

**Departamento de Lenguajes y Sistemas Informáticos**  
Escuela Técnica Superior de Ingeniería Informática  
Universidad de Sevilla  
Avda. Reina Mercedes s/n. 41012 Sevilla  
Tif: (+34) 954 555 964, Fax: (+34) 954 557 139, Email: [buzon@lsi.us.es](mailto:buzon@lsi.us.es)



# **Una propuesta basada en el paradigma dirigido por modelos para la extracción de procesos del software desde sistemas heredados utilizando la perspectiva temporal**

TESIS DOCTORAL

Autor

D. Carlos Arévalo Maldonado

Directoras

Dra. D<sup>a</sup> María José Escalona Cuaresma

Dra. D<sup>a</sup> Isabel Ramos Román

Sevilla, Diciembre 2015



*La alegría está en la lucha, en el esfuerzo,  
en el sufrimiento que supone la lucha, y no en la victoria misma.  
Nuestra recompensa se encuentra en el esfuerzo y no en el resultado.  
Un esfuerzo total es una victoria completa.*

***Mohandas Karamchand Gandhi (1869-1948)***



# CONTENIDO

<b>Contenido .....</b>	<b>iii</b>
<b>Índice de definiciones .....</b>	<b>ix</b>
<b>Índice de expresiones .....</b>	<b>xi</b>
<b>Índice de figuras .....</b>	<b>xiii</b>
<b>Índice de tablas .....</b>	<b>xv</b>
<b>Agradecimientos .....</b>	<b>xvii</b>
<b>Resumen .....</b>	<b>xix</b>

## PARTE I. INTRODUCCIÓN

<b>Capítulo 1. Contexto de la investigación.....</b>	<b>25</b>
<b>1.1 Organizaciones del negocio del software (TI) .....</b>	<b>25</b>
<b>1.2 Motivación .....</b>	<b>29</b>
<b>1.3 Desarrollo del trabajo de investigación .....</b>	<b>36</b>
1.3.1 Contexto.....	36
1.3.2 Metodología .....	38
<b>1.4 Estructura de la memoria .....</b>	<b>41</b>
<b>1.5 Resumen y conclusiones .....</b>	<b>42</b>
<b>Capítulo 2. Retos y objetivos.....</b>	<b>43</b>
<b>2.1 Identificación del problema .....</b>	<b>44</b>
<b>2.2 Retos .....</b>	<b>46</b>
<b>2.3 Un ejemplo motivador .....</b>	<b>47</b>
2.3.1 Planificación de un proyecto como sistema origen .....	47
2.3.1.1 Restricciones temporales.....	48
2.3.1.2 Dependencias temporales .....	49
2.3.2 Proceso BPMN para el plan del proyecto .....	49
2.3.3 Discusión.....	52
<b>2.4 Hipótesis y objetivos .....</b>	<b>52</b>
2.4.1 Legacy systems como sistemas origen.....	53
2.4.2 Reglas temporales sobre procesos.....	55
2.4.3 Lenguajes para representación de procesos.....	56
2.4.4 Extracción de procesos desde legacy databases .....	57
2.4.5 Verificación del enfoque en un caso de estudio .....	58
2.4.6 Resumen hipótesis vs objetivos de la investigación.....	60

<b>2.5</b>	<b>Arquitectura de la solución .....</b>	<b>60</b>
<b>2.6</b>	<b>Resumen y conclusiones .....</b>	<b>62</b>

## **PARTE II. ANTECEDENTES**

<b>Capítulo 3.</b>	<b>Estado del arte .....</b>	<b>67</b>
--------------------	------------------------------	-----------

