEL MCG

Tres de los cuatro formalismos de representación del MCG, en concreto el Mapa de Conceptos, el Diccionario de Identificación y el Diccionario de Descripción son similares, esto es, las capacidades de representación del MC, DI y DD son prácticamente equivalentes, al menos desde un punto de vista *naïve*.

En la práctica, sin embargo, los formalismos gráficos se manejan de distinta forma [Vessey, 1991] que los tabulares y textuales, por lo que MC, DI y DD se perciben como formalismos distintos y con distintas capacidades de representación y manipulación, aunque en el fondo se trate del mismo formalismo, expresado bajo formas distintas y a distintos niveles de refinamiento.

Los siguientes epígrafes se dedicarán a precisar la relación existente entre los formalismos MC, DI y DD, así como el procedimiento a aplicar para, a partir de la información recogida en un formalismo determinado (por ejemplo, el Mapa de Conceptos), derivar otro formalismo (por ejemplo, el Diccionario de Descripción).

5.5.1. RELACIÓN ENTRE EL MAPA DE CONCEPTOS Y EL DICCIONARIO DE IDENTIFICACIÓN

Como se ha indicado anteriormente, el Diccionario de Identificación (DI) posee una potencia expresiva similar a la del Mapa de Conceptos (MC). Ello se debe a que el formato del DI es isomorfo al MC, esto es, ambos registran la misma información con distinto formato.

A continuación, se ilustrará con un ejemplo el isomorfismo existente entre ambos formalismos de representación. Más concretamente, se tratará el isomorfismo existente entre el DI y el MC preliminar, ya

que ambos se utilizan conjuntamente, esto es, el DI y el MC exhaustivo son utilizados en distintos pasos (Análisis Superficial y exhaustivo, respectivamente) de MAON. Supóngase, por lo tanto, el fragmento del MC preliminar mostrado en la figura 5.15.

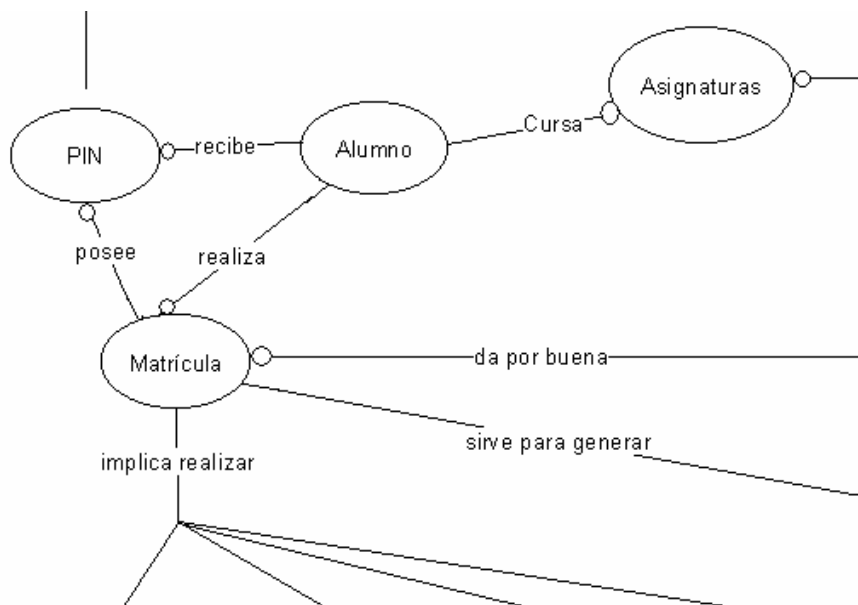


Figura 5.15. Fragmento de un Mapa de Conceptos preliminar

Existe un isomorfismo entre el MC preliminar y el DI debido a que: (1) todos los conceptos que aparecen en el MC preliminar se registran en la columna “Concepto” del DI, tal y como muestra la figura 5.16.

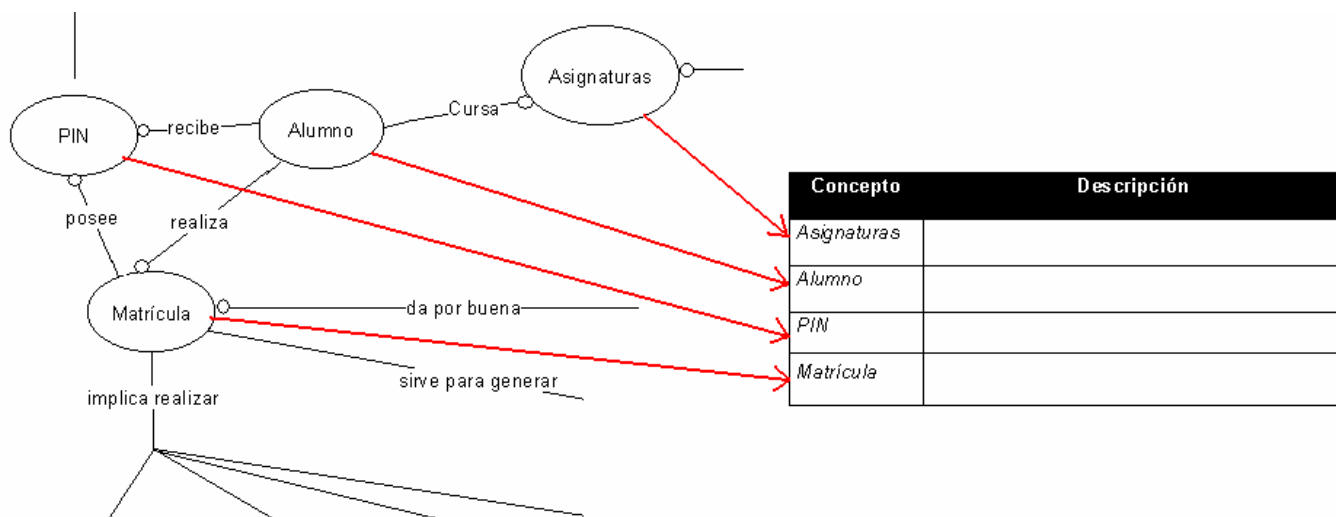


Figura 5.16. Traslación de los conceptos desde el Mapa de Conceptos preliminar al Diccionario de Identificación

Y (2), todos los predicados del MC preliminar se registran en la columna “Descripción”, asociados al

concepto que califican, tal y como muestra la figura 5.17.

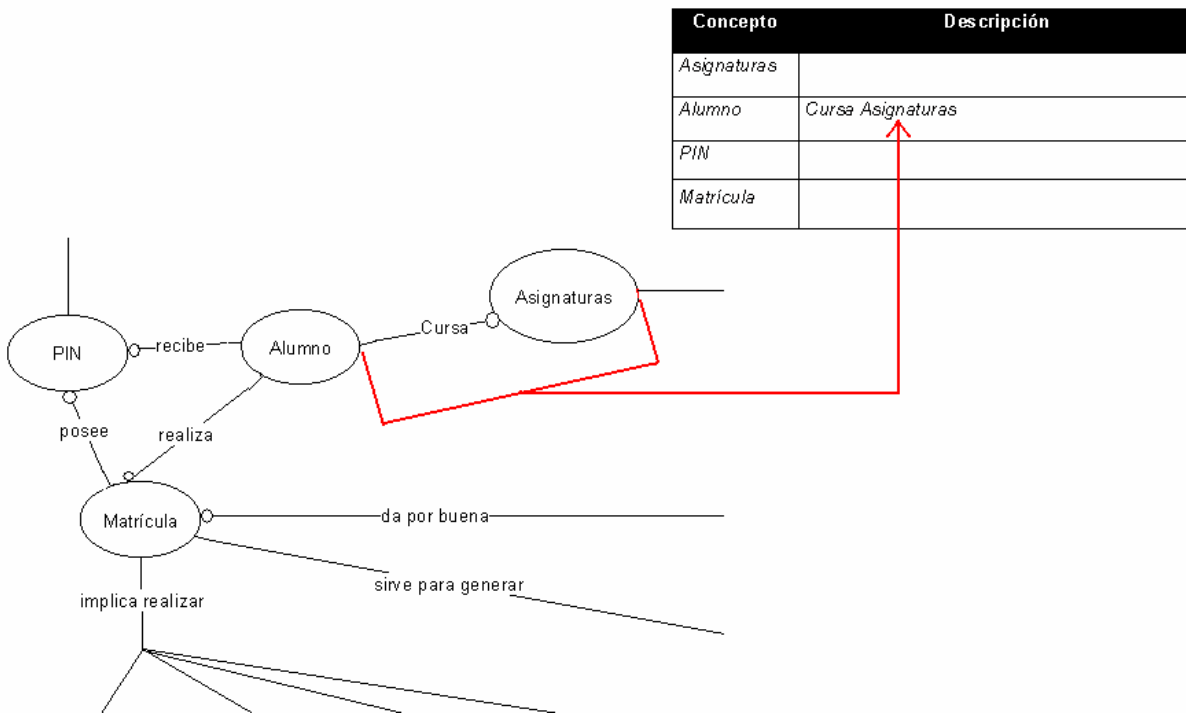


Figura 5.17. Traslación de los predicados desde el Mapa de Conceptos preliminar al Diccionario de Identificación

La relación existente entre el MC preliminar y el DI produce que cada fila del DI contenga una proposición. De esta forma, el DI puede entenderse como una versión lineal del MC preliminar. Aun así, es necesario disponer de ambos formalismos de representación en el MCG ya que, como se ha indicado al inicio de la presente sección, el formato de representación de cada formalismo altera o modula la facilidad de uso que se desee dar al mismo.

No obstante, aunque el MC preliminar y el DI son isomorfos, ello no implica que, para cualquier problema del usuario, ambos formalismos registren la misma cantidad de información. Existen dos razones que justifican esta afirmación:

1. El DI registra la información de forma más compacta que el MC preliminar. Por ello, para problemas no triviales, es preferible registrar en el DI toda la información identificada durante el Análisis Preliminar, reservando el MC preliminar para representar los conceptos que se creen más importantes, así como las asociaciones más destacadas en las que toman parte.
2. En los problemas no triviales, el ingeniero identifica una gran cantidad de información que inicialmente no parece importante y finalmente se manifiesta inútil. Es preferible relegar esta información al relativo “olvido” del DI y reservar el MC preliminar para focalizar el estudio en los aspectos principales del problema del usuario.

Para el mantenimiento del DI, así como su consistencia con el MC preliminar, lo preferible es utilizar algún tipo de herramienta informática, básicamente porque el volumen de la información manejada es considerable en problemas no triviales y, por otra, es muy fácil cometer errores por la enorme cantidad de detalle que se puede llegar a registrar.

5.5.2. RELACIÓN ENTRE EL MAPA DE CONCEPTOS Y EL DICCIONARIO DE DESCRIPCIÓN

Al igual que el DI, el Diccionario de Descripción (DD) es isomorfo al MC, en concreto, y dado que el DD se utiliza durante el paso de Análisis Exhaustivo, el DD es isomorfo al MC exhaustivo. Sin embargo, y a diferencia del DI, la construcción del DD es una tarea compleja, tanto por el formato que posee, como por la cantidad y calidad de la información que registra.

En la aplicación práctica de MAON, dada la complejidad de construcción del DD, una de las mejores alternativas es realizar el Análisis Exhaustivo utilizando el MC exhaustivo y, con posterioridad, derivar el DD a partir de dicho MC exhaustivo. Ello es posible gracias al isomorfismo existente entre ambas representaciones.

El DD se construye, por lo tanto, una vez se ha finalizado el MC exhaustivo, mediante la aplicación de una serie de reglas. En los epígrafes siguientes se describirán dichas reglas. Nótese, no obstante, que lo automático de tales reglas permite algún tipo de implementación automatizada, la cual favorece sobremanera la confección del DD.

5.5.2.1. CONFECCIÓN DE LA PARTE DE DECLARACIONES DEL DICCIONARIO DE DESCRIPCIÓN

La confección de la parte de declaraciones se realiza siguiendo las reglas indicadas en la Tabla 5.6.

Nótese un suceso que ocurre con mucha frecuencia durante la construcción del DD. En el MC exhaustivo, todo concepto se considera distinto de otro aunque tenga el mismo nombre. Por esta razón, al confeccionar el DD es necesario reescribir con mucha frecuencia los nombres de conceptos (no ocurre lo mismo con las asociaciones, ya que van ligadas a conceptos). Se ha optado en este trabajo de Tesis en realizar la reescritura utilizando un número distintivo al final del nombre del concepto (como en el caso de Asignatura_1 y Asignatura_2).

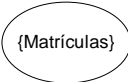
Mapa de Conceptos	Declaraciones del Diccionario de Descripción
	Conjuntos: Matrícula

Tabla 5.6. Confección de la parte de declaraciones del Diccionario de Descripción (parte 1)

Mapa de Conceptos	Declaraciones del Diccionario de Descripción
	Subconjuntos: Asignaturas_2 subs Asignaturas_1
	Individuos: Matrícula_2 ∈ Matrícula_1

Tabla 5.6. Confección de la parte de declaraciones del Diccionario de Descripción (parte 2)

5.5.2.2. CONFECCIÓN DE LA PARTE DE DEFINICIONES DEL DICCIONARIO DE DESCRIPCIÓN

La confección de la parte de definiciones se realiza siguiendo las reglas indicadas en la Tabla 5.8.

Mapa de Conceptos	Definiciones del Diccionario de Descripción
	Obligatorias Spec Asignaturas
	Asignaturas pof Matrícula
	Profesor rel Asignatura
	Calificación attr Asignatura
	Calificación val Aprobado

Tabla 5.7. Confección de la parte de definiciones del Diccionario de Descripción

Para el caso de los predicados, funciones y transiciones, la transformación es más complicada, debido a que pueden poseer una complejidad arbitraria. En líneas generales, un predicado, función o transición (en el MCG) es una estructura arborescente de la forma indicada en la figura 5.18.

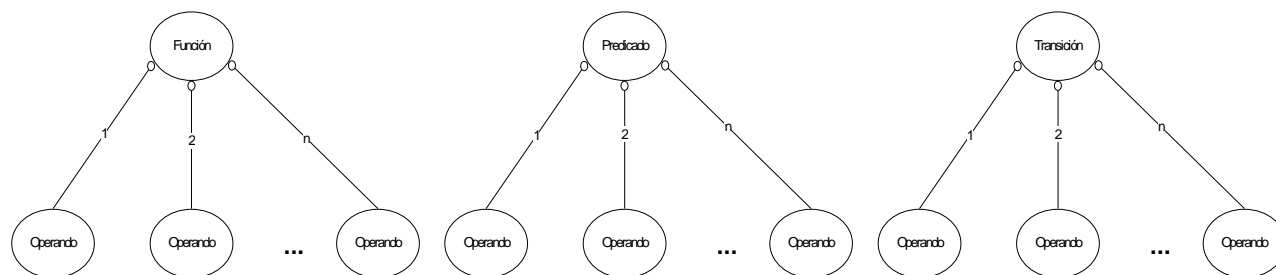


Figura 5.18. Estructura general de las funciones y predicados en el MCG

Dicha estructura arborescente (la cual puede ser una estructura arborescente de múltiples niveles, definida de forma recursiva), se deriva al DD de la forma siguiente:

Predicab(Operando, Operando, ..., Operando)

Función(Operando, Operando, ..., Operando)

Transición(Operando, Operando, ..., Operando)

Utilizando los índices 1, 2, ..n para ordenar los operandos en el predicado, función o transición. Nótese que la notación para las funciones es completamente habitual (en la mayoría de los casos). No obstante, la notación de los predicados (y transiciones) puede ser bastante difícil de entender; por ejemplo:

$\gt=(5, 4)$ ó $+(3, 7)$

no son fáciles de interpretar en la mayoría de los casos. Para ello, cuando el DD vaya a ser utilizado por seres más o menos humanos, es preferible utilizar la notación infija para los predicados (y funciones) que la admitan; por ejemplo:

$5 \gt= 4$ ó $3 + 7$

Nótese, adicionalmente, que cualquier función o predicado del MCG es arbitrario, esto es, **no existe un catálogo de funciones o predicados**. No obstante, ello no supone una limitación en la medida en que el ingeniero defina que un concepto determinado se corresponde con un predicado o función. Dicha definición se produce **antes** de finalizar la tarea de Análisis Preliminar, en la medida en que son fácilmente reconocibles.

5.5.2.3. CASOS ESPECIALES EN LA CONFECCIÓN DE LA PARTE DE DEFINICIONES DEL DICCIONARIO DE DESCRIPCIÓN

En algunos casos, los predicados, funciones y transiciones poseen operandos que son proposiciones, como es el caso mostrado en la figura 5.19. En este caso, las proposiciones estarán registradas en la parte de Proposiciones o Definiciones del DD. Para registrar adecuadamente el predicado, función o transición, se utilizará el índice de la proposición como operando.

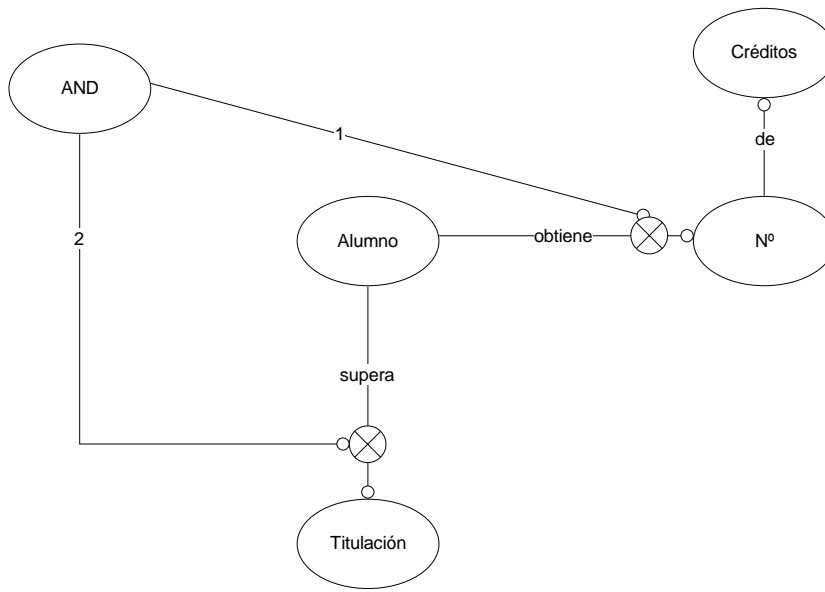


Figura 5.19. Predicado con operandos definidos sobre predicados construidos con asociaciones genéricas

Para este predicado, las proposiciones Alumno obtiene Nº y Alumno supera Titulación deberán estar registradas en la parte de Proposiciones del DD. Supóngase que dichas proposiciones poseen índices P6 y P7, respectivamente. Entonces el predicado se derivaría de la forma siguiente:

$$AND(P6, P7)$$

O, de nuevo para el consumo humano:

$$P6 AND P7$$

5.5.2.4. CONFECCIÓN DE LA PARTE DE PROPOSICIONES DEL DICCIONARIO DE DESCRIPCIÓN

La confección de la parte de Proposiciones del DD se realiza siguiendo las reglas indicadas en la Tabla 5.8.

Mapa de Conceptos	Proposiciones del Diccionario de Descripción
	P1 Permite consultar PIN Preacta

Tabla 5.8. Confección de la parte de Proposiciones del Diccionario de Descripción

5.5.2.5. EFECTOS DEL OPERADOR DE DEFINICIÓN SOBRE EL DICCIONARIO DE DESCRIPCIÓN

Una definición es, en el MCG, simplemente una regla de reescritura de proposiciones. Por lo tanto, para confeccionar el DD a partir del MC exhaustivo, es simplemente necesario reescribir la definición con la proposición o proposiciones de las que se compone. Por, ejemplo, sea el MC exhaustivo de la figura 5.20.

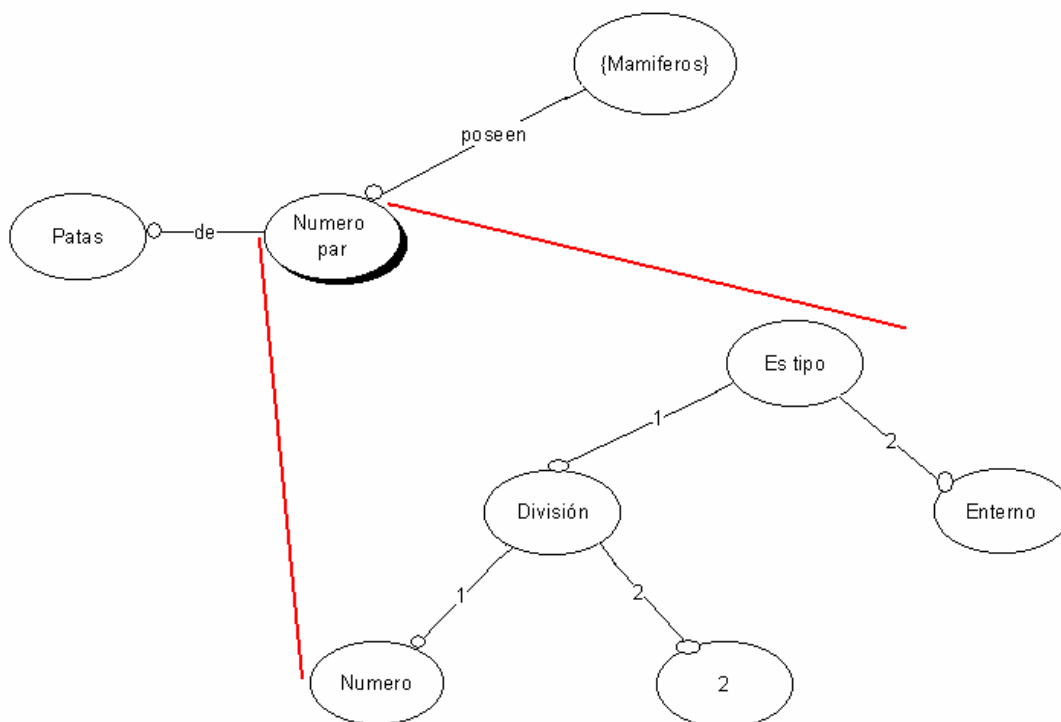


Figura 5.20. Mapa de Conceptos exhaustivo que contiene una definición

Este MC exhaustivo es equivalente, tras la reescritura, al MC exhaustivo mostrado en la figura 5.21.

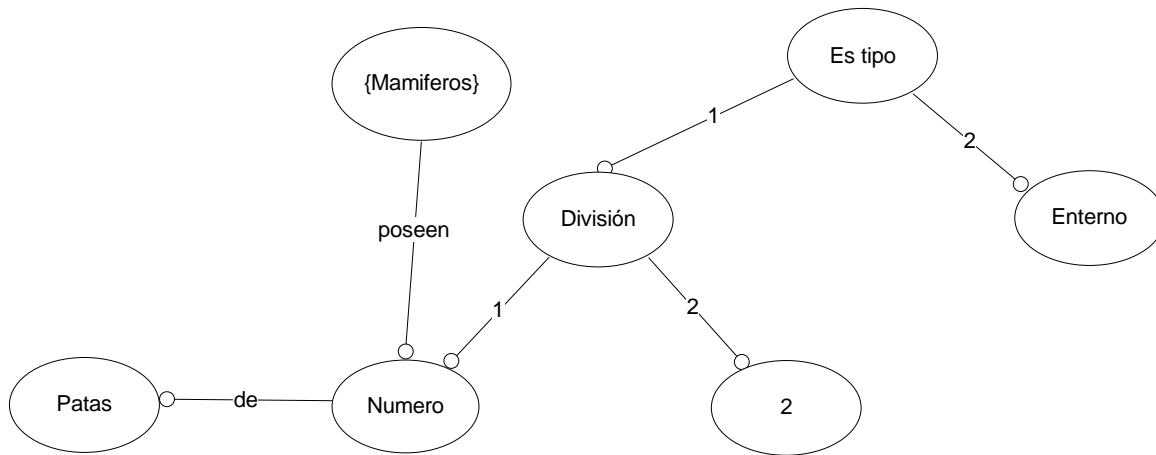


Figura 5.21. Mapa de Conceptos exhaustivo una vez deshecha la definición

Y, por lo tanto, para derivar de este MC exhaustivo el DD, simplemente se seguirían las reglas enunciadas anteriormente.

5.5.2.6. EFECTOS DEL OPERADOR DE ABSTRACCIÓN SOBRE EL DICCIONARIO DE DESCRIPCIÓN

Hasta el momento, se han expuesto las normas de derivación usuales para construir el DD. No obstante, no se ha tratado la complicación que añade la existencia del operador de abstracción, el cual, como ya se ha expuesto, permite dar profundidad al MCG, pero no conserva las asociaciones entre niveles.

El operador de abstracción afecta a las tres partes del DD, de forma distinta. En lo que sigue se expone cómo afecta a cada una de dichas partes y qué acciones hay que realizar para derivar el DD.

5.5.2.6.1. DECLARACIONES

Un concepto construido mediante el operador de abstracción puede ocultar la declaración de conjuntos, subconjuntos e individuos. Tómese como ejemplo la figura 5.22.

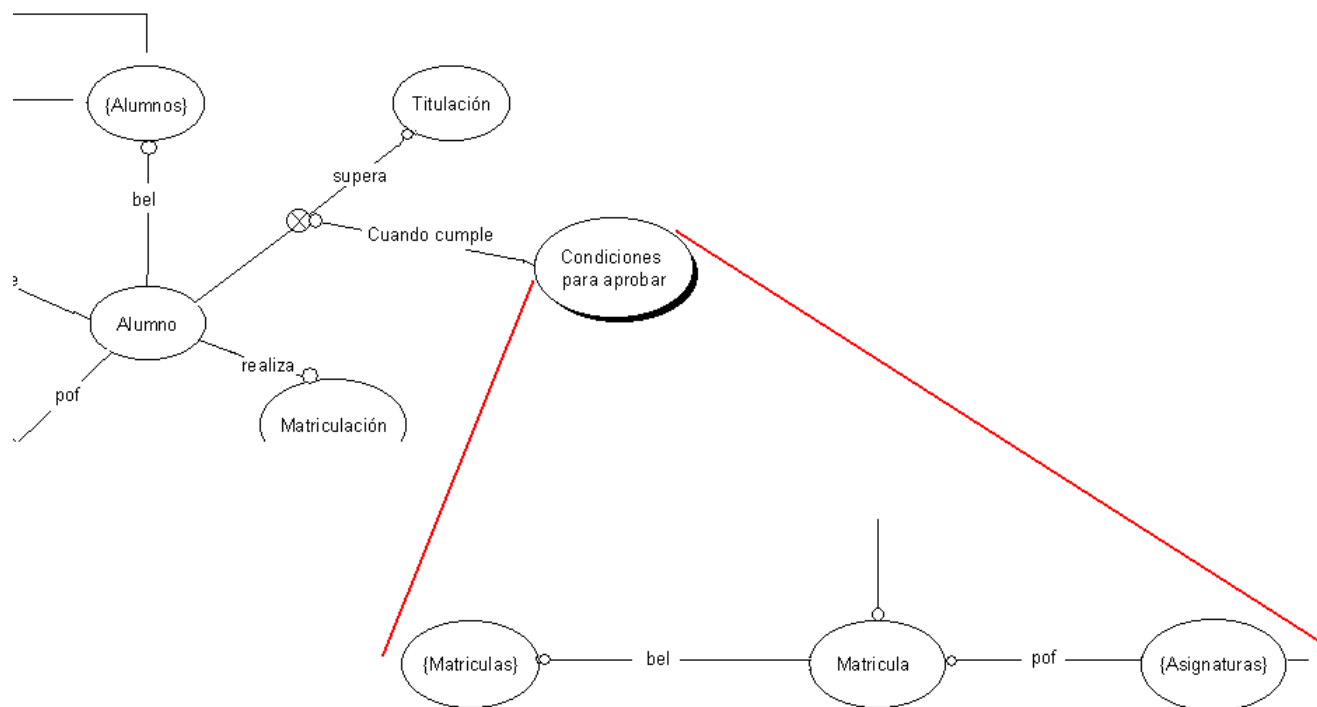


Figura 5.22. Abstracción que oculta declaraciones

En dicho caso, todos los conjuntos, subconjuntos e individuos que aparezcan dentro de la abstracción deben registrarse en el DD, tal y como muestra la figura 5.23.

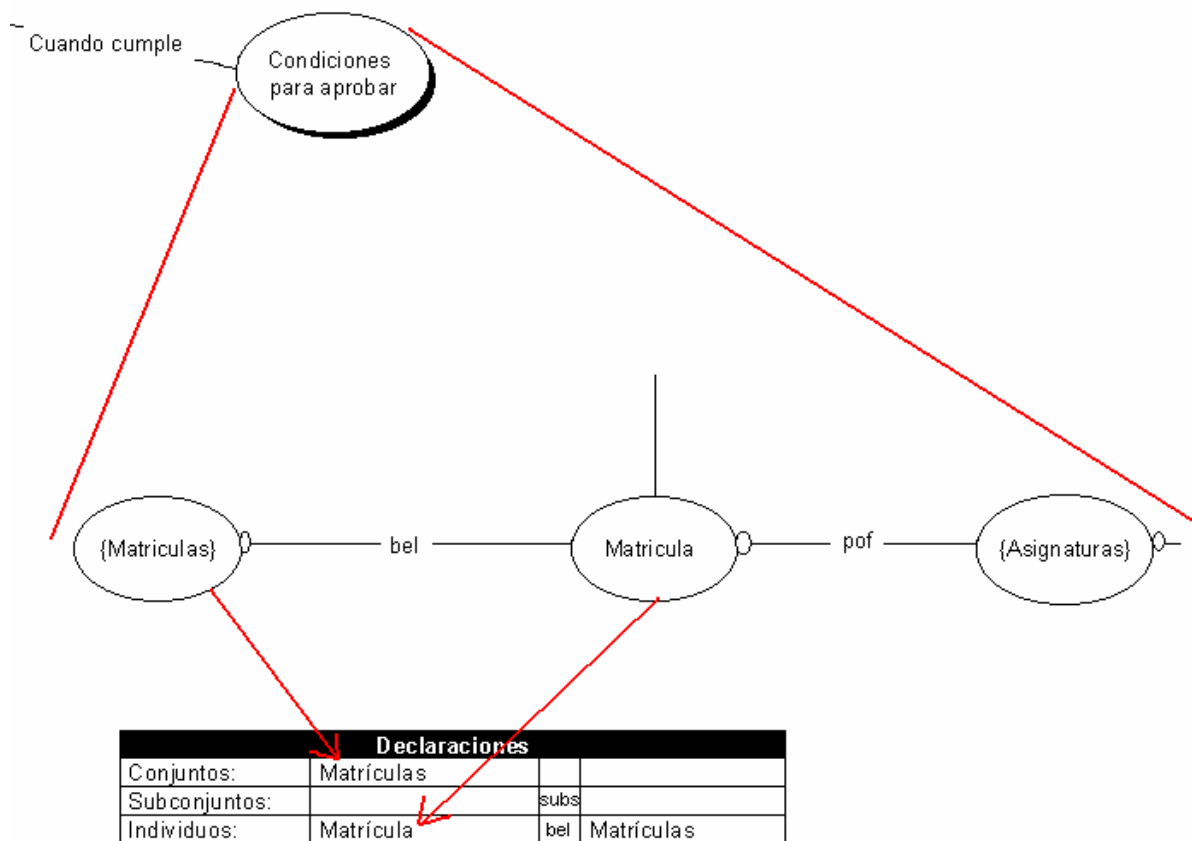


Figura 5.23. Registro de las declaraciones en el Diccionario de Descripción

5.5.2.6.2. DEFINICIONES

Una abstracción puede intervenir en proposiciones construidas con la asociación pof. No es posible, por el contrario, que intervenga en asociaciones spec, rel, attr o val, ya que la semántica del modelo, en estos caso, sería muy compleja de entender y, probablemente, de escasa utilidad. Un ejemplo se muestra en la figura 5.24.

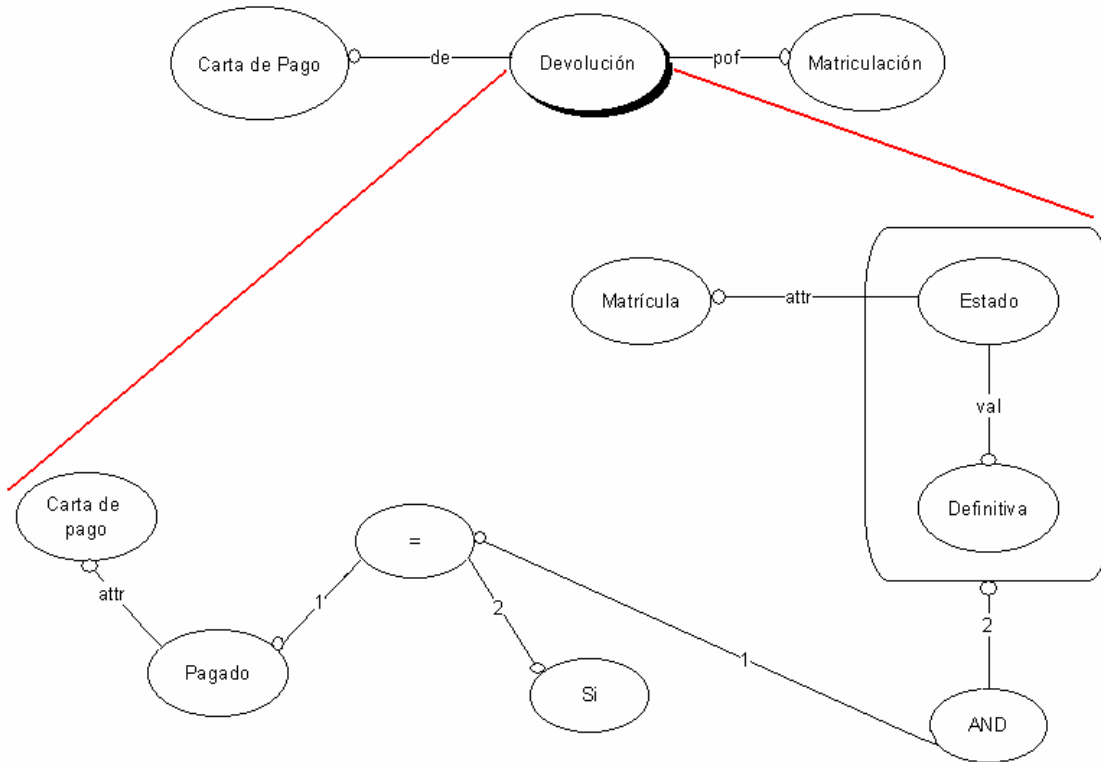


Figura 5.24. Abstracción que forma una proposición mediante la asociación pof

En este caso, se registraría en el DD que “Matriculación” contiene (pof) todos los elementos de la abstracción. Este caso es similar, por ejemplo, a lo que ocurre durante la descomposición de un DFD; todos los procesos de nivel inferior están incluidos (pof) en el proceso de nivel superior descompuesto.

No obstante, existe una restricción durante la derivación del DD, que consiste en que las asociaciones (dentro de la abstracción) spec, rel, pof y attr no son consideradas. Esta restricción se debe a que spec, pof y attr determinan aspectos estructurales que son válidos en todo el modelo. No tendría sentido que formaran parte de un concepto mediante la asociación pof, ya que se verificarían de todas formas. Con esta restricción, el ejemplo de la figura 5.24 se derivaría al DD de la forma indicada en la figura 5.25.

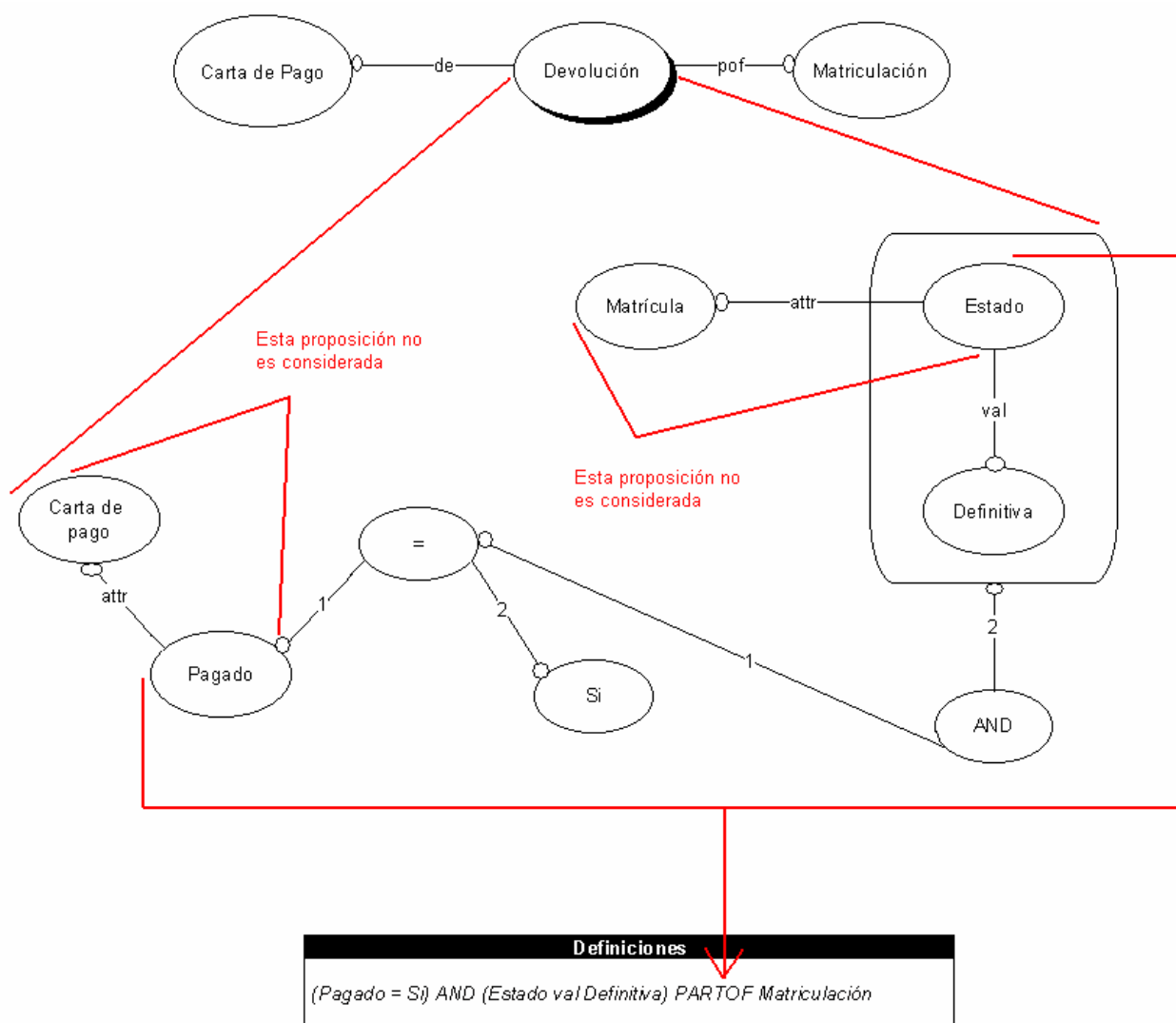


Figura 5.25. Derivación al Diccionario de Descripción

Asimismo, es posible que una abstracción funcione como operador de un predicado (aunque nunca de una función, ya que la semántica sería muy compleja). El mecanismo de la derivación, en este caso, es muy similar al anterior. Véase el ejemplo de la figura 5.26. Aunque el ejemplo de la figura 5.26 está muy forzado, permite distinguir las características principales de la derivación: Al igual que en el caso anterior, todas las proposiciones incluidas dentro de la abstracción forman parte del predicado. También al igual que en el caso anterior, no se consideran las asociaciones (dentro de la abstracción) spec, rel, pof y attr, por la complejidad que implicaría en la semántica del modelo. De esta forma, la derivación al DD sería la siguiente:

AND(Alumno supera Titulación, (Nº de Créditos, AND(Alumno obtiene Nº, Alumno supera Titulación)))

Las proposiciones indicadas anteriormente (Alumno supera Titulación, Nº de Créditos, Alumno obtiene Nº, Alumno supera Titulación), al ser proposiciones construidas mediante asociaciones genéricas,

deberán estar registradas en la parte de Proposiciones del DD. Por ello, en lugar de registrar la proposición completa, se utilizará su índice. Suponiendo que este índice es P1, P2, P3 y P4, respectivamente, la definición sería de la forma siguiente (ya en formato de consumo humano):

$$P1 \text{ AND } (P2, P3 \text{ AND } P4)$$

Evidentemente, el significado de P2, P3 AND P4 es confuso. Ello se debe a que P2 no posee una interpretación, por lo que el predicado P1 AND (P2, P3 AND P4) deberá evaluarse posteriormente, una vez que P2 (y, en general, que P1-P4) hayan sido interpretados.

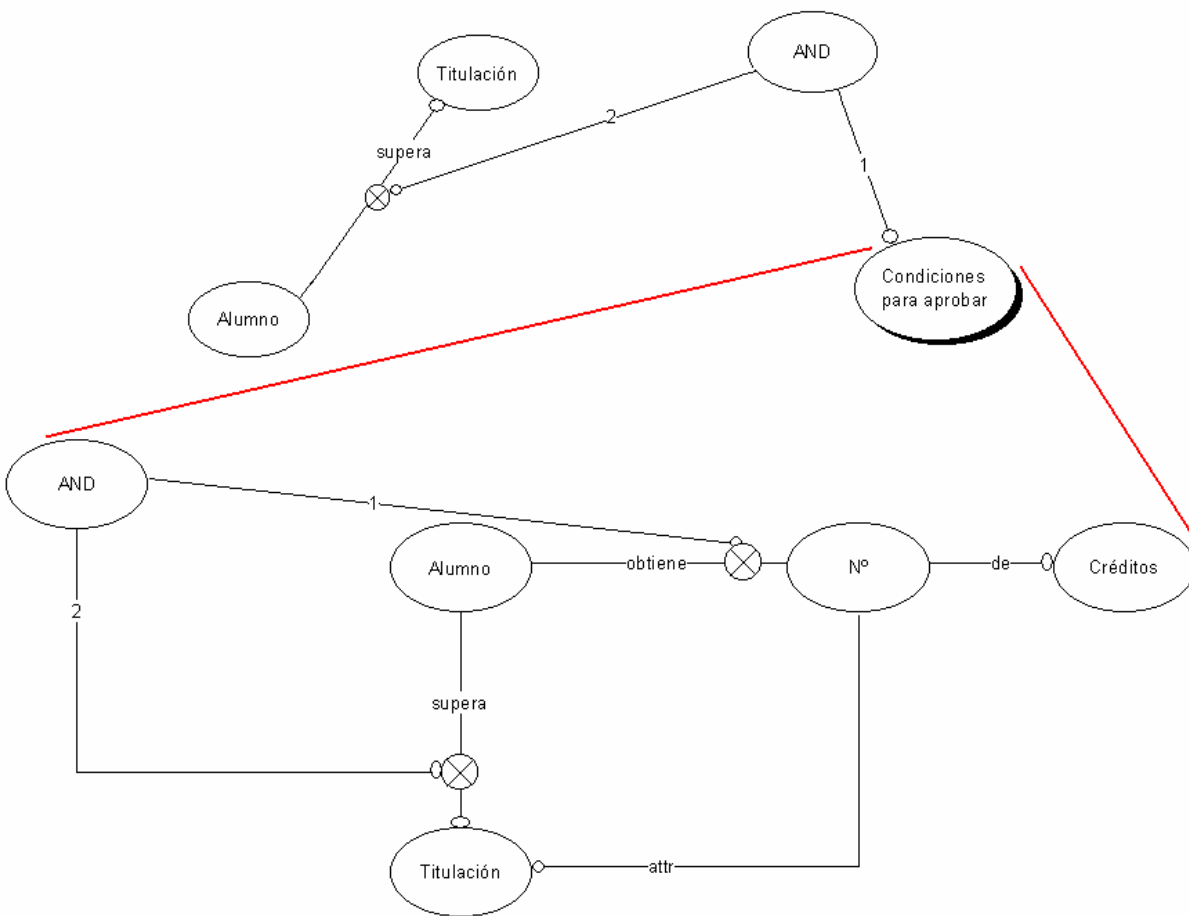


Figura 5.26. Abstracción que forma parte de un predicado

5.5.2.6.3. PROPOSICIONES

Una abstracción puede intervenir en proposiciones construidas con conceptos y asociaciones genéricas. En este caso, la derivación debe realizarse mediante la conexión de los conceptos y proposiciones abstraídos con los conceptos y proposiciones de mayor nivel, no abstraídos. Supóngase el ejemplo de la figura 5.27, ya presentado anteriormente.

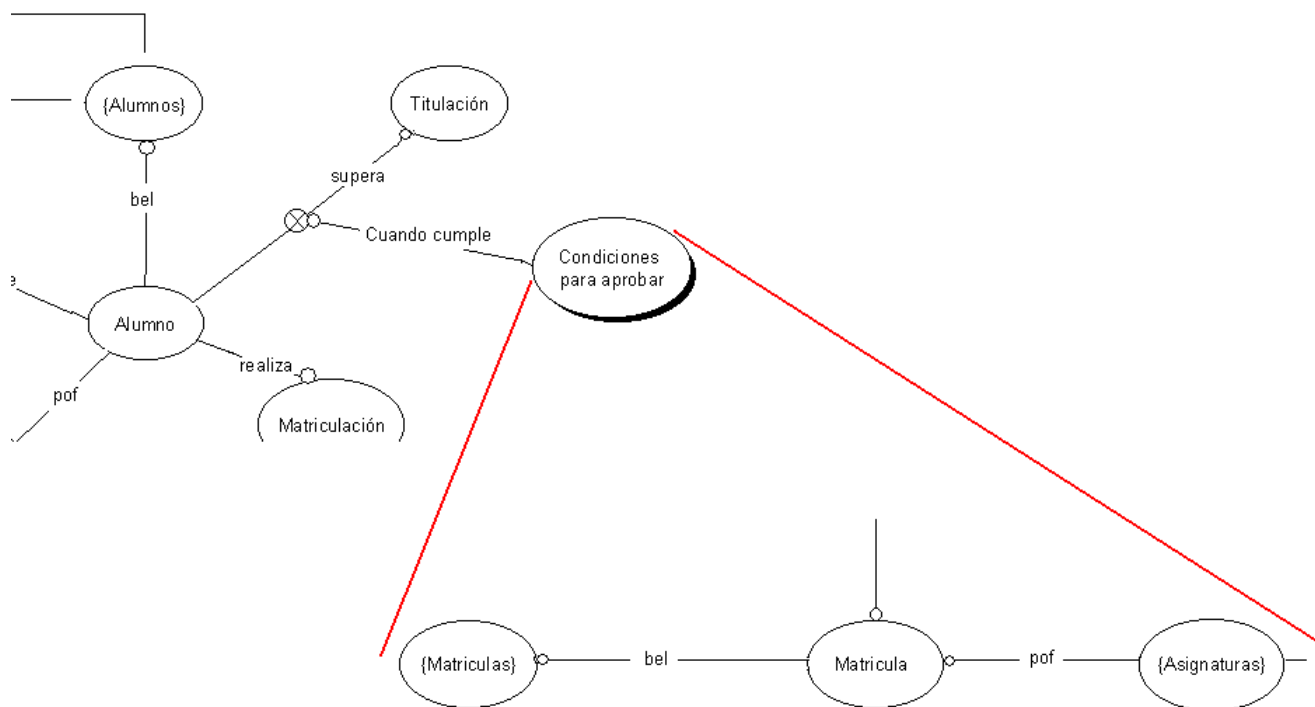


Figura 5.27. Abstracción que forma parte de una proposición genérica

La asociación entre la proposición “Alumno supera Titulación” y la abstracción “Condiciones para aprobar” es, simplemente, una asociación que: (1) facilita la legibilidad del Mapa de Conceptos y (2) denota una cierta proximidad entre los conceptos abstraídos y la proposición “Alumno supera Titulación”. *Per se*, esta asociación no puede poseer significado alguno, ya que en la abstracción se pierden todas las asociaciones con conceptos inferiores. En muchos casos, la asociación posee una etiqueta muy simple, como en el caso de la figura 5.28.

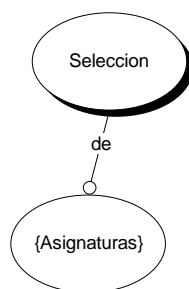


Figura 5.28. Asociación entre un concepto y una abstracción

Ó incluso no poseer ninguna etiqueta en absoluto, como en el caso de la figura 5.29.

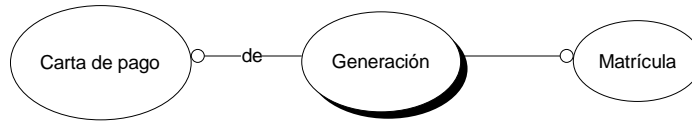


Figura 5.29. Otra asociación entre un concepto y una abstracción

Volviendo al caso que tenemos entre manos, cuando una abstracción forma parte de una proposición, es necesario especificar una conexión entre los conceptos abstraídos y los conceptos de mayor nivel. Esta conexión se realiza mediante el operador de equivalencia, y permite indicar que dos conceptos, a diferentes niveles de abstracción, son realmente el mismo concepto. En el ejemplo de la figura 5.27 se pueden indicar las equivalencias de la Tabla 5.9.

Nivel Superior	EQ	Nivel Superior
Matrícula	EQ	Matrícula
Asignaturas	EQ	Asignaturas

Tabla 5.9. Equivalencias

Una vez dadas las equivalencias, los dos niveles de abstracción pueden mezclarse, con el fin de derivar el DD. La mezcla de ambos niveles de abstracción se presenta en la figura 5.30.

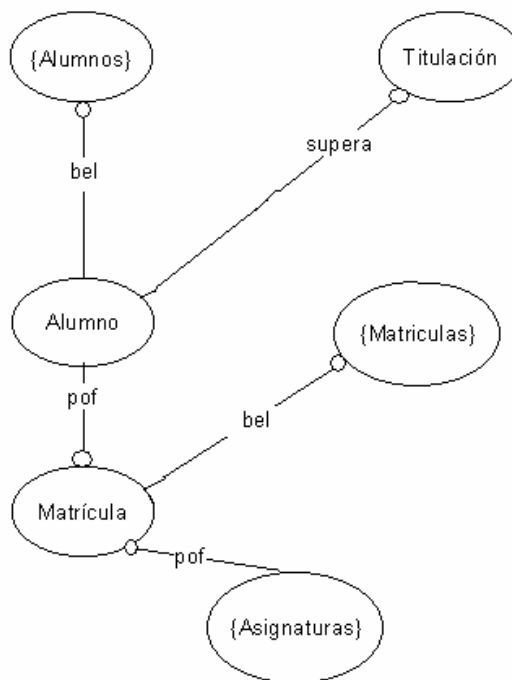


Figura 5.30. Mezcla de niveles de abstracción

Una vez realizada la mezcla de niveles, pueden utilizarse las reglas anteriormente mencionadas para realizar la derivación del DD. Nótese que las equivalencias no tienen por qué especificarse en el MC exhaustivo, ya que el único fin de dichas equivalencias es permitir una correcta derivación del DD.

5.5.3. RELACIÓN ENTRE EL TEXTO NARRATIVO Y LOS RESTANTES FORMALISMOS DE REPRESENTACIÓN

En líneas generales, el Texto Narrativo (TN) se construye como un conjunto secuencial de frases (oraciones) que forman un texto. Cada frase del TN se corresponde 1:1 con una proposición del MC, del DI o del DD. Las reglas para construir el TN son evidentes; para cada proposición del MC, DI o DD se construye una oración o frase del TN donde:

- El concepto principal forma el sujeto de la oración.
- La asociación forma el verbo o frase verbal.
- El concepto subordinado forma el predicado.

En el caso de proposiciones complejas, construidas recursivamente, el concepto subordinado es una proposición. En este caso, la oración o frase del TN se construiría de igual forma que en el caso de las proposiciones no recursivas, con la única excepción del que el predicado es una oración subordinada. A modo de ejemplo, supóngase el MC mostrado en la figura 5.31.

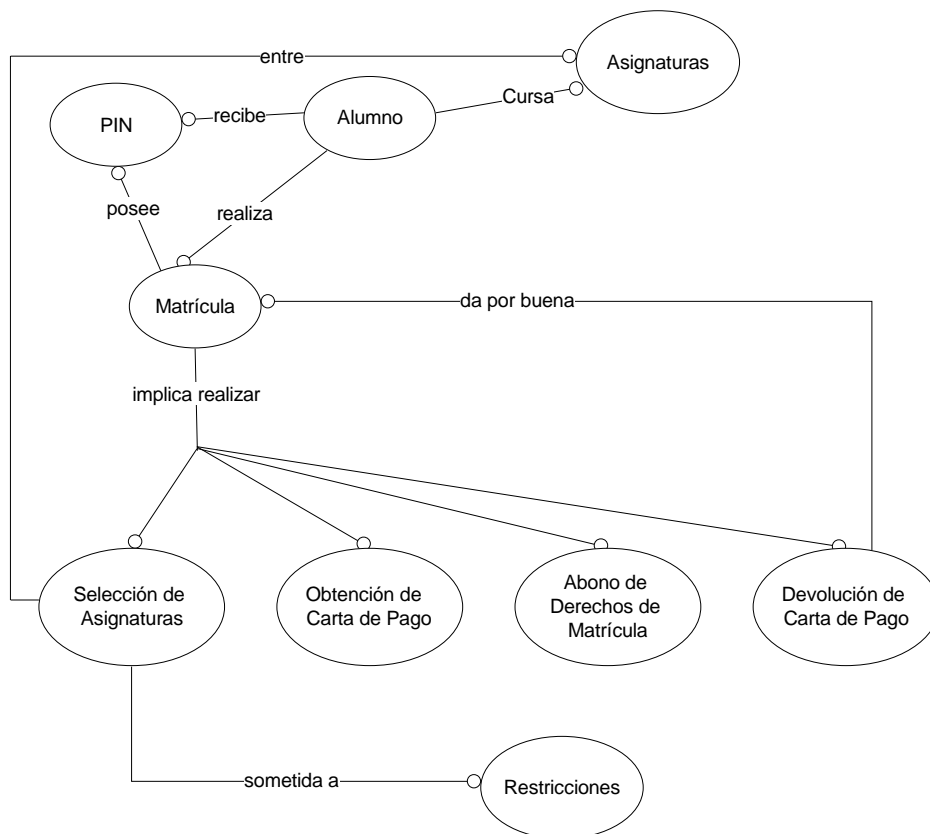


Figura 5.31. Mapa de Conceptos

El MC de la figura 5.31 podría transformarse en el TN mostrado en el cuadro 6.1. Nótese que dado el isomorfismo entre MC, DI y DD, los mismos resultados podrían obtenerse a partir del DI o del DD. Hay que mencionar, no obstante, que la obtención del TN a partir del DD es más compleja y, con frecuencia, menos útil, por lo que no se expondrá en toda su complejidad.