<b>3.1</b>	<b>Ingeniería del software dirigida por modelos.....</b>	<b>67</b>
3.1.1	Problemas del desarrollo tradicional de software.....	67
3.1.2	Model-Driven Engineering (MDE).....	71
<b>3.2</b>	<b>Gestión de procesos del software .....</b>	<b>78</b>
<b>3.3</b>	<b>Gestión de procesos de negocio (BPM) .....</b>	<b>78</b>
3.3.1	Conceptos.....	78
3.3.2	Ciclo de vida BPM.....	79
3.3.3	Beneficios del enfoque BPM .....	81
3.3.4	Estándares BPM.....	82
3.3.4.1	Business Process Definition Meta-Model (BPDM).....	84
3.3.4.2	Business Process Model and Notation (BPMN).....	85
3.3.4.3	Xml Process Definition Language (XPDL).....	87
3.3.4.4	Evolución histórica.....	87
3.3.5	Sistemas de gestión de procesos (PAIS) .....	88
<b>3.4</b>	<b>Navigational Development Techniques (NDT).....</b>	<b>91</b>
<b>3.5</b>	<b>Modernización de legacy systems .....</b>	<b>96</b>
3.5.1	Conceptos y problemas en legacy systems.....	96
3.5.2	Arquitectura ADM para modernización de sistemas.....	98
<b>3.6</b>	<b>Resumen y conclusiones .....</b>	<b>99</b>

<b>Capítulo 4.</b>	<b>Trabajo relacionado.....</b>	<b>101</b>
--------------------	---------------------------------	------------

<b>4.1</b>	<b>Introducción .....</b>	<b>101</b>
<b>4.2</b>	<b>Reglas temporales sobre procesos .....</b>	<b>102</b>
<b>4.3</b>	<b>Procesos de negocio del software .....</b>	<b>105</b>
<b>4.4</b>	<b>Modernización de legacy systems .....</b>	<b>106</b>
<b>4.5</b>	<b>Otras perspectivas BPM: Organizacional y Casos .....</b>	<b>109</b>
<b>4.6</b>	<b>Minería de procesos .....</b>	<b>110</b>
<b>4.7</b>	<b>Resumen y conclusiones .....</b>	<b>111</b>

<b>Capítulo 5.</b>	<b>Legacy systems en el sector de las TI .....</b>	<b>113</b>
--------------------	--	------------

<b>5.1</b>	<b>Sistemas de gestión en organizaciones orientadas al software .....</b>	<b>113</b>
5.1.1	Sistemas de planificación y control de proyectos.....	114
5.1.2	Sistemas de gestión documental (ECMs).....	115
5.1.3	ERPs, CRMs, SCMs y desarrollos a medida.....	115
<b>5.2</b>	<b>Perspectivas de procesos vs legacy systems .....</b>	<b>116</b>
<b>5.3</b>	<b>Dimensión temporal en un legacy system .....</b>	<b>117</b>
<b>5.4</b>	<b>Resumen y conclusiones .....</b>	<b>118</b>

# PARTE III. SOLUCIÓN PROPUESTA

<b>Capítulo 6. Reglas temporales sobre procesos .....</b>	<b>121</b>
<b>6.1 Perspectiva temporal en los procesos de negocio .....</b>	<b>121</b>
<b>6.2 Taxonomía de reglas temporales .....</b>	<b>122</b>
6.2.1 Restricciones temporales.....	123
6.2.2 Dependencias temporales.....	126
<b>6.3 Enfoques para la extensión de BPMN 2.0 con reglas temporales.....</b>	<b>127</b>
6.3.1 Capacidades y limitaciones de BPMN 2.0 para representar la perspectiva temporal.....	127
6.3.2 Extensiones a BPMN mediante decoradores.....	129
6.3.3 Patrones BPMN para representación de reglas temporales .....	132
<b>6.4 Meta–Modelos estándar para representación de procesos .....</b>	<b>136</b>
6.4.1 Meta–Modelo UML Activity Diagrams.....	136
6.4.2 Meta–Modelo BPMN 2.0.....	139
6.4.3 Meta–Modelo XPDL 2.2 .....	140
6.4.4 Comparación de Meta–Modelos de procesos .....	141
<b>6.5 Un Meta–Modelo de procesos con reglas temporales (PIM-up) .....</b>	<b>142</b>
6.5.1 Meta–Modelo UML de procesos .....	142
6.5.1.1 Diagrama de clases.....	142
6.5.1.2 Enumeraciones .....	144
6.5.1.3 Extensión del ciclo de vida de una actividad bajo CPM.....	145
6.5.2 Definición de la dimensión temporal con OCL.....	146
6.5.2.1 Restricciones sobre el ciclo de vida de procesos y actividades.....	146
6.5.2.2 Restricciones temporales sobre una actividad .....	148
6.5.2.3 Dependencias temporales entre actividades .....	152
<b>6.6 Resumen y conclusiones .....</b>	<b>153</b>
<b>Capítulo 7. Extracción de procesos desde Legacy Databases .....</b>	<b>155</b>
<b>7.1 Enfoque MDE para la extracción de procesos .....</b>	<b>155</b>
<b>7.2 Meta–Modelo de procesos (PIM-upl).....</b>	<b>159</b>
<b>7.3 Plataforma origen piloto: MS Project Server .....</b>	<b>160</b>
7.3.1 Arquitectura de MS Project Server .....	161
7.3.2 Meta–Modelo PSM de tareas.....	162
<b>7.4 Extracción de procesos desde Legacy Databases .....</b>	<b>164</b>
7.4.1 Extracción de modelos PSM→PIM-upl.....	165
7.4.1.1 Algoritmo .....	167
7.4.1.2 Reglas de Mapeo .....	168
7.4.2 Actividades replicadas en legacy systems.....	171
7.4.2.1 Identificación del problema.....	171
7.4.2.2 Resolución mediante extensión del Meta–Modelo PIM-upl .....	171
7.4.2.3 Escenario ejemplo .....	175
7.4.3 Transformación de modelos PIM-upl→CIM .....	178
7.4.3.1 Algoritmo.....	182
7.4.3.2 Reglas de Mapeo .....	183
<b>7.5 Resumen y conclusiones .....</b>	<b>185</b>
<b>Capítulo 8. Caso de estudio: Aqua-WS project .....</b>	<b>187</b>
<b>8.1 AQUA-WS Project.....</b>	<b>187</b>
8.1.1 Proyectos definidos en projectserver_Published_AQUA-WS .....	188

8.1.2	Extracción de tareas, restricciones y relaciones de precedencia.....	189
8.1.2.1	Extracción E1: tareas generales y fases de la planificación general .....	189
8.1.2.2	Extracción E2: Grupo de tareas ligadas a un subsistema .....	190
8.1.2.3	Extracción E3: Tarea elemental de la planificación general .....	191
8.1.2.4	Extracción E4: Patrón NDT ASI (Análisis del sistema) .....	193
<b>8.2</b>	<b>Generación de modelos de procesos .....</b>	<b>194</b>
8.2.1	Extracción E1: tareas generales y fases de la planificación general.....	197
8.2.2	Extracción E2: Grupo de tareas ligadas a un subsistema .....	198
8.2.3	Extracción E3: Tarea elemental de la planificación general .....	199
8.2.4	Extracción E4: Patrón NDT ASI.....	200
<b>8.3</b>	<b>Discusión de resultados .....</b>	<b>202</b>
8.3.1	Proceso general y fases del proyecto.....	202
8.3.2	Fase NDT ASI.....	203
8.3.3	Nivel de abstracción de los modelos .....	204
8.3.4	Extensión y reutilización de modelos.....	204
<b>8.4</b>	<b>Resumen y conclusiones .....</b>	<b>205</b>

## PARTE IV. CONCLUSIONES

<b>Capítulo 9.</b>	<b>Conclusiones y trabajo futuro.....</b>	<b>209</b>
--------------------	---	------------

<b>9.1</b>	<b>Contribuciones .....</b>	<b>209</b>
9.1.1	Estudio del estado del arte (ob.0.).....	209
9.1.2	Meta-Modelos de tareas para legacy system de uso habitual en «TI» (ob.1.).....	211
9.1.3	Taxonomía de reglas temporales (ob.2.).....	212
9.1.4	Meta-Modelo de procesos PIM-up con reglas temporales (ob.3.).....	212
9.1.5	Extensión al Meta-Modelo BPMN (ob.4.).....	212
9.1.6	Ingeniería inversa MDE para extraer procesos desde legacy systems (ob.5.).....	213
9.1.6.1	Meta-Modelo PIM-upl (ob.5.1).....	213
9.1.6.2	Algoritmo de transformación PSM→PIM-upl (ob.5.2).....	214
9.1.6.3	Algoritmo de transformación PIM-upl→CIM (ob.5.3).....	214
9.1.7	Verificación del enfoque en el caso AQUA-WS Project (ob.6).....	214
<b>9.2</b>	<b>Trabajo futuro.....</b>	<b>216</b>
<b>9.3</b>	<b>Conclusiones .....</b>	<b>217</b>
<b>Referencias .....</b>	<b>219</b>	