Alumno cursa asignaturas. Alumno recibe PIN. Alumno realiza matrícula.
 Matrícula posee PIN. Matrícula implica realizar selección de asignaturas.
 Matrícula implica realizar obtención de carta de pago. Matrícula implica realizar abono de derechos de matrícula. Matrícula implica realizar devolución de carta de pago. Selección de asignaturas entre asignaturas.
 Selección de asignaturas sometida a restricciones. Devolución de carta de pago da por buena matrícula.

Cuadro 5.1. Texto Narrativo derivado del Mapa de Conceptos mostrado en la figura 5.31

El mecanismo básico de confección del TN admite muchas mejoras a nivel estilístico. Algunas de ellas son:

1. La utilización de artículos determinados (no se permiten indeterminados) antepuestos a un concepto, si fuese preciso para facilitar la lectura. Por ejemplo, es preferible:
 - La devolución de la carta de pago da por buena la matrícula.
 - a
 - Devolución de la carta de pago da por buena matrícula.

2. La separación de las proposiciones referidas a distintos conceptos. Por ejemplo, es preferible el texto del cuadro 6.2 al texto brutal del cuadro 6.1.

Alumno cursa asignaturas. Alumno recibe PIN. Alumno realiza matrícula.
 Matrícula posee PIN. Matrícula implica realizar selección de asignaturas.
 Matrícula implica realizar obtención de carta de pago. Matrícula implica realizar abono de derechos de matrícula. Matrícula implica realizar devolución de carta de pago.
 Selección de asignaturas entre asignaturas. Selección de asignaturas sometida a restricciones.
 Devolución de carta de pago da por buena matrícula.

Cuadro 5.2. Texto Narrativo mejorado para el ejemplo del cuadro 6.1

3. Ordenación de las proposiciones en algún orden secuencial, indicado, por ejemplo, por la ordenación de los hechos, objetos, situaciones, etc. en el mundo real. Por ejemplo, es preferible:
 - o Alumno realiza matrícula. Alumno recibe PIN. Alumno cursa asignaturas.
 - a
 - o Alumno cursa asignaturas. Alumno recibe PIN. Alumno realiza matrícula.
4. Utilización de conjunciones copulativas o adversativas. Por ejemplo, es preferible:
 - o Alumno realiza matrícula y recibe PIN. Alumno cursa asignaturas.
 - a
 - o Alumno realiza matrícula. Alumno recibe PIN. Alumno cursa asignaturas.

Podrían idearse otras normas, referidas a la colocación de las jerarquías, manejo de singulares y plurales, tiempo de los verbos, etc. No se proponen dichas normas ya que ello excedería el propósito del presente trabajo de Tesis, e **incluso podría suponer un trabajo de investigación totalmente independiente que consistiera en el análisis de un texto con el objetivo de derivar un MCG.**

Por último, conviene indicar que en principio no existe ningún impedimento para usar el TN como un mecanismo de representación en pie de igualdad con el MC o los DI y DD. No obstante, el TN es más útil como **mecanismo de comunicación** que de representación. Esto es, el TN permite representar las información registrada en el MC, DI o DD de forma que sea más fácilmente comprensible, ya sea para una ingeniero o para cualquier otro participante en la actividad de análisis.

6. Análisis Orientado a la Solución

El presente capítulo tiene como finalidad profundizar en la exposición iniciada en el capítulo 4, focalizándose en la fase de Análisis Orientado a la Solución de MAON.

La fase de Análisis Orientada a la Solución posee como finalidad identificar qué aproximación de desarrollo es más adecuada para resolver el problema planteado por el usuario bajo estudio, así como derivar los modelos conceptuales utilizados por la aproximación seleccionada. Para alcanzar ambos objetivos, es necesario realizar dos pasos procedimentales del proceso propuesto: la Identificación del Modelo Conceptual Idóneo y la Derivación del Modelo Conceptual Seleccionado. Ambos pasos se describirán en las secciones 6.1 y 6.2. Dicha descripción se ajustará al siguiente esquema:

- Descripción, en la que se indican las tareas que se deben realizar para completar los pasos de Identificación del Modelo Conceptual Idóneo y Derivación del Modelo Conceptual Seleccionado.
- Técnicas utilizadas, donde se describen aquellas técnicas, recetas o procedimientos objetivos (esto es, no sometidos a interpretación) que facilitan la terminación de cada paso.
- Productos de entrada, en la que se describen los productos de pasos anteriores que se utilizan en cada paso.
- Productos de salida, donde se describen los productos generados en cada paso.
- Excepciones, en la que se describen las posibles alteraciones del curso normal de cada paso, en especial las condiciones de retorno a un paso previo.

Los pasos de Identificación del Modelo Conceptual Idóneo y Derivación del Modelo Conceptual Seleccionado utilizan dos técnicas: la Técnica de Identificación del Modelo Conceptual Idóneo (IMCI) y la Técnica de Derivación del Modelo Conceptual Seleccionado (DMCS), respectivamente. Por lo tanto, para realizar una exposición completa de dichos pasos procedimentales, es necesario exponer las técnicas IMCI y DMCS. Ambas técnicas se describen en las secciones 6.3 y 6.4 respectivamente.

6.1. IDENTIFICACIÓN DEL MODELO CONCEPTUAL IDÓNEO

El paso de Identificación del Modelo Conceptual Idóneo tiene como finalidad identificar el modelo más adecuado para proseguir con el desarrollo una vez finalizado el pre-Análisis. El modelo más adecuado podría ser cualquiera de los utilizados por las distintas aproximaciones de desarrollo, esto es, un diagrama de flujo de datos, un diagrama de casos de uso, etc.

6.1.1. DESCRIPCIÓN

El paso de Identificación del Modelo Conceptual Idóneo consiste, casi en su totalidad, en la aplicación de la Técnica de Identificación del Modelo Conceptual Idóneo (IMCI). La Técnica IMCI permite conseguir dos resultados fundamentales:

1. Obtener el Modelo Canónico de Requisitos.
2. Identificar el Modelo Conceptual Idóneo.

6.1.1.1. MODELO CANÓNICO DE REQUISITOS

El Modelo Canónico de Requisitos (MCR) es un formalismo de representación que no forma parte del MCG, sino que se obtiene a partir del MCG, en concreto, a partir del Diccionario de Descripción (DD). El MCR se obtiene mediante el etiquetado del DD, empleando el procedimiento de interpretación, que forma parte de la Técnica IMCI.

El etiquetado del DD consiste en poner en relación cada concepto y asociación del DD con los constructores del denominado Modelo Canónico. El Modelo Canónico es un formalismo de transcripción que permite reescribir distintos modelos conceptuales como, por ejemplo, los diagramas de flujo de datos o los diagramas entidad-relación.

El Modelo Canónico, el cual se presenta en el anexo A, se inspira en el trabajo de [Davis et al, 1997], con profundas modificaciones realizadas por el autor del presente trabajo de Tesis. El corazón del Modelo Canónico son un conjunto de elementos y enlaces, los cuales se corresponden con los distintos constructores de los modelos conceptuales. El etiquetado del DD consistirá, por lo tanto, en relacionar cada concepto del DD con un elemento del Modelo Canónico y cada asociación del DD con un enlace del Modelo Canónico.

El etiquetado del DD se consigue mediante el procedimiento de interpretación. El procedimiento de interpretación se compone de una serie de pasos, en su mayoría completamente automatizables, que permiten etiquetar los conceptos y asociaciones del DD.

Una vez que todos los conceptos y asociaciones del DD han sido etiquetados, el DD se ha puesto en relación 1:1 con el Modelo Canónico. El DD completamente etiquetado es lo que se ha denominado Modelo Canónico de Requisitos.

6.1.1.2. IDENTIFICACIÓN DEL MODELO CONCEPTUAL IDÓNEO

El Modelo Conceptual Idóneo es el modelo conceptual que puede representar con mayor amplitud la información recogida en el MCG.

El Modelo Conceptual Idóneo se calcula a partir del MCR utilizando un procedimiento muy simple. Dicho procedimiento emplea un conjunto de tablas derivadas del Modelo Canónico que ponen en relación cada elemento y enlace de dicho modelo con el conjunto de modelos conceptuales utilizados por las distintas aproximaciones de desarrollo.

Dichas tablas, por lo tanto, indican qué elementos y enlaces (y, en consecuencia, qué constructores) poseen los distintos modelos conceptuales. Como en el MCR cada concepto y asociación del MCG ha sido etiquetado, es posible establecer una correspondencia entre el MCR y los distintos modelos conceptuales. El Modelo Conceptual Idóneo se corresponderá con el modelo conceptual que más proposiciones del MCR es capaz de registrar. La Técnica IMCI posee, adicionalmente, una Métrica de Adecuación. La Métrica de Adecuación permite valorar cuánto de adecuado es el modelo conceptual considerado idóneo. La necesidad de esta métrica viene dada por el hecho de que un modelo puede ser el idóneo (es decir, el mejor de todos los modelos), pero aún así representar muy pobremente la información recogida del dominio.

6.1.2. TÉCNICAS UTILIZADAS

En el presente paso de MAON se utiliza la Técnica IMCI. Esta técnica consta de tres componentes:

- El procedimiento de interpretación, que permite construir el MCR.
- El cálculo del Modelo Conceptual Idóneo, que permite identificar el modelo conceptual más adecuado para representar la información del MCG.
- La Métrica de Adecuación, que permite determinar cuánto de adecuado es el Modelo Conceptual Idóneo.

6.1.3. PRODUCTOS DE ENTRADA

El producto de entrada de esta paso es el Modelo Preliminar del Problema del Usuario. Este modelo está compuesto por dos formalismos de representación:

- Diccionario de Descripción.

- Mapa de Conceptos Exhaustivo.

De ambos formalismos de representación, el más importante para este paso es el Diccionario de Descripción (DD). El DD se utiliza para obtener, en este paso, el MCR. El Mapa de Conceptos Exhaustivo puede utilizarse para obtener la versión gráfica del MCR, tal y como se indica en el anexo A. No obstante, el MCR derivado del DD es el producto más importante del presente paso y será el único considerado en el presente trabajo de Tesis.

6.1.4. PRODUCTOS DE SALIDA

Los productos de salida del paso de Identificación del Modelo Conceptual Idóneo son los siguientes:

- Modelo Conceptual Idóneo. Es el modelo conceptual más adecuado para proseguir el proceso de implementación del sistema software.
- Modelo Canónico de Requisitos. Es un formalismo de representación, derivado del DD, donde se han etiquetado los distintos conceptos y asociaciones del MCG con los elementos y enlaces del Modelo conceptual.

El Modelo Canónico de Requisitos es especialmente importante, ya que es el producto de entrada para el paso de Derivación del Modelo Conceptual Seleccionado.

6.1.5. EXCEPCIONES

La Técnica IMCI propone la Métrica de Adecuación para medir la bondad del Modelo Conceptual Idóneo. La Métrica de Adecuación se basa en la distancia entre los conceptos que el problema del usuario exige representar y los conceptos que el modelo conceptual seleccionado puede representar efectivamente.

Dicha distancia está determinada, sobre todo, por los conceptos elegidos en el Análisis Preliminar y refinados durante el Análisis Exhaustivo. Si la adecuación del modelo conceptual seleccionado es muy alta, es posible proseguir al siguiente paso de MAON (Derivación del Modelo Conceptual Seleccionado). Por el contrario, si la adecuación es muy baja, será necesario volver a la tarea de Análisis Exhaustivo, o incluso a la tarea de Análisis Preliminar en los casos donde la Métrica de Adecuación de valores muy bajos.

6.2. DERIVACIÓN DEL MODELO CONCEPTUAL SELECCIONADO

El paso de Derivación del Modelo Conceptual Seleccionado tiene como finalidad derivar el modelo conceptual seleccionado como idóneo en el paso anterior. Este paso de MAON se basa en la utilización de la Técnica de Derivación del Modelo Conceptual Seleccionado, la cual es completamente automatizable.

6.2.1. DESCRIPCIÓN DE LA TAREA

Una vez obtenido el Modelo Canónico de Requisitos (MCR), sólo resta obtener el modelo conceptual identificado como idóneo en el paso anterior de MAON.

La derivación del Modelo Conceptual Idóneo se realiza aplicando la Técnica de Derivación del Modelo Conceptual Seleccionado (Técnica DMCS). Esta técnica se fundamenta en el Modelo Canónico.

Como ya se ha indicado anteriormente, el Modelo Canónico permite poner en correlación los distintos constructores de los modelos conceptuales utilizados por las distintas aproximaciones de desarrollo con el MCG. Dicha correlación se lleva a cabo en el MCR.

Gracias a dicha correlación, es posible derivar los constructores del Modelo Conceptual Idóneo a partir de las proposiciones del MCR. Para ello, es necesario utilizar un conjunto de tablas, similares en cierta medida a las tablas utilizadas en la Técnica IMCI. Dichas tablas permiten generar fragmentos del Modelo Conceptual Idóneo, uno para cada proposición del MCR. El modelo conceptual se obtiene finalmente fusionando todos los fragmentos obtenidos para cada proposición por separado.

No obstante, es necesario indicar que, en determinadas ocasiones, por razones organizativas o debido a las preferencias del analista, el modelo deseado para proseguir con el subsiguiente proceso de desarrollo de software no tiene por qué ser el modelo identificado como idóneo. Así, por ejemplo, si una organización puede haber decidido utilizar un método orientado a objetos, y MAON identifica como idóneo un diagrama de flujo de datos.

En este caso, la Técnica DMCS permite obtener no el Modelo Conceptual Idóneo, sino el modelo que el analista desee utilizar posteriormente. Esta facilidad de la Técnica DMCS posibilita que MAON pueda ser utilizado en cualquier caso para facilitar la comprensión del problema del usuario a resolver, independientemente de que la organización o el analista haya prescrito de antemano un modelo o método para continuar con el desarrollo posterior del producto software.

6.2.2. TÉCNICAS UTILIZADAS

En el presente paso de MAON se utiliza la Técnica DMCS. Esta técnica permite obtener, a partir del MCR, el Modelo Conceptual Idóneo o, alternativamente, el modelo conceptual seleccionado por el analista.

6.2.3. PRODUCTOS DE ENTRADA

Los productos de entrada de este paso son el Modelo Canónico de Requisitos y la indicación del modelo conceptual considerado idóneo.

No obstante, y debido a motivos organizativos o, simplemente, a las preferencias del mismo analista, también se considera como entrada el modelo conceptual que organización o analista desean utilizar en las tareas siguientes de desarrollo.

6.2.4. PRODUCTOS DE SALIDA

Como salida de este paso se obtiene el Modelo Conceptual Idóneo o, alternativamente, el modelo conceptual seleccionado por el analista. Dicho modelo conceptual representa el punto final de aplicación de MAON y, al mismo tiempo, el punto de inicio para el subsiguiente desarrollo del producto software.

6.2.5. EXCEPCIONES

No existen excepciones en este paso, ya que el filtro se ha establecido en el paso anterior, de Identificación del Modelo Conceptual Idóneo.

6.3. TÉCNICA DE IDENTIFICACIÓN DEL MODELO CONCEPTUAL IDÓNEO

La técnica de Identificación del Modelo Conceptual Idóneo (Técnica IMCI) tiene como objetivo identificar el modelo más adecuado (esto es, el modelo idóneo) para representar la información registrada en el MCG y, por lo tanto, permitir continuar el proceso de desarrollo de software.

El modelo idóneo se identifica calculando la **adecuación** de los distintos modelos. Sin embargo, para calcular la adecuación es necesario aplicar, previamente, el **procedimiento de interpretación**. El procedimiento de interpretación tiene como objetivo etiquetar los conceptos y asociaciones del Diccionario de Descripción (DD) de tal forma que a cada concepto y asociación le corresponda un constructor único del Modelo Canónico.

La exposición de la Técnica IMCI se estructurará de la forma siguiente. El epígrafe 6.3.1 se recoge ciertas consideraciones previas necesarias para una adecuada exposición de la Técnica IMCI. El epígrafe 6.3.2 presenta el procedimiento de interpretación. Finalmente, el epígrafe 6.3.3 se dedica a presentar la Identificación del Modelo Conceptual Idóneo, así como la Técnica de Adecuación.

6.3.1. ASPECTOS PRELIMINARES

En los siguientes epígrafes se exponen algunas consideraciones previas que es necesario detallar previamente a la exposición de la Técnica de Identificación del Modelo Conceptual Idóneo. Estas consideraciones previas se tratan en los siguientes epígrafes y hacen referencia a:

- Cómo facilita el Diccionario de Descripción la derivación del Modelo Canónico de Requisitos.
- Cuál es la naturaleza del Modelo Canónico.
- Qué son las etiquetas del Modelo Canónico de Requisitos y cómo se representan.

6.3.1.1. ESTRUCTURA DEL DICCIONARIO DE DESCRIPCIÓN

Una vez se ha obtenido el DD, toda la información recogida acerca del dominio está representada de forma compacta y, lo más importante, estructurada, en tres partes (declaraciones, definiciones y proposiciones), las cuales permiten una aplicación rápida y eficaz del procedimiento de interpretación.

Como puede observarse en la figura 6.1, la estructura del DD se caracteriza por presentar, en primer lugar, los conocimientos más formalizados (conjuntos, subconjuntos e individuos, registrados en la parte de declaraciones), en segundo lugar los conocimientos estructurantes, declarativos y procedimentales (asociaciones spec, rel, pof, attr, val y predicados, funciones y transiciones) y, por último, los conocimientos no interpretados, en la forma de proposiciones creadas a partir de asociaciones genéricas.

De modo simplificado, se puede indicar que la estructura del DD permite una construcción secuencial del Modelo Canónico de Requisitos (MCR), empezando por los conjuntos, subconjuntos e individuos, progresando posteriormente al conjunto de definiciones (relaciones estructurales, asignación de valores, predicados y funciones) e interpretando, por último, las proposiciones construidas con asociaciones genéricas.

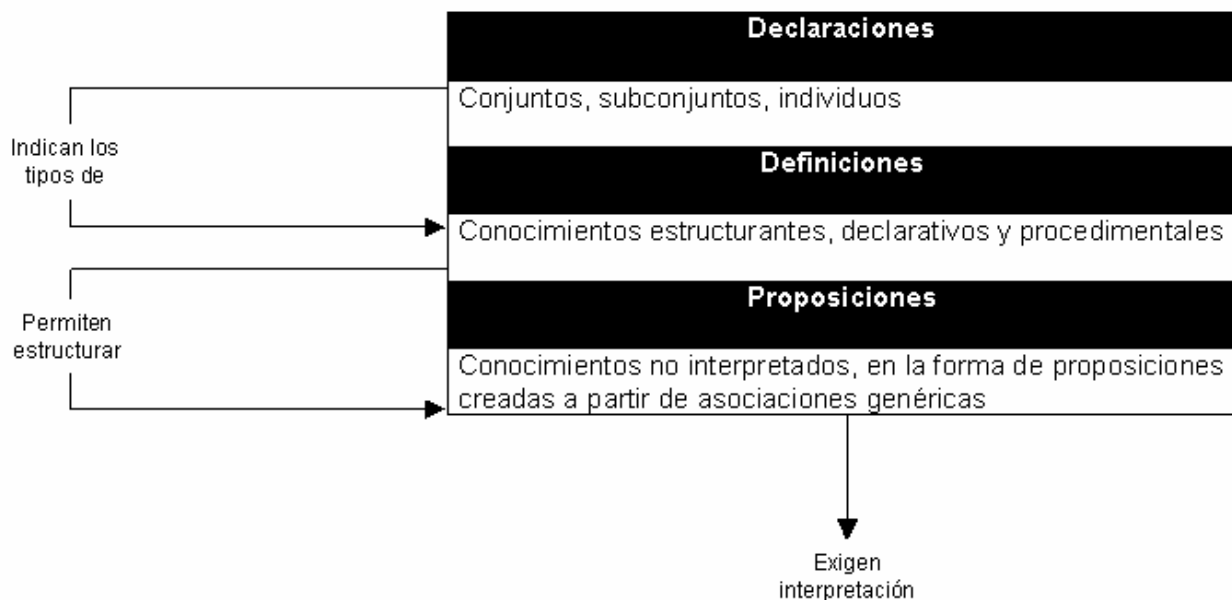


Figura 6.1. Estructura del Diccionario de Descripción

6.3.1.2. MODELO CANÓNICO

El Modelo Canónico es un formalismo de representación que permite reescribir cualquier modelo conceptual en función de un conjunto de elementos y enlaces. Los elementos representan diversos constructores de distintos modelos conceptuales, tales como las entidades, funciones o estados, mientras que los enlaces representan otros tipos de constructores, tales como los flujos de datos o relaciones.

El Modelo Canónico se debe originalmente a Davis [Davis et al., 1997]. El objetivo de este autor fue definir un mecanismo de representación que permitiese transcribir los distintos modelos conceptuales utilizados durante la especificación de requisitos. Dicha transcripción permitiría, a juicio de Davis, describir

los requisitos de diversas maneras, al permitir obtener un modelo conceptual a partir de la información registrada en otros modelos.

Deben indicarse, no obstante, diversas carencias del enfoque de Davis. En primer lugar, el Modelo Canónico no se definió nunca con la intención de crear un nuevo modelo conceptual, es más, el Modelo Canónico, tal y como se expone en el anexo A, es altamente fragmentario, esto es, está formado por pequeñas unidades, las cuales apenas proveen significado para un intérprete humano.

En segundo lugar, y en coherencia con la carencia antes señalada, la intención de Davis fue que el Modelo Canónico sirviese no para modelar el problema del usuario, sino la solución. Es decir, el Modelo Canónico sirve para modelar fragmentos de la solución del problema del usuario recogidos en los modelos conceptuales, de tal forma que dichos modelos conceptuales fuesen intercambiables.

En otras palabras, el Modelo Canónico de Davis puede considerarse, en concordancia con lo expuesto en la sección 2.3.4, un modelo centrado exclusivamente en los Requisitos y no adecuado, por lo tanto, para el estudio del problema del usuario en el mundo real.

Ahora bien; aunque el Modelo Canónico de Davis sea inadecuado para analizar el problema plantado por un usuario, sí parece adecuado como mecanismo de interconexión entre MAON y los modelos conceptuales propios de las distintas aproximaciones de desarrollo. Esta interconexión es posible porque el MCG y el Modelo Canónico poseen estructuras superficiales parecidas, esto es, los conceptos y asociaciones del MCG conforman topologías parecidas a las del Modelo Canónico, aunque esta similitud es únicamente superficial.

Después de la realización de profundas modificaciones al Modelo Canónico, las cuales se presentan en el anexo A, ha sido posible conseguir que el MCG pueda hacerse concordar con el Modelo Canónico, lo cual forma el núcleo del procedimiento de interpretación. Aquí es adecuado reiterar que el hecho de poder poner en común no significa que el MCG y el Modelo Canónico sean similares. Utilizando una analogía, el cuerpo de una persona y un robot que simule este cuerpo no son similares, aunque algunas partes de cada uno, persona y robot, puedan desarrollar funciones similares.

La correspondencia entre conceptos y asociaciones del DD, por un lado, y elementos y enlaces del Modelo conceptual, por otro, se establece mediante el etiquetado de los distintos conceptos y asociaciones del DD. Dicho etiquetado se obtiene al finalizar la aplicación del procedimiento de interpretación.

6.3.1.3. ESTRUCTURAS DE LAS ETIQUETAS

La obtención del MCR se realizará aplicando el procedimiento de interpretación y asignando, a cada concepto y asociación del MCR, una etiqueta. Dicha etiqueta se corresponderá con alguno de los constructores del Modelo Canónico. En el presente trabajo de tesis se ha definido el siguiente formato de etiquetado:

<Etiqueta>[<Calificadores>]:<concepto ó asociación>

El formato de etiquetado consta de los siguientes elementos.

- <Etiqueta> puede ser cualquiera de los 8 elementos o 15 enlaces del Modelo Canónico.
- <Calificadores> en principio podrían ser uno o varios de los *statespaces* predefinidos en el Modelo Canónico, asignables a aquellos constructores que los soporten. Los *statespaces* predefinidos en el Modelo Canónico son los siguientes:
 - Repl/notrepl
 - Data/cntl
 - Int/ext
 - Contin/disc

No obstante, es necesario indicar que tres de los cuatro calificadores anteriores (Data/cntl, Int/ext y Contin/disc) hacen referencia a aspectos puramente computacionales que no pueden ser derivados de ninguna forma a partir del MCG. Por ello, el único calificador que se utilizará en el presente trabajo de Tesis será el *statespace* repl/notrepl, que permitirá distinguir los conjuntos de los individuos en el MCR. Dicha distinción facilitará la derivación de los modelos conceptuales (tales como el diagrama de clases o el diagrama de flujo de datos) desde el MCR.

- <Concepto o asociación> se refiere a cualquiera de los conceptos o asociaciones del DD.

Con el propósito de exponer más claramente el formato de etiquetado, supóngase el siguiente ejemplo. Si el concepto “cliente” se etiquetara como:

Entity[repl]:Cliente

Significaría que “cliente” **se interpreta** como una entidad replicable (esto es, un conjunto o subconjunto), en el MCR, y como tal se derivaría al modelo conceptual identificado como idóneo.

6.3.2. PROCEDIMIENTO DE INTERPRETACIÓN

El procedimiento de interpretación es el núcleo fundamental de la Técnica IMCI. Este procedimiento consta de ocho pasos, todos los cuales, a excepción del paso 6, son totalmente automatizables, esto es, no dependen del analista que aplica la Técnica IMCI. Los pasos del procedimiento de interpretación son los siguientes:

1. Etiquetado de conjuntos, subconjuntos, predicados y funciones.
2. Etiquetado de asociaciones especiales.
3. Etiquetado de conceptos y asociaciones de interpretación única.
4. Vuelta a (3) hasta que no pueda etiquetarse ningún concepto o asociación. En caso de que no

- queden conceptos o asociaciones por interpretar, ir a 9.
5. Etiquetado de conceptos o asociaciones de interpretación múltiple (esto es, genéricos, no interpretados).
 6. Estudiar la sustitución de proposiciones.
 7. Definir proposiciones no interpretables.
 8. Vuelta a (3) hasta que todos los conceptos y asociaciones hayan sido etiquetados.
 9. Fin de la interpretación.

Al finalizar el procedimiento de interpretación, todos los conceptos y asociaciones de DD han sido etiquetados. El conjunto de conceptos y asociaciones etiquetadas forman el MCR. A continuación, se detallan cada uno de los pasos del procedimiento de interpretación.

6.3.2.1. ETIQUETADO DE CONJUNTOS, SUBCONJUNTOS, PREDICADOS, FUNCIONES Y TRANSICIONES

Los conjuntos, subconjuntos, funciones y transiciones poseen una interpretación bien definida, la cual se indica en la tabla 6.1.

Concepto	Interpretación
Conjunto	<i>Entity[repl]</i>
Subconjunto	<i>Entity[notrepl]</i>
Función	<i>Process</i>
Transición	<i>Transition</i>

Tabla 6.1. Etiquetado de conjuntos, subconjuntos y funciones

Las reglas de reescritura del DD aseguran que cada concepto posea un nombre diferente. Ello implica naturalmente que, una vez se interpreta un conjunto, subconjunto o función, todos los conceptos con nombres similares en el DD (esto es, el mismo concepto participante en distintas proposiciones) pueden ser etiquetados igualmente.

A modo de ejemplo de la aplicación del presente paso del Procedimiento de Interpretación, véase el DD mostrado en la figura 6.2.

Por el contrario, los predicados pueden derivarse a restricciones o predicados del MCR. Por esta razón, la interpretación de los predicados debe realizarla el analista. Para cada predicado del DD, el analista debe indicar si la interpretación correcta es la de restricción (*constraint*), esto es, algo que siempre se verifica, o si por el contrario puede o no verificarse, en cuyo caso se interpretaría como un predicado (*predicate*).

Declaraciones			
Conjuntos:	Alumnos Matrículas Asignaturas		
Subconjuntos:		Subs	
Individuos:	Asignatura	bel	Asignaturas
	Matrícula	bel	Matrículas
	Alumno	bel	Alumnos
Definiciones			
Índice	Definición		
1	AND (P1, P2)		
2	$\geq(N^{\circ}, \text{Sumatorio}(\text{Aprobado}, N^{\circ}_1))$		
3	Nº attr Titulación		
4	Matrícula pof Alumno		
5	Asignaturas pof Matrícula		
6	Nº_1 attr Asignatura		
7	Calificación attr Asignatura		
8	Calificación val Aprobado		
Proposiciones			
Índice	Concepto – 1	Asociación	Concepto – 2
1	Alumno	Supera	Titulación
2	Alumno	Obtiene	Nº
3	Nº	De	Créditos
4	Nº_1	De	Créditos
5	Alumno	Realiza	Matrícula

Figura 6.2. Diccionario de Descripción que se utilizará a modo de ejemplo a lo largo de la presente sección

Durante la ejecución del presente paso procedimental, lo primero que se debe realizar es el etiquetado indicado en la tabla 6.1. Tras este etiquetado (las etiquetas se muestran en negrita), el DD quedaría de la forma indicada en la figura 6.3.

Declaraciones			
Conjuntos:	Entity[repl]: Alumnos Entity[repl]: Matrículas Entity[repl]: Asignaturas		
Subconjuntos:		Subs	
Individuos:	Asignatura	bel	Entity[repl]: Asignaturas
	Matrícula	bel	Entity[repl]: Matrículas
	Alumno	bel	Entity[repl]: Alumnos
Definiciones			
Índice	Definición		
1	AND (P1, P2)		
2	$\geq(N^{\circ}, \text{Process: Sumatorio}(\text{Aprobado}, N^{\circ}_1))$		
3	Nº attr Titulación		
4	Matrícula pof Alumno		
5	Entity[repl]: Asignaturas pof Matrícula		
6	Nº_1 attr Asignatura		
7	Calificación attr Asignatura		
8	Calificación val Aprobado		
Proposiciones			
Índice	Concepto – 1	Asociación	Concepto – 2
1	Alumno	Supera	Titulación
2	Alumno	Obtiene	Nº
3	Nº	De	Créditos
4	Nº_1	De	Créditos
5	Alumno	Realiza	Matrícula

Figura 6.3. Primer etiquetado

Nótese que durante el Análisis Exhaustivo, se han identificado los conceptos que hacen referencia a funciones y a predicados. Por ello ha sido posible identificar “Sumatorio” como una función, y también por ello es posible conocer que “AND” y “>=” son predicados.

Para interpretar los predicados anteriores, es necesario establecer si se tratan de predicados o restricciones del MCR. En el ejemplo de la figura 6.3, es claro que el predicado “AND” se trata de un predicado del MCR (esto es, no se verifica en todo momento), mientras que el predicado “>=” es una restricción del MCR (se verifica siempre). Por ello, se antepone la etiqueta “Predicate” al predicado “AND” y la etiqueta “Constraint” al predicado “>=”, tal y como se muestra en la tabla figura 6.4.

Declaraciones			
Conjuntos:	Entity[repl]: Alumnos Entity[repl]: Matrículas Entity[repl]: Asignaturas		
Subconjuntos:		Subs	
Individuos:	Asignatura	bel	Entity[repl]: Asignaturas
	Matrícula	bel	Entity[repl]: Matrículas
	Alumno	bel	Entity[repl]: Alumnos
Definiciones			
Índice	Definición		
1	Predicate: AND (P1, P2)		
2	Constraint: >=(Nº, Process: Sumatorio(Aprobado, Nº_1))		
3	Nº attr Titulación		
4	Matrícula pof Alumno		
5	Entity[repl]: Asignaturas pof Matrícula		
6	Nº_1 attr Asignatura		
7	Calificación attr Asignatura		
8	Calificación val Aprobado		
Proposiciones			
Índice	Concepto – 1	Asociación	Concepto – 2
1	Alumno	Supera	Titulación
2	Alumno	Obtiene	Nº
3	Nº	De	Créditos
4	Nº_1	De	Créditos
5	Alumno	Realiza	Matrícula

Figura 6.4. Segundo etiquetado

6.3.2.2. ETIQUETADO DE ASOCIACIONES ESPECIALES

Las asociaciones especiales, al igual que los conjuntos, subconjuntos y funciones, únicamente pueden interpretarse en un sentido, esto es, son etiquetables unívocamente. La interpretación de las asociaciones especiales se indica en la tabla 6.2.

Asociación	Interpretación
Bel	<i>bel</i>
Subs	<i>subs</i>
Spec	<i>spec</i>
Pof	<i>pof</i>
Rel	<i>rel</i>
Attr	<i>pof</i>
Val	<i>hval</i>

Tabla 6.2. Reglas de etiquetado de asociaciones especiales

La aplicación de las reglas de interpretación de la tabla 6.2 al ejemplo que se ha venido manejando hasta el momento da como resultado el etiquetado mostrado en la figura 6.5.

Declaraciones			
Conjuntos:	Entity[repl]: Alumnos Entity[repl]: Matrículas Entity[repl]: Asignaturas		
Subconjuntos:		Subs	
Individuos:	Asignatura	Bel: bel	Entity[repl]: Asignaturas
	Matrícula	Bel: bel	Entity[repl]: Matrículas
	Alumno	Bel: bel	Entity[repl]: Alumnos
Definiciones			
Índice	Definición		
1	Predicate: AND (P1, P2)		
2	Constraint: >=(Nº, Process: Sumatorio(Aprobado, Nº_1))		
3	Nº Pof: attr Titulación		
4	Matrícula Pof: pof Alumno		
5	Entity[repl]: Asignaturas Pof: pof Matrícula		
6	Nº_1 Pof: attr Asignatura		
7	Calificación Pof: attr Asignatura		
8	Calificación Hval: val Aprobado		
Proposiciones			
Índice	Concepto – 1	Asociación	Concepto – 2
1	Alumno	Supera	Titulación
2	Alumno	Obtiene	Nº
3	Nº	De	Créditos
4	Nº_1	De	Créditos
5	Alumno	Realiza	Matrícula

Figura 6.5. Etiquetado de asociaciones especiales

6.3.2.3. ETIQUETADO DE CONCEPTOS Y ASOCIACIONES DE INTERPRETACIÓN ÚNICA

Una vez realizados los dos pasos anteriores, es necesario realizar múltiples iteraciones hasta obtener el MCR. El primer paso de la iteración consiste en determinar qué conceptos y asociaciones poseen una interpretación única.

Los conceptos y asociaciones que poseen una interpretación única son aquellos que pueden

etiquetarse unívocamente. Esto es posible debido a que el Modelo Canónico no admite combinaciones arbitrarias de elementos y enlaces, sino que dichas combinaciones se restringen a las indicadas en la tabla 6.3

En el caso de que las proposiciones estén construidas recursivamente, la interpretación es más compleja. En dicho caso, deben utilizarse las tablas 6.4 y 6.5, las cuales relacionan proposiciones y conceptos por la izquierda y por la derecha, respectivamente.

Cuando las proposiciones están construidas recursivamente por la izquierda y por la derecha, al mismo tiempo, es necesario refinar la proposición así construida antes de aplicar el procedimiento de interpretación. Por ejemplo, véase la figura 6.6. La figura 6.6 muestra la proposición [Alumno (supera) Titulación] (cuando) [Alumno (obtiene) N°]. Esta proposición es recursiva por la izquierda y por la derecha debido a que, tanto a la izquierda como a la derecha de la asociación “cuando” existe una proposición.

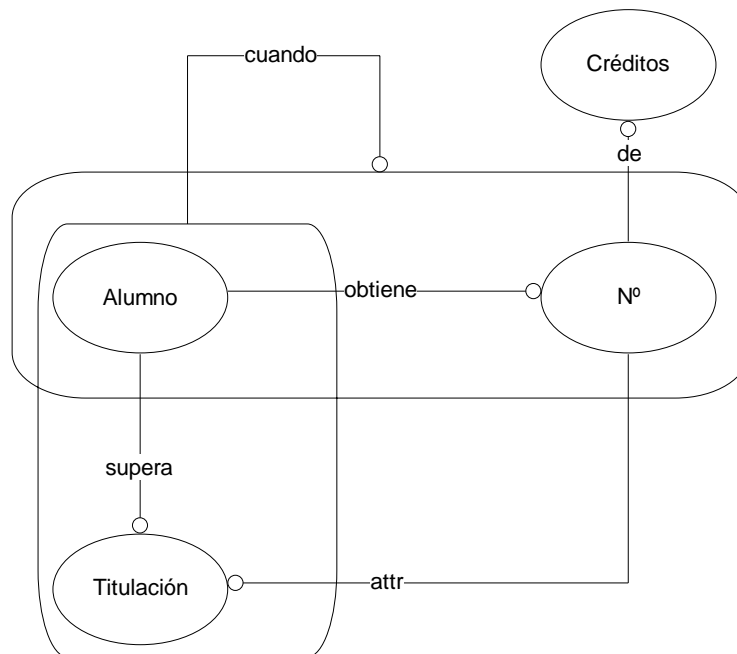


Figura 6.6. Proposición recursiva por la izquierda y por la derecha

En este caso, es necesario refinar la asociación mediante la introducción de un concepto; esto es, se sustituye la proposición recursiva por la izquierda por dos proposiciones: una recursiva por la izquierda y otra recursiva por la derecha. En el caso de la figura 6.6, lo que ha ocurrido es una falta de precisión. Realmente, el hecho de que el [Alumno (obtiene) N°] y [Alumno (supera) Titulación] ocurre simultáneamente. Por ello, la asociación “cuando” puede sustituirse por un predicado “AND”, tal y como muestra la figura 6.7.

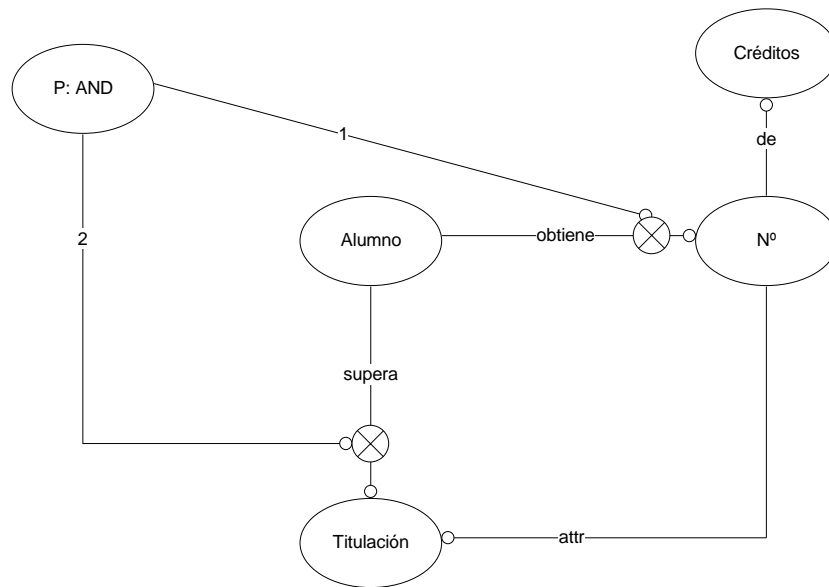


Figura 6.7. Descomposición de una proposición recursiva a izquierda y derecha

Una vez descompuestas las proposiciones recursivas a izquierda y derecha, se procede al etiquetado de los conceptos de interpretación única. Para identificar los conceptos que poseen una interpretación única, debe seguirse el siguiente procedimiento:

- a) Caso 1. Proposiciones construidas no recursivamente. En este caso, para cada proposición o definición que haya sido parcialmente interpretada (por ejemplo, donde el concepto principal y la asociación, o los conceptos principal y subordinado ya tengan asignadas etiquetas), se debe localizar en la tabla 6.3 la interpretación del concepto o asociación que no haya sido interpretada.

La tabla 6.3 es simétrica, esto es, el concepto principal de la proposición puede leerse por filas y el subordinado por columnas (que es la forma en que la tabla está construida), o viceversa. En el caso de que sea necesario leer el concepto principal por columnas, debe invertirse el "signo" del enlace, esto es, si el enlace posee un signo "-", éste desaparece, y viceversa.

	entity [repl]	entity [notrepl]	process	predicate	transition	message	constraint	value	statespace
entity [repl]	spec subs pof rel activate								
entity [notrepl]	spec subs bel pof rel activate	spec subs pof rel activate							
process	pof sends receives -activate	pof sends receives -activate	spec pof activate						
predicate	operand	operand	activate	operand					
transition			stimulus response	stimulus	stimulus response				
message	-sends -receives -pof	-sends -receives -pof	-sends -receives	-operand	-stimulus -response	pof			
constraint	operand	operand	pof	operand	operand	operand	operand		
value	-sends -receives	-sends -receives	-sends -receives pof	-operand	-stimulus -response	-operand	-operand	pof	
statespace	-sends -receives pof	-sends -receives pof	-sends -receives pof	-operand	-stimulus -response	-operand	-operand	hval	spec pof

Tabla 6.3. Combinaciones permitidas en el Modelo Canónico entre elementos y enlaces

Para ilustrar el presente paso, retómese el ejemplo manejado, cuyo último paso se reflejó en la figura 6.5. En dicha figura, se puede observar que las proposiciones:

- Asignatura **Bel:** bel **Entity[repl]:** Asignaturas
- Matrícula **Bel:** bel **Entity[repl]:** Matrículas
- Alumno **Bel:** bel **Entity[repl]:** Alumnos

statespace	value	constraint	message	transition	predicate	process	entity [notrepl]	entity [repl]	message	-receives	entity [repl]
				stimulus			sends	sends	message	-	entity [repl]
				stimulus		receives	sends	sends	message	-sends	entity [notrepl]
				stimulus		receives	receives	receives	message	-receives	entity [notrepl]
				stimulus		receives	activate	activate	message	-sends	process
				stimulus			sends	sends	message	-receives	process
				stimulus			sends	sends	message	-operand	predicate
				stimulus			sends	sends	message	-stimulus	transition
				stimulus			sends	sends	message	-response	transition
						Sends receives	sends	sends	message	po	message
		operand							constraint	operand	entity [repl]
		operand							constraint	operand	entity [notrepl]
		operand							constraint	po	process
		operand							constraint	operand	predicate
		operand							Constraint	operand	transition
		operand							constraint	operand	message
									value	-sends	constraint
				Stimulus response	operand	receives			value	-sends	process
				Stimulus response	operand	sedns			value	-receives	process
					operand				value	po	process
					operand				value	-operand	predicate
				stimulus					value	-stimulus	transition
				stimulus					value	-response	transition
									value	-operand	message
									value	-operand	constraint
					operand	Sends receives			value	po	value
					operand				statespace	po	entity [repl]
					operand				statespace	po	entity [notrepl]
				stimulus	operand				Statespace	-sends	process
				stimulus	operand				Statespace	-receives	process
					operand				Statespace	po	process
					operand				Statespace	-operand	predicate
				stimulus					Statespace	-stimulus	transition
				stimulus					Statespace	-response	transition
									Statespace	-operand	message
									Statespace	-operand	constraint
				response					Statespace	hval	value
							-po	-po	Statespace	spec	statespace
					operand		-po	-po	Statespace	po	statespace

Tabla 6.5. Combinaciones permitidas en el Modelo Canónico entre elementos, enlaces y proposiciones (parte 2)

Están parcialmente evaluadas. Acudiendo a la tabla 6.3, puede observarse que una proposición del tipo:

<concepto> **Bel Entity[repl]**

sólo es posible si <concepto> es un elemento del tipo **Entity[notrepl]**, tal y como muestra la figura 6.8.

	entity [repl]	entity [notrepl]	
entity [repl]	spec subs pof rel activate		
entity [notrepl]	spec pof rel bel activate	spec pof rel activate	
			nof

Figura 6.8. Interpretación de <concepto> **Bel Entity[repl]**

- b) Caso 2. Proposiciones construidas recursivamente. En este caso, la interpretación es un poco más compleja. Por ejemplo, supóngase la proposición construida recursivamente que se muestra en la figura 6.9.

Para interpretar la proposición completa, es previamente necesario interpretar la proposición interna, esto es, [Profesor (corrige) Trabajo] o, al menos, que la asociación esté interpretada, ya que las tablas 6.4 y 6.5 relacionan enlaces (propios de las proposiciones) y conceptos.

Supóngase, por lo tanto, que la proposición [Profesor (corrige) Trabajo] esté interpretada de la forma **Entity[notrepl]:** Profesor **Rel:** corrige **Entity[notrepl]:** Trabajo. Supóngase, adicionalmente, que el concepto “nota” fuese un **Statespace**. Entonces, acudiendo a la tabla 6.4, la asociación “pone” sólo puede interpretarse como el enlace **-Pof**.

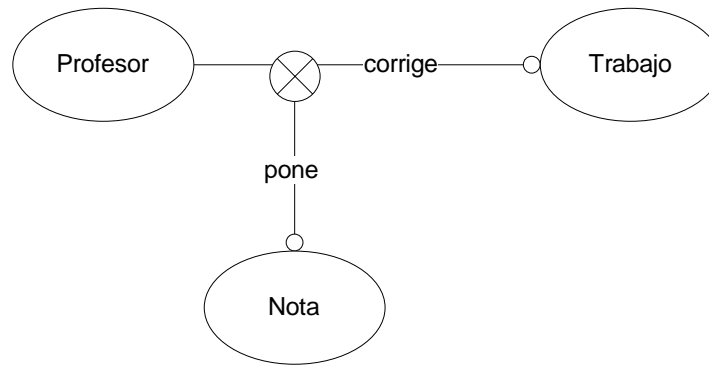


Figura 6.9. Ejemplo de proposición construida recursivamente por la izquierda

Finalmente, es necesario, para cada concepto interpretado, etiquetar con el mismo elemento todos aquellos conceptos que posean igual nombre. Con ello se finaliza el presente paso del procedimiento de interpretación.

Una vez aplicado el presente paso del procedimiento de interpretación, el resultado del ejemplo que se ha venido manejando, y que se mostró por última vez en la figura 6.5, aparecería tal y como se indica en la figura 6.10.

Declaraciones			
Conjuntos:	Entity[repl]: Alumnos Entity[repl]: Matrículas Entity[repl]: Asignaturas		
Subconjuntos:		Subs	
Individuos:	Entity[notrepl]: Asignatura	Bel: bel	Entity[repl]: Asignaturas
	Entity[notrepl]: Matrícula	Bel: bel	Entity[repl]: Matrículas
	Entity[notrepl]: Alumno	Bel: bel	Entity[repl]: Alumnos
Definiciones			
Índice	Definición		
1	Predicate: AND (Operand: P1, Operand: P2)		
2	Constraint: >=(Nº, Operand: Process: Sumatorio(Aprobado, Nº_1))		
3	Nº Pof: attr Titulación		
4	Entity[notrepl]: Matrícula Pof: pof Entity[notrepl]: Alumno		
5	Entity[repl]: Asignaturas Pof: pof Entity[notrepl]: Matrícula		
6	Statespace: Nº_1 Pof: attr Entity[notrepl]: Asignatura		
7	Calificación Pof: attr Entity[notrepl]: Asignatura		
8	Calificación Hval: val Aprobado		
Proposiciones			
Índice	Concepto – 1	Asociación	Concepto – 2
1	Entity[notrepl]: Alumno	Supera	Titulación
2	Entity[notrepl]: Alumno	Obtiene	Nº
3	Nº	De	Créditos
4	Nº_1	De	Créditos
5	Entity[notrepl]: Alumno	Realiza	Entity[notrepl]: Matrícula

Figura 6.10. Etiquetado de conceptos de interpretación única

6.3.2.4. ETIQUETADO DE CONCEPTOS O ASOCIACIONES DE INTERPRETACIÓN MÚLTIPLE

Una vez realizados los pasos anteriores, todos los conceptos y asociaciones restantes son de interpretación múltiple, esto es, no se puede asignar inequívocamente una interpretación a los mismos en función de las restricciones expuestas en las tablas 6.3, 6.4 y 6.5.

Para interpretar dichos conceptos y asociaciones no existe procedimiento alguno. Es el propio analista el que debe, conforme a la semántica de los conceptos y asociaciones, y de los elementos y enlaces sobre los que debe realizar la correspondencia, indicar cuál es la interpretación correcta de los distintos conceptos y asociaciones.

No obstante, el analista no tiene por qué interpretar de una vez todos los conceptos y asociaciones, sino que puede interpretar únicamente los más evidentes y realizar de nuevo los pasos anteriores del procedimiento de interpretación, con el objetivo de que las restricciones del Modelo Canónico permitan aclarar la interpretación de otros conceptos o asociaciones.

Nótese que este hecho posee una gran relevancia. En problemas que representan necesidades complejas, donde existen muchas proposiciones, la interpretación “incremental” que permite el procedimiento de interpretación favorece el escalado de MAON, esto es, la degradación de la utilidad de MAON para problemas grandes es progresiva, pero no posee un punto a partir del cual deja de ser útil.

A modo de ejemplo, el analista podría interpretar los conceptos N^0 y N^0_1 como **Statespace** (ya que parece que pueden albergar varios valores), y una vez hecho esto, volver a realizar los pasos anteriores del procedimiento de interpretación. Ello permitiría que en la proposición:

Constraint: \geq (**Statespace:** N^0 , **Operand:** **Process:** Sumatorio(Aprobado, **Statespace:** N^0_1))

podieran interpretarse la asociación entre “ \geq ” y “ N^0 ”, que sólo puede tratarse de un enlace **Operand**, tal y como se indica a continuación.

Constraint: \geq (**Operand:** **Statespace:** N^0 , **Operand:** **Process:** Sumatorio(Aprobado, **Statespace:** N^0_1))

6.3.2.5. ESTUDIAR LA SUSTITUCIÓN DE PROPOSICIONES

En determinadas ocasiones, las proposiciones son difíciles de interpretar en términos del Modelo Canónico, hasta el punto de que ello es imposible. Dicho caso se trata en el siguiente epígrafe. Sin embargo, en muchos casos, las citadas dificultades surgen por una colisión entre el MCG y el Modelo Canónico en lo que respecta a las asociaciones. Quizás un ejemplo sea más ilustrativo que una larga exposición. Supóngase la siguiente proposición:

Entity[notrepl]: Cliente realiza **Entity[notrepl]:** Pedido

Las posibles interpretaciones para la asociación “realiza”, recogidas en la tabla 6.3, son:

spec subs pof rel activate

Sin embargo, pudiera ser que el ingeniero no creyera adecuada ninguna de estas interpretaciones, ya que su intención, al enunciar la proposición **Entity[notrepl]: Cliente realiza Entity[notrepl]: Pedido**, fue expresar algo netamente dinámico, y no algo estático tal y como se deriva de los enlaces disponibles.

En este caso, el ingeniero podría, en MAON, sustituir la proposición **Entity[notrepl]: Cliente realiza Entity[notrepl]: Pedido** por otra proposición distinta, donde la asociación “realiza” puede ser interpretada como:

- Predicate
- Constraint
- Process
- Transition

Por lo que la proposición (en caso, por ejemplo, de que “realiza” se interpretase como un Process) resultante sería:

Process: realiza (Entity[notrepl]: Cliente, Entity[notrepl]: Pedido)

La sustitución de proposiciones es necesaria porque los aspectos dinámicos, en los lenguajes naturales, se expresa mediante los verbos de las sentencias o frases, y dichas sentencias se acostumbran a reflejar sin modificación en el MCG. Sin embargo, el Modelo Canónico únicamente permite expresar dinamismo mediante los elementos, y no mediante los enlaces (nótese que las asociaciones del MCG siempre se interpretan como enlaces). Esta falta de acoplamiento hace necesaria la sustitución de proposiciones en algunas ocasiones.

Adicionalmente, la sustitución de una proposición conlleva, habitualmente, que dicha proposición pase a formar parte de las declaraciones del Diccionario de Descripción. No obstante, y dada la complejidad que implican los cambios en el Diccionario de Descripción, cuando éste se confecciona manualmente es preferible no realizar ningún cambio en la ubicación de las distintas declaraciones y proposiciones. El resultado final, en cualquier caso, será exactamente el mismo.

6.3.2.6. DEFINIR PROPOSICIONES NO INTERPRETABLES

Durante la realización del procedimiento de interpretación, existe la posibilidad de que un determinado concepto u asociación no pueda ser interpretado debido a que no admitan ninguna

interpretación plausible. En este caso, se ha definido la etiqueta especial **nev:** (por no-evaluable), la cual permite realizar una interpretación, aunque no sea posible derivar, a partir de dichas proposiciones, ningún tipo de modelo conceptual.