## PARTE V. ANEXOS

<b>Anexo I: Acrónimos .....</b>	<b>251</b>
<b>Anexo II: Meta-Modelos de tareas en legacy databases (LDB) .....</b>	<b>259</b>
<b>Anexo II.a: LDB MS Access/MS Project.....</b>	<b>261</b>
<b>Anexo II.b: LDB SQL*Server/MS Project Server Published instance .....</b>	<b>261</b>
<b>Anexo II.c: LDB PostgreSQL/ECM Alfresco .....</b>	<b>262</b>
<b>Anexo II.d: LDB MySQL/RedMine.....</b>	<b>264</b>
<b>Anexo III: Meta-Modelos relacionales SQL .....</b>	<b>265</b>



<b>Anexo III.a: Meta–Modelos relacionales OMG:IMM .....</b>	<b>267</b>
<b>Anexo III.b: Extensiones a Meta–Modelos relacionales. Restricciones .....</b>	<b>273</b>
<b>Anexo IV: Proyecto AQUA-WS.....</b>	<b>275</b>
<b>Anexo IV.a: Planificación del proyecto (MS Project) .....</b>	<b>277</b>
Anexo IV.a1. Planificación general del Proyecto (Tareas generales y subsistemas).....	277
Anexo IV.a2. Planificación Control Calidad Tarea (64): (ASI) Instalaciones y equipos .....	283
Anexo IV.a3. Patrón de Planificación para Control Calidad de la fase NDT ASI .....	284
<b>Anexo IV.b: Diagrama Relacional BD SQL*Server Published-AQUA .....</b>	<b>285</b>
<b>Anexo IV.c: Tablas BD SQL*Server Published-AQUA .....</b>	<b>286</b>
<b>Anexo V. Publicaciones y proyectos .....</b>	<b>299</b>
<b>Capítulos de libros .....</b>	<b>301</b>
<b>Publicaciones en revistas internacionales.....</b>	<b>302</b>
<b>Publicaciones en conferencias internacionales .....</b>	<b>305</b>
<b>Publicaciones en conferencias nacionales.....</b>	<b>306</b>
<b>Proyectos .....</b>	<b>307</b>
Proyectos de investigación .....	307
Proyectos de transferencia tecnológica.....	308
Redes nacionales .....	309
Redes internacionales .....	310



# ÍNDICE DE DEFINICIONES

Definición. 1.1. Proceso de negocio	30
Definición. 1.2. Business Process Reengineering (BPR)	30
Definición. 1.3. Business Process Management (BPM)	31
Definición. 1.4. Ingeniería del Software	31
Definición. 1.5. Proceso software	32
Definición. 1.6. Interoperabilidad (IEEE)	33
Definición. 1.7. Interoperabilidad (CEE)	33
Definición. 3.1. Modelo	71
Definición. 3.2. Meta-Modelo	72
Definición. 3.3. Instancia de un proceso	79
Definición. 3.4. Traza de ejecución de un proceso	79
Definición. 3.5. Process Aware Information System (PAIS)	88



# ÍNDICE DE EXPRESIONES

Expresión. 6.1. Duración de una actividad perteneciente a un proceso	124
Expresión. 6.2. Duración flexible de una actividad de un proceso	124
Expresión. 6.3. Restricciones temporales de inicio y terminación flexibles	125
Expresión. 6.4. Restricción temporal de cardinalidad	126
Expresión. 6.5. Restricción de ausencia de una actividad	126
Expresión. 6.6. Dependencia de ausencia entre actividades	127
Expresión. 6.7. Dependencias temporales SS, SF, FS, FF	127