La etiqueta especial **nev:**, a efectos del procedimiento de interpretación, es equivalente a cualquier concepto o asociación, esto es, **nev:** es una etiqueta comodín que permite que las proposiciones que la contengan puedan ser libremente interpretadas. Por ejemplo, supóngase la proposición:

Entity[notrepl]: Alumno **Nev:** tiene Novia

“Novia”, en la proposición anterior, no ha sido interpretada. Dado que la etiqueta **nev:** no aparece en ninguna de las tablas desde la 6.3 a 6.5, sería imposible llegar a interpretar, coherentemente, dicho concepto. Sin embargo, si consideramos que **nev:** es una etiqueta comodín, dicha proposición podría considerarse equivalente a **Entity[notrepl]:** Alumno **Spec:** tiene Novia, o **Entity[notrepl]:** Alumno **Pof:** tiene Novia, etc., y realizar la interpretación en base a alguna de dichas proposiciones.

Nótese, no obstante, que la proposición **Entity[notrepl]:** Alumno **Nev:** tiene Novia es intocable. Simplemente, el carácter de comodín de **Nev:** nos permite asignar coherentemente una interpretación al concepto “Novia”.

Ya, finalmente, el ejemplo que se ha venido manejando aparece completamente interpretado en la figura 6.11.

Declaraciones			
Conjuntos:	Entity[repl]: Alumnos Entity[repl]: Matrículas Entity[repl]: Asignaturas		
Subconjuntos:		Subs	
Individuos:	Entity[notrepl]: Asignatura	Bel: bel	Entity[repl]: Asignaturas
	Entity[notrepl]: Matrícula	Bel: bel	Entity[repl]: Matrículas
	Entity[notrepl]: Alumno	Bel: bel	Entity[repl]: Alumnos
Definiciones			
Índice	Definición		
1	Predicate: AND (Operand: P1, Operand: P2)		
2	Constraint: >=(Operand: Statespace: N ^o , Operand: Process: Sumatorio(Receives: Value: Aprobado, Receives: Statespace: N ^o _1))		
3	Statespace: N ^o Pof: attr Entity[notrepl]: Titulación		
4	Entity[notrepl]: Matrícula Pof: pof Entity[notrepl]: Alumno		
5	Entity[repl]: Asignaturas Pof: pof Entity[notrepl]: Matrícula		
6	Statespace: N ^o _1 Pof: attr Entity[notrepl]: Asignatura		
7	Statespace: Calificación Pof: attr Entity[notrepl]: Asignatura		
8	Statespace: Calificación Hval: val Value: Aprobado		
Proposiciones			
Índice	Concepto – 1	Asociación	Concepto – 2
1	Entity[notrepl]: Alumno	Rel: Supera	Entity[notrepl]: Titulación
2	Entity[notrepl]: Alumno	-Pof: Obtiene	Statespace: N ^o
3	Statespace: N ^o	Spec: De	Statespace: Créditos
4	Statespace: N ^o _1	Spec: De	Statespace: Créditos
5	Entity[notrepl]: Alumno	Rel: Realiza	Entity[notrepl]: Matrícula

Figura 6.11. Modelo Canónico de Requisitos

6.3.3. SELECCIÓN DEL MODELO CONCEPTUAL IDÓNEO

Una vez obtenido el MCR, todos los conceptos y asociaciones han sido etiquetados, esto es, cada concepto del DD ha sido interpretado como un elemento, y cada asociación como un enlace.

El Modelo Canónico posee la propiedad de que relaciona los distintos constructores de los modelos conceptuales, de tal forma que es posible reescribir cualquier modelo conceptual en términos del Modelo Canónico. En otras palabras: dado un modelo conceptual cualquiera, por ejemplo, un diagrama entidad-relación, o un diagrama de flujo de datos, es posible construir un Modelo Canónico de Requisitos traduciendo los constructores del diagrama entidad-relación a los constructores del Modelo Canónico.

Utilizando esta propiedad, es muy simple seleccionar el modelo conceptual más adecuado para representar el problema del usuario analizado y, lo más importante, esta selección es completamente objetiva, ya que se funda en la utilización de un simple cálculo. Para realizar dicho cálculo, se necesitan dos entradas. Por una parte, el MCR (por ejemplo, el MCR de la figura 6.11, el cual se utilizará de modo ilustrativo). Por otra parte, son necesarias un conjunto de tablas que relacionen las etiquetas del MCR con los distintos modelos conceptuales.

Las tablas que relacionan las etiquetas del MCR con los distintos modelos conceptuales se muestran en el anexo B. Existen tantas tablas como enlaces en el Modelo Canónico, razón para no incluir dichas tablas en esta sección. La confección del conjunto de tablas anterior es muy simple. Para cada enlace del Modelo Canónico, se construye una tabla separada. Dicha tabla (de doble entrada) permite relacionar todos los elementos entre sí. En cada celda de la tabla se indican los modelos conceptuales que permiten expresar la combinación de ambos elementos y el enlace correspondiente. Véase, a modo de ejemplo, la figura 6.12.

	entity [repl]	entity [notrepl]
entity [repl]	Diagrama de Clases	
entity [notrepl]		

Figura 6.12. Fragmento de la tabla para el enlace spec

La figura 6.12 muestra un fragmento de la tabla para el enlace spec. En la intersección entre “Entity[repl]” y “entity[repl]” aparecen listados un único modelo conceptual clásico: el diagrama de clases. Ello significa que, si existiera una combinación Entity[repl]-Spec-Entity[repl] en el MCR, esta combinación sólo podría expresarse mediante un diagrama de clases.

Este hecho (que una determinada combinación elemento-enlace-elemento pueda expresarse utilizando uno o varios modelos conceptuales) es la base del procedimiento de cálculo del Modelo Conceptual Idóneo.

Este cálculo se realiza como sigue:

1. Para cada proposición del MCR (esto es, una determinada combinación elemento-enlace-elemento), acúdase a la tabla correspondiente al enlace y apúntense los modelos conceptuales clásicos que permitan la expresión de dicha combinación elemento-enlace-elemento.
2. Una vez que se han revisado todas las proposiciones del MCR, se habrá obtenido una tabla que, si se sigue el esquema propuesto en la tabla 6.6, mostrada a continuación, contendrá qué modelos conceptuales soportan qué proposiciones del MCR. Adicionalmente, la última fila de la tabla contendrá el **número** de proposiciones del MCR que soporta cada modelo conceptual.
3. El Modelo Conceptual Idóneo será aquel que soporte la expresión de mayor número de proposiciones del MCR.

Para facilitar el cálculo, puede utilizarse una tabla como la mostrada en la tabla 6.6. Esta tabla de doble entrada permite relacionar cada proposición del MCR con el modelo conceptual que permite su expresión y realizar un cálculo sencillo del Modelo Conceptual Idóneo.

Proposición del M. Canónico	DFD DeMarco	DFD Ward	DFD Hatley	CFD Hatley	Mini- especificación Diccionario de Datos	DTE	FSM Hatley	Statecharts	Entidad- Relación Diagrama de Clases	Casos de Uso	PAT Hatley
entity[notrepl]: Matrícula pof: pof entity[notrepl]: Alumno					X				X		
Proposiciones MCR soportadas por cada M. Conceptual.....					1				1		

Tabla 6.6. Ejemplo de tabla para el cálculo del Modelo Conceptual Idóneo

Antes de pasar a mostrar un ejemplo de aplicación del procedimiento de cálculo, conviene realizar una precisión al procedimiento anterior. En el caso de los predicados, funciones y transiciones, existen dos cuestiones a tener en cuenta para identificar, en las tablas del anexo B, los modelos conceptuales que los soportan. Éstas son las siguientes:

- Los predicados, funciones y transiciones son, normalmente, proposiciones construidas recursivamente.
- Los predicados, funciones y transiciones no son proposiciones formadas mediante la regla concepto-asociación-concepto, ya que la existencia de múltiples operadores obliga a utilizar múltiples asociaciones para representar el predicado o la función (las ya indicadas asociaciones etiquetadas como 1, 2, ...n).

Ambas situaciones aparecen simultáneamente, ya que es normal que las proposiciones sean operadores de predicados y funciones. Por ello, para realizar el cálculo del Modelo Conceptual Idóneo es necesario **separar** o **expandir** todos los componentes de los predicados y funciones simultáneamente. Por

ejemplo, el predicado:

Predicate: AND (Operand: P1, Operand: P2)

esconde dos proposiciones distintas:

Predicate: AND Operand: P1

Predicate: AND Operand: P2

Nótese que el símbolo “P” hace referencia a una proposición.

Como segundo ejemplo, supóngase el predicado:

Constraint: >=(Operand: Statespace: N⁰, Operand: Process: Sumatorio(Receives: Value: Aprobado, Receives: Statespace: N⁰_1))

Para una persona es un tanto difícil de identificar las proposiciones. Sin embargo, estas son fáciles de visualizar en una versión gráfica del MCR, tal y como se muestra en la figura 6.13. La versión gráfica del Modelo Canónico se trata en el anexo A.

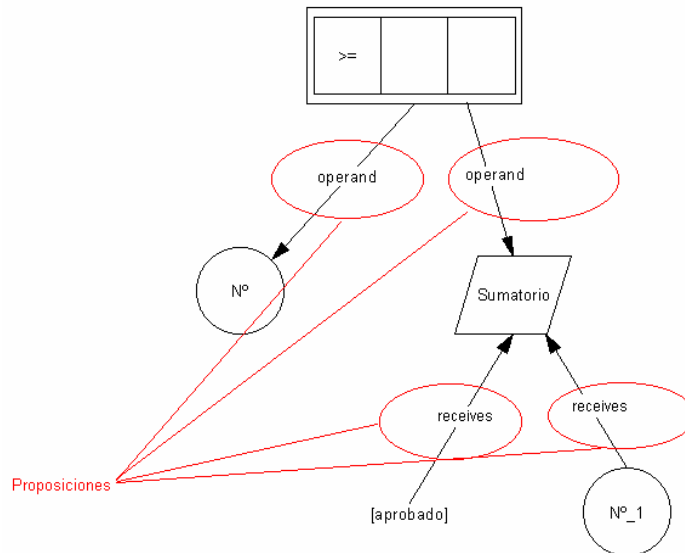


Figura 6.13. MCR en formato gráfico para el predicado **Constraint: >=(Operand: Statespace: N⁰, Operand: Process: Sumatorio(Receives: Value: Aprobado, Receives: Statespace: N⁰_1))**

Las proposiciones simples serían, por lo tanto, las siguientes:

Constraint: >= Operand: Statespace N⁰

Constraint: >= Operand: Process: Sumatorio

Process: Sumatorio **Receives:** Value: Aprobado

Process: Sumatorio **Receives:** Statespace N°_1

Aunque la identificación de las proposiciones simples que componen un predicado o parece una labor ardua, se debe indicar que es posible definir un procedimiento automatizado para identificar dichas proposiciones.

A continuación, se seguirá con el ejemplo planteado en el MCR de la figura 6.11. Los predicados utilizados a modo de ejemplo anteriormente pertenecen a dicho Modelo Canónico de Requisitos, por lo que se debe a las transformaciones expuestas la diferencia entre las proposiciones de la figura 6.11 y de la tabla 6.7, donde se muestra el cálculo del Modelo Conceptual Idóneo.

Proposición del Modelo Canónico de Requisitos	DFD DeMarco	DFD Ward	DFD Hatley	CFD Hatley	Mini- especificación Diccionario de Datos	DTE	FSM Hatley	Statecharts	Entidad- Relación Diagrama de Clases	Casos de Uso	PAT Hatley		
Entity[repl]: Alumnos					X				X	X			
Entity[repl]: Matrículas					X				X	X			
Entity[repl]: Asignaturas					X				X	X			
D1-1: Predicate: AND Operand: P1					X								
D1-2: Predicate: AND Operand: P2					X								
D2-1: Constraint: >= Operand: Statespace N°					X								
D2-2: Constraint: >= Operand: Process: Sumatorio					X								
D2-3: Process: Sumatorio Receives: Value: Aprobado	X	X	X		X								
D2-4: Process: Sumatorio Receives: Statespace N°_1	X	X	X		X								
D3: Statespace N° Pof: attr Entity[notrepl]: Titulación					X				X	X			
D4: Entity[notrepl]: Matrícula Pof: pof Entity[notrepl]: Alumno					X					X			
D5: Entity[repl]: Asignaturas Pof: pof Entity[notrepl]: Matrícula					X					X			
D6: Statespace: N°_1 Pof: attr Entity[notrepl]: Asignatura					X				X	X			
D7: Statespace: Calificación Pof: attr Entity[notrepl]: Asignatura					X				X	X			
D8: Statespace: Calificación Hval: val Value: Aprobado					X	X	X	X					
P1: Entity[notrepl]: Alumno Rel: Supera Entity[notrepl]: Titulación									X	X			
P2: Entity[notrepl]: Alumno Attr: Obtiene Statespace: N°									X	X			
P3 : Statespace: N° Spec: De Statespace: Créditos					X			X					
P4 : Statespace: N°_1 Spec: De Statespace: Créditos					X			X					
P5: Entity[notrepl]: Alumno Rel: Realiza Entity[notrepl]: Matrícula									X	X			
Proposiciones MCR soportadas por cada M. Conceptual.....	2	2	2	0	7	10	1	1	3	9	11	0	0

Tabla 6.7. Cálculo del Modelo Conceptual Idóneo para el ejemplo de la figura 6.11

Tras realizar el cálculo, puede observarse que el Modelo Conceptual Idóneo es el diagrama de clases. Ello parece lógico, dada la naturaleza altamente estática del ejemplo.

6.3.3.1. ESTUDIO DE LA MÉTRICA DE ADECUACIÓN

En la mayoría de los casos, el Modelo Conceptual Idóneo no es lo suficientemente potente como para expresar toda la información obtenida acerca del problema del usuario, esto es, para representar todas las proposiciones del MCR. Es más, el hecho de que un modelo se considere idóneo después de la aplicación del procedimiento de interpretación significa que es el modelo con mayor capacidad de representación para un análisis determinado de un problema concreto del usuario.

Es posible definir una métrica que permite medir la bondad del Modelo Conceptual Idóneo. Dicha métrica, que se ha denominado *Métrica de adecuación*, se define como:

$$\text{Adecuación} = \frac{\text{Nº de proposiciones representadas en el Modelo Conceptual Idóneo}}{\text{Nº de proposiciones totales del Modelo Canónico de Requisitos}}$$

La adecuación se expresa, como es evidente dada la fórmula anterior, como un valor entre 0 y 1 o, alternativamente, como un porcentaje. Cuanto mayor es el valor de la adecuación, mejor se expresa el problema del usuario en términos del Modelo Conceptual Idóneo.

A modo de ejemplo, la adecuación del diagrama de clases, que es el Modelo Conceptual Idóneo seleccionado en el ejemplo de la tabla 6.7, es del:

$$\frac{11}{20} = 0,55$$

o lo que es lo mismo, el 55%. Cuando el Modelo Conceptual Idóneo posee una adecuación muy baja, es posible que se den una de estas dos situaciones:

- Que el análisis se haya realizado deficientemente, esto es, que el Modelo Preliminar del Problema del Usuario y, por lo tanto, el Modelo Canónico no posean la calidad suficiente.
- Que el problema del usuario no pueda representarse adecuadamente con un único modelo conceptual.

La primera afirmación es muy difícil de probar. La segunda, sin embargo, admite una comprobación relativamente sencilla. En la mayoría de las ocasiones, durante un proyecto de desarrollo no se utiliza únicamente un modelo conceptual en la actividad de Análisis, sino que se utilizan varios modelos

conceptuales de forma conjunta, en el marco de un método de desarrollo, con el fin de representar mejor toda la información recogida durante el análisis. Un caso bien conocido de método de análisis podría ser, por ejemplo, el análisis estructurado moderno [Yourdon, 1989], el cual utiliza 5 modelos conceptuales distintos o el Método de Hatley [Hatley, 1984], el cual utiliza 7 modelos distintos.

MAON puede ser utilizado para seleccionar y derivar modelos conceptuales singulares o, por el contrario, seleccionar o derivar el conjunto de modelos conceptuales que exige un método de desarrollo. Para esto último, es únicamente necesario realizar una mínima modificación a la tabla de cálculo del Modelo Conceptual Idóneo, mostrada en la tabla 6.8.

Proposición del Modelo Canónico de Requisitos	Yourdon					Larman		
	DFD DeMarco	Mini-especificación de Diccionario de Datos	DTE	Entidad-Relación	Casos de Uso	Contratos (similar a mini-especificaciones)	Diagrama de Clases	
Entity[repl]: Alumnos			X	X			X	
Entity[repl]: Matrículas			X	X			X	
Entity[repl]: Asignaturas			X	X			X	
D1-1: Predicate: AND Operand: P1		X				X		
D1-2: Predicate: AND Operand: P2		X				X		
D2-1: Constraint: >= Operand: Statespace N°		X				X		
D2-2: Constraint: >= Operand: Process: Sumatorio		X				X		
D2-3: Process: Sumatorio Receives: Value: Aprobado	X	X				X		
D2-4: Process: Sumatorio Receives: Statespace N°_1	X	X				X		
D3: Statespace N° Pof: attr Entity[notrepl]: Titulación			X	X			X	
D4: Entity[notrepl]: Matrícula Pof: pof Entity[notrepl]: Alumno			X				X	
D5: Entity[repl]: Asignaturas Pof: pof Entity[notrepl]: Matrícula			X				X	
D6: Statespace: N°_1 Pof: attr Entity[notrepl]: Asignatura			X	X			X	
D7: Statespace: Calificación Pof: attr Entity[notrepl]: Asignatura			X	X			X	
D8: Statespace: Calificación Hval: val Value: Aprobado		X		X		X		
P1: Entity[notrepl]: Alumno Rel: Supera Entity[notrepl]: Titulación				X			X	
P2: Entity[notrepl]: Alumno Attr: Obtiene Statespace: N°				X			X	
P3: Statespace: N° Spec: De Statespace: Créditos			X					
P4: Statespace: N°_1 Spec: De Statespace: Créditos			X					
P5: Entity[notrepl]: Alumno Rel: Realiza Entity[notrepl]: Matrícula				X			X	
Proposiciones MCR soportadas por cada M. Conceptual.....	20				18			

Tabla 6.8. Cálculo del Método Idóneo para el ejemplo de figura 6.11

Dicha modificación consiste, básicamente, en la agrupación de los distintos modelos conceptuales individuales conforme a lo exigido por los métodos particulares que pueden ser utilizados en el desarrollo posterior. El cálculo de la tabla es similar que en el caso de los modelos considerados individualmente, con la única excepción de que las sumas finales no se calculan para cada modelo, sino para el método en su totalidad. Para ello, se considera que el método puede representar una determinada proposición del MCR **cuando alguno** de los modelos que lo forman puede representar dicha proposición.

De igual forma que con los modelos conceptuales individuales, el método seleccionado será aquel que puede representar más proposiciones del MCR.

A modo de ejemplo, se utilizará de nuevo el MCR reflejado en la figura 6.11 y, además, dos métodos como son el análisis estructurado moderno [Yourdon, 1989] y el análisis orientado a objetos de Larman [Larman, 1999]. En este caso, la selección del método idóneo se realizaría como indica la tabla 6.8, lo cual es simplemente una extensión del cálculo de la adecuación de modelos. Ésta se define como:

$$\text{Adecuación} = \frac{\text{Nº de proposiciones representadas en el Método Idóneo}}{\text{Nº de proposiciones totales del Modelo Canónico de Requisitos}}$$

En el ejemplo de la tabla 6.8, se observa que el número de proposiciones del MCR que es capaz de representar el método de Yourdon es mayor que el número de proposiciones que representa el método de Larman. El método de Yourdon sería, por lo tanto, el método idóneo.

Es necesario recordar, no obstante, que los distintos métodos de desarrollo poseen modelos dominantes. El modelo dominante, como ya se ha indicado en el Capítulo 2, guía todo el proceso de desarrollo prescrito por cada método particular. **Es por ello que la adecuación de métodos es mucho menos significativa que la adecuación de los modelos conceptuales dominantes de cada método.** En general, para calcular la adecuación de un método, es preferible calcular la adecuación de los modelos dominantes de dicho método y, como mucho, la adecuación de los modelos compatibles más importantes.

Así, por ejemplo, para el caso del análisis estructurado moderno [Yourdon, 1989], sería mucho más efectivo calcular la adecuación del diagrama de flujo de datos (dominante) y del diagrama entidad-relación (compatible), sin tener en cuenta las miniespecificaciones, diccionario de datos y DTE, mucho menos significativos. Para el caso del método de Larman, los modelos a considerar serían los casos de uso (dominante) y Diagrama de clases (compatible). De esta forma, el cálculo de la adecuación de ambos métodos sería el indicado en la tabla 6.9.

Como puede observarse, considerando únicamente los dos modelos indicados para cada método, la adecuación de ambos es similar. Idénticas consideraciones se pueden realizar para la adecuación teórica.

Proposición del M. Canónico	Yourdon		Larman	
	DFD DeMarco	Entidad-Relación	Casos de Uso	Diagrama de Clases
Entity[repl]: Alumnos		X		X
Entity[repl]: Matrículas		X		X
Entity[repl]: Asignaturas		X		X
D1-1: Predicate: AND Operand: P1				
D1-2: Predicate: AND Operand: P2				
D2-1: Constraint: >= Operand: Statespace N°				
D2-2: Constraint: >= Operand: Process: Sumatorio				
D2-3: Process: Sumatorio Receives: Value: Aprobado	X			
D2-4: Process: Sumatorio Receives: Statespace N°_1	X			
D3: Statespace N° Pof: attr Entity[notrepl]: Titulación		X		X
D4: Entity[notrepl]: Matrícula Pof: pof Entity[notrepl]: Alumno				X
D5: Entity[repl]: Asignaturas Pof: pof Entity[notrepl]: Matrícula				X
D6: Statespace: N°_1 Pof: attr Entity[notrepl]: Asignatura		X		X
D7: Statespace: Calificación Pof: attr Entity[notrepl]: Asignatura		X		X
D8: Statespace: Calificación Hval: val Value: Aprobado				
P1: Entity[notrepl]: Alumno Rel: Supera Entity[notrepl]: Titulación		X		X
P2: Entity[notrepl]: Alumno Attr: Obtiene Statespace: N°		X		X
P3 : Statespace: N° Spec: De Statespace: Créditos				
P4 : Statespace: N°_1 Spec: De Statespace: Créditos				
P5: Entity[notrepl]: Alumno Rel: Realiza Entity[notrepl]: Matrícula		X		X
Proposiciones MCR soportadas por cada M. Conceptual.....	11		11	

Tabla 6.9. Nuevo cálculo del Método Idóneo para el ejemplo de figura 6.11

6.4. DERIVACIÓN DEL MODELO CONCEPTUAL SELECCIONADO

Una vez que se ha obtenido el Modelo Canónico de Requisitos (MCR), y se ha identificado el Modelo Conceptual Idóneo, solo resta aplicar la Técnica de Derivación del Modelo Conceptual Seleccionado (Técnica DMCS) para obtener el modelo conceptual clásico que se ha identificado como Modelo Idóneo. La Técnica DMCS se basa en la utilización de un conjunto de tablas, que se han denominado Tablas de Derivación, las cuales se presentan en el anexo C debido al gran número de las mismas (existen tantas tablas como modelos conceptuales distintos se han considerado en el presente trabajo de Tesis).

Existe una tabla para cada modelo conceptual. En cada tabla, se listan exhaustivamente todas las combinaciones elemento-enlace-elemento que soporta el modelo conceptual. Para cada combinación

elemento-enlace-elemento, se determina qué fragmento del modelo conceptual permite expresar dicha combinación. En algunos casos es necesario, adicionalmente, aplicar algunas reglas que afectan a la obtención de fragmentos. Dichas reglas, particulares para cada modelo conceptual, se indican, igualmente, en el anexo C.

Las tablas anteriormente indicadas son el núcleo de la Técnica DMCS. Utilizando dichas tablas, la derivación del Modelo Conceptual Idóneo es prácticamente trivial. Únicamente es necesario realizar tres pasos procedimentales que se detallan en las secciones siguientes:

1. Aislar las proposiciones del MCR expresables en el modelo conceptual considerado Modelo Idóneo.
2. Para cada proposición, aplicar la transformación indicada en la tabla de derivación correspondiente. En caso de que existan reglas de derivación, aplicar éstas antes de obtener cada una de los fragmentos.
3. Una vez obtenidos todos los fragmentos, unir todos ellos para obtener el modelo conceptual definitivo.

6.4.1. AISLAMIENTO DE PROPOSICIONES

El aislamiento de las proposiciones del MCR expresables en el modelo conceptual considerado idóneo es trivial, ya que dichas proposiciones son, precisamente, las que se han utilizado en el cálculo del Modelo Conceptual Idóneo. Por ejemplo, la figura 6.14 muestra las proposiciones derivables a un diagrama de clases, el cual fue el Modelo Conceptual Idóneo identificado en el ejemplo que se ha utilizado a lo largo del presente capítulo.

Declaraciones			
Conjuntos:	Entity[rep]: Alumnos Entity[rep]: Matrículas Entity[rep]: Asignaturas		
Subconjuntos:		Subs	
Individuos:	Entity[notrep]: Asignatura	Bel: bel	Entity[rep]: Asignaturas
	Entity[notrep]: Matrícula	Bel: bel	Entity[rep]: Matrículas
	Entity[notrep]: Alumno	Bel: bel	Entity[rep]: Alumnos
Definiciones			
Índice	Definición		
1	Predicate: AND (Operand: P1, Operand: P2)		
2	Constraint: >=(Operand: Statespace: Nº, Operand: Process: Sumatorio(Receives: Value: Aprobado, Receives: Statespace: Nº 1))		
3	Statespace: Nº Pof: attr Entity[notrep]: Titulación		
4	Entity[notrep]: Matrícula Pof: pof Entity[notrep]: Alumno		
5	Entity[rep]: Asignaturas Pof: pof Entity[notrep]: Matrícula		
6	Statespace: Nº 1 Pof: attr Entity[notrep]: Asignatura		
7	Statespace: Calificación Pof: attr Entity[notrep]: Asignatura		
8	Statespace: Calificación Hval: val Value: Aprobado		
Proposiciones			
Índice	Concepto – 1	Asociación	Concepto – 2
1	Entity[notrep]: Alumno	Rel: Supera	Entity[notrep]: Titulación
2	Entity[notrep]: Alumno	-Pof: Obtiene	Statespace: Nº
3	Statespace: Nº	Spec: De	Statespace: Créditos
4	Statespace: Nº 1	Spec: De	Statespace: Créditos
5	Entity[notrep]: Alumno	Rel: Realiza	Entity[notrep]: Matrícula

Proposiciones derivables a un Diagrama de Clases

Figura 6.14. Proposiciones derivables a un diagrama de clases en el ejemplo que se ha manejado en este capítulo (figura 6.11)

6.4.2. OBTENCIÓN DE FRAGMENTOS

Una vez que se han separado las proposiciones representables en el Modelo Conceptual Idóneo, el paso siguiente es obtener los fragmentos correspondientes, para cada proposición, de dicho modelo conceptual. Para ello se utilizan las Tablas de Derivación, las cuales se exponen en el anexo C.

La utilización de las tablas de derivación es la siguiente. Para cada proposición del MCR, se busca la combinación elemento-enlace-elemento en la tabla de derivación. La columna "Deriva a:" de la tabla de derivación indica el fragmento de modelo conceptual que se deriva de dicha combinación elemento-enlace-elemento. La única precaución necesaria es asignar correctamente, a cada fragmento del modelo conceptual, los nombres adecuados. Dichos nombres se corresponden con los nombres de los conceptos y asociaciones del MCR. Adicionalmente, en el caso de que existan reglas de derivación, éstas deben aplicarse antes de la derivación de cada fragmento.

Un ejemplo ayudará a comprender mejor la aplicación de las Tablas de Derivación. Considérese, a modo de ejemplo, la siguiente proposición, incluida en el Modelo Canónico de Requisitos mostrado en la figura 6.11. Téngase en cuenta, adicionalmente, que el modelo idóneo es el diagrama de clases, cuya Tabla de Derivación se adjunta en el anexo C.

Statespace: Calificación **Pof:** attr **Entity[notrepl]:** Asignatura

Esta proposición contiene un elemento Entity[notrepl]. Siguiendo la regla (1), comprobamos que existe un elemento Entity[repl] (esto es, **Entity[repl]:** Asignaturas) que cumple que:

Entity[notrepl]: Asignatura **Bel:bel** **Entity[repl]:** Asignaturas

tal y como puede comprobarse en el Modelo Canónico de Requisitos mostrado en la figura 6.11. Por ello, la regla (1) es aplicable, y en lugar de **Statespace:** Calificación **Pof:** attr **Entity[notrepl]:** Asignatura, para la derivación del modelo conceptual deberemos utilizar:

Statespace: Calificación **Pof:** attr **Entity[repl]:** Asignatura.

Puede realizarse otro ejemplo de la aplicación de las reglas de derivación mediante la proposición:

Statespace: N^o **Pof:** attr **Entity[notrepl]:** Titulación

Esta proposición contiene un elemento de tipo Entity[notrepl] (titulación), para el cual no es posible aplicar las reglas (1) o (2), ya que dicho elemento no pertenece ni es subconjunto de ningún otro.

Entra en juego, por lo tanto, la regla (3), que nos indica que en caso de que las reglas (1) o (2) no sean aplicables, consideremos si, semánticamente, es coherente:

Entity[repl]: Titulación

O, lo que es lo mismo, que consideremos si deseamos incluir una clase “Titulación” en el modelo. En este caso, es el ingeniero el que debe responder afirmativa o negativamente a dicha pregunta.

Ya para finalizar, y tras la aplicación de la Técnica DMCS, los fragmentos de diagrama de clases obtenidos para las proposiciones derivables del ejemplo manejado a lo largo del capítulo, las cuales se mostraron en la figura 6.14, se indican en la tabla 6.10. Puede comprobarse que se ha decidido considerar “Titulación” como una Entity[repl]:.


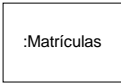
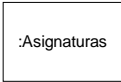
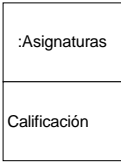
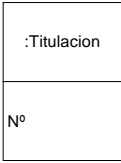
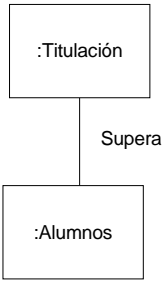
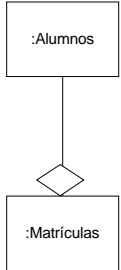
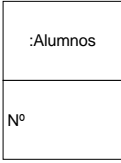
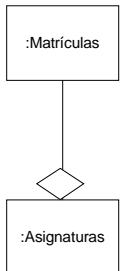
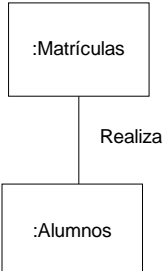
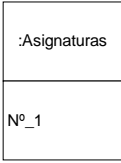
<p>Entity[repl]: Alumnos</p>		<p>Entity[repl]: Matrículas</p>	
<p>Entity[repl]: Asignaturas</p>		<p>Statespace: Calificación</p> <p>Pof: attr</p> <p>Entity[notrepl]: Asignatura</p>	
<p>Statespace: N°</p> <p>pof: attr</p> <p>entity[notrepl]: Titulación</p>		<p>Entity[notrepl]: Alumno</p> <p>Rel: Supera</p> <p>Entity[notrepl]: Titulación</p>	
<p>entity[notrepl]: Matrícula</p> <p>Pof: pof</p> <p>Entity[notrepl]: Alumno</p>		<p>Entity[notrepl]: Alumno</p> <p>-Pof: Obtiene</p> <p>Statespace: N°</p>	
<p>Entity[repl]: Asignaturas</p> <p>Pof: pof</p> <p>Entity[notrepl]: Matrícula</p>		<p>Entity[notrepl]: Alumno</p> <p>Rel: Realiza</p> <p>Entity[notrepl]: Matrícula</p>	
<p>Statespace: N°_1</p> <p>Pof: attr</p> <p>Entity[notrepl]: Asignatura</p>			

Tabla 6.10. Fragmentos del diagrama de clases

6.4.3. UNIÓN DE FRAGMENTOS

Una vez obtenidos los fragmentos individuales, el último paso de la Técnica DMCS es unir dichos fragmentos para obtener el modelo deseado. La unión de dichos fragmentos puede realizarse automáticamente, gracias a que cada se ha asignado el nombre correcto a cada constructor del Modelo Conceptual Idóneo.

A modo de ejemplo, el diagrama de clases que se obtendría a partir de los fragmentos de la tabla 6.10 sería el mostrado en la figura 6.15.

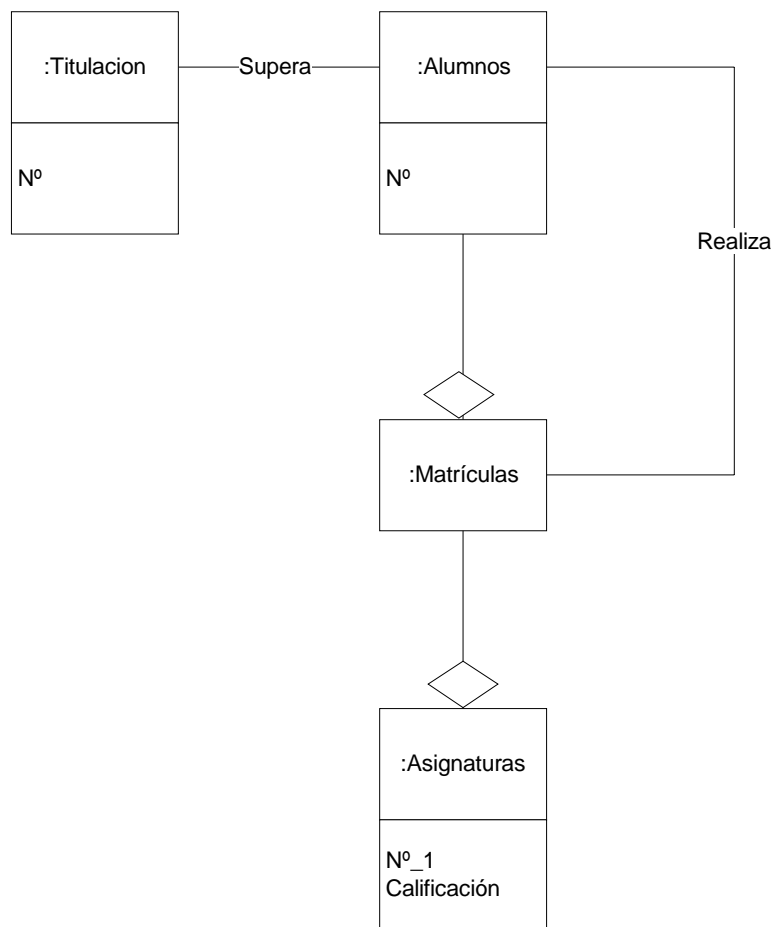


Figura 6.15. Diagrama de clases obtenido por la unión de fragmentos

6.4.4. REFINAMIENTO DEL MODELO OBTENIDO

Cabe realizar dos consideraciones a la Técnica DMCS:

1. En la exposición realizada hasta el momento, se ha considerado únicamente la derivación del Modelo Conceptual Idóneo. No obstante, es necesario indicar que en el primer paso de la Técnica DMCS es posible aislar las proposiciones de dicho modelo o de cualquier otro que el

analista deseara derivar. Ello permite aplicar la Técnica DMCS no para obtener el Modelo Conceptual Idóneo, sino aquel modelo deseado por el analista.

2. La Técnica DMCS no genera, en todos los casos, el modelo conceptual "perfecto". Es el analista el que, en el proceso de desarrollo posterior, debe "refinar" el modelo conceptual obtenido.

Por ejemplo, el diagrama de objetos mostrado en la figura 6.15 posee dos relaciones entre "Matrícula" y "Alumno". Ambas relaciones son, lógicamente, redundantes. El analista, estudiando la semántica del modelo, deberá decidir qué relación sobra y qué relación debe mantenerse. Una posible decisión sería eliminar la agregación, la cual parece un tanto forzada, y mantener únicamente la asociación. El diagrama de clases resultante después de tomar esta decisión se muestra en la figura 6.16.

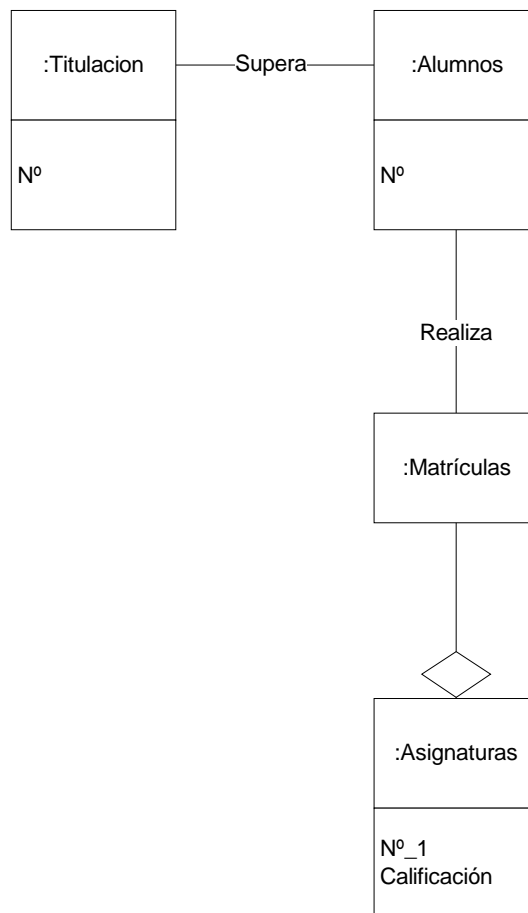


Figura 6.16. Diagrama de clases después del refinamiento

PARTE III

VALIDACIÓN

7. VALIDACIÓN

7.1. PLANTEAMIENTO DE LA VALIDACIÓN

En el presente capítulo, se describirá la validación a la que ha sido sometido el Método de Análisis Orientado a la Necesidad (MAON). La validación tiene por objetivo comprobar que las hipótesis de trabajo, enunciadas en el capítulo 3, Planteamiento del Problema, se verifican. A modo de recordatorio, dichas hipótesis de trabajo son:

HG: Es viable obtener definir un método de pre-análisis capaz de determinar la aproximación de desarrollo más adecuada para la necesidad del usuario que el sistemas software pretende atender, y permite continuar el desarrollo con las aproximaciones tradicionales: Estructurada, Orientada a Objetos y de Tiempo Real.

Esta hipótesis, demasiado compleja para ser validada en conjunto, ha sido subdividida en 3 sub-hipótesis, las cuales son las que se validarán efectivamente:

- SH1: El método de pre-Análisis es capaz de determinar qué aproximación de desarrollo (estructurada, orientada a objetos, de tiempo real) es la más adecuada para proseguir el proceso de desarrollo.
- SH2: El método de pre-Análisis permite derivar los modelos utilizados por las aproximaciones de desarrollo existentes en la actualidad (estructurada, orientada a objetos y de tiempo real).

SH3: El método de pre-Análisis tiene la capacidad de representación suficiente para ser utilizado en las circunstancias en las que serían utilizables las aproximaciones estructuradas, orientadas a objetos o de tiempo real.

Para validar las hipótesis, es necesario identificar un conjunto de problemas, cada uno de los cuales pueda resolverse utilizando una aproximación de desarrollo determinada, o una combinación de aproximaciones simultáneamente. Una vez identificados dichos problemas, y asociado a cada uno de ellos la(s) aproximación(es) de desarrollo adecuadas, las hipótesis SH1, SH2 y SH3 admiten una validación inmediata, tal y como muestra la figura 7.1:

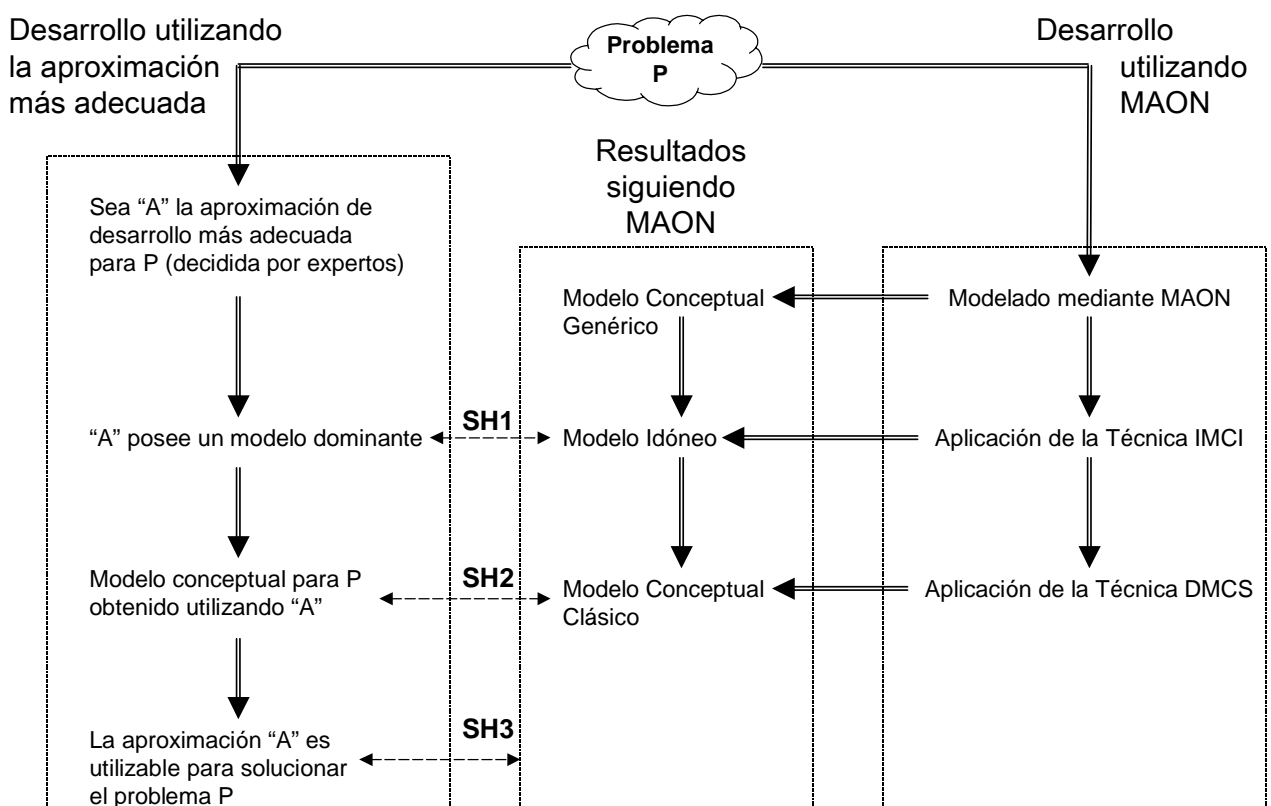


Figura 7.1. Planteamiento de la validación

- SH1 se validará cuando, para cada problema, el Modelo Conceptual Idóneo, obtenido mediante la técnica IMCI, coincida con el modelo conceptual utilizado por la aproximación de desarrollo adecuada para dicho problema. La adecuación de una aproximación se decide, como se explicará más adelante, mediante consenso de expertos y comparación de los modelos obtenidos por las tres aproximaciones de desarrollo para el mismo problema.

- SH2 se validará cuando sea posible, a partir del Modelo Conceptual Genérico, obtener los modelos propios de las aproximaciones de desarrollo estructurada, orientada a objetos y tiempo real.
- SH3 se validará cuando se aplique MAON a los problemas planteados, es decir, cuando MAON pueda aplicarse sobre problemas propios de las aproximaciones Estructurada, Orientada a Objetos y de Tiempo Real. Ello equivale a decir que SH3 se validará cuando SH1 (identificación del modelo idóneo) y SH2 (derivación de los modelos conceptuales) se hayan efectivamente validado, como se explicará más adelante, en la sección 7.5.

El paso más delicado del procedimiento de validación, esbozado anteriormente, es la identificación de los problemas para ser resueltos utilizando una u otra aproximación de desarrollo, así como la adecuación de tal aproximación al problema. Este objetivo es difícil de conseguir, debido a la ausencia de esquemas bien definidos de clasificación de problemas en la literatura [Glass et al., 1995]. Por ello, para conseguir tal conjunto de problemas, se ha optado por utilizar los esquemas de clasificación de modelos conceptuales, de los que sí existen varias propuestas, tales como [Webster, 1988] [Zave, 1990] [Davis, 1993] [Blum, 1996]. La correspondencia que existe entre las aproximaciones de desarrollo y los modelos conceptuales que estas aproximaciones utilizan permite realizar esta trasposición ya que, tal y como se ha indicado en el capítulo 2, de Estado de la Cuestión, las distintas aproximaciones de desarrollo se caracterizan por la utilización de un modelo conceptual (el modelo dominante) que guía todo el proceso de desarrollo utilizando dicha aproximación.

Se ha utilizado el esquema de clasificación de [Davis, 1993], ya que es el más sencillo y, probablemente, el más difundido de todos, hasta tal punto que es isomórfico a los grandes tipos de aproximaciones de desarrollo (estructurada, orientada a objetos y tiempo real). Davis clasifica los modelos en tres grupos: Orientados a funciones, orientados a objetos y orientados a estados. A partir de ahora se denominarán modelos tipo F, O y E, respectivamente. Cada tipo de modelos es adecuado para aquellos problemas cuya estructura corresponda, aproximadamente, con los elementos constructores de dichos modelos (llámense problemas F, O y E, respectivamente). Ello permite construir, razonablemente, casos de prueba representativos de dichos tipos de problemas, que se corresponden directamente con los problemas abordables bajo las aproximaciones estructurada, orientada a objetos y tiempo real, respectivamente. Adicionalmente, también pueden construirse casos de prueba para problemas mixtos (FO, FE, OE, FOE), los cuales podrán resolverse utilizando varias aproximaciones de desarrollo simultáneamente.

En total, se han diseñado 7 casos de prueba utilizando la estrategia enunciada anteriormente (tipos F, O, E, FO, FE, OE y FOE). Dichos casos de prueba se muestran en la tabla 7.1.

La estrategia de validación contempla utilizar **varios grupos de personas, con diferentes capacidades:**

- 4 profesores, los cuales resolverán los casos de prueba utilizando los modelos de todas las aproximaciones de desarrollo. La comparación entre modelos para el mismo problema realizada por estos expertos permitirá decidir qué aproximación(es) de desarrollo es(son) más adecuada(s).

- 5 desarrolladores con experiencia en una o varias aproximaciones de desarrollo, los cuales utilizarán aquella aproximación que consideren más adecuada.
- 18 estudiantes de grado y postgrado, los cuales utilizarán, igualmente, la aproximación de desarrollo que consideren más adecuada.
- 3 alumnos de grado y postgrado que obtendrán los modelos conceptuales para cada caso utilizando MAON.

El primer grupo (grupo de control –GC-) tiene como objetivo verificar que cada caso realmente se corresponde con un problema F, O, E, FO, FE, OE y FOE. El segundo y tercer grupo (grupos de prueba 1 y 2 –GP1 y GP2-) representan el modo actual de realizar la modelización conceptual. Es contra estos dos grupos con quién se comparará la actuación del cuarto grupo (Grupo de Prueba 3 –GP3-) utilizando MAON. Se ha querido observar dos grupos utilizando el modo tradicional para comparar MAON tanto con desarrolladores inexpertos (estudiantes) como con desarrolladores expertos.

En las secciones siguientes, se presentarán los resultados de la validación realizada. En la sección 7.2, se presentarán los casos de prueba, junto con modelos confeccionados por el GC. La sección 7.3 se dedicará a analizar la actuación de los GP1, GP2 y GP3 a la hora de identificar el modelo más adecuado para cada caso de prueba. Ambas secciones, conjuntamente, permitirán validar la sub-hipótesis SH1.

La sección 7.4 se ocupará de las generación de los modelos conceptuales, propios de cada aproximación de desarrollo, para los casos de prueba 1-7 utilizando la técnica DMCS. Ello permitirá validar la sub-hipótesis SH2.

Finalmente, la sección 7.5 presentará la validación de la sub-hipótesis SH3, la cual se deriva de la validación de las sub-hipótesis SH1 y SH2.

Nº	Tipo	Enunciado del caso
1	F	<p>Los clientes de la compañía IJK realizan pedidos de productos, habitualmente a diario. Los pedidos son comunicados al almacén, donde se recogen los productos, se empaquetan y se preparan para su envío posterior al cliente. Los pedidos que no se pueden servir por falta de mercancía se archivan a la espera de que llegue la mercancía necesaria para servirlos.</p> <p>Los pedidos servidos al cliente se envían al departamento de Contabilidad para su facturación. Contabilidad confecciona las facturas y las remite al cliente para que procedan a su abono. Cuando el banco comunica que un cliente ha abonado una factura, Contabilidad envía al cliente un recibo del pago.</p>
2	O	<p>La facultad XYZ permite que los alumnos se matriculen libremente de las asignaturas que desean cursar en cada año académico. Cada asignatura posee un profesor coordinador y un número variable de profesores que imparten docencia en la misma.</p> <p>A lo largo del curso académico, cada alumno realiza un conjunto de trabajos teóricos y prácticos. La calificación de dichos trabajos corre a cargo de alguno de los profesores que imparten docencia. El examen final, sin embargo, es realizado y calificado por el profesor coordinador. La calificación final del alumno es la media de todas las calificaciones de los trabajos y de la calificación del examen. La nota media se registra en el expediente del alumno para el cálculo de la nota media de la carrera.</p>
3	E	<p>El cajero automático del banco ACME debe poder realizar reintegros, imposiciones e informes de los saldos.</p> <p>El cajero debería comenzar a funcionar una vez un cliente introduce una tarjeta de crédito. Después de leer los datos de la tarjeta y comprobar que es la adecuada, se pedirá el Número Identificativo Personal del cliente (el PIN). En el caso de que el PIN sea correcto, el cliente podrá realizar reintegros, imposiciones e informes de saldo tantas veces como quiera.</p> <p>El cliente dispondrá de un botón TERMINAR en el panel del cajero. El botón TERMINAR servirá para que el cliente finalice de operar con el cajero, devolviéndolo éste la tarjeta que el cliente había previamente introducido.</p>
4	FO	<p>El hospital 123 posee dos procedimientos para la admisión de pacientes. El primero es el ingreso de los pacientes en lista de espera. El segundo es el ingreso de los pacientes que acuden a urgencias.</p> <p>Cuando se ingresa un paciente en lista de espera, se le asigna una habitación en la planta correspondiente a la dolencia de la que se le va a tratar. Por ejemplo, si un paciente va a ser operado para colocar un <i>by-pass</i> coronario, se le asignaría una habitación en la planta de cardiología.</p> <p>Los pacientes ingresados por lista de espera tiene un médico de referencia asignado. Este médico puede ser cualquiera de los pertenecientes a la especialidad de que va a ser tratado.</p> <p>Por el contrario, los pacientes que ingresan por Urgencias son, antes de cualquier trámite administrativo, atendidos de forma inmediata. Una vez atendidos, se les asigna una habitación en las tres horas siguientes a su ingreso, siguiendo las mismas pautas que los pacientes admitidos por lista de espera. La única diferencia es que su médico de referencia no aquel médico que lo atendió en Urgencias.</p>
5	FE	<p>El Videocasete PQR realiza una diversidad de funciones. Las funciones básicas son el manejo de la cinta (rebobinar, adelantar y parar) y la reproducción (play).</p> <p>Adicionalmente, el videocasete PQR permite realizar la grabación de películas. La grabación puede ser inmediata (siempre y cuando no se esté reproduciendo una cinta) o planificada. El videocasete permite planificar hasta 5 sesiones de grabación simultáneamente.</p> <p>Por último, como en todo video, PQR permite establecer la fecha y la hora actuales. Ello es necesario para que las sesiones planificadas de grabación se inicien y terminen en la fecha y hora que espera el usuario.</p>
6	OE	<p>La reparación de las máquinas HBC sigue un orden predeterminado. En primer lugar, debe verificarse que cada componente está situado en el slot correspondiente. Es un error bastante frecuente que las HBC fallen porque un determinado componente se sitúa en un slots erróneo. Una vez comprobados los slots, se debe revisar la versión de cada componente, porque no todas las versiones de HBC soportan los mismos componentes.</p> <p>Las revisiones anteriores son de rutina. La verdadera revisión se inicia comprobando el cuadro de luces de los componentes. En caso de funcionamiento correcto de los componentes, el cuadro de luces del componente muestra una secuencia única de luces, propia de cada componente. Alguna luz del cuadro no coincide, dicha luz indica el PIN del slot que debe ser cambiado.</p>
7	FOE	<p>El sistema de ataque del avión JFK deberá poseer la capacidad de discriminar entre los distintos tipos de aviones de combate existentes. Los tipos de aviones son los "amigos", esto es, los aviones de las fuerzas aéreas aliadas y "enemigos", esto es, los aviones de los países potencialmente hostiles.</p> <p>Cuando el sistema de tipo detecte un avión, comprobará si es amigo o enemigo. Después de esta comprobación, identificará qué tipo de armamento es más adecuado para realizar el ataque, ya que para cada tipo de avión, existe una arma más eficaz que otra.</p> <p>El sistema de ataque preparará para el disparo el arma más eficaz disponible. Ello implica verificar que existe munición para realizar el ataque. En caso de que no exista munición, se utilizará la segunda arma más eficaz, y así sucesivamente.</p> <p>Antes de disparar, se deberá comprobar si el piloto ha establecido el modo automático de disparo o el modo manual. En el modo automático, el sistema realizará automáticamente el disparo si el avión detectado es enemigo. En el modo manual, el sistema esperará por una orden del piloto antes de realizar el disparo. Adicionalmente, si el avión es amigo, el sistema pedirá una confirmación de la orden antes de disparar.</p>

Tabla 7.1. Casos de prueba

7.2. RESOLUCIÓN DE LOS CASOS DE PRUEBA POR EL GRUPO DE CONTROL

El GC, formado por 4 profesores, resolvió conjuntamente los casos de prueba. Tal y como se ha indicado, los casos de prueba fueron resueltos utilizando todos los modelos dominantes de las aproximaciones de desarrollo estructurada, orientada a objetos y de tiempo real. Posteriormente, dichos modelos fueron comparados para decidir cuáles eran más expresivos en cada caso particular, de modo que se decidiera antes la aproximación más adecuada para cada caso de prueba. La totalidad de los modelos se muestra en el anexo AV1 del volumen II de la Tesis Doctoral "Datos de la Validación". Según la opinión de los miembros del CG, los modelos más expresivos para cada caso son los mostrados en la tabla 7.2.

Nº	Tipo	Modelos más expresivos
1	F	- Diagrama de Flujo de Datos
2	O	- Diagrama de Clases
3	E	- Diagrama de Transición de Estados
4	FO	- Diagrama de Flujo de Datos - Diagrama de Clases
5	FE	- Diagrama de Flujo de Datos - Diagrama de Transición de Estados
6	OE	- Diagrama de Clases - Diagrama de Transición de Estados
7	FOE	- Diagrama de Flujo de Datos - Diagrama de Clases - Diagrama de Transición de Estados

Tabla 7.2. Decisión del GC

Como puede observarse, se produce una perfecta ratificación del diseño de los casos de prueba por los resultados obtenidos por el GC. Concretamente, los miembros del GC han identificado como modelos adecuados para cada caso aquellos que coincidan con el tipo del caso. Por ejemplo, para el caso nº 4, los miembros del GC han identificado el Diagrama de Flujo de Datos y el Diagrama de Clases como modelos más adecuados, lo cual está en perfecta concordancia con el tipo FO del caso. Lo mismo puede afirmarse para el resto de los casos de prueba.

Puede afirmarse, por lo tanto, que los casos de prueba han sido diseñados adecuadamente. Ello permite la resolución de los casos de prueba por los GP1, GP2 y GP3, con la finalidad de comparar la capacidad de MAON a la hora de identificar el modelo más adecuado frente a las capacidades, fundadas en criterios subjetivos de los sujetos.

7.3. VALIDACIÓN DE SH1: SELECCIÓN DEL MODELO MÁS ADECUADO

7.3.1. IDENTIFICACIÓN DE LOS MODELOS MÁS ADECUADOS POR GP1, GP2 Y GP3

Los casos de prueba han sido resueltos, independientemente, por los GP1, GP2 y GP3. Las respuestas individualizadas de los sujetos de los grupos GP1, GP2 y GP3 pueden consultarse en los

anexos AV2, AV3 y AV4 del volumen II de la Tesis Doctoral “Datos de la Validación”, respectivamente. La tabla 7.3 muestra los resultados obtenidos. Los datos obtenidos por el GP3 se han agrupado debido a la similitud de los Modelos Conceptuales Genéricos obtenidos, los cuales, tras su interpretación y derivación, generan, igualmente, modelos conceptuales similares. Este resultado era esperable, puesto que uno de los objetivos de MAON es su alto grado de independencia del sujeto, de modo similar a lo que ocurre en las técnicas y métodos de las ingenierías clásicas.

		CP 1	CP 2	CP 3	CP 4	CP 5	CP 6	CP 7	Índice de aciertos	
		P	O	E	PO	PE	OE	POE	Total	%
GP1	Especialista 1	O	O	O	O	O	O	O	4	57%
	Especialista 2	O	O	O	F	O	O	O	4	57%
	Especialista 3	F	O	E	O	F	E	E	7	100%
	Especialista 4	F	O	E	F	F	O	F	7	100%
	Especialista 5	F	O	F	F	F	O	F	6	86%
	Acierto	60%	100%	20%	100%	60%	100%	100%	5,6	80%
GP2	Alumno 1	F	O	E	F	E	E	E	7	100%
	Alumno 2	F	O	E	O	E	E	E	7	100%
	Alumno 3	F	O	O	O	O	O	E	5	71%
	Alumno 4	E	O	E	O	E	E	F	6	86%
	Alumno 5	O	O	O	O	O	O	E	4	57%
	Alumno 6	F	O	E	F	E	F	O	6	86%
	Alumno 7	F	O	E	O	E	F	E	6	86%
	Alumno 8	F	O	E	O	E	E	F	7	100%
	Alumno 9	O	O	F	O	F	E	F	5	71%
	Alumno 10	F	O	E	O	O	O	E	6	86%
	Alumno 11	F	O	E	---	O	O	E	5	71%
	Alumno 12	F	O	E	F	O	E	E	6	86%
	Alumno 13	O	F	O	O	O	E	E	3	43%
	Alumno 14	F	O	E	O	E	E	E	7	100%
	Alumno 15	F	O	E	F	E	F	E	6	86%
	Alumno 16	F	F	E	F	O	E	O	5	71%
	Alumno 17	F	O	E	O	E	E	E	7	100%
	Alumno 18	F	O	E	O	E	E	O	7	100%
Acierto	78%	89%	78%	94%	61%	83%	100%	5,8	83%	
GP3	Alumnos	F	O	E	O	F	O	E	7	100%
	Acierto	100%	100%	100%	100%	100%	100%	100%	7,0	100%

Tabla 7.3. Resultados de los GP1, GP2 y GP3. Las siglas F, O y E aparecen en la tabla cuando un determinado sujeto experimental ha seleccionado un modelo de tipo F, O o E, respectivamente.

Puede observarse que el GP3, el cual utilizó MAON, ha sido el grupo con mejores resultados, **ya que en un 100% de los casos identifica el modelo idóneo**, en concordancia con el diseño del caso de prueba y la decisión tomada por el GC. Esto es, estos datos **validan la sub-hipótesis SH1**.

Por el contrario, el GP1 (esto es, los desarrolladores) obtiene un 80% de aciertos (un 20% menor que el GP3), mientras que el GP2 (alumnos de grado y postgrado) obtiene un 83% de aciertos (17 puntos por debajo del acierto obtenido por GP3).

Merece especial mención que el porcentaje de acierto de GP1 sea menor que el de GP2 (en 3 puntos), en la medida en que los desarrolladores con experiencia deberían ser mejores que los alumnos a la hora de juzgar qué aproximación es más adecuada para un problema determinado, esto es, de relacionar problemas con soluciones. Sin embargo, entraba dentro de lo previsible que la realidad fuera la contraria, como de hecho ha ocurrido. Dado que los desarrolladores están generalmente acostumbrados a utilizar su aproximación preferida en todos los proyectos de desarrollo, los modelos de dicha aproximación actúan a modo de gafas que sólo permiten ver el dominio del problema de un modo determinado, esto es, utilizando los conceptos que utilizan los modelos conceptuales de una aproximación de desarrollo concreta. En general, los alumnos seleccionan mejor no porque tengan más conocimiento que los desarrolladores, sino porque están menos polarizados respecto a la utilización de una u otra aproximación de desarrollo.

No obstante, y a la vista de los datos obtenidos, la capacidad del GP1 y GP2, a la hora de identificar el modelo más adecuado, no ha sido tremendamente distinta a la de GP3, ya que la diferencia en el porcentaje de aciertos es relativamente reducida. Este hecho, el cual parece cierto a la vista de los datos no resulta tan claro si los datos se analizan respecto a tres criterios que se estudiarán en las secciones siguientes:

- Capacidad de los grupos de prueba a la hora de identificar el modelo más adecuado en los casos puros (F, O, E).
- Capacidad de los grupos de prueba a la hora de identificar el modelo más adecuado en los casos mixtos (FO, FE, OE, FOE).
- Tendenciosidad de los sujetos, esto es, en qué medida las respuestas de los sujetos están sesgadas hacia una aproximación de desarrollo determinada.

7.3.2. ESTUDIO DEL MODELO MÁS ADECUADO EN LOS CASOS PUROS

Se han denominado casos puros a aquellos casos de prueba que pertenecen a un único tipo, esto es, F, O o E. Al analizar separadamente los casos puros, se obtienen los resultados mostrados en la tabla 7.4.

Excepto para el GP3 (que ha identificado el modelo idóneo en el 100% de los casos) los porcentajes de acierto de GP1 y GP2 disminuyen. La disminución más acusada se produce en el GP1, esto es, el grupo de los desarrolladores con experiencia, mientras que GP2 sufre una leve caída de dos puntos. Aunque el GP1 ha obtenido un índice de acierto del 80% en los siete casos, al restringir el análisis a los casos puros, el porcentaje desciende al 67%, esto es, 33 puntos menos que el GP3 y 14 puntos menos que GP2. Un 67% de acierto en 3 casos significa que, en términos medios, GP1 no acierta dos de cada tres casos.

Esta disminución se debe a que los casos de prueba puros poseen únicamente un modelo adecuado, por lo que la probabilidad de acierto con una selección aleatoria disminuye. Esto es, la

probabilidad de acertar un caso puro es de 1/3, mientras que, en líneas generales, cuando es posible más de una solución, la probabilidad de acertar aleatoriamente es mayor o igual a 2/3 (esto es, el número de soluciones posibles dividido por 3 –los distintos tipos de modelos-). Dicho de otro modo: Identificar un modelo adecuado en un problema razonablemente complejo es sencillo, dado que es muy probable que dicho problema admita modelos de distintos tipos. Si el problema es lo suficientemente grande, además, cualquier modelo será lo suficientemente complejo como para poder ser considerado idóneo (en ausencia de los restantes modelos) sin demasiados problemas. No obstante, esta estrategia se demuestra fallida cuando el problema es lo suficientemente pequeño como para que el efecto producido por el tamaño disminuya hasta permitir apreciar claramente qué modelos son, o no, adecuados.

		CP 1	CP 2	CP 3	Índice de aciertos	
		P	O	E	Total	%
GP1	Especialista 1	O	O	O	1	33%
	Especialista 2	O	O	O	1	33%
	Especialista 3	F	O	E	2	67%
	Especialista 4	F	O	E	3	100%
	Especialista 5	F	O	F	2	67%
	Acierto	60%	100%	40%	2	67%
GP2	Alumno 1	F	O	E	3	100%
	Alumno 2	F	O	E	3	100%
	Alumno 3	F	O	O	2	67%
	Alumno 4	E	O	E	2	67%
	Alumno 5	O	O	O	1	33%
	Alumno 6	F	O	E	3	100%
	Alumno 7	F	O	E	3	100%
	Alumno 8	F	O	E	3	100%
	Alumno 9	O	O	F	1	33%
	Alumno 10	F	O	E	3	100%
	Alumno 11	F	O	E	3	100%
	Alumno 12	F	O	E	3	100%
	Alumno 13	O	F	O	0	0%
	Alumno 14	F	O	E	3	100%
	Alumno 15	F	O	E	3	100%
	Alumno 16	F	F	E	2	67%
	Alumno 17	F	O	E	3	100%
	Alumno 18	F	O	E	3	100%
Acierto	78%	89%	78%	2,4	81%	
GP3	Alumnos	F	O	E	3	100%
	Acierto	100%	100%	100%	3,0	100%

Tabla 7.4. Resultados de los GP1, GP2 y GP3 para los casos puros

Es de suponer que esta estrategia es utilizada por muchos profesionales en procesos de desarrollo reales, habida cuenta de que dos de los cinco sujetos del GP1 (concretamente, los sujetos 1 y 2), han

seleccionado el mismo modelo en todos los casos. Ello puede ser representativo de una carencia ya señalada en el Capítulo 1, Introducción: que los profesionales no se basan en criterios objetivos a la hora de seleccionar los métodos y técnicas de desarrollo para un problema determinado, sino que utilizan, o bien aquella a la que están más acostumbrados, o bien aquella que está de moda, sin analizar la adecuación de dichos métodos al problema en cuestión.

7.3.3. ESTUDIO DEL MODELO IDÓNEO EN LOS CASOS MIXTOS

Se han denominado casos mixtos a aquellos casos de prueba que pertenecen a dos o más tipos, esto es, FO, FE, OE y FOE. Al analizar separadamente los casos mixtos, se obtienen los resultados mostrados en la tabla 7.5.

Los resultados obtenidos por el GP1 y GP2 son del 90 y 85% de acierto, respectivamente. Esto es, la mayor parte de los sujetos identifica el modelo idóneo en los 4 casos.

¿Qué ha cambiado respecto a las casos puros, para que haya aumentado en tal medida el índice de acierto (sobre todo en el caso de GP1)? La respuesta es el criterio de acierto que se la utilizado para evaluar los resultados. Como ya se ha indicado anteriormente, los casos de prueba 4-7 son más complejos que los casos de prueba 1 a 3, y pertenecen como mínimo a dos tipos (FO, FE, OE y FOE). El criterio de acierto utilizado ha sido considerar correcta la selección cuando los sujetos identifican correctamente *al menos* uno de los modelos adecuados. Ello implica que la probabilidad de acertar, aleatoriamente, algún modelo idóneo para cada caso es de 2/3 (casos FO, FE, OE) y de 1 para FOE. Esto es, una persona que seleccionase siempre el mismo modelo acertaría, de media, tres de los cuatro casos, que es precisamente lo que ocurre en los casos de prueba 4 a 7.

Como puede observarse en la tabla 7.5, cuatro de los cinco sujetos del GP1 (esto es, un 80%) selecciona el mismo modelo en, al menos, tres de los cuatro casos (75% de los casos). En el GP2, la situación no es tan acusada, pero aún así es significativa, ya que siete de los dieciocho sujetos (esto es, un 39%) eligen, en tres de los cuatro casos (75% de los casos), el mismo modelo.

Puede afirmarse que, por lo tanto, el porcentaje de acierto que el GP1 y GP2 ha tenido a la hora de identificar el modelo idóneo en los casos mixtos se debe, en gran parte, al hecho de que una selección aleatoria (o constante, como es el caso) parece tan efectiva como una selección informada. Como ya se ha indicado anteriormente, si el problema es lo suficientemente grande, cualquier modelo será lo suficientemente complejo como para poder ser considerado parcialmente idóneo, independientemente de que dicho modelo sea totalmente idóneo (esto es, cuando puede existir otro modelo claramente mejor, pero no se pueda afirmar su existencia debido a que sólo se ha confeccionado un único modelo conceptual para el problema bajo estudio).

		CP 4	CP 5	CP 6	CP 7	Índice de aciertos	
		PO	PE	OE	POE	Total	%
GP1	Especialista 1	O	O	O	O	3	75%
	Especialista 2	F	O	O	O	3	75%
	Especialista 3	O	F	E	E	4	100%
	Especialista 4	F	F	O	F	4	100%
	Especialista 5	F	F	O	F	4	100%
	Acierto	100%	60%	100%	100%	3,6	90%
GP2	Alumno 1	F	E	E	E	4	100%
	Alumno 2	O	E	E	E	4	100%
	Alumno 3	O	O	O	E	3	75%
	Alumno 4	O	E	E	F	4	100%
	Alumno 5	O	O	O	E	3	75%
	Alumno 6	F	E	F	O	3	75%
	Alumno 7	O	E	F	E	3	75%
	Alumno 8	O	E	E	F	4	100%
	Alumno 9	O	F	E	F	4	100%
	Alumno 10	O	O	O	E	3	75%
	Alumno 11	---	O	O	E	2	50%
	Alumno 12	F	O	E	E	3	75%
	Alumno 13	O	O	E	E	3	75%
	Alumno 14	O	E	E	E	4	100%
	Alumno 15	F	E	F	E	3	75%
	Alumno 16	F	O	E	O	3	75%
	Alumno 17	O	E	E	E	4	100%
	Alumno 18	O	E	E	O	4	100%
Acierto	94%	61%	83%	100%	3,4	85%	
GP3	Alumnos	O	F	O	E	4	100%
	Acierto	100%	100%	100%	100%	4,0	100%

Tabla 7.5. Resultados de los GP1, GP2 y GP3 para los casos mixtos

Resultan muy diferentes los resultados de los casos mixtos cuando se aplican criterios más restrictivos para evaluar el acierto. Por ejemplo, y dado que los casos 4-7 poseen más de un modelo adecuado, podría considerarse como acierto identificar no uno, sino *todos* los modelos adecuados para cada caso. La tabla 7.6 muestra los resultados obtenidos por el GP2, único grupo en el que se estimó posible realizar este tipo de estudio (dada la reducida habitualidad de los sujetos con una determinada aproximación de desarrollo, hecho que no se produce en GP1), utilizando este nuevo criterio.

		CP 4	CP 5	CP 6	CP 7	Índice de aciertos	
		PO	PE	OE	POE	Total	%
GP2	Alumno 1	PO	PE	PE	PE	2	50%
	Alumno 2	O	PE	E	POE	2	50%
	Alumno 3	PO	OE	POE	POE	2	50%
	Alumno 4	PO	PE	E	PO	2	50%
	Alumno 5	O	PO	OE	OE	1	25%
	Alumno 6	PO	E	POE	PO	1	25%
	Alumno 7	O	OE	PE	PE	0	0%
	Alumno 8	OE	OE	OE	PE	1	25%
	Alumno 9	PO	PO	PE	POE	2	50%
	Alumno 10	PO	PO	O	PE	1	25%
	Alumno 11	---	O	OE	PE	1	25%
	Alumno 12	POE	O	PE	PE	0	0%
	Alumno 13	POE	POE	PE	PE	0	0%
	Alumno 14	O	E	E	PE	0	0%
	Alumno 15	P	E	PO	E	0	0%
	Alumno 16	P	OE	OE	O	1	25%
	Alumno 17	OE	E	E	E	0	0%
	Alumno 18	OE	E	E	OE	0	0%
	Acierto		35%	17%	22%	17%	0,9

Tabla 7.6. Resultados de GP2 considerando como acierto la identificación de *todos* los modelos adecuados

Como puede observarse en la tabla 7.6, el porcentaje de acierto, aplicando el nuevo criterio, es muy reducido, pasando de un 85% a un 22%. Esto es, para casos de tipo mixto, resulta relativamente fácil para los sujetos identificar un modelo razonablemente adaptado al caso pero, sin embargo, es considerablemente más difícil identificar el espectro completo de modelos adecuados. MAON, por el contrario, permite identificar el modelo idóneo independientemente del ingeniero que lo utiliza, sin atender a consideraciones subjetivas como la bondad relativa de un modelo conceptual en función del tamaño o complejidad del problema. Adicionalmente, MAON es más efectivo que los sujetos a la hora de identificar el modelo más adecuado, ya que el índice de acierto de los sujetos que utilizan MAON (esto es, el GP3) es del 100%, mayor, por lo tanto, que los índices de acierto de GP1 y GP2, independientemente del criterio de acierto utilizado.

7.3.4. TENDENCIOSIDAD EN LA SELECCIÓN DEL MODELO MÁS ADECUADO

7.3.4.1. Estudio de las Preferencias de los Sujetos

La tabla 7.7 muestra el número de veces que cada sujeto experimental ha seleccionado un mismo tipo de modelos, expresado como un porcentaje sobre el total. Los datos de la tabla 7.6 evidencian los siguientes hechos:

		Porcentaje de modelos seleccionados por tipos		
		F	O	E
GP1	Especialista 1	0%	100%	0%
	Especialista 2	14%	86%	0%
	Especialista 3	29%	29%	43%
	Especialista 4	57%	29%	14%
	Especialista 5	71%	29%	0%
	Acierto	40%	57%	3%
	GP2	Alumno 1	29%	14%
Alumno 2		14%	29%	57%
Alumno 3		14%	71%	14%
Alumno 4		14%	29%	57%
Alumno 5		0%	86%	14%
Alumno 6		43%	29%	29%
Alumno 7		29%	29%	43%
Alumno 8		29%	29%	43%
Alumno 9		43%	43%	14%
Alumno 10		14%	57%	29%
Alumno 11		14%	43%	29%
Alumno 12		29%	29%	43%
Alumno 13		14%	57%	29%
Alumno 14		14%	29%	57%
Alumno 15		43%	14%	43%
Alumno 16		43%	29%	29%
Alumno 17		14%	29%	57%
Alumno 18		14%	43%	43%
Acierto		23%	38%	38%
GP3	Alumnos	29%	43%	29%
	Acierto	29%	43%	29%

Tabla 7.7. Número de veces que cada sujeto ha seleccionado un mismo tipo de modelo

- Los sujetos experimentales del GP1 seleccionan prácticamente siempre modelos de tipo funcional o modelos orientados a objetos. Este hecho, una vez revisados los índices de acierto en las secciones 7.3.2 y 7.3.3, y las conclusiones que de ellos pueden extraerse, era previsible, ya que los desarrolladores están seleccionando modelos propios de aquellas aproximaciones que les son más familiares. La selección de modelos de tipo funcional y orientado a objetos

muestra únicamente, que los desarrolladores que forman el GP1 poseen principalmente experiencia con las aproximaciones estructurada y orientada a objetos. Asimismo, cuatro de los cinco sujetos pertenecientes a GP1 seleccionan el mismo modelo en al menos cuatro de los siete casos de prueba (57%). También se puede observar que uno de los sujetos selecciona siempre el mismo modelo en 5/7, otro en 6/7 e incluso existe un sujeto que selecciona el mismo modelo en 7/7 ocasiones.

- Los sujetos pertenecientes a GP2 muestra una mayor dispersión en los modelos seleccionados. La media total, de hecho, muestra un claro balanceo entre los tres tipos de modelos. No obstante, nueve de los dieciocho sujetos seleccionan el mismo modelo en al menos cuatro de siete casos de prueba (57%). Al igual que en el GP1, sujetos particulares sobrepasan este índice.
- Los sujetos que utilizan MAON muestran un balanceo entre los modelos identificados como idóneos, no destacando ningún modelo en particular.

Por lo tanto, una gran parte de los sujetos manifiesta una tendencia a seleccionar, en una mayoría de casos, modelos similares para resolver problemas diferentes, independientemente de que dicho problema pueda resolverse utilizando otros modelos.

7.3.4.2. Estudio de la Sensibilidad de MAON Frente a la Tendenciosidad

Los datos obtenidos en la solución de los distintos casos de prueba muestran que la selección realizada por los distintos sujetos es, en gran medida, tendenciosa, tal y como se vislumbra en la sección anterior. Ello es todavía más sorprendente cuando se analiza el desempeño del GP1 (desarrolladores con experiencia), en la medida en que deberían ser dichos sujetos, precisamente, los que mejor rendimiento deberían obtener en la solución de los casos de prueba. Adicionalmente, dicha tendenciosidad está basada en una descripción totalmente ecuánime de los casos de prueba, esto es, utilizando una descripción del dominio realizada por un elicitador absolutamente imparcial, la cual es sesgada por el modelador.

Con la finalidad de comprobar si efectivamente existe tendenciosidad, nos planteamos qué ocurriría en el caso de que un modelador recibiera información proveniente de un educador sesgado hacia una perspectiva particular y que, por lo tanto, destacara unos elementos del problema en detrimento de otros. A priori, dos resultados son posibles: (1) que el modelador fuera en realidad no-tendencioso, esto es, realizara el modelado absorbiendo el sesgo introducido por el educador o (2) que el modelador fuese efectivamente tendencioso, con lo cual el modelo obtenido contendrá principalmente el sesgo del modelador y, en mucha menor medida, el sesgo del educador.

Para realizar tal comprobación, un educador tendencioso ha realizado una modificación de uno de los casos de prueba, concretamente, del caso de prueba 3 (el cual pertenecía al tipo E), sesgando el caso de tal forma que fuese preferible una solución bajo una aproximación basada en funciones. Lo que se ha hecho es, en realidad, contaminar el caso de prueba, introduciendo aspectos funcionales en mayor proporción que los aspectos de estados, convirtiendo el caso de prueba en el tipo FE, aunque el tipo F sea el dominante.

El enunciado transformado es el que sigue:

El cajero automático del banco ACME debe poder realizar tres tipos de operaciones sobre las cuentas de los clientes: reintegros, imposiciones e informes de los saldos. La idea es que el cajero posea un funcionamiento similar a otros cajeros ya existentes.

El cajero debería comenzar a funcionar una vez un cliente introduce una tarjeta de crédito. Después de leer los datos de la tarjeta y comprobar que es la adecuada, se pedirá el Número Identificativo Personal del cliente (el PIN). En el caso de que el PIN sea correcto, el cajero permitirá al cliente que realice cualquiera de las tres operaciones indicadas anteriormente tantas veces como quiera.

El cliente dispondrá de un botón TERMINAR en el panel del cajero. El botón TERMINAR servirá para que el cliente finalice de operar con el cajero. Cuando el cliente pulse el botón TERMINAR, el cajero devolverá la tarjeta al cliente y quedará inactivo hasta la llegada de otro cliente.

Se ha seleccionado este caso de prueba debido a que una gran cantidad de los sujetos del GP2 parecen mostrar una clarísima tendencia a seleccionar modelos orientados a estados, mientras que los miembros del GP1 poseen una tendencia en sentido inverso, esto es, hacia modelos orientados a funciones y modelos orientados a objetos. Por ello, en caso de que se produzca tendenciosidad, es de esperar que los miembros del GP2 seleccionen mayoritariamente modelos orientados a estados, mientras que los miembros de GP1 deberían seleccionar modelos orientados a funciones.

Este caso ha sido entregado a los sujetos miembros de todos los grupos de prueba (GC, GP1, GP2 y GP3), con el objetivo de estudiar cómo sorteaban el problema de la información tendenciosa que poseían como material base. Los resultados de la resolución de este caso de prueba por el GC, GP1, GP2 y GP3, los cuales se muestran en la tabla 7.8, muestran los siguientes hechos:

- Todos los grupos de prueba, con excepción del GP3 (los alumnos que han utilizado MAON), muestran tendenciosidad, aunque de diverso cuño. El GP3, por el contrario, ha realizado un análisis imparcial, reflejando en el modelo obtenido tanto los aspectos orientados a estados como los aspectos funcionales, más numerosos, del problema, obteniendo congruentemente como modelo idóneo el DFD. Los resultados del GP3 se muestran en el anexo AV5 del volumen II de la Tesis Doctoral "Datos de la Validación".
- El GP1 se muestra tendencioso, con el mismo comportamiento que en los casos de prueba 1-7 y en congruencia con las hipótesis planteadas para el presente "caso contaminado". Concretamente, los modelos seleccionados por este grupo han sido los orientados a funciones (60%) y orientados a objetos (40%).
- El GP2 se muestra claramente tendencioso hacia los modelos orientados a estados (78% de los sujetos), con la agravante de que solo un sujeto ha seleccionado como idóneo los modelos orientados a funciones. Ello también está en perfecta coherencia con las hipótesis del presente caso.

- El GC también se ha visto afectado por los aspectos superficiales del problema (recuérdese que el caso de prueba fue inicialmente de tipo E), los cuales han oscurecido los aspectos, más numerosos, de tipo funcional.

GC	Profesores	E
GP1	Especialista 1	O
	Especialista 2	O
	Especialista 3	F
	Especialista 4	F
	Especialista 5	F
GP2	Alumno 1	E
	Alumno 2	E
	Alumno 3	O
	Alumno 4	E
	Alumno 5	O
	Alumno 6	E
	Alumno 7	E
	Alumno 8	E
	Alumno 9	F
	Alumno 10	E
	Alumno 11	E
	Alumno 12	E
	Alumno 13	O
	Alumno 14	E
Alumno 15	E	
Alumno 16	E	
Alumno 17	E	
Alumno 18	E	
GP3	Alumnos	F

Tabla 7.8. Resultados del caso contaminado

El cambio que se ha producido en la selección del GP3 es significativo, ya que demuestra la sensibilidad de MAON ante cambios en el dominio, sensibilidad que no se produce en los sujetos de los GC, GP1 y GP2, los cuales se ven más afectados no por la aparición de nuevos componentes en el dominio, sino por aquellos aspectos a los que cada sujeto da mayor importancia.

De hecho, si se analiza más en profundidad este caso, se pueden apreciar otros aspectos importantes del mismo. El Diagrama de Flujo de Datos obtenido mediante la aplicación de la técnica DMCS, para este caso de prueba, se muestra en la figura 7.2. A simple vista, en el diagrama 7.2 pueden observarse diversas irregularidades (flujos no nombrados, falta de conexión entre procesos, etc.) y, asimismo, varios errores, los cuales no se pueden apreciar fácilmente debido a lo poco estricto de la semántica del Diagrama de Flujo de Datos. Concretamente, es poco habitual que los procesos se comuniquen directamente mediante flujos de datos, como ocurre en el caso de “introducir tarjeta” y “leer PIN”. Igualmente, no está clara la relación entre el proceso “Leer PIN” y “Realizar reintegro”, “Realizar imposición” y “Consultar saldo”.

Lo que ha ocurrido, en realidad, es que el Diagrama de Flujo de Datos es poco adecuado para modelar el presente caso de prueba. El Diagrama 7.2 refleja este hecho en su falta de conexión y en su compleja semántica.

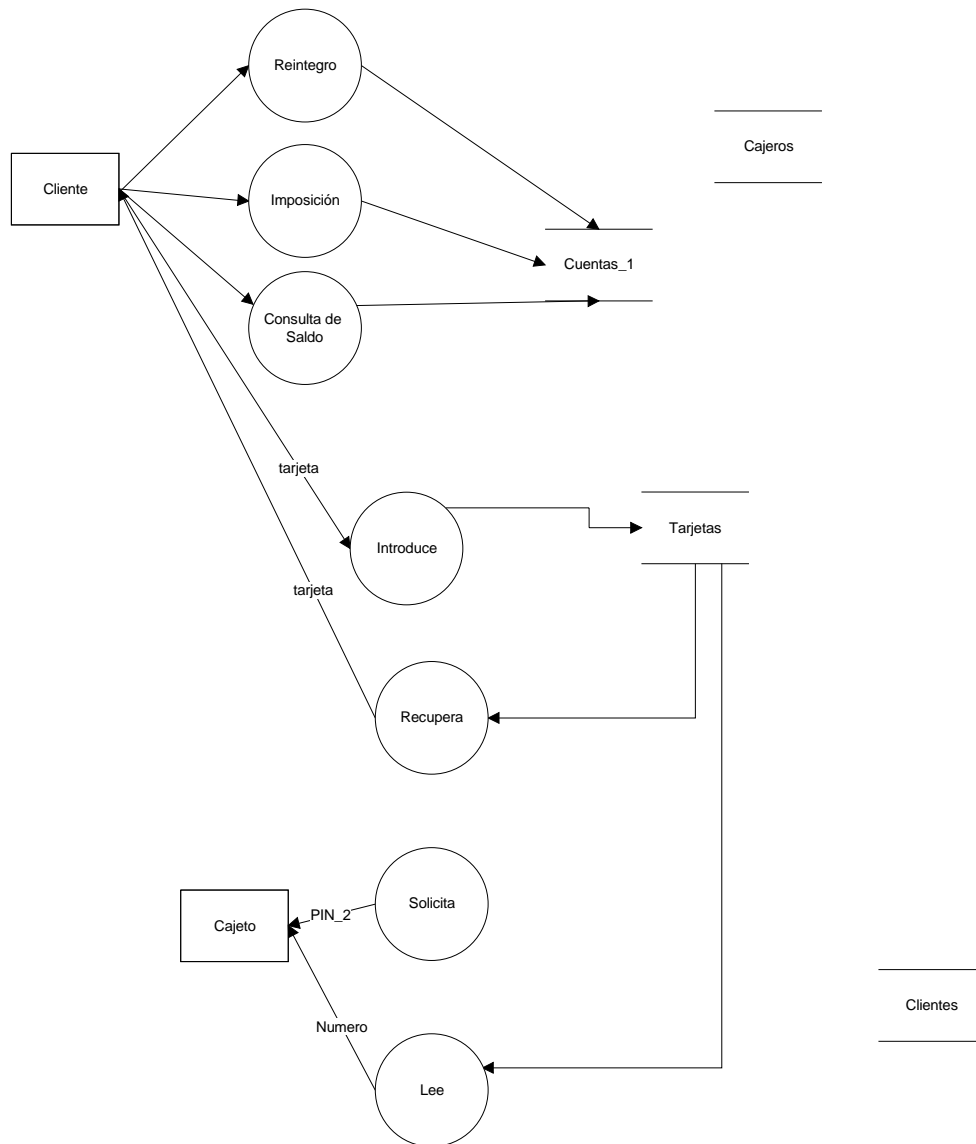


Figura 7.2. Diagrama de Flujo de Datos obtenido mediante la aplicación de la técnica DMCS

Tras la re aplicación de la Técnica IMCI al Modelo Canónico de Requisitos, se ha comprobado que el modelo más adecuado es el Diagrama de Flujo de Datos/Tiempo Real de Ward. Ello es coherente, ya que el Diagrama de Flujo de Datos/Tiempo Real posee todas las capacidades expresivas de un Diagrama de Flujo de Datos, además de la capacidad de expresar control. El aspecto de control, por otra parte, ha sido el único detectado por los grupos GC, GP1 y GP2, lo cual explica su decisión de escoger como modelo idóneo el diagrama de transición de estados, un modelo de tipo E.

La idoneidad del Diagrama de Flujo de Datos/Tiempo Real es del 66%, mucho mayor que la del Diagrama de Flujo de Datos (42%) y la del Diagrama de transición de estados (26%). El modelo obtenido

tras la aplicación de la técnica DMCS para el Diagrama de Flujo de Datos/Tiempo Real se muestra en la figura 7.3. Su parecido con el Diagrama de Flujo de Datos es lógico, ya que las capacidades de representación del Diagrama de Flujo de Datos/Tiempo Real incluyen las del Diagrama de Flujo de Datos.

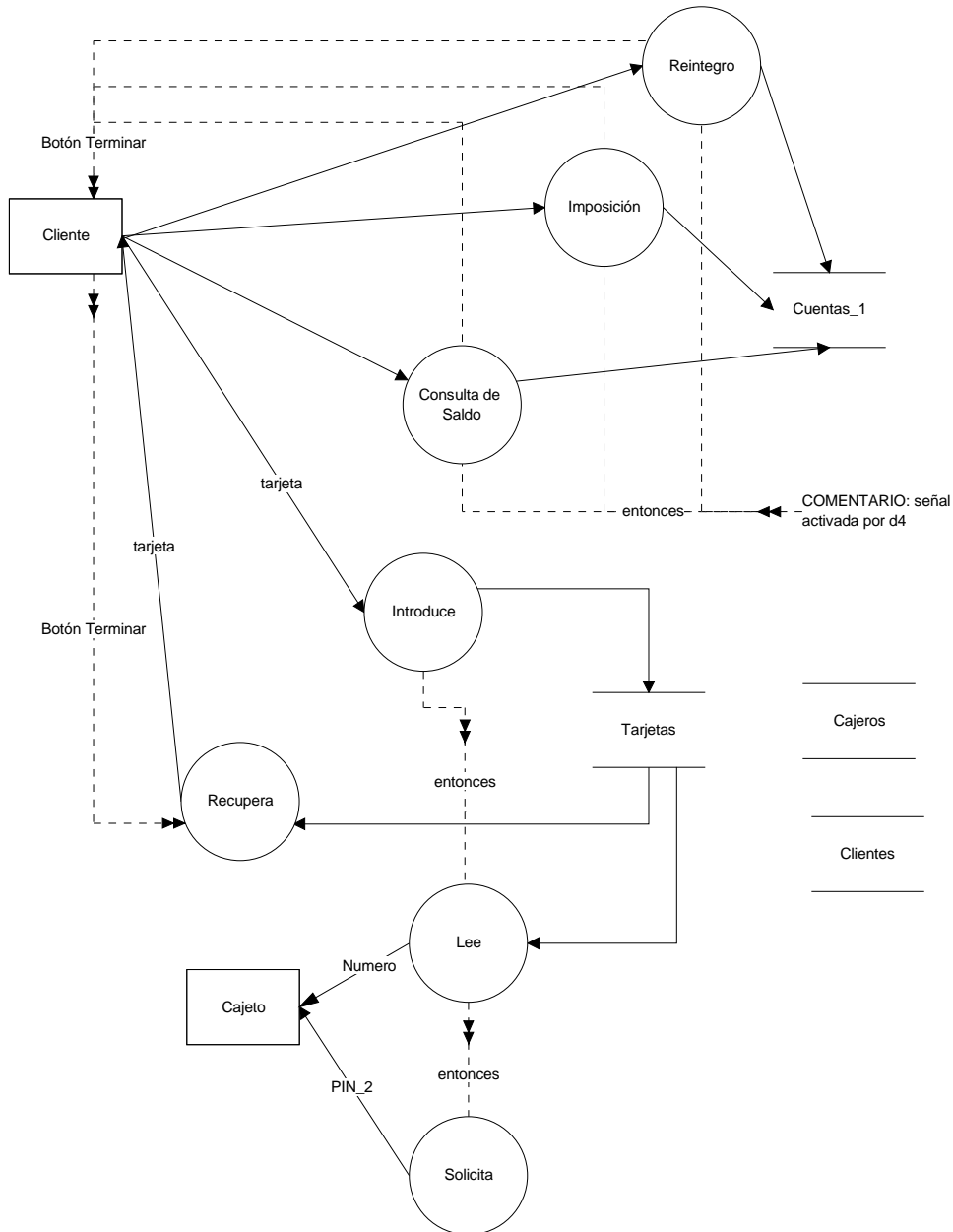


Figura 7.3. Diagrama de Flujo de Datos/Tiempo Real obtenido tras la aplicación de la técnica DMCS para el Diagrama de Flujo de Datos/Tiempo Real

El Diagrama de Flujo de Datos/Tiempo Real obtenido mediante la Técnica DMCS es muy similar a un Diagrama de Flujo de Datos/Tiempo Real realizado *ex profeso* para este caso de prueba, el cual se muestra en el anexo AV5 del volumen II de la Tesis Doctoral “Datos de la validación”. No obstante, subyacen varias diferencias de escasa importancia, las cuales se refieren, primordialmente, a aspectos

formales (refinamiento de las entradas y salidas de los procesos, refinamiento de los flujos de control, etc.), los cuales pueden ser difícilmente considerados por un método orientado a la necesidad como MAON.

Para concluir, se puede afirmar que MAON posee mucho mejores capacidades predictivas que los sujetos. Los sujetos son muy sensibles a los aspectos superficiales (forma) de los problemas y a su propia habitualidad a la hora de desarrollar software. MAON, sin embargo, permite considerar los aspectos profundos del problema, esto es, los elementos y asociaciones que mantienen dichos elementos en el dominio del problema, de tal forma que es capaz de identificar con mayor eficacia el modelo idóneo sin estar sometido a consideraciones subjetivas de los sujetos experimentales.

7.4. VALIDACIÓN DE SH2: DERIVACIÓN DE LOS MODELOS CONCEPTUALES DE LAS APROXIMACIONES TRADICIONALES

Los modelos conceptuales propios de las aproximaciones estructurada, orientada a objetos y tiempo real se obtienen tras la aplicación de la técnica DMCS al Modelo Canónico de Requisitos. Los resultados de esta aplicación, para los casos de prueba 1 a 7, se muestran en el anexo AV6 del Volumen II de la Tesis Doctoral "Datos de la Validación".

Los resultados obtenidos permiten realizar las siguientes afirmaciones:

- MAON, mediante la técnica DMCS, permite generar cualquier tipo de modelo conceptual, aunque habitualmente se genere sólo aquel de mayor adecuación.
- Los modelos generados mediante la técnica DMCS poseen una capacidad de representación similar a los modelos confeccionados independientemente por el GC y por el GP1.
- Los modelos generados mediante la técnica DMCS poseen, en líneas generales, una mayor capacidad de representación que los modelos generados por el GP2. Los sujetos pertenecientes a GP2 (alumnos de último curso de licenciatura o alumnos de postgrado) tienden a realizar modelos parciales, que únicamente recogen parte de la información que, a priori, podría representarse. Ello se debe, probablemente, a que los aspectos sintácticos de los modelos conceptuales hacen difícil, para ellos, la expresión de todos los elementos del problema, hecho que no ocurre con MAON.

Por todo ello puede afirmarse que MAON permite generar adecuadamente los modelos conceptuales propios de las aproximaciones de desarrollo estructurada, orientada a objetos y de tiempo real, por lo que **se valida, por lo tanto, la sub-hipótesis SH2.**

7.5. VALIDACIÓN DE SH3: VERSATILIDAD DE MAON

SH3 establece que MAON podrá ser utilizado en aquellas circunstancias en las que sean aplicables las aproximaciones de desarrollo Estructuradas, Orientadas a Objetos y de Tiempo Real. Dado que las aproximaciones de desarrollo se caracterizan por los modelos que utilizan, tal y como se ha justificado en el

capítulo 2 Estado de la Cuestión, la sub-hipótesis SH3 equivale a afirmar que MAON es capaz de generar los modelos propios de las aproximaciones de desarrollo Estructurada, Orientada a Objetos y de Tiempo Real en las mismas circunstancias en que dichas aproximaciones podrían ser utilizadas. Es decir, para validar la sub-hipótesis SH3, es necesario:

- a. que MAON genere los modelos propios de cada aproximación de desarrollo y
- b. que MAON sea adecuado en las mismas situaciones en que una determinada aproximación de desarrollo sería utilizable.

En las secciones 7.2 a 7.4, se ha expuesto la validación de las sub-hipótesis SH1 y SH2. La validación de la sub-hipótesis SH1 ha permitido comprobar que MAON permite determinar el modelo conceptual más adecuado (modelo idóneo) para un determinado problema. Ello es equivalente, dada la correspondencia existente entre modelos conceptuales y aproximaciones de desarrollo, a identificar la aproximación más adecuada para un problema determinado.

La validación de la sub-hipótesis SH2 ha permitido, asimismo, comprobar que MAON posee la capacidad suficiente para generar, a partir del Modelo Canónico de Requisitos, los modelos conceptuales propios de las distintas aproximaciones de desarrollo.

La validación de la sub-hipótesis SH2 equivale a validar (a), esto es, que MAON genera los modelos propios de cada aproximación de desarrollo. Asimismo, la validación de la sub-hipótesis SH1 equivale a validar (b), es decir, que MAON es adecuado en las mismas situaciones en que una determinada aproximación de desarrollo sería utilizable. (a) y (b), conjuntamente, permiten **validar por lo tanto la sub-hipótesis SH3**.

8. Caso Práctico

El presente capítulo expone la resolución de un caso práctico, de alta complejidad, en el que se ha utilizado MAON. El dominio que se ha estudiado es el de una empresa de publicidad, de la cual, por motivos de confidencialidad, no se suministrarán más detalles. Para hacer referencia a dicha empresa, se utilizarán únicamente las siglas MP.

En lo que respecta al caso práctico en sí, éste contempla todas las actividades productivas de MP, desde la petición inicial del cliente hasta la puesta en marcha de la campaña publicitaria. El volumen de la información recogida, así como las interrelaciones que ésta presenta, justifican el calificativo de “alta complejidad” que se ha atribuido a este caso práctico. No en vano, el MCG contiene más de 100 proposiciones.

Además de la utilización de MAON, en el presente caso práctico también ha sido empleada para su resolución una aproximación de desarrollo tradicional. Concretamente, dicha aproximación podría calificarse de “Bases de Datos” o, en los términos que se han utilizado en el presente trabajo de Tesis, de Objetos. La resolución independiente del mismo caso práctico se realizó con un propósito doble: (1) llevar a cabo el proyecto de desarrollo en el caso de que la aplicación de MAON, que se encontraba en fase de prueba en aquellos momentos, resultase fallida y (2) proporcionar material suficiente para poder llevar a cabo una comparación entre los resultados obtenidos mediante ambos métodos.

En las secciones siguientes se presentan los resultados obtenidos mediante la aplicación de MAON. Dicha presentación se realizará por fases, en los diversos epígrafes de la sección 8.1. Los resultados que se expondrán serán el Mapa de Conceptos, el Diccionario de Descripción y la interpretación del Diccionario de Descripción.

La sección 8.2 se ocupará de la aplicación de la Técnica IMCI, la cual, como se expondrá en dicha sección, ha predecido como Modelo Idóneo el Diagrama de Clases. Dicha elección es perfectamente

compatible con la elección, independiente, de la aproximación de desarrollo llevada a cabo por los ingenieros de MP, lo cual es un nuevo argumento a favor de las capacidades predictivas de MAON.

La sección 8.3 muestra todos los fragmentos obtenidos mediante la aplicación de la Técnica DMCS. Los distintos fragmentos son combinados posteriormente y el modelo final se presenta en la sección 8.4.

8.1. LOS ENUNCIADOS

Los epígrafes siguientes, del 8.1.1 al 8.1.9, describen los resultados de la aplicación de MAON a las partes constituyentes del proceso de negocio de MP. La presentación por partes se ha realizado únicamente con el propósito de facilitar la legibilidad del caso pero, no obstante, la unión de todas estas partes representa el proceso de negocio completo de MP.

8.1.1. BRIEFING

8.1.1.1. Descripción

El BRIEFING es el primer paso del proceso de negocio de MP. El BRIEFING es una reunión de los clientes con el equipo comercial de MP, con el objetivo de presentar las peticiones de campañas publicitarias. Cada petición de campaña puede referirse a distintos medios publicitarios (por ejemplo, distintas cadenas de televisión), y, asimismo, a distintos productos a publicitar.

Para cada medio, se determina la inversión total a realizar, el periodo de la campaña y el target de planificación deseado (por ejemplo, amas de casa, varones de 20-50 años, etc.).

8.1.1.2. Mapa de conceptos

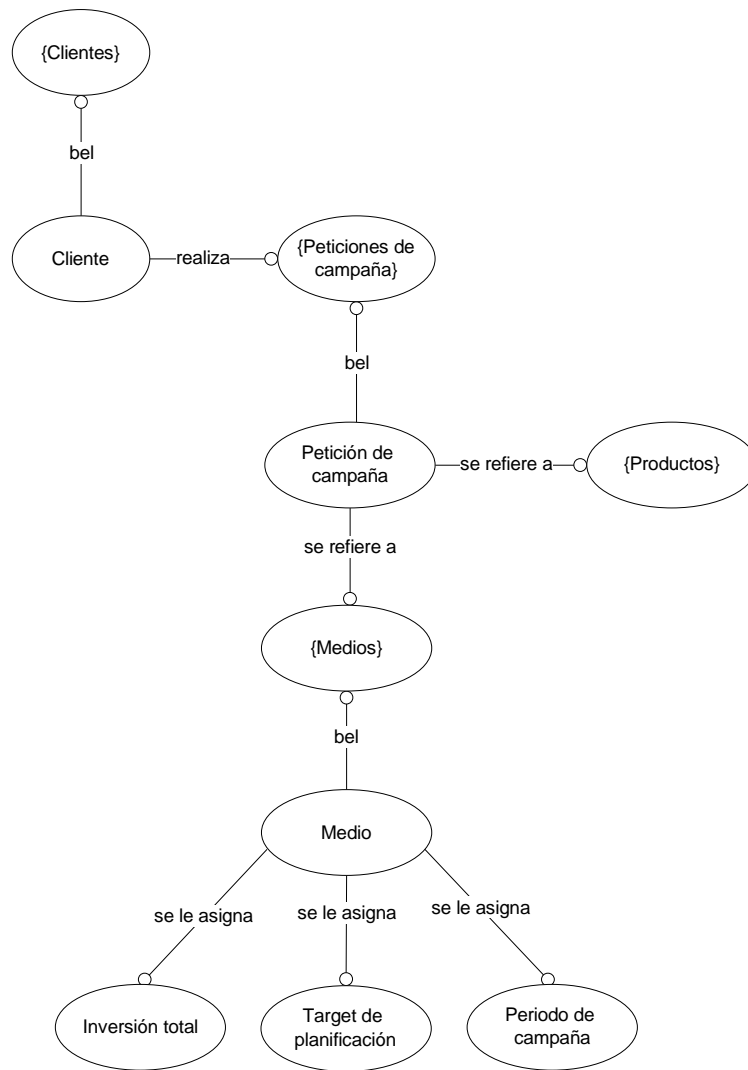


Figura 8.1. Mapa de Conceptos correspondiente al BRIEFING

8.1.1.3. Diccionario de descripción

Declaraciones			
Conjuntos:	Clientes		
	Petición de campaña		
	Productos		
	Medios_1		
Subconjuntos:		subs	
Individuos:	Cliente	bel	Clientes
	Petición de campaña	bel	Petición de campaña
	Medio_1	bel	Medios_1
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Cliente	Realiza	Petición de campaña
2	Petición de campaña	Se refiere a	Productos
3	Petición de campaña	Se refiere a	Medios_1
4	Medio_1	Se le asigna	Inversión total
5	Medio_1	Se le asigna	Target de planificación_1
6	Medio_1	Se le asigna	Periodo de campaña

Tabla 8. 1. Diccionario de Descripción del BRIEFING

8.1.1.4. Interpretación parcial

Declaraciones			
Conjuntos:	Entity[repl]: Clientes		
	Entity[repl]: Petición de campaña		
	Entity[repl]: Productos		
	Entity[repl]: Medios		
Subconjuntos:		subs	
Individuos:	Entity[notrepl]: Cliente	Bel: bel	Entity[repl]: Clientes
	Entity[notrepl]: Petición de campaña	Bel: bel	Entity[repl]: Petición de campaña
	Entity[notrepl]: Medio_1	Bel: bel	Entity[repl]: Medios
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Entity[notrepl]: Cliente	Rel: Realiza	Entity[repl]: Petición de campaña
2	Entity[notrepl]: Petición de campaña	Rel: Se refiere a	Entity[repl]: Productos
3	Entity[notrepl]: Petición de campaña	Rel: Se refiere a	Entity[repl]: Medios
4	Entity[notrepl]: Medio_1	-Pof: Se le asigna	Statespace: Inversión total
5	Entity[notrepl]: Medio_1	-Pof: Se le asigna	Statespace: Target de planificación_1
6	Entity[notrepl]: Medio_1	-Pof: Se le asigna	Statespace: Periodo de campaña

Tabla 8.2. Interpretación del Diccionario de Descripción del BRIEFING

8.1.2. PLANIFICACIÓN

8.1.2.1. Descripción

La PLANIFICACIÓN es el proceso de MP en el cual se confeccionan las campañas publicitarias. Para cada petición de campaña, se crea una o más planificaciones. Cada planificación está asociada a un conjunto de pases tipo, un conjunto de planes y un conjunto de alternativas.

Una planificación siempre posee un periodo de vigencia, el cual debe estar comprendido en el periodo de vigencia de la petición de campaña. Asimismo, la planificación posee un target de planificación, el cual puede coincidir, o no, con el target de planificación de la petición de campaña, y una franja de planificación, la cual forma parte de un conjunto predefinido de franjas de planificación.