# ÍNDICE DE FIGURAS

Figura. 1.1. Extracción de procesos de legacy systems	36
Figura. 1.2. Cronología del desarrollo de la tesis	38
Figura. 2.1. Un ejemplo de planificación de actividades con MS Project	48
Figura. 2.2. Restricciones temporales sobre tareas en MS Project, p. ej.: MSON, SNET	49
Figura. 2.3. Dependencias temporales sobre tareas en MS Project, p. ej.: FS	49
Figura. 2.4. Tareas del proyecto como actividades del proceso BPMN	51
Figura. 2.5. Arquitectura MDE para extraer procesos desde legacy databases	61
Figura. 3.1. Informe Chaos 2012	68
Figura. 3.2. Informe Chaos: serie temporal 1994-2012	69
Figura. 3.3. Conceptos fundamentales MDE	73
Figura. 3.4. Meta-Modelo: sintaxis abstracta vs sintaxis concreta	74
Figura. 3.5. MOF y niveles de abstracción	77
Figura. 3.6. Ciclo de mejora continuo BPM	80
Figura. 3.7. Minería de procesos (process mining)	81
Figura. 3.8. Arquitectura de referencia para la gestión de procesos	83
Figura. 3.9. Principales estándares para la gestión de procesos	83
Figura. 3.10. Principales estándares para la gestión de procesos	84
Figura. 3.11. BPDM	85
Figura. 3.12. Capas de XPDL 2.2	87
Figura. 3.13. Evolución histórica de estándares BPM	88
Figura. 3.14. Arquitectura PAIS de cuatro capas	89
Figura. 3.15. Arquitectura de un BPMS según WfMC	90
Figura. 3.16. Generación NDT de la fase de Análisis desde la fase de Requisitos	92
Figura. 3.17. Meta-Modelo de procesos NDTQ-Framework	94
Figura. 3.18. Meta-Modelo InRoma	95
Figura. 3.19. Modelo de herradura ADM «Horseshoe Model»	98
Figura. 6.1. Taxonomía de reglas temporales	123
Figura. 6.2. Análisis de sensibilidad de eventos al aplicar CPM	125
Figura. 6.3. Dependencia de ausencia entre actividades	126
Figura. 6.4. Dependencias temporales SS, SF, FS, FF	127
Figura. 6.5. Ciclo de vida de una actividad BPMN	128
Figura. 6.6. Sintaxis abstracta «UML Activity Diagrams (AD)»	137
Figura. 6.7. Meta-Modelo para intercambio de diagramas UML	138
Figura. 6.8. Meta-Modelo de procesos y actividades BPMN 2.0	139
Figura. 6.9. Meta-Modelo para intercambio de diagramas BPMN 2.0	140
Figura. 6.10. Meta-Modelo de procesos y actividades XPDL 2.2	141
Figura. 6.11. Meta-Modelo PIM-up	143
Figura. 6.12. Enumeraciones del Meta-Modelo PIM-up	144
Figura. 6.13. Ciclo de vida de proceso y actividades (cálculo CPM)	145
Figura. 6.14. Restricciones OCL sobre el ciclo de vida de procesos y actividades MM PIM-up	147
Figura. 6.15. Restricciones de tiempo OCL (a)	148
Figura. 6.16. Restricciones de tiempo OCL (b)	149
Figura. 6.17. Dependencias temporales OCL	152
Figura. 7.1. MOF e ingeniería inversa aplicado al enfoque	157
Figura. 7.2. Plataformas para ingeniería inversa	158
Figura. 7.3. Meta-Modelo pivote PIM-upl para ingeniería inversa desde legacy systems	160
Figura. 7.4. Arquitectura de MS Project Server	161
Figura. 7.5. Meta-Modelo de tareas MS Project Server	163