8.1.2.2. Mapa de conceptos

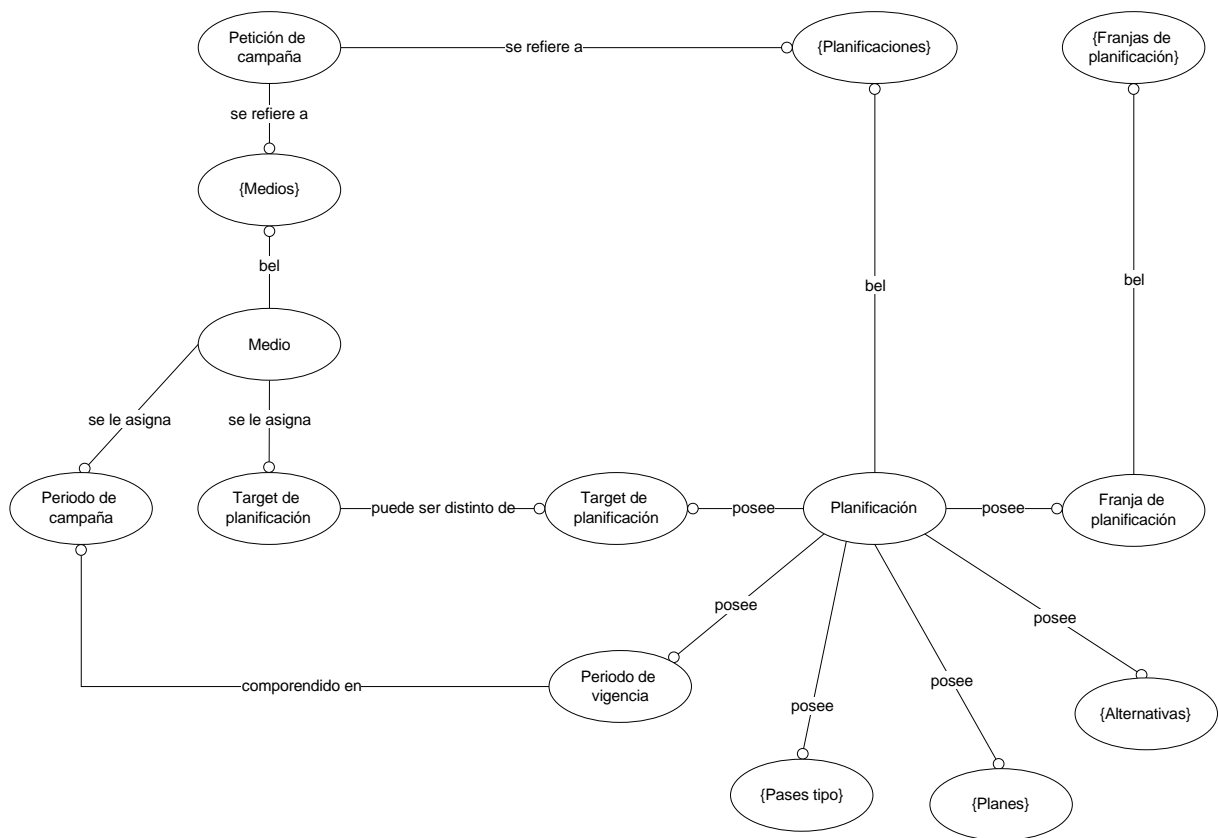


Figura 8.2. Mapa de Conceptos correspondiente a la PLANIFICACIÓN

8.1.2.3. Diccionario de descripción

Declaraciones			
Conjuntos:	Planificaciones		
	Franjas de planificación		
	Alternativas_1		
	Pases tipo_1		
	Planes_1		
Subconjuntos:		subs	
Individuos:	Planificación	bel	Planificaciones
	Franja de planificación	bel	Franjas de planificación
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Planificación	Posee	Franja de planificación
2	Planificación	Posee	Periodo de vigencia
3	Planificación	Posee	Target de planificación_2
4	Planificación	Posee	Pases tipo_1
5	Planificación	Posee	Planes_1
6	Planificación	Posee	Alternativas_1
7	Periodo de vigencia	Comprendido en	Periodo de campaña

Tabla 8.3. Diccionario de Descripción de la PLANIFICACIÓN

8.1.2.4. Interpretación parcial

Declaraciones			
Conjuntos:	Entity[repl]: Planificaciones		
	Entity[repl]: Franjas de planificación		
	Entity[repl]: Alternativas_1		
	Entity[repl]: Pases tipo_1		
	Entity[repl]: Planes_1		
Subconjuntos:		subs	
Individuos:	Entity[notrepl]: Planificación	Bel: bel	Entity[repl]: Planificaciones
	Entity[notrepl]: Franja de planificación	Bel: bel	Entity[repl]: Franjas de planificación
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Entity[notrepl]: Planificación	-Pof: Posee	Entity[notrepl]: Franja de planificación
2	Entity[notrepl]: Planificación	-Pof: Posee	Statespace: Periodo de vigencia
3	Entity[notrepl]: Planificación	-Pof: Posee	Statespace: Target de planificación_2
4	Entity[notrepl]: Planificación	-Rel: Posee	Entity[repl]: Pases tipo_1
5	Entity[notrepl]: Planificación	-Rel: Posee	Entity[repl]: Planes_1
6	Entity[notrepl]: Planificación	-Rel: Posee	Entity[repl]: Alternativas_1
7	Statespace: Periodo de vigencia	Constraint: Comprendido en	Statespace: Periodo de campaña

Tabla 8.4. Interpretación del Diccionario de Descripción de la PLANIFICACIÓN

8.1.3. PASES TIPO

8.1.3.1. Descripción

Un PASE TIPO es cada posible spot que puede hacerse de un producto. Esto es, un pase tipo asocia un determinado producto con un conjunto de materiales publicitarios (por ejemplo, un video de un anuncio, una cuña de radio) y una duración de spot determinada.

Es necesario hacer notar que la duración del material publicitario es parcialmente independiente de la duración del PASE TIPO. Esto es, el material podrá tener una duración mayor que la duración del PASE TIPO.

Dado que los materiales publicitarios no son intercambiables entre medios (por ejemplo, un video no puede emitirse en radio), cada material publicitario está indisolublemente unido a un medio. Asimismo, cada material publicitario no es intercambiable entre productos (por ejemplo, la publicidad del yogurt de fresa X no puede utilizarse en una campaña del yogurt de fresa Y), cada material publicitario está igualmente asociado a un producto determinado y, por ende, a un cliente concreto.

Por último, un material publicitario posee un periodo de vigencia y una procedencia, la cual indica el estudio de producción donde se ha creado dicho material.

8.1.3.2. Mapa de conceptos

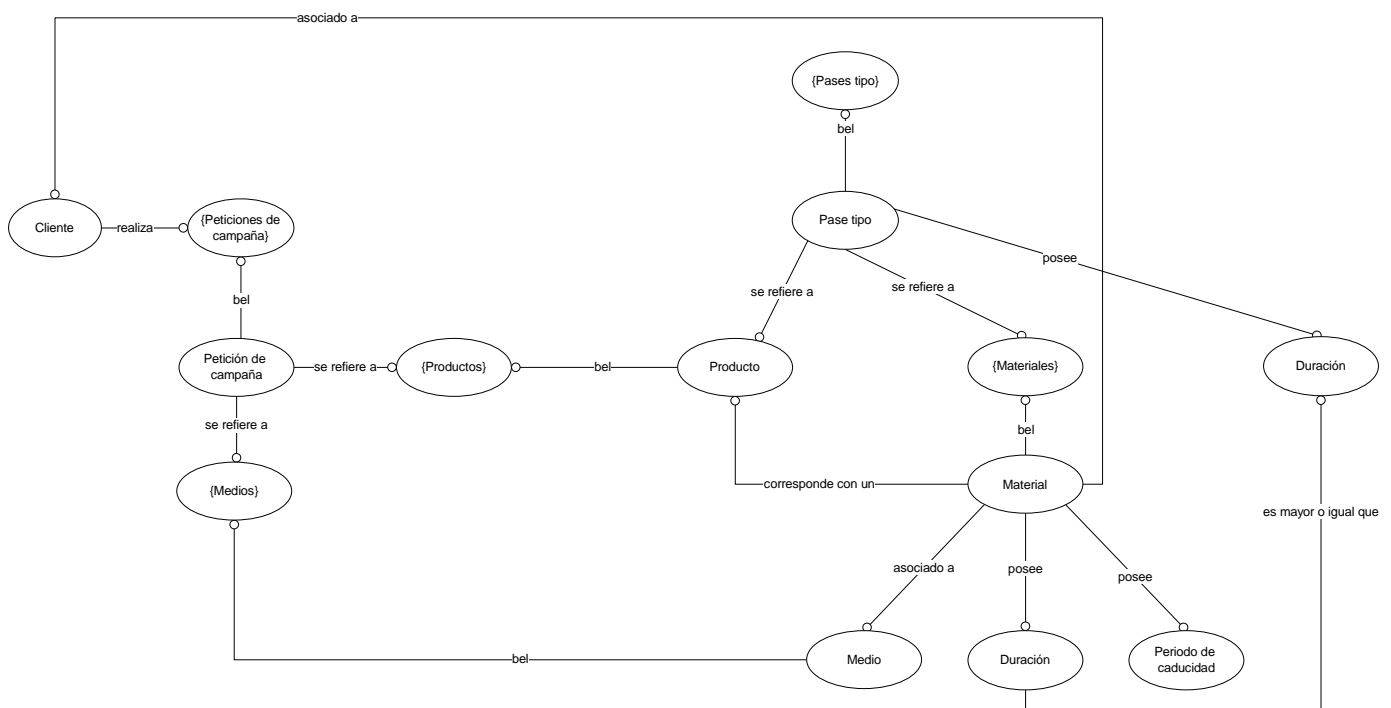


Figura 8.3. Mapa de Conceptos correspondiente a los PASES TIPO

8.1.3.3. Diccionario de descripción

Declaraciones			
Conjuntos:	Pases tipo_1		
	Materiales		
	Productos		
	Medios_1		
Subconjuntos:		subs	
Individuos:	Pase tipo_1	bel	Pases tipo_1
	Material	bel	Materiales
	Producto	bel	Productos
	Medio_2	bel	Medios_1
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Material	Asociado a	Cliente
2	Pase tipo_1	Se refiere a	Producto
3	Pase tipo_1	Se refiere a	Materiales
4	Pase tipo_1	Posee	Duración_1
5	Material	Corresponde con un	Producto
6	Material	Posee	Medio_2
7	Material	Posee	Duración_2
8	Material	Posee	Periodo de caducidad
9	Duración_2	Es mayor o igual que	Duración_1

Tabla 8.5. Diccionario de Descripción de los PASES TIPO

8.1.3.4. Interpretación parcial

Declaraciones			
Conjuntos:	Entity[repl]: Pases tipo_1		
	Entity[repl]: Materiales		
	Entity[repl]: Productos		
	Entity[repl]: Medios		
Subconjuntos:		subs	
Individuos:	Entity[notrepl]: Pase tipo_1	Bel: bel	Entity[repl]: Pases tipo_1
	Entity[notrepl]: Material	Bel: bel	Entity[repl]: Materiales
	Entity[notrepl]: Producto	Bel: bel	Entity[repl]: Productos
	Entity[notrepl]: Medio_2	Bel: bel	Entity[repl]: Medios
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Entity[notrepl]: Material	-Pof: Asociado a	Entity[notrepl]: Cliente
2	Entity[notrepl]: Pase tipo_1	Rel: Se refiere a	Entity[notrepl]: Producto
3	Entity[notrepl]: Pase tipo_1	Rel: Se refiere a	Entity[repl]: Materiales
4	Entity[notrepl]: Pase tipo_1	-Pof: Posee	Statespace: Duración_1
5	Entity[notrepl]: Material	-Pof: Corresponde con un	Entity[notrepl]: Producto
6	Entity[notrepl]: Material	-Pof: Posee	Entity[notrepl]: Medio_2
7	Entity[notrepl]: Material	-Pof: Posee	Statespace: Duración_2
8	Entity[notrepl]: Material	-Pof: Posee	Statespace: Periodo de caducidad
9	Statespace: Duración_2	Constraint: Es mayor o igual que	Statespace: Duración_1

Tabla 8.6. Interpretación del Diccionario de Descripción de los PASES TIPO

8.1.4. PLANES

8.1.4.1. Descripción

Un PLAN es la asociación de PASES TIPO a slots. La asociación de un PASE TIPO a un slot se le denomina inserción. Un slot es el intervalo de 15 minutos en el que se divide cada día de la programación de una cadena de televisión. En la terminología de MP, a cada cadena de televisión se le denomina soporte. Las inserciones pueden tener una posición. Una inserción con posición es un pase tipo, situado en un determinado slot y en una determinada posición dentro del slot. Esta posición puede tomar los valores de 1 a 5, y los tipos PB1, PB2 y PB3.

Cada PLAN está asociado a una franja de compra, la cual puede ser distinta a la franja de planificación de la PLANIFICACIÓN. Igualmente, el PLAN está asociado a un target de compra, el cual puede ser distinto del target de planificación.

El propósito del PLAN es dual: Por una parte, el PLAN permite prever la audiencia que tendrán las distintas inserciones. Para ello, cada PLAN se asocia a un MODELO DE PREVISIÓN, el cual puede ser Mitra o Sofres (preferentemente este último). El planificador deberá seleccionar que periodo anterior de audiencia utilizará para estimar la audiencia futura de cada slot mediante la selección de un periodo de predicción de audiencia.

La audiencia se puede calcular de distintas formas:

1. Por slot.
2. Por audiencia media de todos los slots de un programa determinado.
3. Por punto horario.

Por otro lado, el PLAN permite prever el coste de las inserciones. Para ello, cada soporte tiene asociada una TARIFA BASE. No obstante, la TARIFA puede ser alterada mediante una NEGOCIACIÓN o mediante una deflactación de compra y una deflactación de venta. La deflactación de compra y venta son porcentajes que afectan positiva o negativamente a la tarifa.

8.1.4.2. Mapa de conceptos

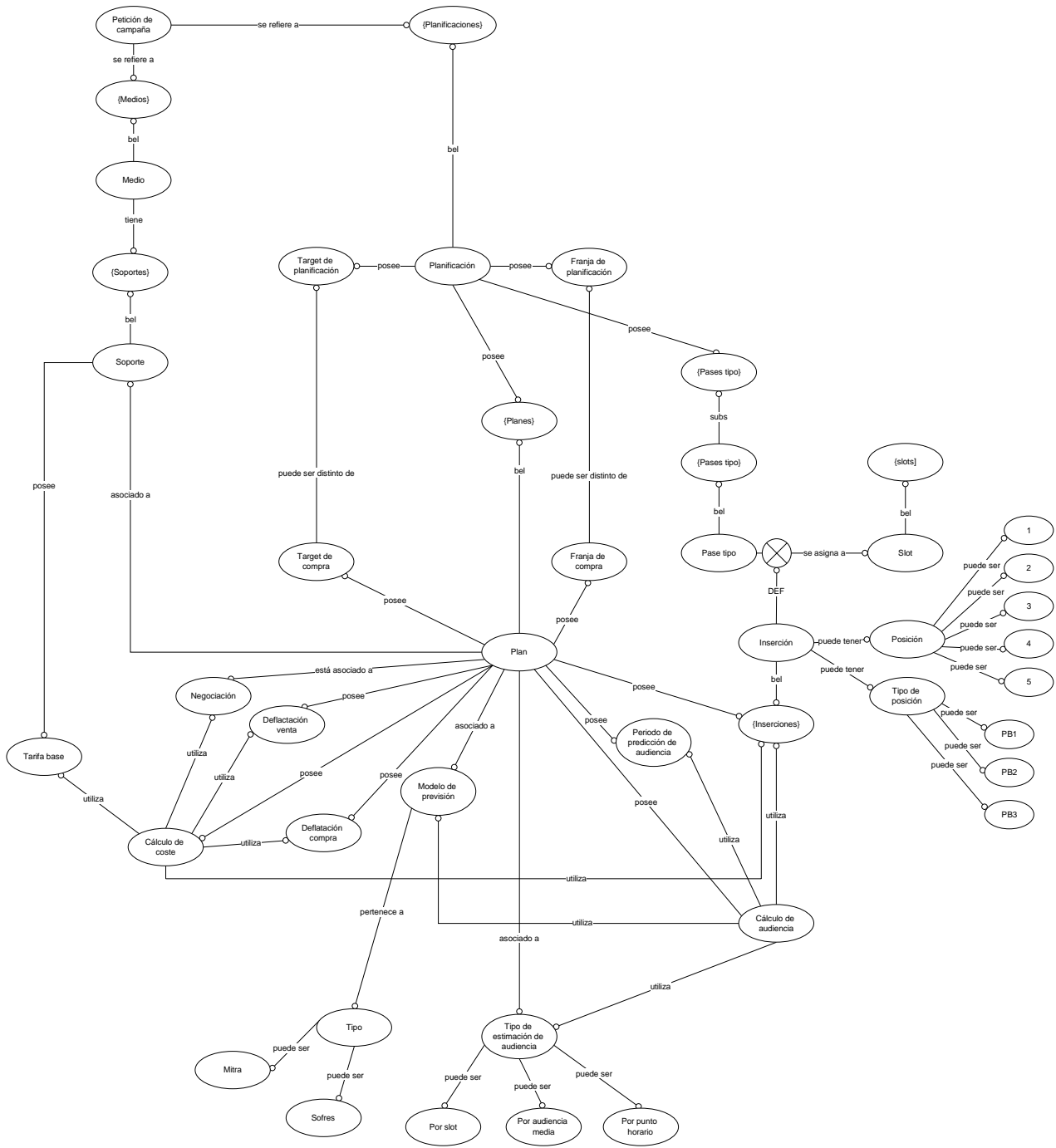


Figura 8.4. Mapa de Conceptos correspondiente a los PLANES

8.1.4.3. Diccionario de descripción

Declaraciones			
Conjuntos:	Pases tipo_1		
	Pases tipo_2		
	Soportes		
	Slots		
	Planes_1		
	Inserciones		
Subconjuntos:	Pases tipo_2	subs	Pases tipo_1
Individuos:	Pase tipo_2	bel	Pases tipo_2
	Slot	bel	Slots
	Inserción	bel	Inserciones
	Plan_1	bel	Planes_1
	Soporte	bel	Soportes
Definiciones			
Índice	Definición		
1	Inserción DEF p1		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Pase tipo_2	Se asigna a	Slot
2	Medio	Tiene	Soportes
3	Soporte	Tiene	Tarifa base
4	Plan_1	Posee	Target de compra
5	Plan_1	Posee	Franja de compra
6	Plan_1	Posee	Periodo de predicción de audiencia
7	Plan_1	Posee	Deflactación de compra
8	Plan_1	Posee	Deflactación de venta
9	Plan_1	Posee	Inserciones
10	Plan_1	Está asociado a	Negociación
11	Plan_1	Está asociado a	Modelo de previsión
12	Plan_1	Asociado a	Tipo de estimación de audiencia
13	Modelo de previsión	Pertenece a	Tipo
14	Tipo	Puede ser	Sofres
15	Tipo	Puede ser	Mitra
16	Tipo de estimación de audiencia	Puede ser	Por slot
17	Tipo de estimación de audiencia	Puede ser	Por audiencia media
18	Tipo de estimación de audiencia	Puede ser	Por punto horario
19	Cálculo de coste	Utiliza	Tarifa base
20	Cálculo de coste	Utiliza	Negociación
21	Cálculo de coste	Utiliza	Deflactación de compra
22	Cálculo de coste	Utiliza	Deflactación de venta
23	Cálculo de coste	Utiliza	Inserciones
24	Cálculo de audiencia	Utiliza	Inserciones
25	Cálculo de audiencia	Utiliza	Periodo de predicción de audiencia
26	Cálculo de audiencia	Utiliza	Modelo de previsión
27	Plan_1	Posee	Cálculo de coste
28	Plan_1	Posee	Cálculo de audiencia
29	Inserción	Puede tener	Posición
30	Inserción	Puede tener	Tipo de posición
31	Posición_1	Puede ser	1
32	Posición_1	Puede ser	2
33	Posición_1	Puede ser	3
34	Posición_1	Puede ser	4
35	Posición_1	Puede ser	5
36	Tipo de posición_1	Puede ser	PB1
37	Tipo de posición_1	Puede ser	PB2
38	Tipo de posición_1	Puede ser	PB3

Tabla 8.7. Diccionario de Descripción correspondiente a los PLANES

8.1.4.4. Interpretación parcial

Declaraciones			
Conjuntos:	Entity[repl]: Pases tipo_1		
	Entity[repl]: Pases tipo_2		
	Entity[repl]: Soportes		
	Entity[repl]: Slots		
	Entity[repl]: Planes_1		
	Entity[repl]: Inserciones		
Subconjuntos:	Entity[repl]: Pases tipo_2	Subs: subs	Entity[repl]: Pases tipo_1
Individuos:	Entity[notrepl]: Pase tipo_2	Bel: bel	Entity[repl]: Pases tipo_2
	Entity[notrepl]: Slot	Bel: bel	Entity[repl]: Slots
	Entity[notrepl]: Inserción	Bel: bel	Entity[repl]: Inserciones
	Entity[notrepl]: Plan_1	Bel: bel	Entity[repl]: Planes_1
	Entity[notrepl]: Soporte	Bel: bel	Entity[repl]: Soportes
Definiciones			
Índice	Definición		
1	Inserción DEF p1		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Entity[notrepl]: Pase tipo_2	Rel: Se asigna a	Entity[notrepl]: Slot
2	Entity[notrepl]: Medio	Rel: Tiene	Entity[repl]: Soportes
3	Entity[notrepl]: Soporte	-Pof: Tiene	Entity[notrepl]: Tarifa base
4	Entity[notrepl]: Plan_1	-Pof: Posee	Statespace: Target de compra
5	Entity[notrepl]: Plan_1	-Pof: Posee	Statespace: Franja de compra
6	Entity[notrepl]: Plan_1	-Pof: Posee	Statespace: Periodo de predicción de audiencia
7	Entity[notrepl]: Plan_1	-Pof: Posee	Statespace: Deflactación de compra
8	Entity[notrepl]: Plan_1	-Pof: Posee	Statespace: Deflactación de venta
9	Entity[notrepl]: Plan_1	-Pof: Posee	Entity[repl]: Inserciones
10	Entity[notrepl]: Plan_1	Entity[notrepl]: Está asociado a	Entity[notrepl]: Negociación
11	Entity[notrepl]: Plan_1	Entity[notrepl]: Está asociado a	Entity[notrepl]: Modelo de previsión
12	Entity[notrepl]: Plan_1	Entity[notrepl]: Asociado a	Statespace: Tipo de estimación de audiencia
13	Entity[notrepl]: Modelo de previsión	Entity[notrepl]: Pertenece a	Statespace: Tipo
14	Statespace: Tipo	Hval: Puede ser	Value: Sofres
15	Statespace: Tipo	Hval: Puede ser	Value: Mitra
16	Statespace: Tipo de estimación de audiencia	Hval: Puede ser	Value: Por slot
17	Statespace: Tipo de estimación de audiencia	Hval: Puede ser	Value: Por audiencia media
18	Statespace: Tipo de estimación de audiencia	Hval: Puede ser	Value: Por punto horario
19	Process: Cálculo de coste	Receives: Utiliza	Entity[notrepl]: Tarifa base
20	Process: Cálculo de coste	Receives: Utiliza	Entity[notrepl]: Negociación
21	Process: Cálculo de coste	Receives: Utiliza	Statespace: Deflactación de compra
22	Process: Cálculo de coste	Receives: Utiliza	Statespace: Deflactación de venta
23	Process: Cálculo de coste	Receives: Utiliza	Entity[repl]: Inserciones
24	Process: Cálculo de audiencia	Receives: Utiliza	Entity[repl]: Inserciones
25	Process: Cálculo de audiencia	Receives: Utiliza	Statespace: Periodo de predicción de audiencia
26	Process: Cálculo de audiencia	Receives: Utiliza	Entity[notrepl]: Modelo de previsión
27	Entity[notrepl]: Plan_1	-Pof: Posee	Process: Cálculo de coste
28	Entity[notrepl]: Plan_1	-Pof: Posee	Process: Cálculo de audiencia
29	Entity[notrepl]: Inserción	-Pof: Puede tener	Statespace: Posición
30	Entity[notrepl]: Inserción	-Pof: Puede tener	Statespace: Tipo de posición
31	Statespace: Posición	Hval: Puede ser	Value: 1
32	Statespace: Posición	Hval: Puede ser	Value: 2
33	Statespace: Posición	Hval: Puede ser	Value: 3
34	Statespace: Posición	Hval: Puede ser	Value: 4
35	Statespace: Posición	Hval: Puede ser	Value: 5
36	Statespace: Tipo de posición	Hval: Puede ser	Value: PB1
37	Statespace: Tipo de posición	Hval: Puede ser	Value: PB2
38	Statespace: Tipo de posición	Hval: Puede ser	Value: PB3

Tabla 8.8. Interpretación del Diccionario de Descripción correspondiente a los PLANES

8.1.5. ALTERNATIVAS

8.1.5.1. Descripción

Una ALTERNATIVA es un conjunto de pases tipo y un conjunto de planes.

8.1.5.2. Mapa de conceptos

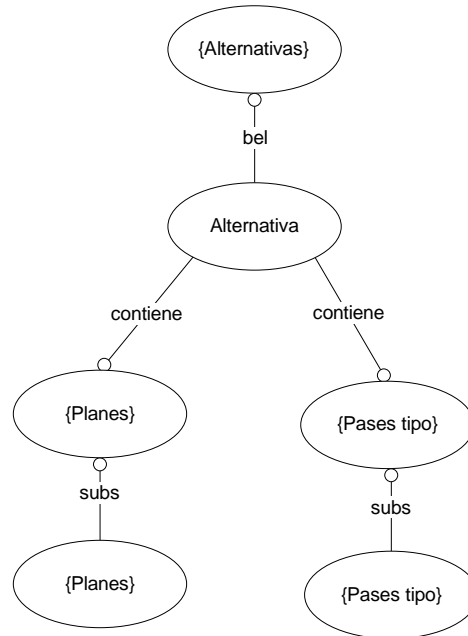


Figura 8.5. Mapa de Conceptos correspondiente a las ALTERNATIVAS

8.1.5.3. Diccionario de descripción

Declaraciones			
Conjuntos:	Alternativas_1		
	Planes_1		
	Planes_2		
	Pases tipo_3		
	Pases tipo_1		
Subconjuntos:	Planes_2	subs	Planes_1
	Pases tipo_3	subs	Pases tipo_1
Individuos:	Alternativa_3	bel	Alternativas_1
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Alternativa_3	Contiene	Pases tipo_3
2	Alternativa_3	Contiene	Planes_2

Tabla 8.9. Diccionario de Descripción correspondiente a las ALTERNATIVAS

8.1.5.4. Interpretación parcial

Declaraciones			
Conjuntos:	Entity[repl]: Alternativas_1		
	Entity[repl]: Planes_1		
	Entity[repl]: Planes_2		
	Entity[repl]: Pases tipo_3		
	Entity[repl]: Pases tipo_1		
Subconjuntos:	Entity[repl]: Planes_2	Subs: subs	Entity[repl]: Planes_1
	Entity[repl]: Pases tipo_3	Subs: subs	Entity[repl]: Pases tipo_1
Individuos:	Entity[notrepl]: Alternativa_3	Bel: bel	Entity[repl]: Alternativas_1
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Entity[notrepl]: Alternativa_3	-Pof: Contiene	Entity[repl]: Pases tipo_3
2	Entity[notrepl]: Alternativa_3	-Pof: Contiene	Entity[repl]: Planes_2

Tabla 8.10. Interpretación del Diccionario de Descripción correspondiente a las ALTERNATIVAS

8.1.6. TARIFA BASE

8.1.6.1. Descripción

Cada soporte proporciona una TARIFA BASE. La TARIFA BASE determina el coste de las distintas franjas de programación, de los distintos programas y de los distintos slots. A efectos de organización de la TARIFA BASE, una franja de programación se considera que contiene a distintos programas o distintos slots.

La tarifa puede ser distinta en cada día, esto es, la tarifa no se repite semanalmente, mensualmente, etc.

8.1.6.2. Mapa de conceptos

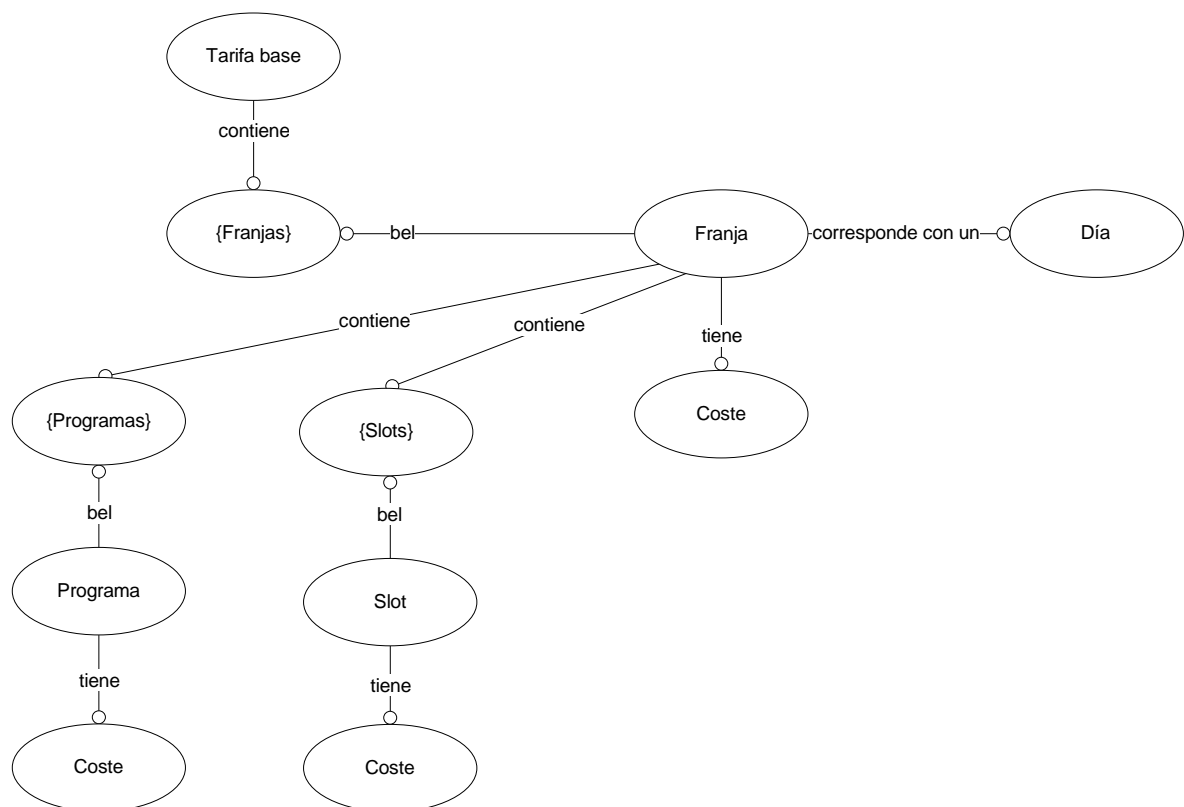


Figura 8.6. Mapa de Conceptos correspondiente a la TARIFA BASE

8.1.6.3. Diccionario de descripción

Declaraciones			
Conjuntos:	Franjas		
	Programas		
	Slots		
Subconjuntos:		subs	
Individuos:	Franja	bel	Franjas
	Programa	bel	Programas
	Slot	bel	Slots
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Franja	Contiene	Programas
2	Franja	Contiene	Slots
3	Franja	Tiene	Coste_1
4	Programa	Tiene	Coste_2
5	Slot	Tiene	Coste_3
6	Franja	Corresponde con un	Día

Tabla 8.11. Diccionario de Descripción correspondiente a la TARIFA BASE

8.1.6.4. Interpretación parcial

Declaraciones			
Conjuntos:	Entity[repl]: Franjas		
	Entity[repl]: Programas		
	Entity[repl]: Slots		
Subconjuntos:		subs	
Individuos:	Entity[notrepl]: Franja	Bel: bel	Entity[repl]: Franjas
	Entity[notrepl]: Programa	Bel: bel	Entity[repl]: Programas
	Entity[notrepl]: Slot	Bel: bel	Entity[repl]: Slots
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Entity[notrepl]: Franja	-Pof: Contiene	Entity[repl]: Programas
2	Entity[notrepl]: Franja	-Pof: Contiene	Entity[repl]: Slots
3	Entity[notrepl]: Franja	-Pof: Tiene	Statespace: Coste_1
4	Entity[notrepl]: Programa_1	-Pof: Tiene	Statespace: Coste_2
5	Entity[notrepl]: Slot	-Pof: Tiene	Statespace: Coste_3
6	Entity[notrepl]: Franja	-Pof: Corresponde con un	Statespace: Día

Tabla 8.12 Interpretación del Diccionario de Descripción correspondiente a la TARIFA BASE

8.1.7. NEGOCIACIÓN

8.1.7.1. Descripción

La NEGOCIACIÓN permite alterar la TARIFA BASE. De hecho, la NEGOCIACIÓN es muy similar a la TARIFA BASE, con escasas diferencias.

La NEGOCIACIÓN permite asignar a una franja de programación (o a un programa), un coste o un descuento. Este coste o descuento se superpone al coste de la TARIFA BASE para la franja o programa. Sin embargo, la negociación, a diferencia de la TARIFA BASE, no tiene una vigencia permanente, sino un periodo de validez.

Adicionalmente, mediante una NEGOCIACIÓN, es posible determinar el coste de una inserción con posición. En cada franja de programación, es posible especificar el coste, o el recargo, de una inserción en una posición y con un tipo de posición determinado.

8.1.7.2. Mapa de conceptos

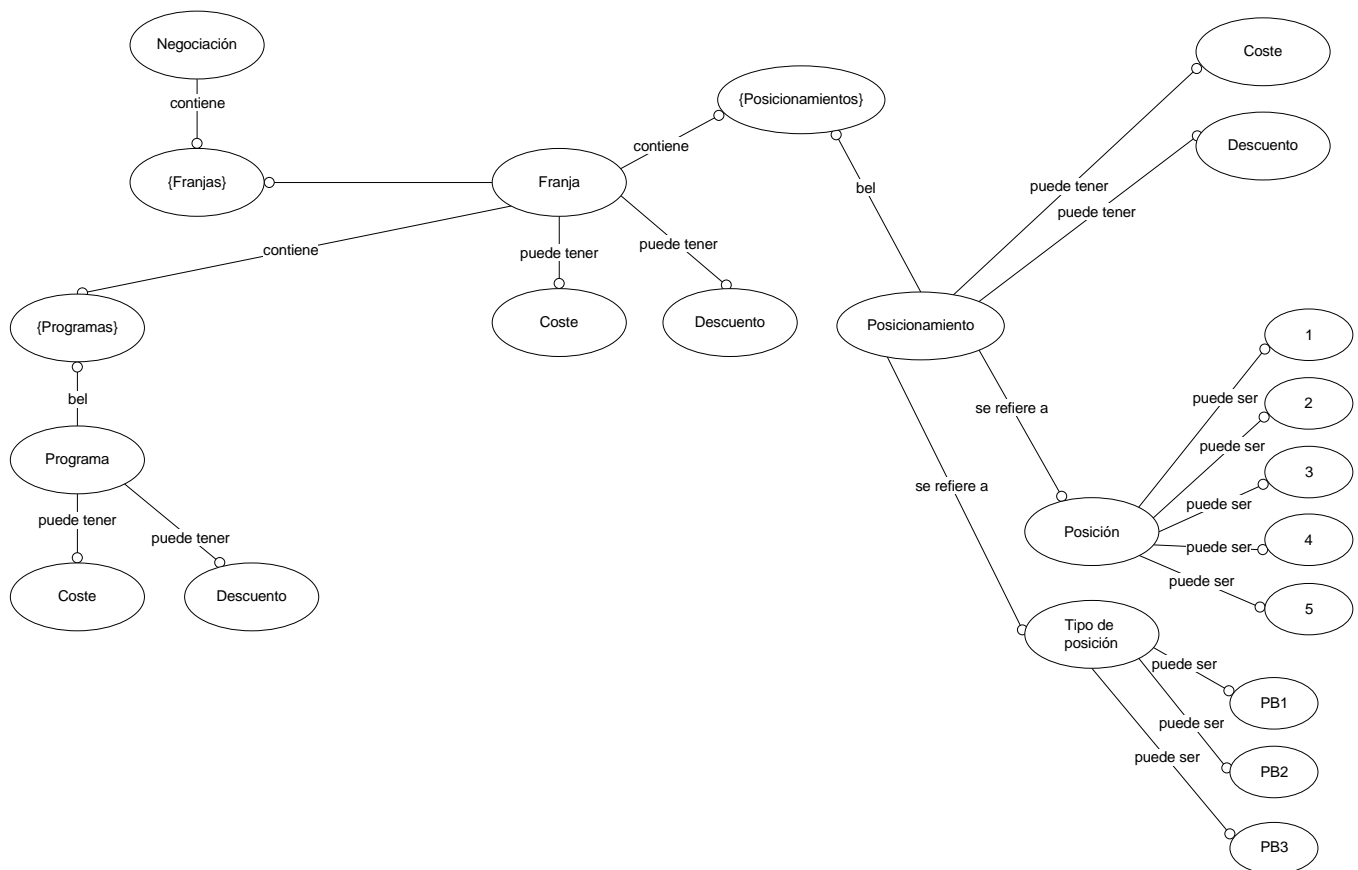


Figura 8.7. Mapa de Conceptos correspondiente a la NEGOCIACIÓN

8.1.7.3. Diccionario de descripción

Declaraciones			
Conjuntos:	Franjas		
	Programas		
	Posicionamientos		
Subconjuntos:		subs	
Individuos:	Franja	bel	Franjas
	Programa	bel	Programas
	Posicionamiento	bel	Posicionamientos
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Franja	Contiene	Programas
2	Franja	Contiene	Posicionamientos
3	Franja	Puede tener	Coste_4
4	Franja	Puede tener	Descuento_1
5	Programa	Puede tener	Coste_5
6	Programa	Puede tener	Descuento_2
7	Programa	Puede tener	Coste_6
8	Programa	Puede tener	Descuento_3
9	Posicionamiento	Se refiere a	Posición_2
10	Posicionamiento	Se refiere a	Tipo de posición_2
11	Posición_2	Puede ser	1
12	Posición_2	Puede ser	2
13	Posición_2	Puede ser	3
14	Posición_2	Puede ser	4
15	Posición_2	Puede ser	5
16	Tipo de posición_2	Puede ser	PB1
17	Tipo de posición_2	Puede ser	PB2
18	Tipo de posición_2	Puede ser	PB3

Tabla 8.13. Diccionario de Descripción correspondiente a la NEGOCIACIÓN

8.1.7.4. Interpretación parcial

Declaraciones			
Conjuntos:	Entity[repl]: Franjas		
	Entity[repl]: Programas		
	Entity[repl]: Posicionamientos		
Subconjuntos:		subs	
Individuos:	Entity[notrepl]: Franja	Bel: bel	Entity[repl]: Franjas
	Entity[notrepl]: Programa	Bel: bel	Entity[repl]: Programas
	Entity[notrepl]: Posicionamiento	Bel: bel	Entity[repl]: Posicionamientos
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Entity[notrepl]: Franja	-Pof: Contiene	Entity[repl]: Programas
2	Entity[notrepl]: Franja	-Pof: Contiene	Entity[repl]: Posicionamientos
3	Entity[notrepl]: Franja	-Pof: Puede tener	Statespace: Coste_4
4	Entity[notrepl]: Franja	-Pof: Puede tener	Statespace: Descuento_1
5	Entity[notrepl]: Programa	-Pof: Puede tener	Statespace: Coste_5
6	Entity[notrepl]: Programa	-Pof: Puede tener	Statespace: Descuento_2
7	Entity[notrepl]: Programa	-Pof: Puede tener	Statespace: Coste_6
8	Entity[notrepl]: Programa	-Pof: Puede tener	Statespace: Descuento_3
9	Entity[notrepl]: Posicionamiento	-Pof: Se refiere a	Statespace: Posición_2
10	Entity[notrepl]: Posicionamiento	-Pof: Se refiere a	Statespace: Tipo de posición_2
11	Statespace: Posición_2	Hval: Puede ser	Value: 1
12	Statespace: Posición_2	Hval: Puede ser	Value: 2
13	Statespace: Posición_2	Hval: Puede ser	Value: 3
14	Statespace: Posición_2	Hval: Puede ser	Value: 4
15	Statespace: Posición_2	Hval: Puede ser	Value: 5
16	Statespace: Tipo de posición_2	Hval: Puede ser	Value: PB1
17	Statespace: Tipo de posición_2	Hval: Puede ser	Value: PB2
18	Statespace: Tipo de posición_2	Hval: Puede ser	Value: PB3

Tabla 8.14. Interpretación del Diccionario de Descripción correspondiente a la NEGOCIACIÓN

8.1.8. MODELO DE PREVISIÓN

8.1.8.1. Descripción

Un MODELO DE PREVISIÓN indica la audiencia que ha tenido cada slot en un día determinado, dividida por targets.

8.1.8.2. Mapa de conceptos

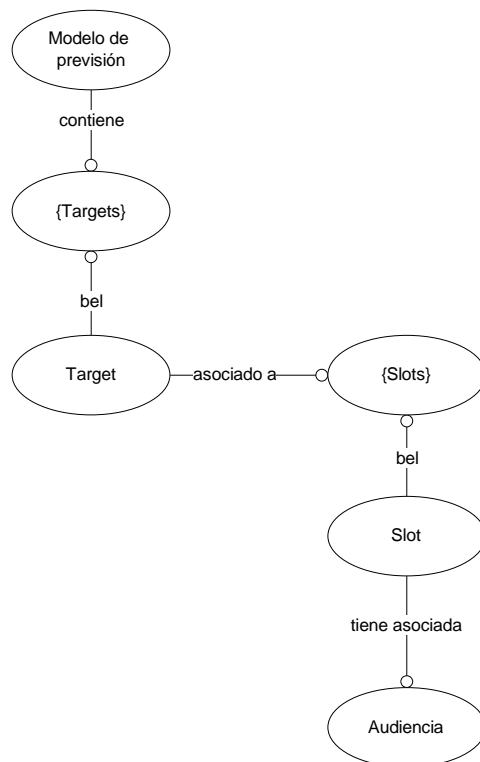


Figura 8.8. Mapa de Conceptos correspondiente al MODELO DE PREVISIÓN

8.1.8.3. Diccionario de descripción

Declaraciones			
Conjuntos:	Targets		
	Slots		
Subconjuntos:		subs	
Individuos:	Slot	bel	Slots
	Target	bel	Targets
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Modelo de previsión	Contiene	Targets
2	Target	Asociado a	Slots
3	Slot	Tiene asociada	Audiencia

Tabla 8.15. Diccionario de Descripción correspondiente al MODELO DE PREVISIÓN

8.1.8.4. Interpretación parcial

Declaraciones			
Conjuntos:	Entity[repl]: Targets		
	Entity[repl]: Slots		
Subconjuntos:		subs	
Individuos:	Entity[notrepl]: Slot	Bel: bel	Entity[repl]: Slots
	Entity[notrepl]: Target	Bel: bel	Entity[repl]: Targets
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Entity[notrepl]: Modelo de previsión	-Pof: Contiene	Entity[repl]: Targets
2	Entity[notrepl]: Target	-Pof: Asociado a	Entity[repl]: Slots
3	Entity[notrepl]: Slot	-Pof: Tiene asociada	Statespace: Audiencia

Tabla 8.16. Interpretación del Diccionario de Descripción correspondiente al MODELO DE PREVISIÓN

8.1.9. CAMPAÑA

8.1.9.1. Descripción

Cuando se ha realizado la PLANIFICACIÓN de una petición de campaña, se seleccionan algunas de las distintas ALTERNATIVAS para realizar la compra de espacios publicitarios. Toda la gestión de la compra de espacios publicitarios se lleva a cabo mediante el concepto de CAMPAÑA.

Una CAMPAÑA está siempre asociada a una petición de campaña concreta, obtenida durante el BRIEFING. A efectos organizativos, una campaña se compone de varias subcampañas, tantas como medios publicitarios se utilicen en una determinada campaña publicitaria. Cada subcampaña, por lo tanto, representa la compra de espacios publicitarios correspondientes a un único medio. Cada subcampaña se divide en uno o varios presupuestos, con el fin de proporcionar una forma adicional de estructurar las subcampañas. Un presupuesto contiene una o varias ALTERATIVAS, con la única restricción de que la ALTERNATIVA debe ser para un único producto y un único medio.

8.1.9.2. Mapa de conceptos

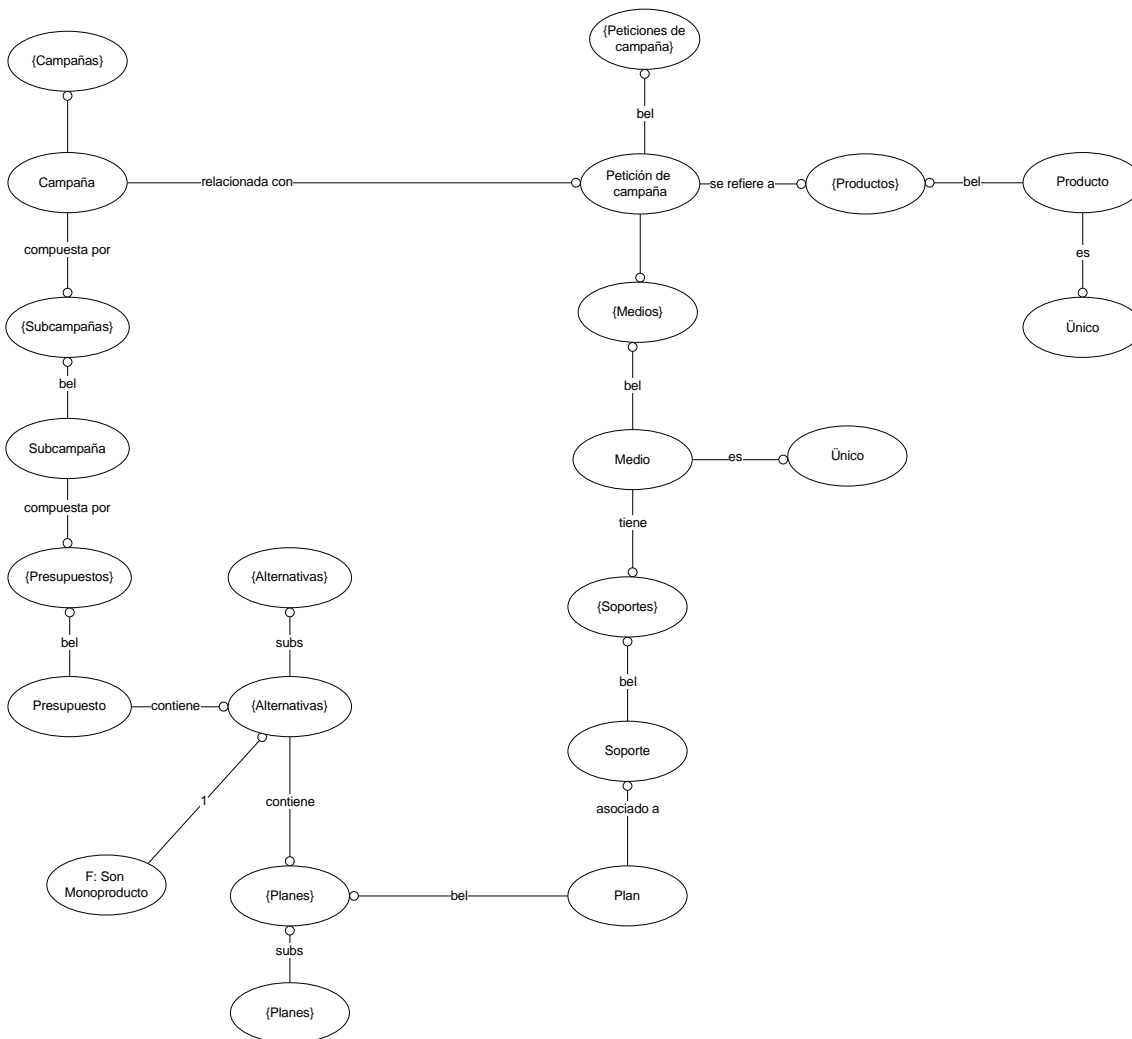


Figura 8.9. Mapa de Conceptos correspondiente a la CAMPAÑA

8.1.9.3. Diccionario de descripción

Declaraciones			
Conjuntos:	Campañas		
	Subcampañas		
	Presupuestos		
	Alternativas_1		
	Alternativas_2		
	Productos		
Subconjuntos:	Alternativas_2	subs	Alternativas_1
Individuos:	Campaña	bel	Campañas
	Subcampaña	bel	Subcampañas
	Presupuesto	bel	Presupuestos
	Producto	bel	Productos
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Campañas	Compuesta por	Subcampañas
2	Campaña	Relacionada con	Petición de campaña
3	Subcampaña	Compuesta por	Presupuestos
4	Presupuesto	Contiene	Alternativas_2
5	Producto	Es	Único
6	Medio	Es	Único
7	P: Son Monoproducto	1	Alternativas_2

Tabla 8.17. Diccionario de Descripción correspondiente a la CAMPAÑA

8.1.9.4. Interpretación parcial

Declaraciones			
Conjuntos:	Entity[repl]: Campañas		
	Entity[repl]: Subcampañas		
	Entity[repl]: Presupuestos		
	Entity[repl]: Alternativas_1		
	Entity[repl]: Alternativas_2		
	Entity[repl]: Productos		
Subconjuntos:	Entity[repl]: Alternativas_2	Subs: Subs	Entity[repl]: Alternativas_1
Individuos:	Entity[notrepl]: Campaña	Bel: bel	Entity[repl]: Campañas
	Entity[notrepl]: Subcampaña	Bel: bel	Entity[repl]: Subcampañas
	Entity[notrepl]: Presupuesto	Bel: bel	Entity[repl]: Presupuestos
	Entity[notrepl]: Producto	Bel: bel	Entity[repl]: Productos
Definiciones			
Índice	Definición		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Entity[notrepl]: Campaña	-Pof: Compuesta por	Entity[repl]: Subcampañas
2	Entity[notrepl]: Campaña	Rel: Relacionada con	Entity[notrepl]: Petición de campaña
3	Entity[notrepl]: Subcampaña	-Pof: Compuesta por	Entity[repl]: Presupuestos
4	Entity[notrepl]: Presupuesto	-Pof: Contiene	Entity[notrepl]: Alternativas_2
5	Entity[notrepl]: Producto	Operand: Es	Constraint: Único
6	Entity[notrepl]: Medio	Operand: Es	Constraint: Único
7	Constraint: Son Monoproducto	Operand: 1	Entity[repl]: Alternativas_2

Tabla 8.18. Interpretación del Diccionario de Descripción correspondiente a la CAMPAÑA

8.1.10. ÓRDENES

8.1.10.1. Descripción

Las inserciones de un plan tienen una consideración distinta cuando pasan a formar parte integrante de un presupuesto. En este caso, las inserciones se consideran “Órdenes”, ya que son los elementos mínimos de compra de espacio en los diferentes soportes.

Las órdenes pueden pasar por diferentes estados. El primero de ellos es “Insertada”, el cual ocurre cuando se crea la orden. Una orden insertada puede pasar al estado “Emitida”, lo cual significa que se ha comprado el slot correspondiente del soporte deseado. Una orden emitida puede pasar a ser “Confirmada”, si se verifica que efectivamente el spot ha sido emitido por el soporte correspondiente. Una orden que no ha sido emitida puede pasar al estado de “Cancelada”.

8.1.10.2. Mapa de conceptos

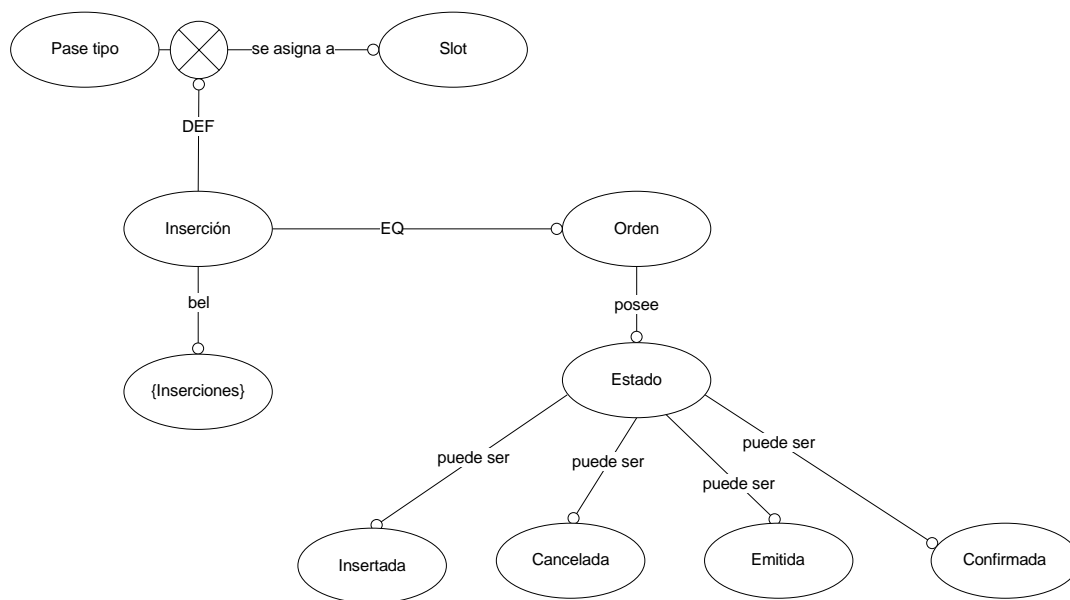


Figura 8.10. Mapa de Conceptos correspondiente a la ORDENES

8.1.10.3. Diccionario de descripción

Declaraciones			
Conjuntos:			
Subconjuntos:		subs	
Individuos:		bel	
Definiciones			
Índice	Definición		
1	Inserción EQ Orden		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Orden	Posee	Estado
2	Estado	Puede ser	Emitida
3	Estado	Puede ser	Confirmada
4	Estado	Puede ser	Insertada
5	Estado	Puede ser	Cancelada

Tabla 8.19. Diccionario de Descripción correspondiente a ORDENES

8.1.10.4. Interpretación parcial

Declaraciones			
Conjuntos:			
Subconjuntos:		subs	
Individuos:		bel	
Definiciones			
Índice	Definición		
1	Inserción EQ Orden		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	P25	-Pof: posee	Statespace: Estado
2	Statespace: Estado	Hval: Puede ser	Value: Emitida
3	Statespace: Estado	Hval: Puede ser	Value: Confirmada
4	Statespace: Estado	Hval: Puede ser	Value: Insertada
5	Statespace: Estado	Hval: Puede ser	Value: Cancelada

Tabla 8.20. Interpretación del Diccionario de Descripción correspondiente a ÓRDENES

8.1.11. FACTURACIÓN

8.1.11.1. Descripción

Una vez que una orden ha sido confirmada, pasa a ser facturada a cliente, esto es, Media Planning factura al cliente el coste de emisión y planificación de la orden. Una factura puede ser abonada a cliente en cualquier momento, si se dieran las condiciones para ello.