Figura. 7.6. Proceso de extracción de la dimensión temporal PSM→PIM	166
Figura. 7.7. Algoritmo para extracción de la dimensión temporal PSM→PIM	167
Figura. 7.8. Extensión al Meta–Modelo PIM–upl para resolver colisiones de actividades replicadas	173
Figura. 7.9. Resolución de colisiones del atributo Activity.name	174
Figura. 7.10. Resolución de colisiones del atributo Activity.unique_Identifier	174
Figura. 7.11. Resolución de colisiones con atributos start y end de la clase Activity	174
Figura. 7.12. Resolución de colisiones con atributos startCPM y endCPM	175
Figura. 7.13. Derivación de reglas para la clase Merged_Activity	175
Figura. 7.14. Escenario de extracción de modelos de procesos con actividades replicadas	177
Figura. 7.15. Proceso de extracción de la dimensión temporal PIM→ CIM	179
Figura. 7.16. Meta–Modelo BPMN 2.0 con extensiones temporales	181
Figura. 7.17. Algoritmo de transformación PIM→CIM	183
Figura. 8.1. E1::Tareas Organizativas, Control de Calidad y Fases AQUA-WS	190
Figura. 8.2. E1::Precedencias en Tareas Organizativas, Control de Calidad y Fases AQUA-WS	190
Figura. 8.3. Tareas de un grupo de actividades (AQUA-WS)	191
Figura. 8.4. Dependencias en tareas de grupo (AQUA-WS)	191
Figura. 8.5. Tabla Msp_Tasks para el subproyecto (64) (AQUA-WS)	192
Figura. 8.6. Tabla Msp_Links para el subproyecto (64) (AQUA-WS)	192
Figura. 8.7. Tabla Msp_Tasks para el subproyecto patrón (1001): Fase NDT ASI	193
Figura. 8.8. Tabla Msp_Links para el subproyecto patrón (1001): Fase NDT ASI	194
Figura. 8.9. Actividades generales y subsistemas AQUA-WS (BPMN)	197
Figura. 8.10. Proceso BPMN (61) Fase Alfa 0.4: Clientes Intervenciones en Red (AQUA-WS)	198
Figura. 8.11. Proceso de negocio BPMN (64) (ASI) Instalaciones y Equipos (AQUA-WS)	199
Figura. 8.12. Proceso de negocio BPMN (1001) Patrón NDT (ASI) Análisis del Sistema	200
Figura. 8.13. Procesos de negocio NDT: Fases de Requerimientos y Análisis	201
Figura. AII.1. Meta–Modelo de tareas de MS Project (MS Access)	261
Figura. AII.2. Meta–Modelo de tareas de MS Project Server (SQL*Server™). Tablas principales	261
Figura. AII.3. Workflow básico de Alfresco	262
Figura. AII.4. Meta–Modelo del ECM:Alfresco (PostgreSQL)	262
Figura. AII.5. Meta–Modelo de tareas de Activity (Integrado con Alfresco en PostgreSQL)	263
Figura. AII.6. Meta–Modelo de tareas de RedMine	264
Figura. III.1. Meta–Modelo OMG:IMM PSM Relacional. Main Schemata	267
Figura. III.2. Meta–Modelo OMG:IMM PSM Relacional. Tables	268
Figura. III.3. Meta–Modelo OMG:IMM PSM Relacional. Columns	269
Figura. III.4. Meta–Modelo OMG:IMM PSM Relacional. Constraint Unique Keys	270
Figura. III.5. Meta–Modelo OMG:IMM PSM Relacional. Constraint Foreign Keys	271
Figura. III.6. Meta–Modelo OMG:IMM PSM Relacional. Constraint Other Constraints	272
Figura. III.7. Meta–Modelo PSM Relacional. Paquetes	273
Figura. III.8. Meta–Modelo PSM Relacional. Extensiones para triggers	273
Figura. IV.1. Planificación General Proyecto AQUA-WS por subsistemas	277
Figura. IV.2. Planificación General Proyecto AQUA-WS por fases	280
Figura. IV.3. Planificación Control Calidad (64) (ASI) Instalaciones y Equipos AQUA-WS	283
Figura. IV.4. Patrón de Planificación para Control Calidad de la fase NDT (ASI)	284
Figura. IV.5. Diagrama Relacional BD SQL*Server Published-AQUA	285
Figura. IV.6. Tabla Ext_Task_Constraint_Types (AQUA-WS)	286
Figura. IV.7. Tabla Ext_Link_Types (AQUA-WS)	286
Figura. IV.8. Tabla Msp_Projects (AQUA-WS)	287
Figura. IV.9. Tabla Msp_Tasks (AQUA-WS)	288
Figura. IV.10. Tabla Msp_Links (AQUA-WS)	293



# ÍNDICE DE TABLAS

Tabla. 2.1. Hipótesis vs Objetivos	60
Tabla. 3.1. Retos de los sistemas legacy systems	97
Tabla. 5.1. Perspectivas BPM soportadas por tipo de legacy system	116
Tabla. 5.2. Reglas temporales soportadas por tipo de legacy system	117
Tabla. 5.3. Dependencias temporales soportadas por tipo de legacy system	118
Tabla. 6.1. Cotas de eventos para restricciones de inicio y terminación flexibles	125
Tabla. 6.2. Extensión temporal a BPMN 2.0 mediante nuevos decoradores	130
Tabla. 6.3. Construcciones BPMN 2.0 para representar Restricciones de Tiempo (TC)	133
Tabla. 6.4. Construcciones BPMN 2.0 para representar Dependencias Temporales (TD)	135
Tabla. 6.5. Comparación de Meta-Modelos de procesos: clases y asociaciones	142
Tabla. 6.6. Restricciones temporales OCL	151
Tabla. 6.7. Dependencias temporales OCL	153
Tabla. 7.1. Tabla Constraint_Type (MS Project Server)	164
Tabla. 7.2. Tabla Link_Type (MS Project Server)	164
Tabla. 7.3. Proyecto como proceso de negocio.	169
Tabla. 7.4. Subproceso como Actividad Resumen y Proceso de Negocio.	169
Tabla. 7.5. Actividades del proceso de negocio.	169
Tabla. 7.6. Restricción de tiempo de duración fija para cada tarea 'FIXD'	170
Tabla. 7.7. Restricciones de tiempo sobre inicio y terminación de la actividad	170
Tabla. 7.8. Dependencias temporales sobre la actividad	171
Tabla. 7.9. Proceso de negocio BPMN	183
Tabla. 7.10. Pool BPMN para el proceso	184
Tabla. 7.11. Actividad BPMN	184
Tabla. 7.12. Calle para la actividad dentro del Pool del proceso	184
Tabla. 7.13. Subproceso BPMN	184
Tabla. 7.14. Subproceso BPMN AdHoc	184
Tabla. 7.15. Restricciones temporales BPMN	185
Tabla. 7.16. Dependencias temporales BPMN	185
Tabla. 8.1. Mapeo de elementos de la planificación AQUA-WS a procesos BPMN	194
Tabla. IV.1. Tabla SQL*Server Msp_Tasks (AQUA-WS)	289
Tabla. IV.2. Tabla SQL*Server Msp_Tasks (AQUA-WS)	294
Tabla. IV.3. Tabla de asignación de recursos (AQUA-WS)	296



# AGRADECIMIENTOS

Este trabajo me ha permitido recibir directrices, contrastar opiniones, orientaciones y contar con la experiencia de personas que han hecho posible su realización, a las que agradezco su colaboración. En primer lugar, a mis directoras de tesis, María José Escalona e Isabel Ramos, por sus sabias orientaciones, motivación, apoyo y acogida en el grupo de investigación IWT2. A María Teresa Gómez, por su empuje en el inicio de la tesis. A Miguel Toro, por su análisis y consejos para enfocar el trabajo. A mi compañera Margarita Cruz, por sus revisiones y oír pacientemente mis reflexiones. A todos mis compañeros de Departamento y del grupo IWT2, especialmente a Julián, Laura y Francis. A mis amigos, y especialmente a mi familia, por apoyarme en la dedicación a este trabajo. A todos ellos, mi reconocimiento y gratitud.



# RESUMEN

En el contexto actual de la globalización, la competencia es cada vez mayor para las organizaciones de cualquier sector. Las «*Tecnologías de la Información y Comunicaciones (TIC)*» representan una clave innovadora para hacer frente a los retos a que se enfrenta el ser humano en las próximas décadas, tanto para las economías avanzadas como para las emergentes.