8.1.11.2. Mapa de conceptos

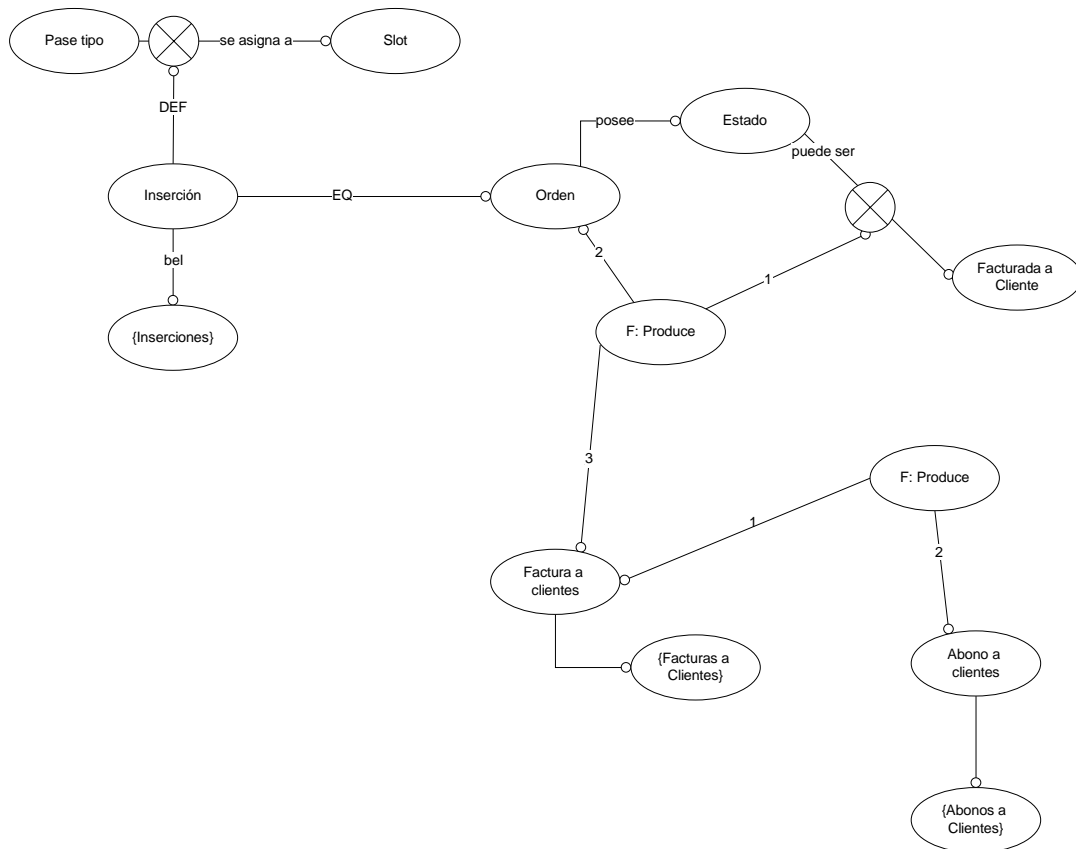


Figura 8.11. Mapa de Conceptos correspondiente a la FACTURACIÓN

8.1.11.3. Diccionario de descripción

Declaraciones			
Conjuntos:	Abonos a Cliente		
	Facturas a Cliente		
Subconjuntos:		subs	
Individuos:	Abono a Cliente		bel
	Factura a Cliente		bel
Definiciones			
Índice	Definición		
1	Inserción EQ Orden		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	Orden	Posee	Estado
2	Estado	Puede ser	Facturada a Cliente
3	F: Produce_1	1	Facturada a Cliente
4	F: Produce_1	2	Orden
5	F: Produce_1	3	Factura a Cliente
6	F: Produce_2	1	Factura a Cliente
7	F: Produce_2	2	Abono a Cliente

Tabla 8.21. Diccionario de Descripción correspondiente a la FACTURACIÓN

8.1.11.4. Interpretación parcial

Declaraciones			
Conjuntos:	Entity[repl]: Abonos a Cliente		
	Entity[repl]: Facturas a Cliente		
Subconjuntos:		subs	
Individuos:	Entity[notrepl]: Abono a Cliente	Bel: bel	Entity[repl]: Abonos a Cliente
	Entity[notrepl]: Factura a Cliente	Bel: bel	Entity[repl]: Facturas a Cliente
Definiciones			
Índice	Definición		
1	Inserción EQ Orden		
Proposiciones			
Índice	Concepto-1	Asociación	Concepto-2
1	P25	-Pof: posee	Statespace: Estado
2	Statespace: Estado	Hval: Puede ser	Value: Facturada a Cliente
3	Process: Produce_1	Receives: 1	P101
4	Process: Produce_1	Receives: 2	P25
5	Process: Produce_1	Sends: 3	Entity[notrepl]: Factura a Proveedor
6	Process: Produce_2	Receives: 1	Entity[notrepl]: Factura a Proveedor
7	Process: Produce_2	Sends: 2	Entity[notrepl]: Abono a Cliente

Tabla 8.22. Interpretación del Diccionario de Descripción correspondiente a la FACTURACIÓN

8.2. APLICACIÓN DE LA TÉCNICA IMCI

La tabla 8.19, mostrada en las páginas siguientes, representa la aplicación de la Técnica IMCI al conjunto de Diccionarios de Descripción parciales presentados en la sección 8.1. El resultado del cálculo del Modelo Conceptual Idóneo ha resultado ser el Diagrama de Clases, con una Adecuación del 61%.

Proposición			Diagrama de Flujo de Datos	Entidad-Relación	Diagrama de Clases	Diagrama de Transición de Estados	Statechart	Casos de Uso
Entity[rep] : Alternativas_1			1	1	1			
Entity[rep] : Alternativas_2			1	1	1			
Entity[rep] : Campañas			1	1	1			
Entity[rep] : Clientes			1	1	1			
Entity[rep] : Franjas			1	1	1			
Entity[rep] : Materiales			1	1	1			
Entity[rep] : Medios			1	1	1			
Entity[rep] : Pases tipo_1			1	1	1			
Entity[rep] : Pases tipo_2			1	1	1			
Entity[rep] : Pases tipo_3			1	1	1			
Entity[rep] : Peticiones de campaña			1	1	1			
Entity[rep] : Planes_1			1	1	1			
Entity[rep] : Planes_2			1	1	1			
Entity[rep] : Posicionamientos			1	1	1			
Entity[rep] : Presupuestos			1	1	1			
Entity[rep] : Productos			1	1	1			
Entity[rep] : Programas			1	1	1			
Entity[rep] : Slots			1	1	1			
Entity[rep] : Soportes			1	1	1			
Entity[rep] : Subcampañas			1	1	1			
Entity[rep] : Targets			1	1	1			
Entity[rep] : Facturas a Cliente			1	1	1			
Entity[rep] : Abonos a Cliente			1	1	1			
Entity[rep] : Pases tipo_2	Subs: subs	Entity[rep] : Pases tipo_1			1			
Entity[rep] : Planes_2	Subs: subs	Entity[rep] : Planes_1			1			
Entity[rep] : Pases tipo_3	Subs: subs	Entity[rep] : Pases tipo_1			1			
Entity[rep] : Alternativas_2	Subs: Subs	Entity[rep] : Alternativas_1			1			
Entity[notrep] : Alternativa_3	Bel: bel	Entity[rep] : Alternativas_1						
Entity[notrep] : Campaña	Bel: bel	Entity[rep] : Campañas						

	Proposición		Diagrama de Flujo de Datos	Entidad-Relación	Diagrama de Clases	Diagrama de Transición de Estados	Statechart	Casos de Uso
	Entity[notrep!] : Cliente	Bel: bel	Entity[repl] : Clientes					
	Entity[notrep!] : Franja	Bel: bel	Entity[repl] : Franjas					
	Entity[notrep!] : Franja de planificación	Bel: bel	Entity[repl] : Franjas de planificación					
	Entity[notrep!] : Material	Bel: bel	Entity[repl] : Materiales					
	Entity[notrep!] : Medio_1	Bel: bel	Entity[repl] : Medios					
	Entity[notrep!] : Medio_2	Bel: bel	Entity[repl] : Medios					
	Entity[notrep!] : Pase tipo_1	Bel: bel	Entity[repl] : Pases tipo_1					
	Entity[notrep!] : Pase tipo_2	Bel: bel	Entity[repl] : Pases tipo_2					
	Entity[notrep!] : Petición de campaña	Bel: bel	Entity[repl] : Peticiones de campaña					
	Entity[notrep!] : Plan_1	Bel: bel	Entity[repl] : Planes_1					
	Entity[notrep!] : Planificación	Bel: bel	Entity[repl] : Planificaciones					
	Entity[notrep!] : Posicionamiento	Bel: bel	Entity[repl] : Posicionamientos					
	Entity[notrep!] : Presupuesto	Bel: bel	Entity[repl] : Presupuestos					
	Entity[notrep!] : Producto	Bel: bel	Entity[repl] : Productos					
	Entity[notrep!] : Programa	Bel: bel	Entity[repl] : Programas					
	Entity[notrep!] : Slot	Bel: bel	Entity[repl] : Slots					
	Entity[notrep!] : Soporte	Bel: bel	Entity[repl] : Soportes					
	Entity[notrep!] : Subcampaña	Bel: bel	Entity[repl] : Subcampañas					
	Entity[notrep!] : Factura a Cliente	Bel: bel	Entity[repl] : Facturas a Cliente					
	Entity[notrep!] : Abono a Cliente	Bel: bel	Entity[repl] : Abonos a Cliente					
P1	Entity[notrep!] : Cliente	Rel: Realiza	Entity[repl] : Peticiones de campaña	1	1			
P2	Entity[notrep!] : Petición de campaña	Rel: Se refiere a	Entity[repl] : Productos	1	1			
P3	Entity[notrep!] : Petición de campaña	Rel: Se refiere a	Entity[repl] : Medios	1	1			
P4	Entity[notrep!] : Medio_1	-Pof: Se le asigna	Statespace: Inversión total	1	1			
P5	Entity[notrep!] : Medio_1	-Pof: Se le asigna	Statespace: Target de planificación_1	1	1			
P6	Entity[notrep!] : Medio_1	-Pof: Se le asigna	Statespace: Periodo de campaña	1	1			
P7	Entity[notrep!] : Planificación	-Pof: Posee	Entity[notrep!] : Franja de planificación		1			
P8	Entity[notrep!] : Planificación	-Pof: Posee	Statespace: Periodo de vigencia	1	1			
P9	Entity[notrep!] : Planificación	-Pof: Posee	Statespace: Target de planificación_2	1	1			
P10	Entity[notrep!] : Planificación	rel: Posee	Entity[repl] : Pases tipo_1	1	1			
p11	Entity[notrep!] : Planificación	rel: Posee	Entity[repl] : Planes_1	1	1			
P12	Entity[notrep!] : Planificación	rel: Posee	Entity[repl] : Alternativas_1	1	1			
P13	Constraint: Comprendido en	Operand: 1	Statespace: Periodo de campaña					
P14	Constraint: Comprendido en	Operand: 2	Statespace: Periodo de vigencia					
P15	Entity[notrep!] : Material	Rel: Asociado a	Entity[notrep!] : Cliente	1	1			
P16	Entity[notrep!] : Pase tipo_1	Rel: Se refiere a	Entity[notrep!] : Producto	1	1			
P17	Entity[notrep!] : Pase tipo_1	Rel: Se refiere a	Entity[repl] : Materiales	1	1			
P18	Entity[notrep!] : Pase tipo_1	-Pof: Posee	Statespace: Duración_1	1	1			
P19	Entity[notrep!] : Material	Rel: Corresponde con un	Entity[notrep!] : Producto	1	1			

Proposición				Diagrama de Flujo de Datos	Entidad-Relación	Diagrama de Clases	Diagrama de Transición de Estados	Statechart	Casos de Uso
P20	Entity[notrepl]: Material	Rel: Posee	Entity[notrepl]: Medio_2		1	1			
P21	Entity[notrepl]: Material	-Pof: Posee	Statespace: Duración_2		1	1			
P22	Entity[notrepl]: Material	-Pof: Posee	Statespace: Periodo de caducidad		1	1			
P23	Contrain: Es mayor o igual que	Operand: 1	Statespace: Duración_1						
P24	Contrain: Es mayor o igual que	Operand: 2	Statespace: Duración_2						
P25	Entity[notrepl]: Pase tipo_2	Rel: Se asigna a	Entity[notrepl]: Slot		1	1			
P26	Entity[notrepl]: Medio	Rel: Tiene	Entity[repl]: Soportes		1	1			
P27	Entity[notrepl]: Soporte	-Pof: Tiene	Entity[notrepl]: Tarifa base			1			
P28	Entity[notrepl]: Plan_1	-Pof: Posee	Statespace: Target de compra		1	1			
P29	Entity[notrepl]: Plan_1	-Pof: Posee	Statespace: Franja de compra		1	1			
P30	Entity[notrepl]: Plan_1	-Pof: Posee	Statespace: Periodo de predicción de audiencia		1	1			
P31	Entity[notrepl]: Plan_1	-Pof: Posee	Statespace: Deflactación de compra		1	1			
P32	Entity[notrepl]: Plan_1	-Pof: Posee	Statespace: Deflactación de venta		1	1			
P33	Entity[notrepl]: Plan_1	-Pof: Posee	p25		1	1			
P34	Entity[notrepl]: Plan_1	Rel: Está asociado a	Entity[notrepl]: Negociación		1	1			
P35	Entity[notrepl]: Plan_1	Rel: Está asociado a	Entity[notrepl]: Modelo de previsión		1	1			
P36	Entity[notrepl]: Plan_1	-pof: Asociado a	Statespace: Tipo de estimación de audiencia		1	1			
P37	Entity[notrepl]: Modelo de previsión	-pof: Pertenece a	Statespace: Tipo		1	1			
P38	Statespace: Tipo	Hval: Puede ser	Value: Sofres				1	1	
P39	Statespace: Tipo	Hval: Puede ser	Value: Mitra				1	1	
P40	Statespace: Tipo de estimación de audiencia	Hval: Puede ser	Value: Por slot				1	1	
P41	Statespace: Tipo de estimación de audiencia	Hval: Puede ser	Value: Por audiencia media				1	1	
P42	Statespace: Tipo de estimación de audiencia	Hval: Puede ser	Value: Por punto horario				1	1	
P43	Process: Cálculo de coste	Receives: Utiliza	Entity[notrepl]: Tarifa base	1					
P44	Process: Cálculo de coste	Receives: Utiliza	Entity[notrepl]: Negociación	1					
P45	Process: Cálculo de coste	Receives: Utiliza	Statespace: Deflactación de compra	1					
P46	Process: Cálculo de coste	Receives: Utiliza	Statespace: Deflactación de venta	1					
P47	Process: Cálculo de coste	Receives: Utiliza	p25	1					
P48	Process: Cálculo de audiencia	Receives: Utiliza	p25	1					
P49	Process: Cálculo de audiencia	Receives: Utiliza	Statespace: Periodo de predicción de audiencia	1					
P50	Process: Cálculo de audiencia	Receives: Utiliza	Entity[notrepl]: Modelo de previsión	1					
P51	Entity[notrepl]: Plan_1	-Pof: Posee	Process: Cálculo de coste			1			
P52	Entity[notrepl]: Plan_1	-Pof: Posee	Process: Cálculo de audiencia			1			
P53	p25	-Pof: Puede tener	Statespace: Posición		1	1			
P54	p25	-Pof: Puede tener	Statespace: Tipo de posición		1	1			
P55	Statespace: Posición	Hval: Puede ser	Value: 1				1	1	
P56	Statespace: Posición	Hval: Puede ser	Value: 2				1	1	
P57	Statespace: Posición	Hval: Puede ser	Value: 3				1	1	
P58	Statespace: Posición	Hval: Puede ser	Value: 4				1	1	

	Proposición			Diagrama de Flujo de Datos	Entidad-Relación	Diagrama de Clases	Diagrama de Transición de Estados	Statechart	Casos de Uso
P59	Statespace: Posición	Hval: Puede ser	Value: 5				1	1	
P60	Statespace: Tipo de posición	Hval: Puede ser	Value: PB1				1	1	
P61	Statespace: Tipo de posición	Hval: Puede ser	Value: PB2				1	1	
P62	Statespace: Tipo de posición	Hval: Puede ser	Value: PB3				1	1	
P63	Entity[notrepl]: Alternativa_3	-Pof: Contiene	Entity[repl]: Pases tipo_3			1			
P64	Entity[notrepl]: Alternativa_3	-Pof: Contiene	Entity[repl]: Planes_2			1			
P65	Entity[notrepl]: Franja	-Pof: Contiene	Entity[repl]: Programas			1			
P66	Entity[notrepl]: Franja	-Pof: Contiene	Entity[repl]: Slots			1			
P67	Entity[notrepl]: Franja	-Pof: Tiene	Statespace: Coste_1		1	1			
P68	Entity[notrepl]: Programa_1	-Pof: Tiene	Statespace: Coste_2		1	1			
P69	Entity[notrepl]: Slot	-Pof: Tiene	Statespace: Coste_3		1	1			
P70	Entity[notrepl]: Franja	-Pof: Corresponde con un	Statespace: Día		1	1			
P71	Entity[notrepl]: Franja	-Pof: Contiene	Entity[repl]: Posicionamientos			1			
P72	Entity[notrepl]: Franja	-Pof: Puede tener	Statespace: Coste_4		1	1			
P73	Entity[notrepl]: Franja	-Pof: Puede tener	Statespace: Descuento_1		1	1			
P74	Entity[notrepl]: Programa	-Pof: Puede tener	Statespace: Coste_5		1	1			
P75	Entity[notrepl]: Programa	-Pof: Puede tener	Statespace: Descuento_2		1	1			
P76	Entity[notrepl]: Programa	-Pof: Puede tener	Statespace: Coste_6		1	1			
P77	Entity[notrepl]: Programa	-Pof: Puede tener	Statespace: Descuento_3		1	1			
P78	Entity[notrepl]: Posicionamiento	-Pof: Se refiere a	Statespace: Posición_2		1	1			
P79	Entity[notrepl]: Posicionamiento	-Pof: Se refiere a	Statespace: Tipo de posición_2		1	1			
P80	Statespace: Posición_2	Hval: Puede ser	Value: 1				1	1	
P81	Statespace: Posición_2	Hval: Puede ser	Value: 2				1	1	
P82	Statespace: Posición_2	Hval: Puede ser	Value: 3				1	1	
P83	Statespace: Posición_2	Hval: Puede ser	Value: 4				1	1	
P84	Statespace: Posición_2	Hval: Puede ser	Value: 5				1	1	
P85	Statespace: Tipo de posición_2	Hval: Puede ser	Value: PB1				1	1	
P86	Statespace: Tipo de posición_2	Hval: Puede ser	Value: PB2				1	1	
P87	Statespace: Tipo de posición_2	Hval: Puede ser	Value: PB3				1	1	
P88	Entity[notrepl]: Modelo de previsión	-Pof: Contiene	Entity[repl]: Targets			1			
P89	Entity[notrepl]: Target	-Pof: Asociado a	Entity[repl]: Slots			1			
P90	Entity[notrepl]: Slot	-Pof: Tiene asociada	Statespace: Audiencia		1	1			
P91	Entity[notrepl]: Campaña	-Pof: Compuesta por	Entity[repl]: Subcampañas			1			
P92	Entity[notrepl]: Campaña	Rel: Relacionada con	Entity[notrepl]: Petición de campaña		1	1			
P93	Entity[notrepl]: Subcampaña	-Pof: Compuesta por	Entity[repl]: Presupuestos			1			
P94	Entity[notrepl]: Presupuesto	-Pof: Contiene	Entity[notrepl]: Alternativas_2			1			
P95	Entity[notrepl]: Producto	Operand: Es	Constraint: Único						
P96	Entity[notrepl]: Medio	Operand: Es	Constraint: Único						
P97	P25	-Pof: posee	Statespace: Estado		1	1			

Proposición				Diagrama de Flujo de Datos	Entidad-Relación	Diagrama de Clases	Diagrama de Transición de Estados	Statechart	Casos de Uso
P98	Statespace: Estado	Hval: Puede ser	Value: Emitida				1	1	
P99	Statespace: Estado	Hval: Puede ser	Value: Confirmada				1	1	
P100	Statespace: Estado	Hval: Puede ser	Value: Insertada				1	1	
P101	Statespace: Estado	Hval: Puede ser	Value: Facturada a Cliente				1	1	
P102	Statespace: Estado	Hval: Puede ser	Value: Cancelada				1	1	
P104	Process: Produce_1	Receives: 1	P101	1					
P105	Process: Produce_1	Receives: 2	P25	1					
P106	Process: Produce_1	Sends: 3	Entity[notrepl]: Factura a Cliente	1					
P107	Process: Produce_2	Receives: 1	Entity[notrepl]: Factura a Cliente	1					
P108	Process: Produce_2	Sends: 2	Entity[notrepl]: Abono a Cliente	1					
P109	Constraint: Son Monoproducto	Operand: 1	Entity[repl]: Alternativas_2						
Idoneidad				36	71	89	26	26	0
Adecuación:				24%	47%	59%	17%	17%	0%

Tabla 8.23. Aplicación de la Técnica IMCI

8.3. FRAGMENTOS

La tabla 8.20, mostrada en las páginas siguientes, contiene la derivación de cada uno de los fragmentos del Diagrama de Clases. La regla de derivación (3) del Diagrama de Clases se ha resuelto siempre, con el fin de producir un modelo lo más completo posible, de forma afirmativa, esto es, siempre se ha considerado que toda entidad de calificador **notrepl** podía ser calificada, igualmente, como **repl**.

Proposición	Fragmento del Diagrama de Clases
Entity[repl]: Alternativas_1	:Alternativas_1
Entity[repl]: Alternativas_2	:Alternativas_2
Entity[repl]: Campañas	:Campañas
Entity[repl]: Clientes	:Clientes
Entity[repl]: Franjas	:Franjas
Entity[repl]: Materiales	:Materiales
Entity[repl]: Medios	:Medios
Entity[repl]: Pases tipo_1	:Pases tipo_1
Entity[repl]: Pases tipo_2	:Pases tipo_2
Entity[repl]: Pases tipo_3	:Pases tipo_3
Entity[repl]: Peticiones de campaña	:Peticiones de campaña

Proposición	Fragmento del Diagrama de Clases
Entity[repl]: Planes_1	:Planes_1
Entity[repl]: Planes_2	:Planes_2
Entity[repl]: Posicionamientos	:Posicionamientos
Entity[repl]: Presupuestos	:Presupuestos
Entity[repl]: Productos	:Productos
Entity[repl]: Programas	:Programas
Entity[repl]: Slots	:Slots
Entity[repl]: Soportes	:Soportes
Entity[repl]: Subcampanías	:Subcampanías
Entity[repl]: Targets	:Targets
Entity[repl]: Facturas a Cliente	:Facturas a Cliente
Entity[repl]: Abonos a Cliente	:Abonos a Cliente

Proposición				Fragmento del Diagrama de Clases
	Entity[repl]: Pases tipo_2	Subs: subs	Entity[repl]: Pases tipo_1	<pre> classDiagram class Pases_tipo_1[":Pases tipo_1"] class Pases_tipo_2[":Pases tipo_2"] Pases_tipo_2 -- > Pases_tipo_1 </pre>
	Entity[repl]: Planes_2	Subs: subs	Entity[repl]: Planes_1	<pre> classDiagram class Planes_1[":Planes_1"] class Planes_2[":Planes_2"] Planes_2 -- > Planes_1 </pre>
	Entity[repl]: Pases tipo_3	Subs: subs	Entity[repl]: Pases tipo_1	<pre> classDiagram class Pases_tipo_1[":Pases tipo_1"] class Pases_tipo_3[":Pases tipo_3"] Pases_tipo_3 -- > Pases_tipo_1 </pre>
	Entity[repl]: Alternativas_2	Subs: Subs	Entity[repl]: Alternativas_1	<pre> classDiagram class Alternativas_1[":Alternativas_1"] class Alternativas_2[":Alternativas_2"] Alternativas_2 -- > Alternativas_1 </pre>
P1	Entity[notrepl]: Cliente	Rel: Realiza	Entity[repl]: Peticiones de campaña	<pre> classDiagram class Clientes[":Clientes"] class Peticiones_de_campaña[":Peticiones de campaña"] Clientes -- Peticiones_de_campaña : realiza </pre>

Proposición				Fragmento del Diagrama de Clases
P2	Entity[notrepl]: Petición de campaña	Rel: Se refiere a	Entity[repl]: Productos	<pre> classDiagram class Productos class Peticones_de_campaña Productos --> Peticones_de_campaña : se refiere a </pre>
P3	Entity[notrepl]: Petición de campaña	Rel: Se refiere a	Entity[repl]: Medios	<pre> classDiagram class Medios class Peticones_de_campaña Medios --> Peticones_de_campaña : se refiere a </pre>
P4	Entity[notrepl]: Medio_1	-Pof: Se le asigna	Statespace: Inversión total	<pre> classDiagram class Medios class Medios { Inversión total } </pre>
P5	Entity[notrepl]: Medio_1	-Pof: Se le asigna	Statespace: Target de planificación_1	<pre> classDiagram class Medios class Medios { Target de Planificación } </pre>
P6	Entity[notrepl]: Medio_1	-Pof: Se le asigna	Statespace: Periodo de campaña	<pre> classDiagram class Medios class Medios { Periodo de campaña } </pre>
P7	Entity[notrepl]: Planificación	-Pof: Posee	Entity[notrepl]: Franja de planificación	<pre> classDiagram class Planificaciones class Franjas_de_planificación Planificaciones o-- Franjas_de_planificación </pre>
P8	Entity[notrepl]: Planificación	-Pof: Posee	Statespace: Periodo de vigencia	<pre> classDiagram class Planificaciones class Planificaciones { Periodo de vigencia } </pre>

Proposición				Fragmento del Diagrama de Clases
P9	Entity[notrepl]: Planificación	-Pof: Posee	Statespace: Target de planificación_2	<pre> classDiagram class Planificaciones { Target de planificación_2 } </pre>
P10	Entity[notrepl]: Planificación	Rel: Posee	Entity[repl]: Pases tipo_1	<pre> classDiagram class Pases_tipo_1 { } class Planificaciones { } Pases_tipo_1 --> Planificaciones : posee </pre>
p11	Entity[notrepl]: Planificación	Rel: Posee	Entity[repl]: Planes_1	<pre> classDiagram class Planes_1 { } class Planificaciones { } Planes_1 --> Planificaciones : posee </pre>
P12	Entity[notrepl]: Planificación	Rel: Posee	Entity[repl]: Alternativas_1	<pre> classDiagram class Alternativas_1 { } class Planificaciones { } Alternativas_1 --> Planificaciones : posee </pre>
P15	Entity[notrepl]: Material	Rel: Asociado a	Entity[notrepl]: Cliente	<pre> classDiagram class Clientes { } class Materiales { } Clientes --> Materiales : asociado a </pre>
P16	Entity[notrepl]: Pase tipo_1	Rel: Se refiere a	Entity[notrepl]: Producto	<pre> classDiagram class Productos { } class Pases_tipo_1 { } Productos --> Pases_tipo_1 : se refiere a </pre>

Proposición				Fragmento del Diagrama de Clases
P17	Entity[notrepl]: Pase tipo_1	Rel: Se refiere a	Entity[repl]: Materiales	<pre> classDiagram class Materiales class Pases_tipo_1 Materiales -- Pases_tipo_1 : se refiere a </pre>
P18	Entity[notrepl]: Pase tipo_1	-Pof: Posee	Statespace: Duración_1	<pre> classDiagram class Pases_tipo_1 Pases_tipo_1 -- Duración_1 </pre>
P19	Entity[notrepl]: Material	Rel: Corresponde con un	Entity[notrepl]: Producto	<pre> classDiagram class Productos class Materiales Productos -- Materiales : corresponde con un </pre>
P20	Entity[notrepl]: Material	Rel: Posee	Entity[notrepl]: Medio_2	<pre> classDiagram class Medios class Materiales Medios -- Materiales : posee </pre>
P21	Entity[notrepl]: Material	-Pof: Posee	Statespace: Duración_2	<pre> classDiagram class Materiales Materiales -- Duración_2 </pre>
P22	Entity[notrepl]: Material	-Pof: Posee	Statespace: Periodo de caducidad	<pre> classDiagram class Materiales Materiales -- Periodo_de_caducidad </pre>
P25	Entity[notrepl]: Pases tipo_2	Rel: Se asigna a	Entity[notrepl]: Slot	<pre> classDiagram class Slots class Pases_tipo_2 Slots -- Pases_tipo_2 : se asigna a </pre>

Proposición				Fragmento del Diagrama de Clases
P26	Entity[notrepl]: Medio	Rel: Tiene	Entity[rep]: Soportes	<pre> classDiagram class Soportes class Medios Soportes -- Medios : tiene </pre>
P27	Entity[notrepl]: Soporte	-Pof: Tiene	Entity[notrepl]:Tarifa base	<pre> classDiagram class Soportes class Tarifa_base Soportes *-- Tarifa_base </pre>
P28	Entity[notrepl]: Plan_1	-Pof: Posee	Statespace: Target de compra	<pre> classDiagram class Planes_1 class Target_de_compra Planes_1 -- Target_de_compra </pre>
P29	Entity[notrepl]: Plan_1	-Pof: Posee	Statespace: Franja de compra	<pre> classDiagram class Planes_1 class Franja_de_compra Planes_1 -- Franja_de_compra </pre>
P30	Entity[notrepl]: Plan_1	-Pof: Posee	Statespace: Periodo de predicción de audiencia	<pre> classDiagram class Planes_1 class Periodo_de_prediccion_de_audiencia Planes_1 -- Periodo_de_prediccion_de_audiencia </pre>
P31	Entity[notrepl]: Plan_1	-Pof: Posee	Statespace: Deflactación de compra	<pre> classDiagram class Planes_1 class Deflactacion_de_compra Planes_1 -- Deflactacion_de_compra </pre>
P32	Entity[notrepl]: Plan_1	-Pof: Posee	Statespace: Deflactación de venta	<pre> classDiagram class Planes_1 class Deflactacion_de_venta Planes_1 -- Deflactacion_de_venta </pre>

Proposición				Fragmento del Diagrama de Clases
P33	Entity[notrepl]: Plan_1	Rel: Posee	p25	<pre> classDiagram class Slots[":Slots"] class Pases_tipo_2[":Pases tipo_2"] class Planes_1[":Planes_1"] Slots -- Pases_tipo_2 Pases_tipo_2 -- Planes_1 : se asigna a </pre>
P34	Entity[notrepl]: Plan_1	Rel: Está asociado a	Entity[notrepl]: Negociación	<pre> classDiagram class Negociacion[":Negociación"] class Planes_1[":Planes_1"] Negociacion -- Planes_1 : está asociado a </pre>
P35	Entity[notrepl]: Plan_1	Rel: Está asociado a	Entity[notrepl]: Modelo de previsión	<pre> classDiagram class Modelo_de_prevision[":Modelo de previsión"] class Planes_1[":Planes_1"] Modelo_de_prevision -- Planes_1 : está asociado a </pre>
P36	Entity[notrepl]: Plan_1	-pof: Asociado a	Statespace: Tipo de estimación de audiencia	<pre> classDiagram class Planes_1[":Planes_1"] Planes_1 -- "Tipo de estimación de audiencia" </pre>
P37	Entity[notrepl]: Modelo de previsión	-pof: Pertenece a	Statespace: Tipo	<pre> classDiagram class Modelo_de_prevision[":Modelo de previsión"] Modelo_de_prevision -- "Tipo" </pre>
P51	Entity[notrepl]: Plan_1	-Pof: Posee	Process: Cálculo de coste	<pre> classDiagram class Planes_1[":Planes_1"] Planes_1 -- "Cálculo de coste" </pre>

Proposición				Fragmento del Diagrama de Clases
P52	Entity[notrepl]: Plan_1	-Pof: Posee	Process: Cálculo de audiencia	<pre> classDiagram class Planes_1 { Cálculo de audiencia() } </pre>
P53	p25	-Pof: Puede tener	Statespace: Posición	<pre> classDiagram class Slots { } class Pases_tipo_2 { } class Posición { } Slots --> Posición : se asigna a Pases_tipo_2 --> Posición : se asigna a </pre>
P54	p25	-Pof: Puede tener	Statespace: Tipo de posición	<pre> classDiagram class Slots { } class Pases_tipo_2 { } class Tipo_de_posición { } Slots --> Tipo_de_posición : se asigna a Pases_tipo_2 --> Tipo_de_posición : se asigna a </pre>
P63	Entity[notrepl]: Alternativa_3	-Pof: Contiene	Entity[repl]: Pases tipo_3	<pre> classDiagram class Pases_tipo_3 { } class Alternativas_1 { } Pases_tipo_3 o-- Alternativas_1 </pre>
P64	Entity[notrepl]: Alternativa_3	-Pof: Contiene	Entity[repl]: Planes_2	<pre> classDiagram class Pases_tipo_3 { } class Alternativas_1 { } Pases_tipo_3 o-- Alternativas_1 </pre>
P65	Entity[notrepl]: Franja	-Pof: Contiene	Entity[repl]: Programas	<pre> classDiagram class Programas { } class Franjas { } Programas o-- Franjas </pre>

Proposición			Fragmento del Diagrama de Clases	
P66	Entity[notrepl]: Franja	-Pof: Contiene	Entity[repl]: Slots	<pre> classDiagram class Slots class Franjas Slots o-- Franjas </pre>
P67	Entity[notrepl]: Franja	-Pof: Tiene	Statespace: Coste_1	<pre> classDiagram class Franjas { Coste_1 } </pre>
P68	Entity[notrepl]: Programa_1	-Pof: Tiene	Statespace: Coste_2	<pre> classDiagram class Programas { Coste_2 } </pre>
P69	Entity[notrepl]: Slot	-Pof: Tiene	Statespace: Coste_3	<pre> classDiagram class Slots { Coste_3 } </pre>
P70	Entity[notrepl]: Franja	-Pof: Corresponde con un	Statespace: Día	<pre> classDiagram class Franjas { Día } </pre>
P71	Entity[notrepl]: Franja	-Pof: Contiene	Entity[repl]: Posicionamientos	<pre> classDiagram class Posicionamientos class Franjas Posicionamientos o-- Franjas </pre>
P72	Entity[notrepl]: Franja	-Pof: Puede tener	Statespace: Coste_4	<pre> classDiagram class Franjas { Coste_4 } </pre>
P73	Entity[notrepl]: Franja	-Pof: Puede tener	Statespace: Descuento_1	<pre> classDiagram class Franjas { Descuento_1 } </pre>

Proposición				Fragmento del Diagrama de Clases
P74	Entity[notrepl]: Programa	-Pof: Puede tener	Statespace: Coste_5	<pre> classDiagram class Programas { Coste_5 } </pre>
P75	Entity[notrepl]: Programa	-Pof: Puede tener	Statespace: Descuento_2	<pre> classDiagram class Programas { Descuento_2 } </pre>
P76	Entity[notrepl]: Programa	-Pof: Puede tener	Statespace: Coste_6	<pre> classDiagram class Programas { Coste_6 } </pre>
P77	Entity[notrepl]: Programa	-Pof: Puede tener	Statespace: Descuento_3	<pre> classDiagram class Programas { Descuento_3 } </pre>
P78	Entity[notrepl]: Posicionamiento	-Pof: Se refiere a	Statespace: Posición_2	<pre> classDiagram class Posicionamientos { Posición_2 } </pre>
P79	Entity[notrepl]: Posicionamiento	-Pof: Se refiere a	Statespace: Tipo de posición_2	<pre> classDiagram class Posicionamientos { Tipo de posición } </pre>
P88	Entity[notrepl]: Modelo de previsión	-Pof: Contiene	Entity[repl]: Targets	<pre> classDiagram class ModeloDePrevision { +Targets } </pre>
P89	Entity[notrepl]: Target	Rel: Asociado a	Entity[repl]: Slots	<pre> classDiagram class Slots class Target Slots -- Target : asociado a </pre>

Proposición				Fragmento del Diagrama de Clases
P90	Entity[notrepl]: Slot	-Pof: Tiene asociada	Statespace: Audiencia	<pre> classDiagram class Slots[":Slots"] class Audiencia Slots -- Audiencia </pre>
P91	Entity[notrepl]: Campaña	-Pof: Compuesta por	Entity[repl]: Subcampañas	<pre> classDiagram class Subcampañas[":Subcampañas"] class Campañas[":Campañas"] Campañas *-- Subcampañas </pre>
P92	Entity[notrepl]: Campaña	Rel: Relacionada con	Entity[notrepl]: Petición de campaña	<pre> classDiagram class Campaña[":Campaña"] class Petición_de_campaña[":Petición de campaña"] Campaña -- Petición_de_campaña : relacionada con </pre>
P93	Entity[notrepl]: Subcampaña	-Pof: Compuesta por	Entity[repl]: Presupuestos	<pre> classDiagram class Subcampañas[":Subcampañas"] class Presupuestos[":Presupuestos"] Subcampañas *-- Presupuestos </pre>
P94	Entity[notrepl]: Presupuesto	-Pof: Contiene	Entity[notrepl]: Alternativas_2	<pre> classDiagram class Presupuestos[":Presupuestos"] class Alternativas_2[":Alternativas_2"] Presupuestos *-- Alternativas_2 </pre>
P97 P25		-Pof: posee	Statespace: Estado	<pre> classDiagram class Pases_tipo_2[":Pases tipo_2"] class Estado Pases_tipo_2 -- Estado : se asigna a </pre>

Tabla 8.24. Fragmentos del Diagrama de Clases

8.4. MODELO DERIVADO

El diagrama resultado de unir todos los fragmentos de la tabla 8.20 se muestra en la figura 8.12. Este diagrama es el resultado final de la aplicación de MAON para el caso de prueba de MP.

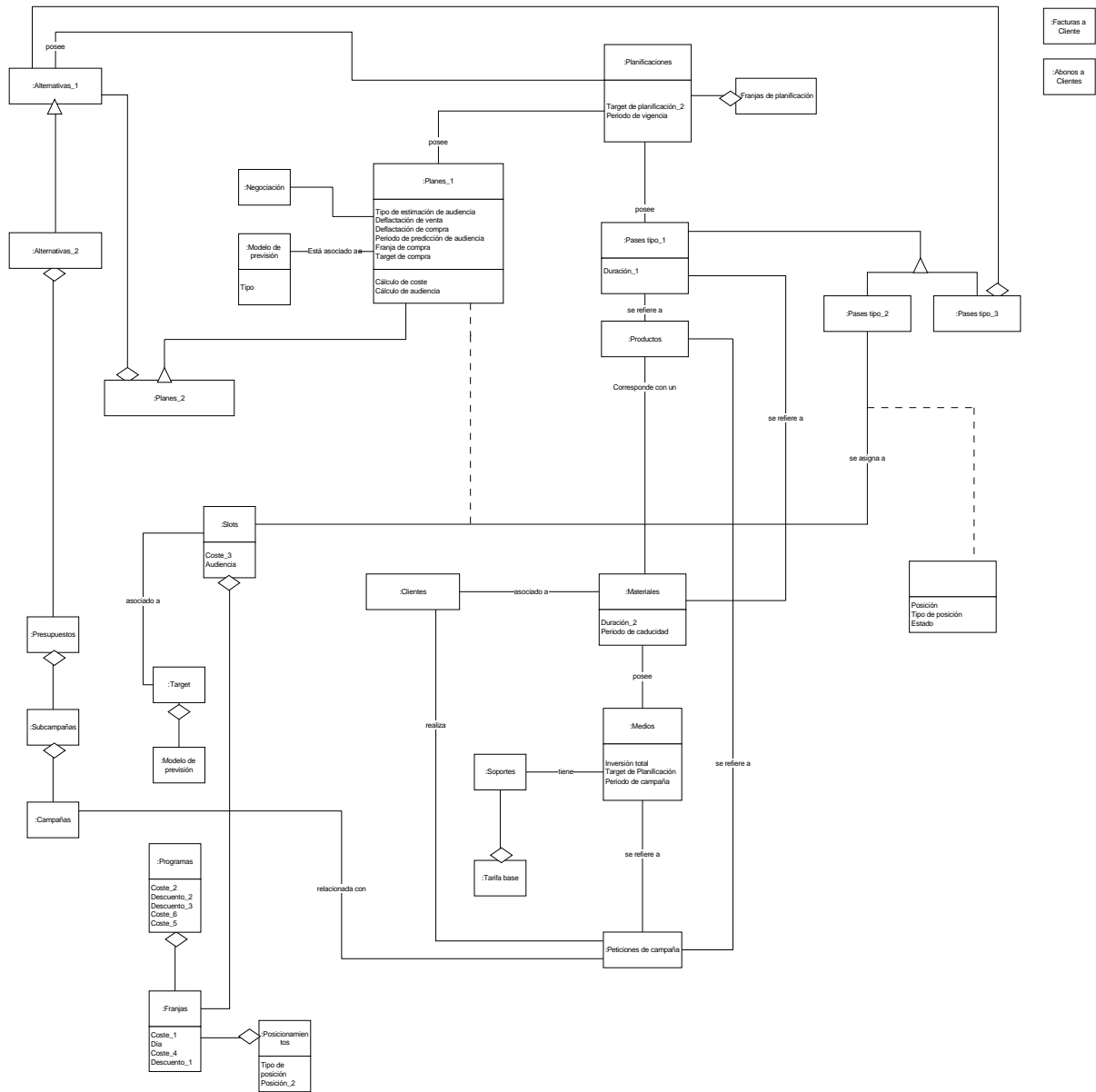


Figura 8.12. Diagrama de Clases confeccionado a partir de los fragmentos de la tabla 8.24

PARTE IV

CONCLUSIONES Y FUTURAS LÍNEAS

9. Conclusiones

En este capítulo se resumen las principales contribuciones de esta investigación y analiza cómo dichas contribuciones han permitido la consecución de los objetivos planteados al comienzo de esta investigación. Así mismo, se indica en qué medida las distintas contribuciones de este trabajo han sido presentadas en diversas publicaciones. Finalmente se presentan las futuras líneas de investigación derivadas de este trabajo.

9.1. PRINCIPALES CONTRIBUCIONES DE LA INVESTIGACIÓN

Tal como se ha especificado en el capítulo 3, la investigación llevada a cabo en esta Tesis Doctoral, tiene como finalidad redefinir la tarea de Análisis de tal forma que:

1. Se realice siempre una clara diferenciación entre los modelos conceptuales y los modelos del sistema en cualquier aproximación de desarrollo.
2. Exista siempre la posibilidad de cambiar la aproximación de desarrollo utilizada en un proyecto si se comprueba que ésta es inadecuada.

Para satisfacer esta finalidad, este trabajo de investigación ha desarrollado **el método de pre-Análisis MAON que ha de aplicarse antes de las tareas clásicas de Análisis, realizadas por las aproximaciones de desarrollo clásicas, y que permite representar la necesidad del usuario de forma independiente del enfoque de desarrollo y estudiar dicha necesidad para derivar de manera formalizada los modelos de desarrollo más adecuados para dicha necesidad.** Esta contribución general se puede descomponer en dos contribuciones más específicas: un modelo conceptual independiente del enfoque de desarrollo, y un método de modelización que permita construir este modelo y derivar el modelo conceptual clásico correspondiente a la aproximación de desarrollo más adecuada.

9.1.1. MODELO CONCEPTUAL GENÉRICO

Tras realizar en el capítulo 2 un extenso estudio sobre el estado del arte en el tema de la modelización conceptual, se ha comprobado que los modelos conceptuales utilizados en la actualidad están fuertemente orientados a definir cómo se implementará el sistema software correspondiente, en lugar de a favorecer la comprensión del dominio. Por otra parte, la falta de procedimientos de traducción entre modelos conceptuales conlleva que la utilización de un modelo conceptual durante el proceso de Análisis determina prácticamente el desarrollo posterior. Es decir, el diseño del sistema se realizará muy posiblemente con el enfoque de desarrollo correspondiente al modelo conceptual utilizado, aunque el desarrollador se percate de que dicho enfoque no es el más adecuado.

Con el fin de resolver este problema, esta Tesis **plantea la utilización de un MCG que permita recoger la información sobre un problema o necesidad planteada por el usuario, con el único propósito de entendimiento, con independencia del enfoque de desarrollo a seguir para resolver la necesidad planteada por el usuario.**

Por lo tanto, el MCG propuesto tiene una capacidad de representación amplia con el fin de representar la mayor cantidad de información posible. Para este fin, el MCG propuesto está formado por tres formalismos de representación complementarios (Mapa de conceptos, Diccionario de Identificación y Diccionario de Descripción), donde complementarios significa que cada formalismo contribuye a la representación de la información del problema del usuario a distintos niveles y que la información de un formalismo puede migrar a otro sin perder información. Estos modelos han sido detallados en el capítulo 5, y como se habrá observado, principalmente se derivan de la psicología ya que proporcionan una representación suave y flexible de los distintos tipos de información sobre un problema. Esto es posible gracias a una categorización muy débil de la información, que permite representar tanto elementos estáticos como dinámicos sobre un problema sin tener que clasificarlos a priori. Esto permite representar una amplia variedad de información.

9.1.2. MÉTODO DE PRE-ANÁLISIS: MAON

El método de pre-Análisis presentado en este trabajo de Tesis MAON (Método de Análisis Orientado a la Necesidad) consta de dos fases: Una primera fase de Análisis Orientado al Problema y una fase siguiente de Análisis Orientado a la Solución, tal como se muestra en la figura 9.1.

Cada una de las fases anteriores ha sido descrita en detalle en los capítulos 5 y 6 respectivamente, indicando los productos de entrada y salida de cada una, y el proceso a seguir para transformar los productos de entrada en los productos de salida correspondientes.

La fase de **Análisis Orientado al Problema contiene un conjunto de tareas** (Análisis Preliminar y Análisis Exhaustivo) **que permiten representar la necesidad del usuario en el MCG.** Este modelo posee, como se ha indicado ya, una alta capacidad de representación, lo que lo hace apto para modelar una gran diversidad de dominios. Adicionalmente, las ligaduras computacionales han sido minimizadas o eliminadas de forma absoluta, por lo que el modelado está libre de cualquier consideración computacional que predetermine un tipo de diseño determinado.

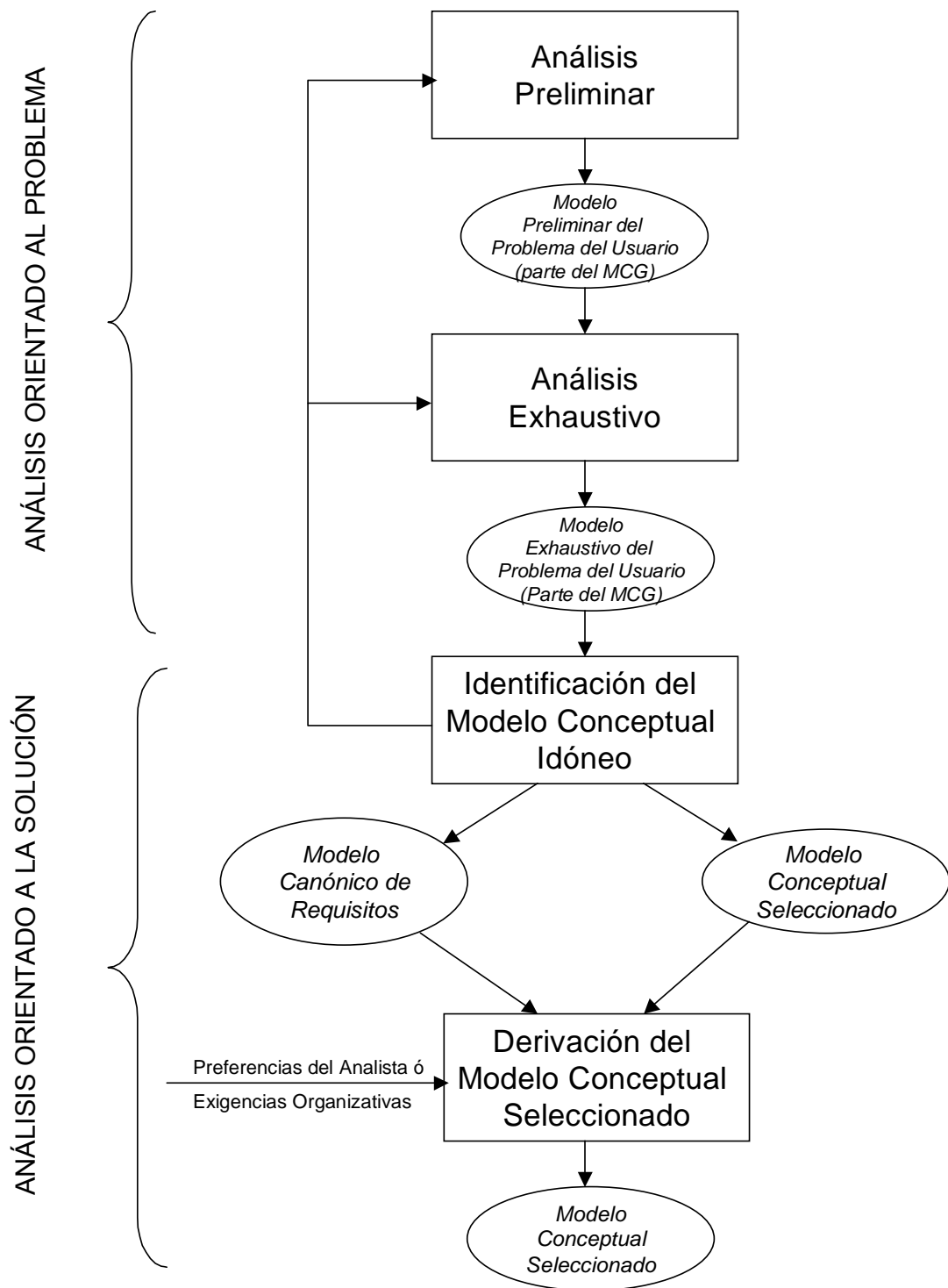


Figura 9.1. Tareas y productos de MAON

La segunda fase, de Análisis Orientado a la Solución, tiene como objetivo permitir que, a partir de la información contenida en el MCG, sea posible proseguir con las restantes actividades de desarrollo. Para ello, durante esta fase se realiza una interpretación del MCG, la cual consiste en la adscripción de

determinadas etiquetas a los distintos constructores del MCG. Estas etiquetas representan aspectos típicos del diseño de software, tales como procesos, objetos o mensajes, que permiten determinar el enfoque computacional más adecuado para resolver la necesidad planteada por el usuario.

Con la finalidad de poder seleccionar la mejor aproximación de desarrollo, MAON introduce dos técnicas de gran relevancia: La Técnica IMCI (Técnica de Identificación del Modelo Conceptual Idóneo) y la Técnica DMCS (Técnica de Derivación del Modelo Conceptual Seleccionado).

Los resultados de la interpretación son empleados por **la Técnica IMCI que permite identificar qué aproximación de desarrollo es más adecuada para la necesidad del usuario representada en el MCG**, y poder proseguir con la construcción del sistema software. La Técnica IMCI se basa en la comparación entre los resultados de la interpretación y las capacidades expresivas de los modelos conceptuales. Dicha comparación permite adscribir a cada modelo una adecuación determinada, la más alta de las cuales determinará el modelo conceptual más adecuado (esto es, el Modelo Idóneo) y, por ende, la aproximación de desarrollo más adecuada.

Finalmente, **la Técnica DMCS tiene como objetivo derivar el Modelo Conceptual Idóneo, esto es el Modelo Conceptual clásico más apropiado para la necesidad representada en el MCG**, a partir de los resultados de la interpretación. Para ello, la Técnica DMCS utiliza un catálogo que permite relacionar fragmentos del Modelo Conceptual Genérico con fragmentos del Modelo Idóneo. La unión de todos los fragmentos resulta en el Modelo conceptual requerido.

Dos son las ventajas principales de la interpretación (y de las Técnicas IMCI y DMCS) frente a la utilización de modelos conceptuales clásicos, esto es, modelos conceptuales con ligaduras de diseño. Por una parte, la lectura en términos computacionales del modelo se retrasa hasta que el estudio del problema está finalizado, y por otra, la interpretación permite derivar distintos tipos de diseño. En otras palabras, los modelos conceptuales clásicos operan al modo de Procusto, acortando o alargando a los invitados –el problema- en función de la cama –el modelo-. MAON permite que cada invitado tenga la cama que le resulte más cómoda, pero además la cama se puede pedir por catálogo después de cenar para que llegue antes de la noche.

La validación realizada en el capítulo 7 ha demostrado la efectividad de la aproximación propuesta. MAON es, por lo tanto, un método que permite modelar todos aquellos problemas que admiten tratamiento bajo las aproximaciones estructurada, orientada a objetos y de tiempo real, permitiendo identificar la aproximación más adecuada en cada caso y derivando los modelos conceptuales correspondientes a la aproximación de desarrollo seleccionada.

9.2. ANÁLISIS DE LA CONSECUCCIÓN DE OBJETIVOS

Tal como se indica en el capítulo 3, este trabajo de investigación se ha planteado para satisfacer el siguiente objetivo:

El desarrollo de un método de pre-Análisis, como tarea previa en el proceso de Análisis, durante el cual será posible realizar un modelo conceptual con capacidad de representación suficiente y sin ligaduras de diseño. Este método permitirá así mismo derivar los modelos propios de la aproximación de desarrollo que se considere más adecuada para el problema en cuestión.

Para la consecución de este objetivo se ha trabajado con la hipótesis general siguiente:

HG: Es viable definir un método de pre-análisis **capaz de determinar la aproximación de desarrollo más adecuada** para la necesidad del usuario que el sistema software pretende resolver ,y **permite continuar el desarrollo con las aproximaciones tradicionales:** Estructurada, Orientada a Objetos y de Tiempo Real.

Esta hipótesis se ha satisfecho mediante la definición del método MAON que permite modelizar un problema con independencia del enfoque de solución elegido para solucionarlo, estudiar las características de dicha modelización para determinar el enfoque de desarrollo más adecuado (Estructurado, Orientado a Objetos y de Tiempo Real), y derivar los modelos propios de dicho enfoque.

No obstante, con objeto de validar esta hipótesis, su formulación ha sido descompuesta en tres sub-hipótesis más concretas:

SH1: El método de pre-Análisis es capaz de **determinar qué aproximación de desarrollo** (estructurada, orientada a objetos, de tiempo real) **es la más adecuada** para proseguir el proceso de desarrollo.

Para satisfacer esta hipótesis se ha propuesto un formalismo de representación (el MCG) así como un método para construirlo dentro de MAON (Análisis Orientado al Problema) que permite recoger de una forma flexible la información necesaria sobre el dominio de un problema o necesidad del usuario. Posteriormente se ha propuesto la técnica IMCI para identificar la aproximación de desarrollo más adecuada, utilizando una métrica de adecuación, que se basa en calcular la proporción de constructores propios de

cada aproximación de desarrollo que se pueden utilizar para representar los componentes del MCG.

Esta hipótesis ha sido validada para cada una de las aproximaciones mencionadas tal como se especifica en el capítulo 7, en el que para distintos problemas se ha comparado la aproximación elegida utilizando el método propuesto en la Tesis y la aproximación elegida por un conjunto de expertos.

SH2: El método de pre-Análisis **permite derivar los modelos utilizados por las aproximaciones de desarrollo** existentes en la actualidad (estructurada, orientada a objetos y de tiempo real).

Una parte fundamental del método de pre-Análisis MAON está constituida por el procedimiento de derivación del modelo conceptual más adecuado para el problema representado en el MCG propiciado por la técnica DMCI. Este procedimiento utiliza un enfoque algorítmico y formalizado para derivar los modelos conceptuales resultantes, mediante el uso de unas Tablas de Derivación que se muestran en el Anexo C, para cada una de las aproximaciones indicadas.

Esta hipótesis se ha validado en el capítulo 7 con los distintos casos prácticos derivando los modelos de las aproximaciones estructurada, orientada a objetos y en tiempo real.

SH3: El método de pre-Análisis **tiene la capacidad de representación suficiente** para ser utilizado en las circunstancias en las que serían utilizables las aproximaciones estructuradas, orientadas a objetos o de tiempo real.

Para satisfacer esta hipótesis se ha propuesto como primer paso del método de pre-Análisis MAON la modelización de la necesidad del usuario empleando un formalismo de representación materializado en el MCG. Este modelo derivado principalmente de modelos utilizados en la psicología permite representar tanto información estática (datos, proposiciones, relaciones, etc.) como información dinámica (funciones, eventos, etc.), pero sin forzar al desarrollador a decidir o hacer distinciones entre ambos tipos de información. Como se ha indicado ya, el MCG es así un modelo flexible y blando que permite representar una amplia variedad de información sin restringir o limitar la información representada.

Esta hipótesis ha sido validada en el capítulo 7 aplicándose MAON a distintos problemas propios de las aproximaciones anteriores.

9.3. RESULTADOS DE LA INVESTIGACIÓN

REVISTAS INTERNACIONALES

O. Dieste, M. Genero, N. Juristo, A. Moreno, J.L. Maté. A Conceptual Model Completely Independent of the Implementation Paradigm. **Journal of System and Software**. En prensa. 2003.

El principal objetivo de este artículo es mostrar cómo MAON puede llegar a generar modelos conceptuales clásicos a partir del Modelo Conceptual Genérico. Para ello, se realiza una descripción somera de la totalidad de los componentes de MAON para, posteriormente, focalizarse en la utilización de las técnicas IMCI y DMCS.

CAPÍTULOS DE LIBRO

O. Dieste, N. Juristo, Ana M. Moreno, J. Pazos, A. Sierra. Conceptual Modelling in Software Engineering and Knowledge Engineering: Concepts, Techniques And Trends. In **Handbook of Software and Knowledge Engineering**. World Scientific Publishing. 2001. pp 763-766.

En este artículo, el objetivo principal no ha sido presentar ningún resultado concreto del presente trabajo de Tesis, sino exponer los argumentos que han llevado a la realización del trabajo de Tesis. Concretamente, en este artículo se realiza una clasificación y descripción de los distintos tipos de modelos conceptuales utilizados tanto en la Ingeniería del Software como en la Ingeniería del Conocimiento para, posteriormente, señalar de forma general sus carencias y, debido a ellas, la necesidad de definir modelos conceptuales distintos, los cuales poseen una orientación más marcada hacia la necesidad del usuario.

CONGRESOS INTERNACIONALES

O. Dieste, M. Genero, A. Moreno. A Problem-Oriented Conceptual Model. **Eighth CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD'03)**. Velden, Austria, 2003.

En el presente artículo, se describe en detalle el Modelo Conceptual Genérico utilizado en MAON. El objetivo es mostrar cómo el Modelo Conceptual Genérico permite modelar toda una diversidad de hechos y situaciones independientemente de cualquier sesgo computacional. Asimismo, se indica que dicho modelo puede ser utilizado para obtener, posteriormente, un modelo conceptual clásico.

O. Dieste, M. Genero. A Problem-Oriented Analysis Method Capable of Identify and Derive the Best-Suited Conceptual Model for the User Need. **International Conference on Software Engineering Research and Applications (SERA '03)**. San Francisco, USA, 2003.

El objetivo del presente artículo es describir cómo mediante la métrica empleada en MAON, es posible identificar el mejor modelo conceptual para un problema determinado. Por ello, este artículo se focaliza en la descripción de las técnicas de identificación (IMCI) y Derivación (DMCS) de MAON.

O. Dieste. Development-Paradigm Independent Conceptual Models. **International Conference on Software Engineering & Knowledge Engineering (SEKE'01)**. Buenos Aires, Argentina, 2001.

Este artículo, elegido como "best paper" de la conferencia, realiza una exposición somera de MAON, tanto en la vertiente del proceso como de los productos (Modelo Conceptual Genérico) empleado. Tras su publicación, se puede considerar que el modelo está maduro y, por lo tanto, listo para su presentación en revistas y congresos internacionales relevantes, así como para la defensa del trabajo de Tesis.

O. Dieste, Ana M. Moreno. Generic Conceptual Models. **International Conference on Software Engineering, Networking and Parallel Distributed Computing (SNPD'01)**. Nagoya, Japon, 2001.

En este artículo, se exponen por primera vez los principales componentes, esto es, el proceso y los modelos, de MAON. El objetivo principal de este artículo ha sido obtener críticas y propuestas de mejora para MAON por parte de la comunidad de la Ingeniería del Software, las cuales pudieran ser revertidas en el capítulo de Resolución.

O. Dieste, Ana Moreno. A Proposal for a Requirements Engineering Process focused on the User Need. **International Conference on Software Engineering, Networking and Parallel Distributed Computing (SNPD'00)**. Reims , Francia, 2000

El principal objetivo de este artículo es presentar ante la comunidad un bosquejo de solución a los problemas identificados en el Estado de la Cuestión. Para ello, después de realizar una breve revisión de los modelos conceptuales y sus carencias, se propone un proceso de pre-análisis que utiliza Modelos Conceptuales Genéricos. Este artículo refleja las Hipótesis de Trabajo de esta Tesis, aunque entre su fecha de publicación y el momento actual el esquema propuesto ha sufrido profundas mejoras.

O. Dieste, A. Moreno. On the Capability of Analysis Techniques in Requirements Engineering. **International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD'99)**. Alemania, Junio, 1999.

En este artículo se realiza una revisión de las carencias de los modelos conceptuales clásicos en lo referente a su capacidad para modelar la necesidad del usuario, lo que corresponde al Estado de la Cuestión del presente trabajo de Tesis.

CONGRESOS NACIONALES O IBEROAMERICANOS

O. Dieste. Estudio de las carencias de los métodos tradicionales de Modelización Conceptual. **Jornadas de Ingeniería del Software y Bases de Datos (IV JISBD)**. Cáceres, 1999.

En el presente artículo, aceptado como "short paper", se realizó una descripción del problema planteado en el presente trabajo de Tesis y de las hipótesis generales para su resolución.

9.4. FUTURAS LÍNEAS DE INVESTIGACIÓN

A juicio del autor, uno de los aspectos más relevantes del presente trabajo de Tesis es la aritmetización, aunque ello pueda haber quedado oculto entre la masa de procesos, técnicas y procedimientos. Parafraseando a Lord Kelvin, "en la ciencia sólo se conoce algo cuando se lo puede medir y cuantificar", y ello (conocer y medir) es algo que en el presente trabajo de Tesis se ha intentado, que no conseguido realmente, hacer. A continuación, se presentan algunos trabajos futuros que pueden contribuir a la deseada cuantificación.

9.4.1. DEFINICIÓN DE NUEVAS MÉTRICAS DE ADECUACIÓN

La Métrica de Adecuación abre un posible camino hacia un mejor conocimiento de qué son, y qué pueden hacer, los modelos conceptuales. Si en el presente trabajo se ha definido que un modelo es Idóneo cuando posee la Adecuación más alta, cabe preguntarse ¿cuánto de alta? La Adecuación ha sido definida de forma relativa a un problema, pero no se ha explorado cuál es su límite superior. Ello posee gran importancia a la hora de decidir la Idoneidad de un modelo y, adicionalmente, podría ser igualmente importante para decidir acerca de la completitud de los modelos realizados, sean estos genéricos, sean éstos clásicos. El autor es plenamente consciente de que la Métrica de Adecuación, por sí sola, no proveerá todos los beneficios anteriormente enunciados, pero si que dicha métrica puede inspirar enfoques para tratar cuantitativamente problemas a los que, hasta ahora, la informática se ha enfrentado de forma mayoritariamente cualitativa.

9.4.2. REFINAMIENTO DEL MODELO CONCEPTUAL OBTENIDO CON LA TÉCNICA DMCS

Como se ha indicado en el capítulo 6, los modelos de desarrollo obtenidos con la técnica DMCS son una primera aproximación que ha de ser refinada por el ingeniero software, en función de los distintos parámetros de calidad que desee potenciar. La calidad de un modelo conceptual, según ISO 9126 (ISO, 1999) está compuesta por distintas características tales como funcionalidad, mantenibilidad, portabilidad, fiabilidad, eficiencia y usabilidad.

Existen diversos estudios [Genero et al, en prensa] [Piattini et al., 2001] que proporcionan métricas determinadas para evaluar estos atributos en los modelos conceptuales. Así una posible línea futura de investigación podría estar dirigida a proporcionar un conjunto de estas métricas y guiar su aplicación para dirigir el posible refinamiento de los modelos conceptuales derivados de MAON.

9.4.3. AMPLIACIÓN DEL ALCANCE DEL PROBLEMA DE INVESTIGACIÓN

Otro aspecto importante que debería ser explorado es la aplicación del MCG a otras disciplinas de la informática. Entre ellas, la más importante es, sin duda, la Ingeniería del Conocimiento. Un Mapa de

Conceptos puede entenderse muy fácilmente en términos de una red semántica o, incluso, en términos de un grafo conceptual, aunque los principios subyacentes en este último sean muy distintos. Y aunque es una suposición, es posible que todo el campo de la Ingeniería Ontológica y MAON descansen en principios fundacionales similares. Si ello es así, ¿podría existir algún medio para aproximar las representaciones de la Ingeniería del Conocimiento y MAON? La exportación de MAON a la Ingeniería del Conocimiento es, por lo tanto, un tema de estudio a considerar.

Otra posible vía de investigación futura que se deriva, directamente, de la aplicación de MAON a la Ingeniería del Conocimiento. Si MAON, o algún método similar o basado en los mismos principios, se pudiera utilizar para desarrollar Sistemas Basados en el Conocimiento, se habría cerrado un ciclo: un método de Análisis (o Conceptualización) común para la Ingeniería del Software y la Ingeniería del Conocimiento. Ello permitiría confeccionar una metodología común que pudiera ser aplicada a cualquier problema y que, únicamente en fase de diseño, llevara a la construcción de uno u otro tipo de sistema.

11. Bibliografía

- [Abrial, 1974] Abrial, J.R.; **Data Semantics**; Data Base Management, North-Holland, 1974.
- [Adelson et al., 1985] Adelson, B., Soloway, E.; **The role of domain experience in software design**; IEEE Transactions on Software Engineering, SE-11, 1985.
- [Adelson et al., 1988] Adelson, B., Soloway, E.; **A model of software design**; *In The nature of Expertise*, Chi, M.T.H. (ed), Lawrence Erlbaum Associates, 1988.
- [Alford, 1977] Alford, M.; **A Requirements Engineering Methodology for Real-Time Processing Requirements**. IEEE Transactions on Software Engineering, Vol SE-3, No. 1, 1977.
- [Alford, 1985] Alford, M.; **SREM at the Age of Eight: The Distributed Computing Design System**; *In System and Software Requirements Engineering*, R.H. Thayer, M. Dorfman (Eds.), IEEE Computer Society Press, 1990.
- [Anderson, 2000] Anderson, J.R.; **Cognitive Psychology and its Implications**; Worth Publishers, 2000.
- [ANSI, 1975] ANSI/X3/SPARC; **Study Group on Data Base Management Systems: Interim Report 75-02-08**; ACM SIGMOD Bulletin, Vol. 32, No. 5, 1989.

- [Avison et al., 1995] Avison, D.E., Fitzgerald, G.; **Information Systems Development: Methodologies, Techniques and Tools**; McGraw-Hill, 1995.
- [Baber, 1997] Baber, R.L.; **Comparison of Electrical "Engineering" of Heaviside's Times and Software "Engineering" of Our Times**; Annals of the History of Computing, Vol. 19, No. 4, 1997.
- [Bachman, 1969] Bachman, C.W.; **Data Structure Diagrams**; Data Bases, Vol. 1, No. 2, 1969.
- [Batini et al., 1986] Batini, C., Lenzerini, M., Navathe, S.B.; **A Comparative Analysis of Methodologies for Database Schema Integration**; ACM Computing Surveys, Vol. 18, No. 4, 1986.
- [Batini et al., 1992] Batini, C., Ceri, S., Navathe, S.B.; **Conceptual Database Design**; Benjamin/Cummings, 1992.
- [Beringer, 1995] Beringer, D.; **The Model Architecture Frame: Quality Management in a Multi Method Environment**; Proceedings of the SQM'95, 1995.
- [Beringer, 1996] Beringer, D.; **The Goals of the Analysis Model**; Technical Report 96/216, 1996.
- [Blum, 1996] Blum, B.I.; **Beyond Programming: To a New Era of Design**. Oxford University Press, 1996.
- [Bonfatti et al., 1994] Bonfatti, F., Morani, P.D.; **Towards a General Purpose Approach to Object-Oriented Analysis**. In *Lecture Notes in Computer Science. Object Oriented Methodologies and Systems*, E. Bertino and S. Urban (Eds.), International Symposium ISOOM'94, 1994.
- [Booch, 1991] Booch, G.; **Object-Oriented Design with Applications**; Benjamin Cummings, 1991.
- [Brodie et al., 1982] Brodie, M.L., Silva, E.; **Active and Passive Component Modelling: ACM/PCM**; In *IFIP WG8.1 Working Conference on Information Systems Design Methodologies: A Comparative Review*, T.W. Olle, H.G. Sol and A.A. Verryn-Stuart (Eds.), North-Holland, 1982.
- [Caine et al, 1975] Caine, S., Gordon, E.; **A Tool for Software Design**, in *Proc. of the AFIPS National Computer Conference*; Vol 44, AFIPS Press, 1975.
- [Champeaux et al., 1993] Champeaux, D., Lea, D., Faure, P.; **Object Oriented System Development**; Addison Wesley, 1993.
- [Chen, 1976] Chen, P.; **The Entity-relationship Model: Towards a Unified View of Data**; *ACM Transactions on Database Systems*, Vol.1, No. 1, 1977.
- [Chen, 1983] Chen, P.; **English Sentence Structure and Entity Relationship Diagrams**; *International Journal of Information Sciences*, Vol. 29, 1983.

- [Chen, 1990] Chen, P.; **Entity-relationship Approach to Data Modeling**. In *System and Software Requirements Engineering*, Thayer RH, Dorfman M (eds). IEEE. Computer Society Press, 1990.
- [Coad et al., 1990] Coad, P., Yourdon, E.; **Object Oriented Analysis**; Yourdon Press, 1990.
- [Coleman et al., 1994] Coleman, D., Arnold, P., Bodoff, S., Dollin, C.; **Object-Oriented Development: The FUSION Method**; Prentice Hall, 1994.
- [CSI, 2000] **METRICA V.3**; Consejo Superior de Informática, Ministerio de Administraciones Publicas, <http://www.map.es/csi/metrica3/>, 2000.
- [Czejdo et al., 1990] Czejdo, B., Elmasri, R., Rusinkiewicz, M., Embley, D.; **A Graphical Data Manipulation Language for an Extended Entity-Relationship Model**; IEEE Computer, Vol. 23, No. 3, 1990.
- [Daniels, 2002] Daniels, J; **Modeling with a sense of purpose**; IEEE Software, Vol. 19, No. 1, 2002.
- [Dardenne et al., 1993] Dardenne, A., Lamsweerde, A., Fickas, S.; **Goal-directed Requirements Adquisition**; Science of Computer Programming, Vol. 20, 1993.
- [Davies, 1991] Davies, S.P.; **Characterizing the program design activity: neither strictly top-down nor globally opportunistic**; Behavioral Information Technology, No. 10, 1991.
- [Davis et al., 1997] Davis, A.M., Jordan, K., Nakajima, T.; **Elements Underlying the Specification of Requirements**; Annals of Software Engineering, Vol. 3, Baltzer Science Publishers, 1997.
- [Davis, 1978] Davis, A.M.; **Requirements Language Processing for the Effective Testing of Real-Time Software**; ACM Software Engineering Notes, Vol. 3, No. 5, 1978.
- [Davis, 1988] Davis, A.M.; **A Taxonomy for the Early Stages of the Software Development Life Cycle**; Journal of Systems and Software, No. 8, 1988.
- [Davis, 1993] Davis, A.M.; **Software Requirements: Objects, Functions and States**; Prentice-Hall International, 1993.
- [Dayal et al., 1984] Dayal, U., Hwang, H.Y.; **View Definition and Generalization for Database Integration in a Multidatabase System**; IEEE Transactions on Software Engineering, Vol. 10, No. 6, 1984.
- [Deaño, 1999] Deaño, A.; **Introducción a la Lógica Formal**; Alianza, 1999.
- [DeMarco, 1979] DeMarco, T.; **Structured Analysis and System Specification**; Prentice-Hall, 1979.
- [Díez et al., 1997] Díez, J.A., Moulines, C.U.; **Fundamentos de Filosofía de la Ciencia**. Ariel, 1997.

- [DIN, 1983] DIN 66001 Beiblatt 1; **Information processing; graphical symbols and their application; layout of graphical symbols on a template**; <http://www.en.din.de/>, 1983.
- [Dori, 1996] Dori, D.; **Unifying Systems Structure and Behaviour Through Object-Process Analysis**; Journal of Object Oriented Programming, July-August 1996. pp. 66-73.
- [Dubois et al., 1989] Dubois, E., hagelstein, J., Rifault, A.; **A Data Model for Requirements Engineering**; 2nd International Conference on Data Engineering, 1986.
- [Eco, 1994] Eco, U.; **La Búsqueda de la Lengua Perfecta**; Grijalbo-Mondadori, 1994.
- [Eco, 1999] Eco, U.; **Kant y el Ornitorrinco**. Lumen, 1999.
- [F3, 1991] F3 Consortium; **From Fuzzy to Formal**; Technical Annex Part II, ESPRIT Project 6612, The F3 Consortium, 1991.
- [Faulk, 1997] Faulk, S.R.; **Software Requirements: A Tutorial**; *In Software Engineering, IEEE Computer Society Press*, 1997. pp 82-101.
- [Feldman et al., 1986] Feldman, P., Miller, D. ; **Entity Model Clustering: Structuring a Data Model by Abstraction**; The Computer Journal, Vol. 29, No. 4, 1986. pp. 348-360.
- [Fickas et al., 1988] Fickas, S., Collins, S., Oliver, S.; **Problem acquisition in problem analysis: a preliminary study**; Report CIS-TR-87-04, Department of Computer and Information Science, University of Oregon, 1988.
- [Fowler et al., 1999] Fowler, M., Scott, K.; **UML Distilled: A Brief Guide to the Standard Object Modeling Language**; Addison-Wesley, 1999.
- [Gane et al., 1979] Gane, C., Sarson, T.; **Structured Systems Analysis: Tools and Techniques**; Prentice-Hall, 1979.
- [Genero et al., en prensa] Genero M., Piattini M. and Jiménez L.; **A Metric-Based Approach For Predicting Conceptual Data Models Maintainability**. Journal of Software Engineering and Knowledge Engineering, 11(6). En prensa.
- [Gentner et al., 1983] Gentner, D., Stevens, A.L.; **Mental Models**; Lawrence Erlbaum Associates, 1983.
- [George et al., 1996] J. George, B. Carter; **A strategy for mapping from function-oriented software models to object oriented software models**; Software Engineering Notes, Vol. 21, No. 2, 1996.
- [Glass et al., 1995] Glass, R.L., Vessey, I.; **Contemporary Application-Domain Taxonomies**. IEEE Software. Vol. 12, No. 4, 1995.
- [Goodland et al., 1990] Goodland, M., Ashworth, C.; **Ssdsm: A Practical Approach**, McGraw-Hill, 1990.

- [Gorman et al., 1991] Gorman, K., Choobineh, J.; **The Object-Oriented Entity-Relationship Model (OOERM)**; Journal on Management Information Systems, Vol. 7, No. 3, 1991.
- [Graham et al., 1997] Graham, I., Henderson-Sellers, B., Younessi, H.; **The OPEN Process Specification**; Addison-Wesley, 1997.
- [Guindon et al., 1987] Guindon, R., Krasner, H., Curtis, B.; **Breakdown and processes during the early activities of software design by professionals**; In *Proc of 2nd Workshop on Empirical Studies of Programmers*, Olson, G., Sheppard, S., Soloway, E. (eds), 1987.
- [Guindon et al., 1988] Guindon, R., Curtis, B.; **Control of cognitive processes during software design: What tools are needed?**; In *Proceedings of CHI'88 Conference: Human Factors in Computing Systems*, Soloway, E., Frye, D., Sheppard, S.B. (Eds), ACM Press, 1988.
- [Gustafsson et al., 1982] Gustafsson, M.R., Karlsson, T., Bubenko, J.A.; **A Declarative Approach to Conceptual Information Modeling**; In *IFIP WG8.1 Working Conference on Information Systems Design Methodologies: A Comparative Review*, T.W. Olle, H.G. Sol, A.A. Verryn-Stuart (Eds.), North-Holland, 1982.
- [Harel, 1981] Harel, D.; **Statecharts: A Visual Formalism for Complex Systems**; Science of Computer programming, Vol. 31, No. 5, 1988.
- [Harel, 1988] Harel, D.; **On Visual Formalisms**; Communications of the ACM, Vol. 31, No. 5, 1988.
- [Hatley et al., 1987] Hatley, D., Pirbhai, I.; **Strategies for Real-Time System Specifications**; Dorset House, 1987.
- [Hatley, 1984] Hatley, D.; **The Use of Structured Methods in the Development of large Software-Based Avionics Systems**; In *AIAA/IEEE 6th Digital Avionics Systems Conference*, American Institute of Aeronautics and Astronautics, 1984.
- [Henderson-Sellers et al., 1990] Henderson-Sellers, B., Edwards, J.; **The Object Oriented Systems Life Cycle**; Communications of the ACM, Vol. 33, No. 9, 1990.
- [Hohenstein et al., 1988] Hohenstein, U., Gogolla, M.; **A Calculus for an Extended Entity-Relationship Model Incorporating Arbitrary Data Operations and Aggregate Functions**; In *Entity-Relationship Approach: A Bridge to the User*. Batini, N. (Ed.), Proc. 7th Int. Conf. On Entity-Relationship Approach, 1988, North-Holland, 1988.
- [Høydalsvik et al., 1993] Høydalsvik, G.M., Sindre, G.; **On the Purpose of Object Oriented Analysis**; *Proc. of the OOPSLA'93*.

- [Hull et al., 1987] Hull, R., King, R.; **Semantic Database Modeling: Survey, Applications and Research Issues**; ACM Computing Surveys, Vol. 13, No. 3, 1987.
- [IEEE, 1998] IEEE Std-830-1998; **IEEE Recommended Practice for Software Requirements Specifications**; IEEE, 1998.
- [i-Logix, 1987] i-Logix; **The Languages of STATEMATE**; Technical Report, i-Logix, 1987.
- [Jackson, 1983] Jackson; M. **Systems Development**. Prentice-Hall, 1983.
- [Jackson, 1994] Jackson, M.; **Problems, Methods and Specialization**; Software Engineering Journal, Vol. 9, No. 6, 1984.
- [Jackson, 1995] Jackson, M; **The world and the machine**; *International Conference on Software Engineering*, ACM Press, 1995.
- [Jackson, 1998] Jackson, M.; **Defining a Discipline of Description**; IEEE Software, Vol. 15, No. 5, 1998.
- [Jackson, 2001] Jackson, M; **Problem Frames: Analysing and Structuring Software Development Problems**; Addison-Wesley, 2001.
- [Jacobson, 1992] Jacobson, I; **Object-Oriented Software Engineering: a Use Case Approach**; Addison-Wesley, 1992.
- [Jaeschke et al., 1993] Jaeschke, P., Oberweis, A., Stucky, W.; **“Extending ER Model Clustering by Relationship Clustering**; Proc. of the 12th Int. Conf. On the Entity-Relationship Approach, 1993.
- [Jalote, 1997] Jalote, P.; **An Integrated Approach to Software Engineering**; Springer-Verlag, 1997.
- [Jeffries et al., 1981] Jeffries, R., Turner, A.A., Polson, P.G., Atwood, M.E.; **The processes involved in designing software**; *In Cognitive Skills and Their Acquisition*, Anderson, J.R. (Ed), Lawrence Erlbaum Associates, 1981.
- [Jones, 1990] Jones, C.B.; **Systematic Software Development Using VDM**; 2nd Edition, Prentice Hall, 1990.
- [Juristo et al., 2000] Juristo, N., Moreno, A.M.; **Introductory paper: Reflections on Conceptual Modeling**; Data and Knowledge Engineering, vol 33, 2000.
- [Kaindl, 1999] Kaindl, H.; **Difficulties in the transition from OO analysis to design**; IEEE Software, Vol. 16, No. 5, 1999.
- [Kappel et al., 1988] Kappel, G., Schrefl, M.; **A Behaviour Integrated Entity-Relationship Approach for the Design of Object-Oriented Databases**; *In Entity-Relationship Approach: A bridge to the User*, C. Batini (Ed.), Proceedings of the 7th International Conference on Entity-Relationship Approach, North-Hollan, 1988.

- [Kerschberg et al., 1976] Kerschberg, L., Pacheco, J.E.S.; **A Funcional Data Base Model**; Technical Report, Universidad Pontificia de Rio de Janeiro, 1976.
- [Kirikova et al., 1994b] Kirikova, M., Bubenko J.A.jr; **Software Requirements Acquisition through Enterprise Modelling**; Software Engineering and Knowledge Engineering - SEKE'94, 1994.
- [Kirikova, et al, 1994a] Kirikova, M., Bubenko J.A. jr; **Enterprise Modelling: Improving the Quality of Requirements Specification**; Information systems Research seminar In Scandinavia, 1994.
- [Kop et al., 1998] Kop, C., Mayr, H.C.; **Conceptual Predesign. Bridging the Gap between Requirements and Conceptual Design**; *In Proceedings of the 3rd International Conference on Requirements Engineering*, IEEE Computer Society Press, 1998.
- [Kotonya et al., 1998] Kotonya, G., Sommerville, I.; **Requirements Engineering: Processes and Techniques**; John Wiley and Sons, 1998.
- [Kuo, 1994] F.Y. Kuo; **A methodology for deriving an entity relationship model based on a data flow diagram**, Journal of Systems and Software, Vol. 24, No. 2, 1994.
- [Lamsweerde et al., 1991] Lamsweerde, A., Dardenne, A., Dubisy, F.; **The KAOS Project: Knowledge Adquisiton in Automated Specification of Software**; Proceedings of the AAI Spring Sysposium Series, 1991.
- [Larman, 1999] Larman, C; **UML y Patrones: Introducción al Análisis y Diseño Orientado a Objetos**; Prentice-Hall, 1999.
- [Loucopoulos et al., 1995] Loucopoulos, P., Karakostas, V; **System Requirements Engineering**; McGraw-Hill, 1995.
- [Marca et al., 1988] Marca, D.A., McGowan, C.L.; **SADT-Structured Analysis and Design Technique**. McGraw-Hill, 1988.
- [Markowitz et al., 1992] Markowitz, V., Shoshani, A.; **Representing Extended Entity-Relationship Structures in Relational Databases: A Modular Approach**; ACM Transactions on Database Systems, Vol.17, No. 3, 1992.
- [Martin, 1990] Martin, J.; **Information Engineering**; Vols. 1-3, Prentice Hall, 1990.
- [McGuinness, 1992] McGinnes, S.; **How objective is object-oriented analysis?**; *Proceedings of the CAISE'92 Advanced Information Systems Engineering*, 1992.
- [Meyer, 1988] Meyer, B.; **Object Oriented Software Construction**; Prentice-Hall, 1988.
- [Mylopoulos et al., 1980] Mylopoulos, J., Bernstein, P.A., Wong, H.K.T. ; **A Language Facility for Designing Database Intensive Applications** ; ACM Transactions on Database Systems, Vol. 15, No. 2, 1980.

- [Mylopoulos et al., 1990] Mylopoulos, J., Borgida, A., Jarke, M., Koubarakis, M.; **TELOS: Representing Knowledge about Information Systems**; ACM Transactions on Office Information Systems, Vol. 8, No. 4, 1990.
- [Navathe et al., 1986] Navathe, S., Elmasri, R., Larson, J.; **Integrating User Views in Database Design**; *IEEE Computer*, Vol.19, No. 1, 1986.
- [Navathe et al., 1988] Navathe, S., Pillalamarri, M.; **Toward Making the ER Approach Object-Oriented**; *In Entity-Relationship Approach: A bridge to the User*, C. batini (Ed.), Proceedings of the 7th International Conference on Entity-Relationship Approach, North-Holland, 1988.
- [Nijssen et al., 1989] Nijssen, G., Halpin, T.A.; **Conceptual Scheme and Relational Database Design – A Fact Oriented Approach**; Prentice-Hall, 1989.
- [Northrop, 1997] Northrop, L.M.; **Object-Oriented Development**; *In Software Engineering*, IEEE Computer Society Press, 1997.
- [Novak et al., 1988] Novak, J.D., Gowin, D.B.; **Aprendiendo a Aprender**; Martínez Roca, 1988.
- [Olivé, 1986] Olivé, A.; **A Comparison of the Operational and Deductive Approaches to Conceptual Information Systems Modelling**; *In Information Processing*, H.J. Kugler (Ed.), North-Holland, 1986.
- [Ontoria et al., 1996] Notoria, A. y otros; **Mapas Conceptuales: Una Técnica para Aprender**; Narcea S.A. de Ediciones, 1996.
- [Orr, 1977] Orr, K.T.; **Structured Systems Development**; Yourdon Press, 1977.
- [Orr, 1981] Orr, K.; **Structured Requirements Definition**; Ken Orr and Associates, 1981.
- [Palmer et al., 1984] Palmer, J., Mcmenamin, S.; **Essential Systems Analysis**; Yourdon Press/Prentice-Hall, 1984.
- [Peterson, 1977] Peterson, J.; **Petri-Nets**; ACM Computing Surveys, Vol. 9, No. 3, 1977.
- [Petri, 1962] Petri, C.; **Kommunikation mit Automation, Schifren des Reinsch-Westfalischen Inst. für Instrumentelle Mathematik an der Universität Bonn**; 1962.
- [Piattini et al., 2001] Piattini M., Genero M., Calero C., Polo M. and Ruiz F. **Metrics for managing information modelling**. In: *Information Modelling in the New Millennium*. Siau, K. and Rossi, M. (eds.), Idea Group Publishing, 2001.
- [Pressman, 2001] Pressman, R.S.; **Software Engineering: A Practitioner's Approach**; McGraw-Hill International, 2001.
- [Rentsch, 1982] Rentsch, T.; **Object Oriented Programming**; SIGPLAN Notices, Vol. 17, No. 9, 1982.

- [Rist, 1991] Rist, R.; **Models of routine and non-routine design in the domain of programming**; *In Artificial Intelligence in Design*, Gero, J. and Sudweeks, F. (Eds), 1991.
- [Rockstrom et al., 1982] Rockstrom, A., Saracco, R.; **SDL-CCITT Specification and Description Language**; IEEE Transaction on Communications, Vol.30, No. 6, 1982.
- [Rolland et al., 1982] Rolland, C., Richard, C.; **The REMORA Methodology for Information Systems Development and Management**; Conference on Comparative Review of Information System Design Methodologies, North-Holland, 1982.
- [Rolland et al., 1992] Rolland, C., Cauvet, C.; **Trends and Perspectives in Conceptual Modelling**; *In Conceptual Modelling, Databases and CASE: An Integrated View of Information Systems Development*, P. Loucopoulos, R. Zicari (Eds.), Willey, 1992.
- [Ross, 1977] Ross D.; **Structured Analysis (SA): A Language for Communicating Ideas**. IEEE Transactions on Software Engineering, SE-3, 1977.
- [Rumbaugh et al., 1991] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W.; **Object-Oriented Modeling and Design**; Prentice-Hall, 1991.
- [Rumbaugh et al., 1998] Rumbaugh, J., Jacobson, I., Booch, G.; **The Unified Modeling Language Reference Manual**; Addison-Wesley Object Technology Series, 1998.
- [Sayani, 1990] Sayani, H.H.; **PSL/PSA at the Age of Fifteen**; *In System and Software Requirements Engineering*, Thayer, R.H. y Dorfman, M. (Eds), IEEE Computer Society Press, 1990.
- [Shipman, 1981] Shipman, D.; **The Functional Data Model and the data language DAPLEX**; ACM Transactions on Database Systems, Vol. 6, No. 1, 1981.
- [Siddiqi, 1994] Siddiqi, J.; **Challenging Universal Truths of Requirements Engineering**; IEEE Software, Vol. 11, No. 2, 1994.
- [Spivey, 1989] Spivey, J.M.; **The Z Notation**; Prentice Hall, 1989.
- [Sudkamp, 1996] Sudkamp, T.A.; **Languages and Machines : An Introduction to the Theory of Computer Science**; Addison-Wesley, 1996.
- [Sutcliffe et al., 1992] Sutcliffe, A.G., Maiden, N.A.M.; **Analysing the Novice Analyst: Cognitive Models in Software Engineering**; International Journal of Man-Machine Studies, Vol. 36, No. 5, 1992.
- [SWEBOK, 2001] **SWEBOK (Software Engineering Body of Knowledge)**. Version 0,95. <http://www.swebok.org>. Mayo 2001.
- [Teichroew et al., 1971] Teichroew, D., Sayani, H.; **Automation of System Building**; Datamation, August 15, 1971.

- [Teichroew et al., 1977] Teichroew, D., Hershey III, E.A.; **PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems**; IEEE Transactions on Software Engineering, Vol. 3, No. 1, 1977.
- [Teorey et al., 1986] Teorey, T., Yang, D., Fry, J.; **A Logical Design Methodology for Relational Databases Using the Extended E-R Model**; ACM Computing Surveys, Vol.18, No. 2, 1986.
- [van Griethuysen, 1982] van Griethuysen, J.J.; **ISO – Concepts and Terminology for the Conceptual Schema and the Information Base**; N695, ISO/TC9/SC5/WG3, 1982.
- [Velasco et al., 1997] Velasco, H., Díaz, A. **La lógica de la Investigación Etnográfica: Un Modelo de Trabajo para Etnógrafos de la Escuela**. Trotta, 1997.
- [Vessey, 1991] Vessey, I.; **A Theory-Based Analysis of the Graph versus Tables Literature**; Decision Sciences, Vol. 22., 1991.
- [Vinceti, 1990] Vinceti, W.G.; **What Engineers Know and How they Know it**; John Hopkins University Press, 1990.
- [Visser, 1987] Visser, W.; **Strategies in programming programmable controllers: a field study on a professional programmer**; *In Proc of 2nd Workshop on Empirical Studies of Programmers*, Olson, G., Sheppard, S., Soloway, E. (eds), 1987.
- [Vitalari et al., 1983] Vitalari, N.P., Dickson, G.W.; **Problem solving for effective systems analysis: an experimental exploration**; Communications of the ACM, Vol. 26, No. 1, 1983.
- [Wand et al., 1989] Wand, Y., Weber, R. ; **An Ontological Evaluation of Systems Analysis and Design Techniques**; *In Information System Concepts: an In-Depth Analysis*, E.D. Falkenberg and P. Lindgreen (Eds.), Elsevier Science Publishers, 1989.
- [Wand, 1996] Wand, Y.; **Ontology as a Foundation for Meta-Modeling and Method Engineering**; Information and Software Technology, vol 38, 1996.
- [Ward et al., 1985] Ward, P., Mellor, S.; **Structured Development for Real-Time Systems.**; Vol. 1-3, Prentice-Hall, 1985.
- [Ward et al., 1989] P. Ward, S. Mellor; **How to integrate object orientation with structured analysis and design**; IEEE Software, Vol. 6, No. 2, 1989.
- [Ward, 1986] Ward, S.; **The Transformation Schema: An Extension of the Data Flow Diagram to Represent Control and Timing**; IEEE Transactions on Software Engineering, vol 12, No. 2, 1986.

- [Webster, 1988] Webster, D.E.; **Mapping the Design Information Representation Terrain**. IEEE Computer. Vol. 21, No. 12, 1988.
- [Webster, 1994] **Webster's New Encyclopedic Dictionary**; Könenmann, Colonia, Alemania, 1994.
- [Wertz, 1993] Wertz, C.; **Relational Database Design**; CRC Press, 1993.
- [Wieringa, 1991] Wieringa, R.; **Object-Oriented Analysis, Structures Analysis, and Jackson System Development**; *In Proceedings of the IFIP WG8.1 Working Conference on the Object-Oriented Approach in Information Systems*, F. van Assche, B. Moulin, C. Rolland (Eds.), North-Holland, 1991.
- [Wieringa, 1995] Wieringa, R.; **Requirements Engineering: Frameworks for Understanding**; John Wiley and Sons, 1995.
- [Wirth, 1971] Wirth, N; **Program Development by Stepwise Refinement**. Communications of the ACM. Vol. 14, No. 4, 1971.
- [Yourdon et al., 1986] Yourdon, E., Constantine, L.; **Structured Design : Fundamentals of a Discipline of Computer Program and Systems Design**; Prentice-Hall, 1986.
- [Yourdon, 1989] Yourdon, E.; **Análisis Estructurado Moderno**; Yourdon Press/Prentice Hall, 1989.
- [Yourdon, 1992] Yourdon, E.; **Decline and Fall of the American Programmer**; Yourdon Press, 1992.
- [Yu et al., 1994] Yu, E., Mylopoulos, J.; **Understanding "Why" in Software Process Modelling, Analysis and Design**; Proceedings of the 16th International Conference on Software Engineering, 1994.
- [Yu, 1995] Yu, E.; **Modelling Strategic Relationships for Process Reengineering**; PhD Thesis, Dept. of Computer Science, University of Toronto, 1995.
- [Zave et al., 1981] Zave, P., Yeh, R.; **Executable Requirements for Embedded Systems**; *In Proc. of the 5th IEEE Int. Conf. On Software Engineering*, IEEE Press, 1981.
- [Zave et al., 1996] Zave, P., Jackson, M.; **Where Do Operations Come From? A Multiparadigm Specification Technique**; IEEE Transactions on Software Engineering, Vol. 22, No. 7, 1996.
- [Zave, 1982] Zave, P.; **An Operational Approach to Requirements Specification for Embedded Systems**; IEEE Transactions on Software Engineering, Vol. 8 , No. 3, 1982.

[Zave, 1990]

Zave, P.; **A Comparison of the Major Approaches to Software Specification and Design**. *In System and Software Requirements Engineering*, Thayer RH, Dorfman M (eds). IEEE. Computer Society Press, 1990.

Anexo A

A.1. MODELO CANÓNICO

El Modelo Canónico de Requisitos (MCR) es un formalismo de representación de tipo tabular, cuya estructura es idéntica a la del Diccionario de Descripción (DD). Ello es natural si se considera el hecho de que el MCR se deriva del DD tras la aplicación de la técnica IMCI o, de forma más precisa, tras la aplicación del procedimiento de interpretación que forma parte de la técnica IMCI.

Las etiquetas utilizadas en el proceso de interpretación se derivan de [Davis et al., 1997]. Dicho trabajo consistió en la identificación de un conjunto de constructores que tuvieran la suficiente potencia como para reescribir, sin pérdida de significado, la información que podían representar un conjunto de modelos conceptuales, tales como el diagrama de flujo de datos, los diagramas de clases, statecharts, etc.

El objetivo de dicho trabajo era, principalmente, proporcionar un mecanismo que facilitase: (1) la posibilidad de comparar e integrar la información proporcionada por diversos modelos conceptuales potencialmente incompatibles y (2) facilitar la realización del análisis o especificación usando dichos modelos una vez que se disponía de un mecanismo de integración entre los mismos.

Dichos objetivos, y la consecución de los mismos, no alteran ni restan vigencia en ninguna medida los objetivos del presente trabajo de Tesis. El trabajo de Davis está situado en una fase posterior a la modelización conceptual, esto es, se sirve de los modelos conceptuales para posteriormente proceder a su reescritura e integración. Por ello, siempre es necesario desarrollar en primer lugar un modelo conceptual. En el presente trabajo de Tesis se pretende evitar este primer paso, habida cuenta de que los modelos conceptuales poseen dependencias computacionales.

Debido a lo anterior, en el presente trabajo de Tesis no se considera el trabajo de Davis como un formalismo sustitutivo de los modelos conceptuales, sino como una *lingua franca* que permite expresar información para, posteriormente, derivar ésta a un modelo conceptual clásico. Nótese que este enfoque no

introduce restricciones computacionales en el modelo del problema, por dos razones:

- La asignación de las etiquetas es posterior a la finalización del modelo del problema. El hecho de que algunas asociaciones (por ejemplo, *bel*, *val*, *subs*, etc.) posean una interpretación bien definida no supone una contradicción, en la medida en que: (1) sería muy discutible que dichas asociaciones supusieran una dependencia computacional, en la medida en que dichas asociaciones ocurren en el dominio del problema en el mundo real, y no únicamente en la máquina y (2) algunas de estas etiquetas (*bel* y *subs*, en concreto) ni siquiera están recogidas en el trabajo de Davis, sino que son modificaciones del autor del presente trabajo de Tesis.
- Como la asignación de etiquetas es independiente de la utilización de cualquier modelo conceptual, esto es, la información recogida en el MCR no es derivable a un único modelo conceptual, sino potencialmente a una diversidad de modelos conceptuales.

En lo que sigue, se presentarán de forma simplificada los aspectos fundamentales del trabajo de Davis (únicamente en lo exigido por el presente trabajo de Tesis), su relación con los modelos conceptuales clásicos y, de forma más pormenorizada, las modificaciones propuestas por el autor.

A.2. ELEMENTOS Y ENLACES

El trabajo original de Davis consiste en la definición de un conjunto de elementos y links. Los elementos denotan ciertos constructores de los modelos conceptuales clásicos, mientras que los links permiten enlazar dichos elementos formando estructuras con significado, potencialmente equivalentes a estructuras parciales de los modelos conceptuales clásicos. Existen 8 elementos y 9 links distintos. Los elementos propuestos por Davis son los siguientes:

Entity: representa una “cosa” existente en el dominio del problema (mundo real) o en el dominio de implementación (sistema informático).

Process: representa una acción, tarea, función o actividad realizada en el mundo real o en el sistema informático.

Message: algo que se mueve o que se intercambia entre dos *entities* o *procesos*, tanto en el mundo real como en el sistema informático.

Predicate: Una proposición que puede ser cierta o falsa, tanto en el mundo real como en el sistema informático.

Constraint: Una proposición que denota una proposición que es siempre cierta, tanto en el mundo real como en el sistema informático.

Statespace: Descriptor de los elementos anteriormente mencionados (*entity*, *process*, *message*, *predicate*, *constraint*), y que puede calificarlos con uno o varios valores. Adicionalmente, existen cuatro tipos de *statespaces* predefinidos:

- *Int/ext*: permite indicar si una *entity* es interna o externa (necesario para, por ejemplo, el

diagrama de flujo de datos).

- *Repl/notrepl*: Permite indicar si una *entity* es replicable (una clase de un diagrama de clases) o no replicable (un objeto, por ejemplo).
- *Data/cntl*: Permite indicar si *process* o un *message* es de datos o control (necesario, por ejemplo, para el DFD de Ward).
- *Contin/disc*: Permite indicar si un *message* es continuo o discreto (necesario, por ejemplo, para el DFD de Ward).

Value: Valor único (1 ó 7, por ejemplo) o múltiple (una tupla como ["Pepe", 17], por ejemplo).

Transition: Captura las causas y efectos de un cambio tanto en el mundo real como en el sistema informático.

Cada uno de estos elementos puede representarse gráficamente, aunque en el presente trabajo de Tesis este hecho no es esencial. La convención de diagramación se muestra en la figura A.1.

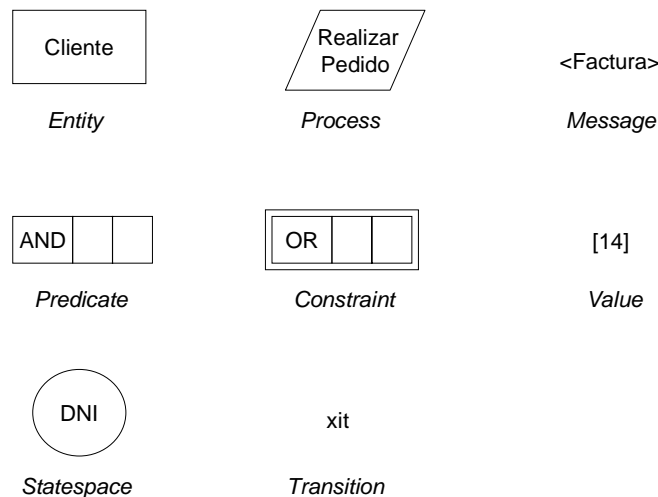


Figura A.1. Convenciones de diagramación

Los elementos indicados anteriormente pueden enlazarse mediante los siguientes links:

Specialisation (o spec): Relación de especialización/generalización.

Part of (o pof): Relación de agregación.

Has value (o hval): Permite asignar un valor a un statespace. Este link no hace referencia a aspectos dinámicos, por lo que la asignación debe entenderse como una "inicialización" (por ejemplo, el valor inicial de un atributo en el mundo real, o el valor inicial de una variable de un programa).

Sends: Indica el envío de un message por parte de un entity o process.

Receives: Indica la recepción de un message por parte de un entity o process.

Operand: Permite relacionar predicates, constraints y messages con sus parámetros o valores.

Stimulus: Define un elemento que desencadena una transition. Existen tres tipos de link stimulus:

- *Stimulus* $\hat{\uparrow}$: Desencadena la transition cuando el elemento *se activa*.
- *Stimulus* \downarrow : Desencadena la transition cuando el elemento *se desactiva*.
- *Stimulus*: Desencadena la transition cuando el elemento *aparece*.

Response: Define qué elementos se ven afectados por una transition. Existen tres tipos de link response.

- *Response* $\hat{\uparrow}$: *Activa* el elemento.
- *Response* \downarrow : *Desactiva* el elemento.
- *Response*: Hace que el elemento sea cierto (básicamente, declara una postcondición).

Equivalence: Define que dos elementos son iguales.

Los links también poseen una notación gráfica. Los links se diagraman como flechas que emanan o apuntan a los distintos elementos, etiquetadas con el nombre correspondiente.

Las posibles combinaciones entre elementos y links poseen restricciones, esto es, no todas las combinaciones están permitidas, sino solo aquellas que son significativas. Las combinaciones significativa son aquellas que pueden traducirse a los constructores de algún modelo conceptual clásico. Ello supone probablemente una limitación expresiva, pero asegura la convertibilidad de elementos y links a algún modelo conceptual clásico. Nótese, no obstante, que:

Una misma combinación de elementos y links puede traducirse a varios modelos conceptuales clásicos.

Dado un conjunto de elementos y links, es sumamente probable que un único modelo conceptual clásico no sea adecuado para representar la totalidad de elementos y links.

A.3. RELACIÓN CON LOS MODELOS CONCEPTUALES

El trabajo original de Davis tenía como objetivo principal proponer un mecanismo para reescribir distintos tipos de modelos conceptuales clásicos, como ya se ha indicado anteriormente. Independientemente de las motivaciones que se esconden detrás de este objetivo principal, es un hecho que la reescritura exige relacionar los distintos constructores de los modelos conceptuales clásicos con los constructores (elementos y links) del formalismo propuesto por Davis. Dicho de otro modo, y tal como muestra la figura A.2, es posible establecer una relación, estricta, entre los elementos y links del formalismo

propuesto por Davis y varios modelos conceptuales clásicos.

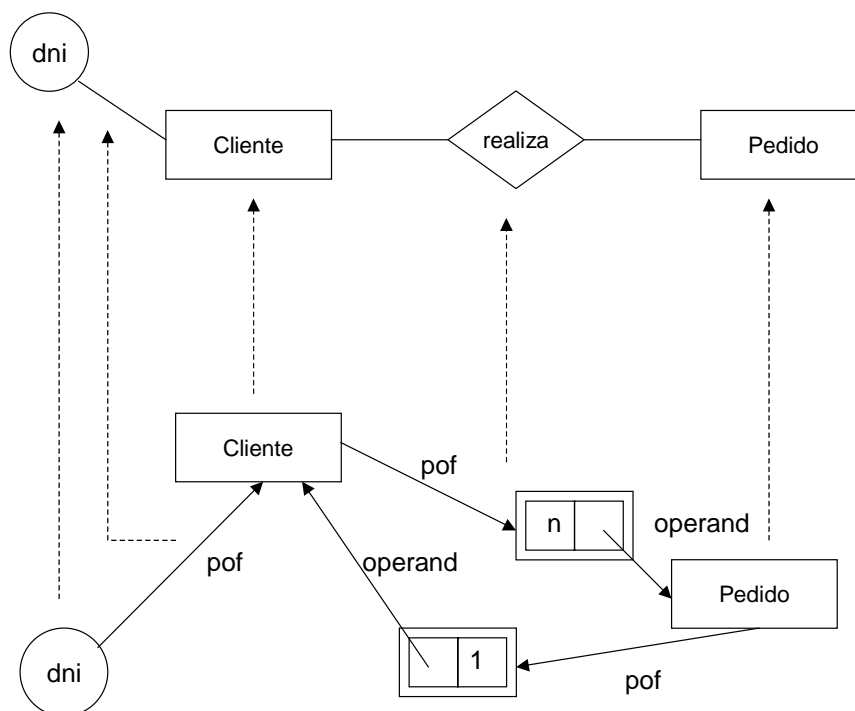


Figura A.2. Relación entre elementos/links y el modelo entidad-relación

A.4. MODIFICACIONES REALIZADAS AL MODELO CANÓNICO

En el presente trabajo de Tesis, se han realizado diversas modificaciones en el Modelo Canónico. El motivo de dichas modificaciones es que Davis, como ya se ha indicado, propuso con su trabajo un mecanismo para reescribir una diversidad de modelos conceptuales clásicos (esto es, actualmente utilizados) en una *lingua franca*, esto es, un mecanismo de expresión que permitiera expresar la diversidad de conocimientos recogidos en los modelos conceptuales utilizados actualmente.

Sin embargo, el presente trabajo de Tesis propone un conjunto de formalismos de representación (el ya citado MCG) libre de restricciones computacionales, esto es, limitaciones provocadas por la atracción que la máquina subyacente ejerce en los modelos conceptuales clásicos. Una vez confeccionado el MCG, es necesario derivar un modelo conceptual clásico para proseguir con el posterior desarrollo del sistema software.

Para ello, se utiliza el trabajo de Davis como un paso intermedio que permite generar una diversidad de modelos conceptuales clásicos, esto es, el lenguaje propuesto por Davis se utiliza de forma inversa a la intención de su autor. Por ello, y al ser dicho lenguaje utilizado de forma inversa, ha sido necesario modificar éste, con la finalidad de que sea capaz de representar aspectos referidos al problema, en lugar de aspectos referidos a la solución, ya que, en la medida en que el modelo conceptual es un lenguaje sustitutivo, posee las mismas ventajas e inconvenientes que los modelos conceptuales a los que sustituye. En concreto, dichas modificaciones han sido las indicadas en las siguientes secciones.

A.4.1. INVOCACIÓN DE PROCESOS

El lenguaje propuesto por Davis contempla 3 formas distintas de registrar una invocación de procesos:

- Invocación de un proceso por una entidad (estilo estructurado): El lenguaje propuesto por Davis propone la estructura mostrada en la figura A.3 para la invocación de un proceso por una entidad externa.

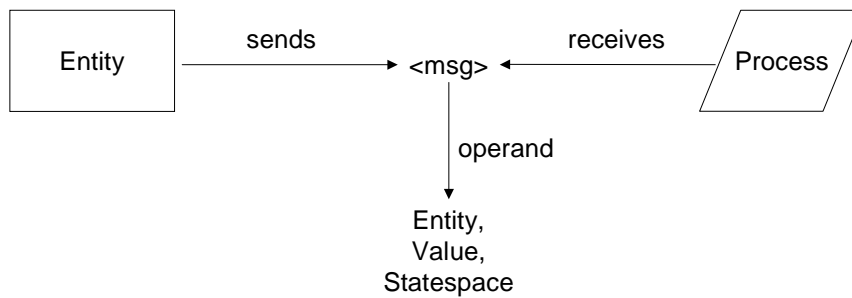


Figura A.3. Estructura de invocación de un proceso por una entidad externa en un DFD

Debido a la definición del MCG, para derivar la estructura anterior (como se comentará en esta sección) sería necesario identificar 4 conceptos, tal y como muestra la figura A.4.

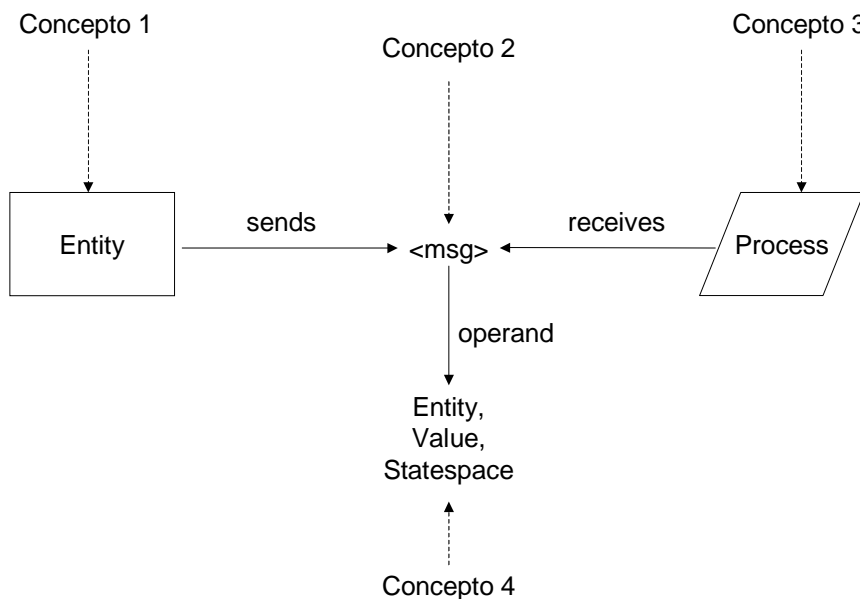


Figura A.4. Conceptos necesarios (del MCG) para derivar la estructura de invocación de un proceso por una entidad externa en un DFD

Sin embargo, aunque posible, es muy poco probable que la estructura de la figura A.4 llegara a encontrarse en el MCG. La razón estriba en el hecho de que la invocación del proceso es excesivamente “computacional”, esto es, no está basada en el hecho de que una entidad inicie un proceso, sino en el intercambio de un token que refleja, únicamente, un flujo de datos entre la entidad y el proceso, que inicia éste. Esta característica computacional es, precisamente, lo que el MCG intenta evitar y, por ello, la estructura reflejada en la figura A.4 difícilmente podrá aparecer en el MCG.

- Invocación de un proceso por un proceso (estilo estructurado), el cual se muestra en la figura A.5.

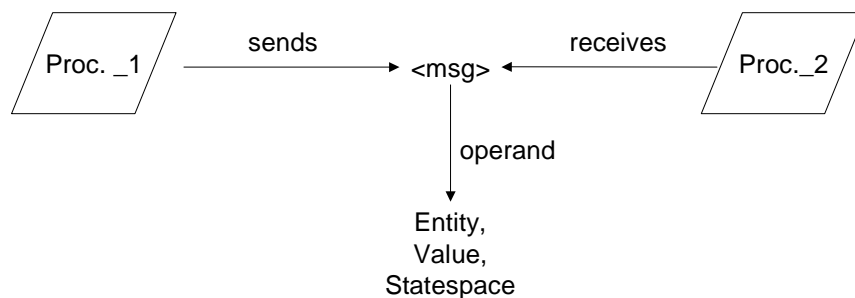


Figura A.5. Estructura de invocación de un proceso por un proceso en un DFD

- Invocación de un proceso por una entidad (estilo objetos): el cual se muestra en la figura A.6. Nótese que la invocación de un proceso por un proceso (estilo objetos) esto es, la invocación de un método por otro método es igual que la invocación de un proceso por una entidad, ya que en ambos casos es el objeto el que envía el mensaje.

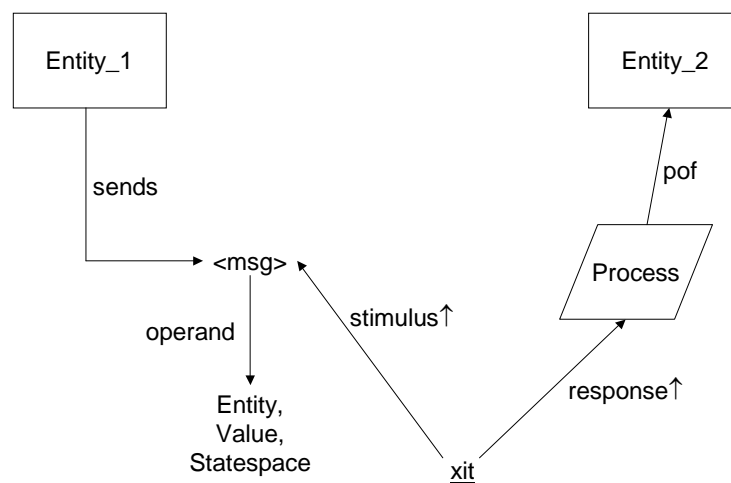


Figura A.6. Invocación de un proceso (método) por una entidad (objeto)

En lugar de los tres modelos de invocación anterior, se ha introducido en el modelo canónico un nuevo link, denominado *activate*. Este link simplifica los tres mecanismos de invocación a uno solo, mostrado en la figura A.7. Nótese que este único mecanismo de invocación lo único que hace es *ocultar* los mecanismos computacionales de invocación de procesos.

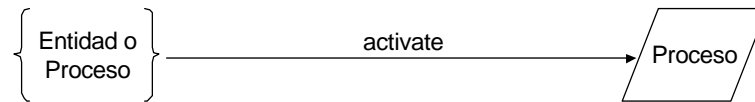


Figura A.7. Mecanismo de invocación de procesos utilizado en el presente trabajo de Tesis. Las llaves se han utilizado (arbitrariamente) para no utilizar un simbolismo concreto del modelo canónico, el cual sería incorrecto

Adicionalmente, nótese que la existencia del link *exec* no implica que no puedan aparecer en el modelo canónico estructuras como las mostradas en las figuras A.4, A.5 y A.6. Simplemente, no es necesario que aparezcan para inferir la existencia de una invocación de un proceso. En el caso de que las estructuras mostradas en las figuras A.4, A.5 y A.6 apareciesen en el MCG, se derivarían a los constructores correctos del modelo conceptual seleccionado.

A.4.2. INVOCACIÓN DE PROCESOS POR UN PREDICADO

Para Davis, un proceso únicamente puede ser invocado de las 3 formas indicadas en la sección 5.8.2.1. Sin embargo, el MCG posee una particularidad que obliga a considerar nuevas formas de invocación.

Dicha particularidad se refiere a que los predicados del MCG pueden contener funciones. Las funciones del MCG siempre se derivan a procesos durante la aplicación de la técnica IMCI (con el objetivo de compatibilizar –interpretar- el MCG en términos del Modelo canónico). Por este motivo, debe permitirse que un predicado (esto es, un predicado o restricción del Modelo canónico) pueda invocar un proceso.

La particularidad de la invocación anterior es que se trata realmente de una reescritura. Dicho de otro modo; supóngase el Mapa de conceptos de la figura A.8.

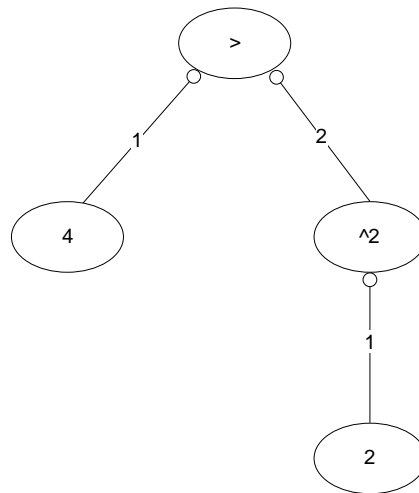


Figura A.8. Estructura que define el predicado $4 > 2^2$

Esta estructura es equivalente a la estructura mostrada en la figura A.9.

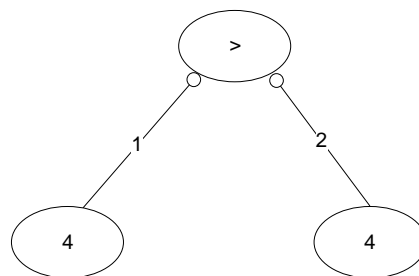


Figura A.9. Reescritura de la función 2^2

Aunque sería posible describir una combinación de elementos y mensajes del Modelo canónico para reflejar la reescritura, dicha combinación poseería los mismos problemas que los indicados para la invocación de procesos. Por ello, parece más conveniente utilizar el link *activate*, definido anteriormente, suponiendo que el proceso invocado, tras su ejecución, se transforma en un valor.

El link *activate* no se utilizará únicamente con predicados. La asignación de un valor a un atributo (que se realiza con la asociación *val*) también se transformará a un link *activate*, cuando el valor deba calcularse mediante una función.

A.4.3. PERTENENCIA A CONJUNTOS

El lenguaje propuesto por Davis no contempla la distinción entre conjuntos e individuos más que de

forma indirecta, mediante la noción de los statespaces repl/notrepl. Esta distinción es suficiente para denotar qué es un individuo y qué es un conjunto, pero no para reflejar la pertenencia de un individuo a un conjunto.

Dentro del lenguaje propuesto por Davis, existe la posibilidad (no enunciada por su autor) de reflejar la pertenencia de un individuo a un conjunto de dos formas distintas: mediante un link spec y mediante un link pof, tal como se muestra en las figuras A.10 y A.11, respectivamente.

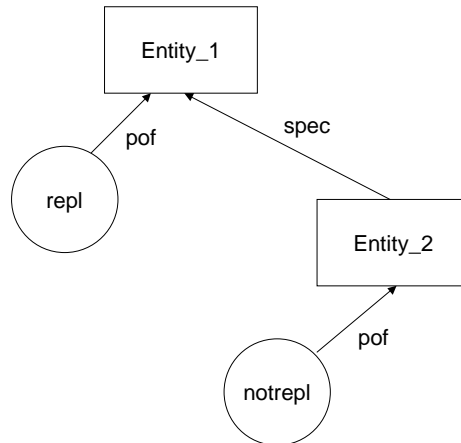


Figura A.10. Relación entre un individuo y un conjunto mediante el link spec

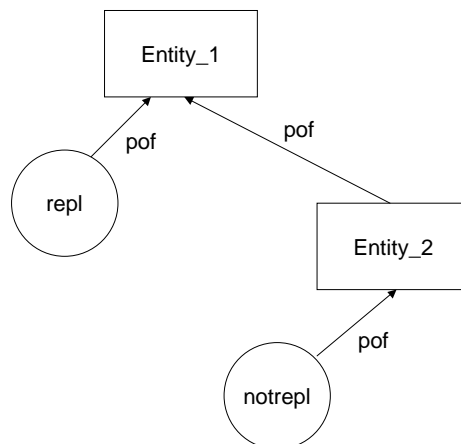


Figura A.11. Relación entre un individuo y un conjunto mediante el link pof

Sin embargo, ambas formas de indicar la pertenencia a un conjunto son insatisfactorias. El link spec es aplicable a priori, pero es difícil de entender que un individuo especialice un conjunto en forma alguna. La utilización del link pof es peor todavía, porque la pertenencia no se puede entender como una agregación de ningún modo como una agregación.

Adicionalmente, indicar la pertenencia a un conjunto como spec o pof complicaría enormemente la derivación del modelo canónico, ya que haría produciría que los links spec o pof poseyeran semánticas distintas en función del modelo conceptual que se pretendiera derivar. Por ejemplo, supóngase que el

ejemplo de la figura A.12 fuera a derivarse a un diagrama de clases. En este caso, se debería ignorar la especialización, ya que un diagrama de clases no puede (al menos inicialmente) considerar individuos, sino conjuntos, tal y como muestra la figura A.12. No obstante, ello ocurriría únicamente en este caso, ya que el link spec si puede derivarse a un diagrama de clases en caso de especialización entre clases.

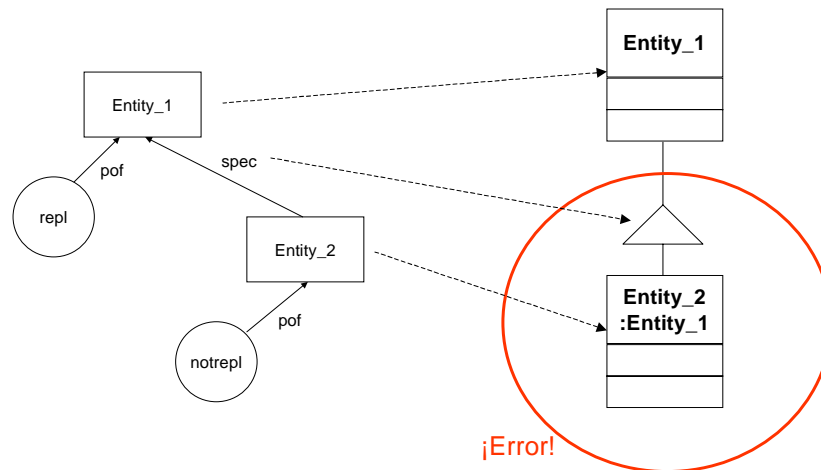


Figura A.12. Derivación del link spec a un diagrama de clases

Ahora bien, si la derivación fuera a realizarse hacia un Diagrama de secuencia, el link spec si debería ser considerado si debería considerarse, tal y como muestra la figura A.13, ya que en el diagrama de secuencia si se tienen en cuenta objetos individuales.

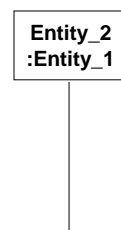


Figura A.13. Derivación del link spec a un diagrama de secuencia

Por ello, la mejor alternativa es definir un nuevo link que indica la pertenencia a conjuntos. Este link, denominado *bel* (de *belongs*), permite representar sin ambigüedad la pertenencia de un individuo a un conjunto.

A.4.4. SUBCONJUNTOS

Idénticas consideraciones que en el caso anterior (utilización de los links spec y pof) pueden establecerse para la inclusión de subconjuntos. Por ello, se ha optado por introducir un nuevo link,

denominado *subs* (de *subset*). Este link permite representar sin ambigüedad la inclusión de conjuntos.

A.4.5. RELACIONES ENTRE ENTIDADES Y CARDINALIDAD

En el lenguaje propuesto por Davis, no se han considerado adecuadamente las relaciones que ocurren entre conjuntos (esto es, entidades u objetos en los modelos conceptuales clásicos). Como el lenguaje propuesto por Davis se deriva a partir de las características de los modelos conceptuales clásicos, las relaciones y la cardinalidad de las mismas se ha considerado como un par de restricciones que enlazan las entidades participantes en la relación, tal y como muestra la figura A.14.

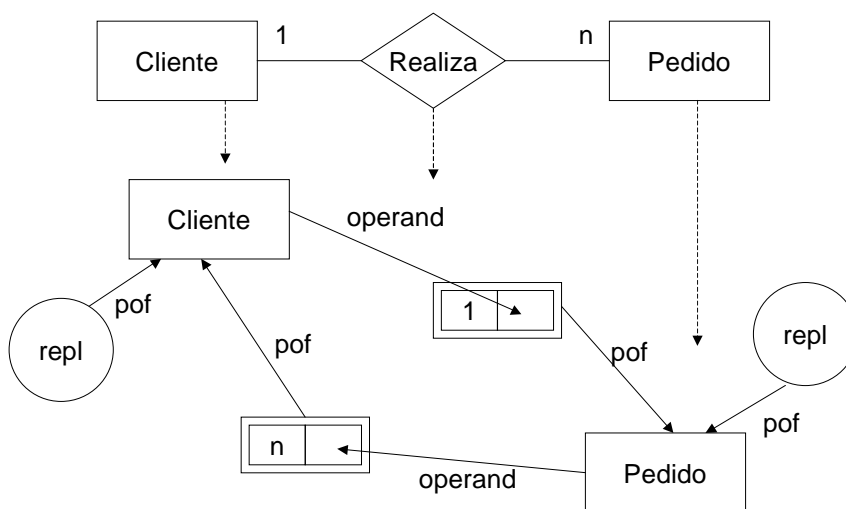


Figura A.14. Relación y cardinalidad en el lenguaje propuesto por Davis

La estructura de las relaciones y cardinalidad de la figura A.14 es muy improbable que aparezca en el MCG, al igual que en el caso de la invocación de procesos, y sin embargo es muy sencillo que aparezca una asociación que denota una relación. Es por ello que se ha introducido un nuevo link, denominado *rel* (de *relation*) que sustituye al par de restricciones de la figura A.14.

De igual modo que en la invocación de procesos, si la estructura de la figura A.14 apareciese en el Modelo canónico, se derivaría a la estructura correspondiente del modelo conceptual clásico seleccionado.

Asimismo, es claro que el link *rel* pierde cierta información acerca de la relación, en concreto, la cardinalidad. No obstante, ello no implica un gran problema, ya que la cardinalidad de las relaciones puede establecerse a posteriori, igual que en el caso de la invocación de procesos.

A.4.6. SIMPLIFICACIÓN DEL LINK STIMULUS

En el lenguaje propuesto por Davis, existen tres tipos de link stimulus:

- stimulus↑

- stimulus↓
- stimulus

La semántica de estos links es distinta. El link stimulus↑ desencadena una transición cuando el elemento que referencia se *activa*; el link stimulus↓ desencadena una transición cuando el elemento que referencia se *desactiva* y, finalmente, el link stimulus desencadena una transición cuando el elemento que referencia *aparece*. A efectos de derivar Modelo canónico, únicamente es necesario considerar un único link stimulus, ya que los Modelos conceptuales clásicos no admiten la distinción stimulus↑ - stimulus↓ - stimulus.

A.4.7. CONCEPTOS ENLAZADOS A PROPOSICIONES

En el MCG es posible (y de hecho, es sumamente frecuente) que un concepto se predique sobre una o varias proposiciones, debido al mecanismo de abstracción. Igualmente, existe la posibilidad de que un predicado se defina sobre una o varias proposiciones.

En ambos casos, al derivar el Modelo canónico, se obtendrá un elemento (el que resulte de la interpretación del concepto) unido por un link a un conjunto de conceptos (elementos) y asociaciones (links).

Desde el punto de vista del lenguaje definido por Davis, el primer hecho puede suceder. Sin embargo, el segundo (un predicado definido sobre una o varias proposiciones), no. El motivo es que la semántica del predicado es difícilmente comprensible.

Sin embargo, en el presente trabajo de Tesis, se ha considerado que para facilitar la aproximación MCG – Modelo canónico, es necesario permitir tal tipo de estructura. Las dificultades semánticas pueden solventarse si se considera que un predicado se cumple cuando:

- Todas las asociaciones de tipo estructural se cumplen, esto es, las asociaciones de tipo estructural existen en el modelo conceptual clásico.
- Todas las asociaciones de asignación de valores se ejecutan.
- Todos los predicados se evalúan ciertos.
- Todas las funciones terminan y generan un resultado correcto.

El tipo de semántica indicada anteriormente mezcla componentes estáticos y dinámicos, y será habitualmente complicado asegurar que el predicado se cumple, sobre todo cuando dicho predicado es interpretado como una restricción. Por ello, lo más fácil será olvidar las condiciones de cumplimiento del predicado durante la derivación del modelo conceptual clásico e incidir en ellas durante la corrección del modelo o incluso durante el diseño.

A.4.8. COMPOUNDS

En el lenguaje propuesto por Davis, existe la posibilidad de utilizar lo que denomina compounds. Un

compound es un mecanismo de expresión que permite reflejar la ambigüedad existente en algunos modelos conceptuales clásicos cuando no especifican claramente el tipo al que pertenece un determinado constructor. Por ejemplo, en un DFD un flujo de datos puede transportar un valor, todos los valores de una entidad o varias entidades, datos de control, etc., esto es, el tipo de constructor flujo de datos no está definido.

En el presente trabajo de Tesis, no es necesario utilizar los compounds debido a que, al ir desde el MCG al Modelo canónico, los tipos de todos los elementos y links están bien definidos gracias al proceso de interpretación.

A.4.9. ASIGNACIÓN DE VALORES

En el lenguaje propuesto por Davis, es posible asignar un valor a un elemento en tiempo de inicialización del sistema (tiempo de especificación), pero no describir la evolución de los valores de dicho elemento, ya que los modelos conceptuales que Davis utilizó como base de su estudio no permitían expresar tal evolución.

El MCG no posee tal restricción, y es posible describir la secuencia de valores de un concepto determinado en el tiempo si se considerase necesario. Por ello, es necesario modificar el concepto de asignación de valores de Davis (link hval), y permitir que especifique valores a elementos no únicamente en tiempo de especificación (esto es inicialización del sistema), sino también durante el funcionamiento del mismo.

A.4.10. LINK EQUIVALENCE

Para Davis, el link equivalence posee dos significados:

- Indicar que dos elementos son realmente el mismo, aunque posean igual nombre.
- Indicar que dos elementos poseen igual valor o, dicho de otra forma, que los statespaces que los componen poseen valores iguales.

El primero de los casos posee la misma semántica que el operador EQ, mientras que el segundo se refiere al predicado de igualdad. Por ello, el link equivalence no se utilizará, para evitar redundancia, en el presente trabajo de Tesis.

Anexo B. Tablas IMCI

La Técnica de Identificación del Modelo Conceptual Idóneo (Técnica IMCI), utiliza un conjunto de tablas. Dichas tablas permiten identificar qué modelos conceptuales permiten la expresión de las proposiciones recogidas en el Modelo Canónico de Requisitos (MCR).

Las tablas IMCI son tablas de doble entrada, existiendo una tabla distinta por cada link del Modelo Canónico. Cada fila de una tabla enumera todos los elementos del Modelo Canónico. Asimismo, dicha enumeración se repite en cada columna. Las tablas construidas de esta forma permiten indicar en cada celda qué modelo conceptual permite la expresión de una proposición interpretada como elemento-link-elemento. Adicionalmente, las tablas IMCI también permiten identificar el Modelo Conceptual Idóneo en el caso de que el Modelo Canónico de Requisitos contenga proposiciones construidas recursivamente.

Las distintas tablas IMCI se presentan en las tablas B.1 a B.25.

Nótese que los modelos conceptuales considerados en las tablas IMCI no son los únicos utilizables en MAON. Concretamente, las tablas IMCI se han construido, únicamente, para el Diagrama de Flujo de Datos (DFD), Diagrama Entidad-Relación (ER), Diagrama de Clases (DC), Diagrama de Transición de Estados (DTE), Statecharts (STC), Casos de Uso (CU) y Diagrama de Flujo de Datos/Tiempo Real (DFDRT). No obstante, tanto el Modelo Canónico como las Tablas IMCI pueden evolucionar, mediante la agregación de elementos y links, y su asociación a los diversos constructores de otros modelos conceptuales, existentes o que se desarrollen en el futuro.

	entity [repl]	entity [notrepl]	process	predicate	transition	message	constraint	value	statespace
entity [repl]	DC								
entity [notrepl]	DC	DC							
process			CU						
predicate									
transition									
message									
constraint									
value									
statespace									STC

Tabla B.1. Tabla IMCI para el link spec

	entity [repl]	entity [notrepl]	process	predicate	transition	message	constraint	value	statespace
entity [repl]	DC								
entity [notrepl]	DC	DC							
process	DC	DC	CU						
predicate									
transition									
message	---	---							
constraint			---						
value			---					---	
statespace	DC ER	DC ER	---						STC

Tabla B.2. Tabla IMCI para el link pof (parte 1/3)

statespace	value	constraint	message	transition	predicate	process	entity [notrepr]	entity [repr]
								message
								receives
								entity [repr]
								-
								message
								sends
								entity [notrepr]
								-
								message
								receives
								entity [notrepr]
								-
								message
								sends
								process
								-
								message
								receives
								predicate
								-
								message
								operand
								stimulus
								transition
								-
								message
								operand
								constraint
								value
								-
								process
								-
								process
								receives
								value
								-
								value
								operand
								pof
								-
								value
								stimulus
								transition
								-
								message
								operand
								constraint
								value
								pof
								-
								statespace
								pof
								entity [repr]
								-
								statespace
								entity [notrepr]
								-
								Statespace
								sends
								process
								-
								Statespace
								receives
								process
								pof
								-
								Statespace
								operand
								predicate
								-
								Statespace
								stimulus
								transition
								-
								Statespace
								response
								message
								operand
								constraint
								value
								hval
								Statespace
								spec
								statespace
								value
								constraint
								message
								Statespace
								DC
								FR
								ER

Tabla B.4. Tabla IMCI para el link pof (parte 3/3)

	entity [repl]	entity [notrepl]	process	predicate	transition	message	constraint	value	statespace
entity [repl]	DC								
entity [notrepl]	DC	DC							
process									
predicate									
transition									
message									
constraint									
value									
statespace									

Tabla B.5. Tabla IMCI para el link subs

	entity [repl]	entity [notrepl]	process	predicate	transition	message	constraint	value	statespace
entity [repl]	DC ER								
entity [notrepl]	DC ER	DC ER							
process									
predicate									
transition									
message									
constraint									
value									
statespace									

Tabla B.6. Tabla IMCI para el link rel (parte 1/2)

	entity [repl]	entity [notrepl]	process	predicate	transition	message	constraint	value	statespace
entity [repl]	---								
entity [notrepl]	---	---							
process	DFD DFDTR	CU DFD DFDTR	DFDTR DFD						
predicate			---						
transition									
message									
constraint									
value									
statespace									

Tabla B.8. Tabla IMCI para el link activate (parte 1/2)

	entity [repl]	entity [notrepl]	process	predicate	transition	message	constraint	value	statespace
entity [repl]									
entity [notrepl]									
process									
predicate	---	---		---					
transition									
message				---					
constraint	---	---		---	---	--	---		
value				---		DFDTR	---		
statespace				---		DFDTR	---		

Tabla B.10. Tabla IMCI para el link operand (parte 1/3)

entity [repl]								entity [repl]	spec	entity [repl]
entity [notrepl]								entity [repl]	subs	entity [repl]
process								entity [repl]	pof	entity [repl]
predicate								entity [repl]	rel	entity [repl]
transition								entity [repl]	activate	entity [repl]
message						---		entity [repl]	spec	entity [repl]
constraint								entity [notrepl]	subs	entity [repl]
value								entity [notrepl]	pof	entity [repl]
statespace								entity [notrepl]	rel	entity [repl]
								entity [notrepl]	activate	entity [repl]
								entity [notrepl]	spec	entity [notrepl]
								entity [notrepl]	subs	nity [notrepl]
								entity [notrepl]	pof	entity [notrepl]
								entity [notrepl]	rel	entity [notrepl]
								entity [notrepl]	activate	entity [repl]
								entity [notrepl]	spec	entity [notrepl]
								entity [notrepl]	subs	nity [notrepl]
								entity [notrepl]	pof	entity [notrepl]
								entity [notrepl]	rel	entity [notrepl]
								entity [notrepl]	activate	entity [notrepl]
								process	pof	entity [repl]
								process	sends	entity [repl]
						---		process	receives	entity [repl]
								process	-activate	entity [repl]
						DFDTR		process	pof	entity [notrepl]
								process	sends	entity [notrepl]
								process	receives	entity [notrepl]
								process	-activate	entity [notrepl]
						DFDTR		process	spec	process
								process	pof	process
								process	activate	process
						DFDTR		predicate	operand	Entity [repl]
								predicate	operand	entity [notrepl]
								predicate	activate	process
						---		predicate	operand	predicate
								transition	stimulus	process
								transition	response	process
								transition	stimulus	predicate
								transition	stimulus	transition
								transition	response	transition
								message	-sends	entity [repl]

Tabla B.11. Tabla IMCI para el link operand (parte 2/3)

statespace	value	constraint	message	transition	predicate	process	entity [notrepl]	entity [repl]
			message				entity [repl]	-receives
			message				entity [notrepl]	-sends
			message				entity [notrepl]	-receives
			message				process	-sends
			message				process	-receives
			message				predicate	-operand
			message				transition	-stimulus
			message				transition	-response
			message				message	pof
			constraint				entity [repl]	operand
		∃	constraint				entity [notrepl]	operand
		∃	constraint				process	pof
		∃	constraint				predicate	operand
		∃	constraint				transition	operand
		∃	Constraint				message	operand
		∃	constraint				constraint	operand
			value		∃		process	-sends
			value		∃		process	-receives
			value		∃		process	pof
			value		∃		predicate	-operand
			value		∃		transition	-stimulus
			value		∃		transition	-response
			value		∃		message	-operand
			value		∃		constraint	-operand
			value		∃		value	pof
			statespace		∃		entity [repl]	pof
			statespace		∃		entity [notrepl]	pof
			Statespace		∃		process	-sends
			Statespace		∃		process	-receives
			Statespace		∃		process	pof
			Statespace		∃		predicate	-operand
			Statespace		∃		transition	-stimulus
			Statespace		∃		transition	-response
			Statespace		∃		message	-operand
			Statespace		∃		constraint	-operand
			Statespace		∃		value	hval
			Statespace		∃		statespace	spec
			Statespace		∃		statespace	pof

Tabla B.12. Tabla IMCI para el link operand (parte 3/3)

	entity [repl]	entity [notrepl]	process	predicate	transition	message	constraint	value	statespace
entity [repl]									
entity [notrepl]									
process									
predicate									
transition			---	STC	---				
message					DTE STC				
constraint									
value					---				
statespace					---				

Tabla B.13. Tabla IMCI para el link stimulus (parte 1/3)

entity [repl]	entity [notrepl]	process	predicate	transition	message	constraint	value	statespace
message	-receives	entity [repl]						
message	-sends	entity [notrepl]						
message	-receives	entity [notrepl]						
message	-sends	process						
message	-receives	process						
message	-operand	predicate						
message	-stimulus	transition						
message	-response	transition						
message	pof	message						
constraint	operand	entity [repl]						
constraint	operand	entity [notrepl]						
constraint	pof	process						
constraint	operand	predicate						
constraint	operand	transition						
Constraint	operand	message						
constraint	operand	constraint						
value	-sends	process						
value	-receives	process						
value	pof	process						
value	-operand	predicate						
value	-stimulus	transition						
value	-response	transition						
value	-operand	message						
value	-operand	constraint						
value	pof	value						
statespace	pof	entity [repl]						
statespace	pof	entity [notrepl]						
Statespace	-sends	process						
Statespace	-receives	process						
Statespace	pof	process						
Statespace	-operand	predicate						
Statespace	-stimulus	transition						
Statespace	-response	transition						
Statespace	-operand	message						
Statespace	-operand	constraint						
Statespace	hval	value		STC				
Statespace	spec	statespace						
Statespace	pof	statespace						

Tabla B.15. Tabla IMCI para el link stimulus (parte 3/3)

	entity [repl]	entity [notrepl]	process	predicate	transition	message	constraint	value	statespace
entity [repl]									
entity [notrepl]									
process									
predicate									
transition			---		---				
message					---				
constraint									
value					---				
statespace					---				

Tabla B.16. Tabla IMCI para el link response (parte 1/3)

entity [repl]	entity [notrepl]	process	predicate	transition	message	constraint	value	statespace
message	entity [repl]	-receives						
message	entity [notrepl]	-sends						
message	entity [notrepl]	-receives						
message	process	-sends						
message	process	-receives						
message	predicate	-operand						
message	transition	-stimulus						
message	transition	-response						
message	message	pof						
constraint	entity [repl]	operand						
constraint	entity [notrepl]	operand						
constraint	process	pof						
constraint	predicate	operand						
constraint	transition	operand						
constraint	message	operand						
constraint	constraint	operand						
value	process	-sends		..				
value	process	-receives		..				
value	process	pof						
value	predicate	-operand						
value	transition	-stimulus						
value	transition	-response						
value	message	-operand						
value	constraint	-operand						
value	value	pof						
statespace	entity [repl]	pof						
statespace	entity [notrepl]	pof						
Statespace	process	-sends						
Statespace	process	-receives						
Statespace	process	pof						
Statespace	predicate	-operand						
Statespace	transition	-stimulus						
Statespace	transition	-response						
Statespace	message	-operand						
Statespace	constraint	-operand						
Statespace	value	hval					DTE STC	
Statespace	statespace	spec						
Statespace	statespace	pof						

Tabla B.18. Tabla IMCI para el link response (parte 3/3)

	entity [repl]	entity [notrepl]	process	predicate	transition	message	constraint	value	statespace
entity [repl]									
entity [notrepl]									
process									
predicate									
transition									
message	---	---	DFDTR						
constraint			---						
value	DFD DFDTR	DFD DFDTR	DFD DFDTR						
statespace	DFD DFDTR	DFD DFDTR	DFD DFDTR						

Tabla B.19. Tabla IMCI para el link sends (parte 1/3)

entity [repl]									entity [repl]	spec	entity [repl]	entity [repl]
entity [notrepl]										subs	entity [repl]	entity [repl]
process		DFD								pof	entity [repl]	entity [repl]
predicate		DFDTR								rel	entity [repl]	entity [repl]
transition										activate	entity [repl]	entity [repl]
message										spec	entity [repl]	entity [repl]
constraint										subs	entity [repl]	entity [repl]
value										pof	entity [repl]	entity [repl]
statespace										rel	entity [repl]	entity [repl]
										activate	entity [repl]	entity [repl]
										spec	entity [repl]	entity [repl]
										subs	entity [repl]	entity [repl]
										pof	entity [repl]	entity [repl]
										rel	entity [repl]	entity [repl]
										activate	entity [repl]	entity [repl]
										spec	entity [notrepl]	entity [notrepl]
										subs	ntity [notrepl]	ntity [notrepl]
										pof	entity [notrepl]	entity [notrepl]
										rel	entity [notrepl]	entity [notrepl]
										activate	entity [notrepl]	entity [notrepl]
										pof	entity [repl]	entity [repl]
										sends	entity [repl]	entity [repl]
										receives	entity [repl]	entity [repl]
										activate	entity [repl]	entity [repl]
										pof	entity [notrepl]	entity [notrepl]
										sends	entity [notrepl]	entity [notrepl]
										receives	entity [notrepl]	entity [notrepl]
										activate	entity [notrepl]	entity [notrepl]
										spec	process	process
										pof	process	process
										activate	process	process
										operand	Entity [repl]	Entity [repl]
										operand	entity [notrepl]	entity [notrepl]
										activate	process	process
										operand	predicate	predicate
										stimulus	process	process
										response	process	process
										stimulus	predicate	predicate
										stimulus	transition	transition
										response	transition	transition
										sends	entity [repl]	entity [repl]
										message		

Tabla B.20. Tabla IMCI para el link sends (parte 2/3)

entity [repl]	DFDTR	DFDTR								message	-receives	entity [repl]
entity [notrepl]	DFDTR	DFDTR								message	-sends	entity [notrepl]
process								DFD DFDTR		message	-receives	entity [notrepl]
predicate										message	-sends	process
transition										message	-sends	process
message										message	-receives	predicate
constraint										message	-operand	predicate
value										message	-stimulus	transition
statespace										message	-response	transition
										message	po	message
										constraint	operand	entity [repl]
										constraint	operand	entity [notrepl]
										constraint	po	process
										constraint	operand	predicate
										constraint	operand	transition
										constraint	operand	message
										value	-sends	constraint
										value	-receives	process
										value	po	process
										value	-operand	predicate
										value	-stimulus	transition
										value	-response	transition
										value	-operand	message
										value	-operand	constraint
										value	po	value
										statespace	po	entity [repl]
										statespace	po	entity [notrepl]
										Statespace	-sends	process
										Statespace	-receives	process
										Statespace	po	process
										Statespace	-operand	predicate
										Statespace	-stimulus	transition
										Statespace	-response	transition
										Statespace	-operand	message
										Statespace	-operand	constraint
										Statespace	hval	value
										Statespace	spec	statespace
										Statespace	po	statespace

Tabla B.21. Tabla IMCI para el link sends (parte 3/3)

	entity [repl]	entity [notrepl]	process	predicate	transition	message	constraint	value	statespace
entity [repl]									
entity [notrepl]									
process									
predicate									
transition									
message	---	---	DFDTR						
constraint			---						
value	DFD DFDTR	DFD DFDTR	DFD DFDTR						
statespace	DFD DFDTR	DFD DFDTR	DFD DFDTR						

Tabla B.22. Tabla IMCI para el link receives (parte 1/3)

entity [repl]									entity [repl]	receives-	message	entity [repl]
entity [notrepl]		DFDTR							entity [notrepl]	-sends	message	entity [notrepl]
process	DFDTR								entity [notrepl]	-receives	message	entity [notrepl]
predicate									process	-sends	message	process
transition									process	-receives	message	process
message									predicate	-operand	message	predicate
constraint									transition	-stimulus	message	transition
value									transition	-response	message	transition
statespace									message	pof	message	message
									constraint	operand	constraint	entity [repl]
									entity [notrepl]	operand	constraint	entity [notrepl]
									process	pof	constraint	process
									predicate	operand	constraint	predicate
									transition	operand	message	transition
									message	operand	message	transition
									constraint	operand	constraint	message
									value	-sends	value	constraint
						DFD	DFDTR		process		value	process
									process	-receives	value	process
									process	pof	value	process
									predicate	-operand	value	predicate
									transition	-stimulus	value	transition
									transition	-response	value	transition
									message	operand	value	message
									constraint	-operand	value	message
									value	pof	value	constraint
								DFD	value		value	value
						DFD	DFDTR		value		value	value
									statespace	pof	statespace	statespace
									entity [repl]		statespace	statespace
									entity [notrepl]		statespace	statespace
									process	-sends	statespace	statespace
									process	-receives	statespace	statespace
									process	pof	statespace	statespace
									predicate	operand	statespace	statespace
									transition	-stimulus	statespace	statespace
									transition	-response	statespace	statespace
									message	operand	statespace	statespace
									constraint	-operand	statespace	statespace
									value	hval	statespace	statespace
									statespace	spec	statespace	statespace
									statespace	pof	statespace	statespace

Tabla B.24. Tabla IMCI para el link receives (parte 3/3)

	entity [repl]	entity [notrepl]	process	predicate	transition	message	constraint	value	statespace
entity [repl]									
entity [notrepl]									
process									
predicate									
transition									
message									
constraint									
value									
statespace								DTE STC	

Tabla B.25. Tabla IMCI para el link hval

Anexo C. Tablas DMCS

Para la correcta aplicación de la Técnica de Derivación del Modelo Conceptual Seleccionado (Técnica DMCS), es necesaria la utilización de una serie de tablas. Dichas tablas poseen como finalidad la derivación de fragmentos del modelo conceptual idóneo a partir de cada una de las proposiciones del Modelo Canónico de Requisitos (MCR). Para ello, cada tabla contiene un subconjunto de las proposiciones del Modelo Canónico (aquellas que el modelo conceptual es capaz de expresar), junto con los fragmentos correspondientes del modelo conceptual que expresan cada una de dichas las proposiciones.

Existe una muy clara relación ente las tablas IMCI y las tablas DMCS. Dicha relación se origina por el hecho de que son dos caras de la misma moneda: Si un modelo conceptual puede expresar una determinada proposición del MCR (que es lo que se investiga con la aplicación de la Técnica IMCI), entonces dicha proposición puede “reescribirse” en dicho modelo (que es lo que se consigue tras la aplicación de la Técnica DMCS).

Las tablas DMCS se construyen para cada modelo conceptual, esto es, una tabla para el diagrama de flujo de datos, otra distinta para el diagrama de clases, etc. En el presente apéndice únicamente se han desarrollado las tablas de los modelos conceptuales más utilizados en la práctica, los cuales son:

- Diagrama de clases
- Diagrama entidad-relación
- Diagrama de flujo de datos
- Diagrama de transición de estados
- Statechart
- Casos de Uso

Las tablas DMCS para los modelos conceptuales citados anteriormente se muestran en las tablas C.1 a C.14.

Las reglas de derivación para cada modelo conceptual se indican, asimismo, en los cuadros C1 a C2.

Adicionalmente, debe notarse que los modelos conceptuales considerados en las tablas DMCS no son los únicos utilizables en MAON. Tanto el Modelo Canónico como las Tablas DMCS pueden evolucionar, mediante la agregación de elementos y links, y su asociación a los diversos constructores de otros modelos conceptuales, existentes o que se desarrollen en el futuro.

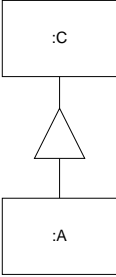

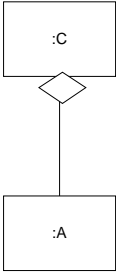
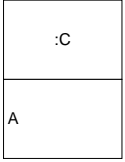
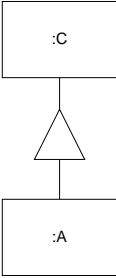
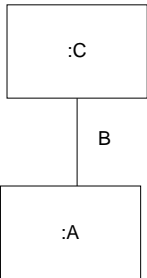
Elemento (A)	Link (B)	Elemento (C)	Se Deriva A:	Elemento (A)	Link (B)	Elemento (C)	Se Deriva A:
Entity[repl]	spec	Entity[repl]		Process	pof	Entity [repl]	
Entity[repl]	pof	Entity[repl]		Statespace	pof	Entity[repl]	
Entity[repl]	subs	Entity[repl]		Entity[repl]	rel	Entity[repl]	

Tabla C.1. Tabla DMCS para el diagrama de clases (parte 1 de 2)

Elemento (A)	Link (B)	Proposición CDE	Se Deriva A:	Elemento (A)	Link (B)	Proposición CDE	Se Deriva A:
Entity[rep]	pof	Entity[rep] Rel Entity[rep]		Statespace]	pof	Entity[rep] Rel Entity[rep]	
Entity[rep]	-pof	Statespace pof Statespace		Entity[rep]	rel	Entity[rep] Rel Entity[rep]	
Entity[rep]	-pof	Statespace spec Statespace					

Tabla C.2. Tabla DMCS para el diagrama de clases (parte 2 de 2)

1. Si un elemento A del tipo Entity[notrep] pertenece (Bel) a un elemento B del tipo Entity[rep], reemplazar A con B.
2. Si un elemento A del tipo Entity[rep] es un subconjunto (Subs) de un elemento B del tipo Entity[rep], reemplazar A con B.
3. Si un elemento A del tipo Entity[notrep] no satisface las (2) o (3), comprobar si sería coherente considerarlo como de tipo Entity[rep].

Cuadro C.1. Reglas de derivación para el diagrama de clases

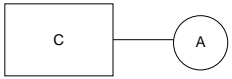
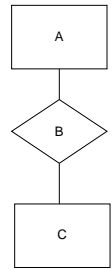
Elemento (A)	Link (B)	Elemento (C)	Se Deriva A:	Elemento (A)	Link (B)	Elemento (C)	Se Deriva A:
Statespace	pof	Entity[repl]		Entity[repl]	rel	Entity[repl]	

Tabla C.3. Tabla DMCS para el diagrama entidad-relación (parte 1 de 2)

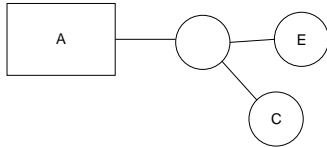
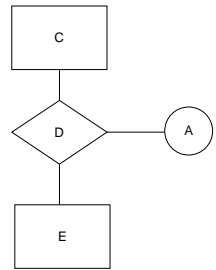
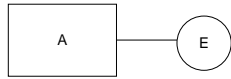
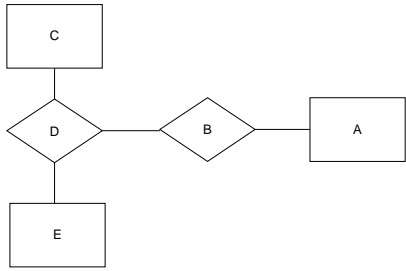
Elemento (A)	Link (B)	Proposición CDE	Se Deriva A:	Elemento (A)	Link (B)	Proposición CDE	Se Deriva A:
Entity[repl]	-pof	Statespace spec Statespace		Statespace	pof	Entity[repl] rel Entity[repl]	
Entity[repl]	-pof	Statespace pof Statespace		Entity [repl]]	rel	Entity[repl] rel Entity[repl]	

Tabla C.4. Tabla DMCS para el diagrama entidad-relación (parte 2 de 2)

1. Si un elemento A del tipo Entity[notrepl] pertenece (Bel) a un elemento B del tipo Entity[repl], reemplazar A con B.
2. Si un elemento A del tipo Entity[repl] es un subconjunto (Subs) de un elemento B del tipo Entity[repl], reemplazar A con B.
3. Si un elemento A del tipo Entity[notrepl] no satisface las (2) o (3), comprobar si sería coherente considerarlo como de tipo Entity[repl].

Cuadro C.2. Reglas de derivación para el diagrama entidad-relación

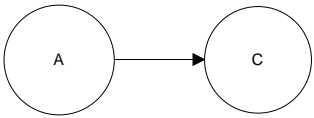
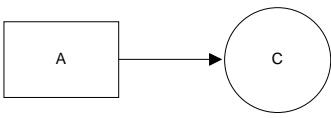
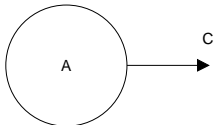
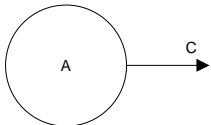
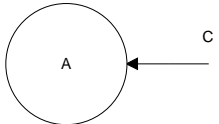
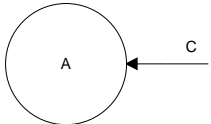
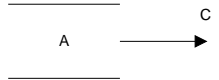
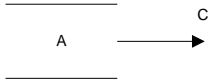
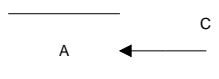
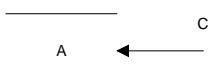
Elemento (A)	Link (B)	Elemento (C)	Se Deriva A:	Elemento (A)	Link (B)	Elemento (C)	Se Deriva A:
Process	activate	Process		Entity[notrepl]	activate	Process	
Process	sends	value		Process	sends	Statespace	
Process	receives	value		Process	receives	Statespace	
Entity[repl]	sends	value		Entity[repl]	sends	Statespace	
Entity[repl]	receives	value		Entity[repl]	receives	Statespace	

Tabla C.5. Tabla DMCS para el diagrama de flujo de datos (parte 1 de 5)

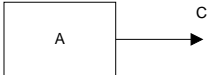
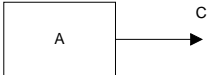
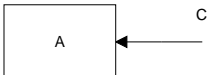
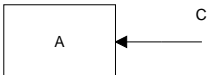
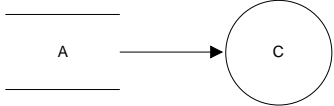
Elemento (A)	Link (B)	Elemento (C)	Se Deriva A:	Elemento (A)	Link (B)	Elemento (C)	Se Deriva A:
Entity[notrepl]	sends	value		Entity[notrepl]	sends	Statespace	
Entity[notrepl]	receives	value		Entity[notrepl]	receives	Statespace	
Entity[repl]	activate	Process					

Tabla C.6. Tabla DMCS para el diagrama de flujo de datos (parte 2 de 5)

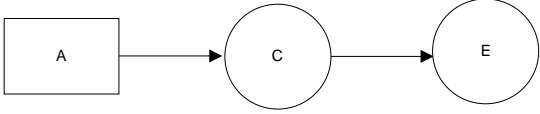
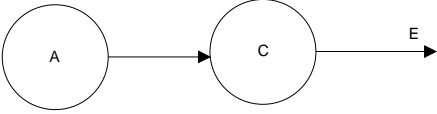
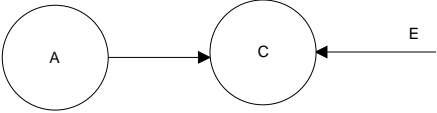
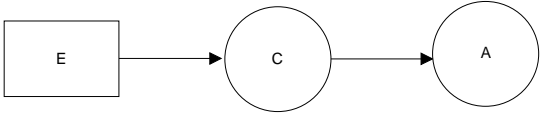
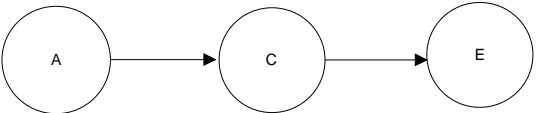
Elemento (A)	Link (B)	Proposición CDE	Se Deriva A:
Entity[repl]	activate	Process activate Process	
Process	Activate	Process sends Entity[repl]	
Process	activate	Process receives Entity[repl]	
Process	- activate	Process -activate Entity[repl]	
Process]	activate	Process activate Process	

Tabla C.7. Tabla DMCS para el diagrama de flujo de datos (parte 3 de 5)

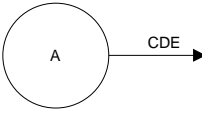
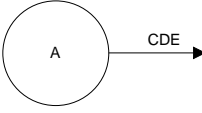
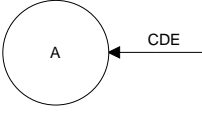
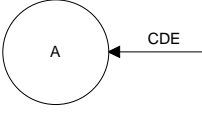
Elemento (A)	Link (B)	Proposición CDE	Se Deriva A:
Process	sends	Entity[rep] rel Entity[rep]	
Process	sends	Entity[rep] pof Entity[rep]	
Process	receives	Entity[rep] rel Entity[rep]	
Process	receives	Entity[rep] pof Entity[rep]	

Tabla C.8. Tabla DMCS para el diagrama de flujo de datos (parte 4 de 5)

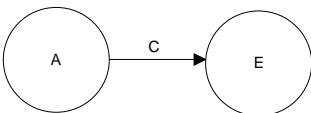
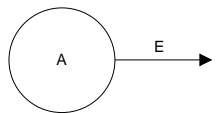
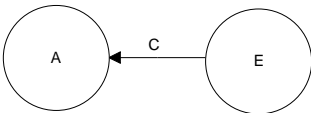
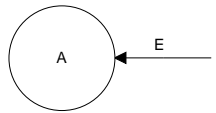
Elemento (A)	Link (B)	Proposición CDE	Se Deriva A:
Process	sends	Value -receives Process	 <pre> graph LR A((A)) -- C --> E((E)) </pre>
Process	sends	Value Pof Value	 <pre> graph LR A((A)) -- E --> Out[] </pre>
Process	receives	Value -sends Process	 <pre> graph LR E((E)) -- C --> A((A)) </pre>
Process	receives	Value Pof Value	 <pre> graph LR In[] -- E --> A((A)) </pre>

Tabla C.9. Tabla DMCS para el diagrama de flujo de datos (parte 5 de 5)

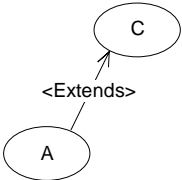
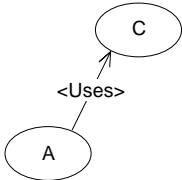
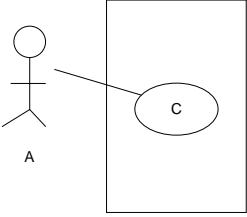
Elemento (A)	Link (B)	Elemento (C)	Se Deriva A:	Elemento (A)	Link (B)	Elemento (C)	Se Deriva A:
Process	spec	Process		Process	pof	Process	
Entity[notrep!]	activate	Process					

Tabla C.10. Tabla DMCS para los casos de uso

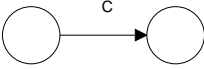
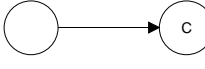
Elemento (A)	Link (B)	Elemento (C)	Se Deriva A:	Elemento (A)	Link (B)	Elemento (C)	Se Deriva A:
Transition	stimulus	Message		Statespace	hval	Value	

Tabla C.11. Tabla DMCS para el diagrama de transición de estados (parte 1 de 2)

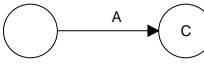
Elemento (A)	Link (B)	Proposición CDE	Se Deriva A:
Transition	Response	Statespace Hval value	

Tabla C.12. Tabla DMCS para el diagrama de transición de estados (parte 2 de 2)

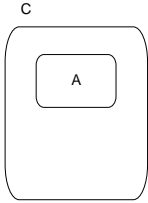
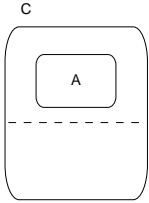
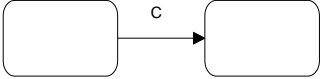
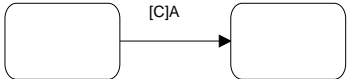

Elemento (A)	Link (B)	Elemento (C)	Se Deriva A:	Elemento (A)	Link (B)	Elemento (C)	Se Deriva A:
Statespace	spec	Statespace		Statespace	pof	Statespace	
Transition	stimulus	Message		Transition	stimulus	Predicate	
Statespace	hval	Value					

Tabla C.13. Tabla DMCS para el statechart (parte 1 de 2)

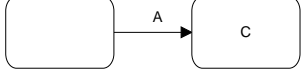
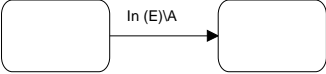
Elemento (A)	Link (B)	Proposición CDE	Se Deriva A:
Transition	Response	Statespace Hval value	 <p>A diagram showing a transition from an empty state to state 'c' labeled 'A'.</p>
Transition	stimulus	Statespace Hval value	 <p>A diagram showing a transition from an empty state to another empty state labeled 'In (E) \ A'.</p>

Tabla C.14. Tabla DMCS para el statechart (parte 2 de 2)