Muchos investigadores consideran el enfoque «*Business Process Management (BPM)*» [van der Aalst 2004; Netjes et al. 2006] como una *ventaja estratégica* [Havey 2005; Trkman 201; Mohapatra 20120; von Rosing et al. 2014] para las organizaciones, pues contribuye a optimizar sus procesos, reduciendo tiempo y costes y aumentando la calidad de sus servicios. La implantación de *BPM* en una organización mejora el conocimiento general de la misma «*Know-how de la organización*», su funcionamiento interno y sus procesos de negocio [Malinova & Mendling 2013]. Las «*organizaciones del software (TI)*» no son una excepción; así aparecen recomendaciones realizadas por instituciones como «*International Standard Office (ISO)*», «*Object Management Group (OMG)*» y «*Workflow Management Consortium (WfMC)*», con distintas normas y estándares relativas a procesos generales y procesos específicos del software, sin olvidar conocidas propuestas como «*Information Technology Infrastructure Library (ITIL)*» [ITIL 2015; Iso/Iec 20000:2012; Clifford & van Bon 2008], *PMBOK* [PMI:PMBOK 2013], *PRINCE2* [SOB 2009] o «*Capability maturity model integration (CMMI)*» [Chrissis et al. 2011], entre otras.

Los procesos de las «*TI*» son más impredecibles y complejos que los de otros sectores, pues están en constante evolución al ser afectados, entre otros aspectos, por nuevos ciclos de vida, nuevas tecnologías y llevados a cabo por grandes equipos de desarrollo multidisciplinares; por esta razón, su establecimiento, control y medición es especialmente costoso [Ruiz-González & Canfora 2004]. Implantar el enfoque *BPM* supone gestionar dichos procesos e integrarlos con sistemas de información destinados a cubrir todas las áreas de la organización: producción de software, gestión de relaciones con clientes, proveedores o colaboradores, gestión económica y financiera, gestión de activos, etc. Algunos de estos sistemas, los más modernos, permiten la especificación y gestión de procesos (*PAIS* [Dumas et al. 2005]), pero otros no contemplan este enfoque (*legacy systems* [Bisbal et al. 1997, 1999; Stavru et al. 2013]); de cualquier manera, todos tienen en común que su capa de persistencia, representada habitualmente por bases de datos relacionales (*legacy databases* [Bisbal et al. 1997,1999; Pérez et al. 2003]), alberga estados que son el resultado de la ejecución de los procesos de negocio de la organización [van der Aalst 2015]. Estos estados están referidos a las distintas perspectivas de los

procesos: *control del flujo, información, temporal, organizacional, operacional, y de casos* [Jablonski 1996; Reichert & Weber 2012; van der Aalst et al. 2004, 2005, 2007, 2012].

La literatura está llena de referencias sobre dos enfoques para el análisis de los sistemas de información: i) *Análisis orientado a datos* y ii) *Análisis orientado a procesos*. En los últimos años han emergido técnicas que intentan trazar un puente entre los dos enfoques, como la *minería de procesos* («*process mining*») [van der Aalst et al. 2007, 2012; van der Aalst 2011, 2013, 2014, 2015; Reichert & Weber 2012] o enfoques que buscan la modernización del software y, en algún caso, la obtención de procesos utilizando el enfoque «*Ingeniería del Software Dirigida por Modelos: Model Driven Engineering (MDE)*» [Kent 2002; Schmidt 2006]. En este trabajo de tesis se va a plantear un enfoque, basado en *MDE*, que extraerá automáticamente aproximaciones a los procesos del software desde *legacy systems*<sup>1</sup> utilizados habitualmente por las «*TI*». La propuesta se va a centrar en el estudio de la *dimensión temporal* de los procesos, analizando los estados representados en la capa de persistencia, especialmente las estructuras y reglas que permiten capturar esta perspectiva: tareas, restricciones y dependencias temporales. Se desarrollará un conjunto de transformaciones, basadas en *Meta-Modelos*, que manejan la perspectiva temporal de los procesos en distintos niveles de abstracción, extrayendo aproximaciones a los procesos del software ejecutados por cada «*TI*», con objeto de ayudar en la implantación del enfoque *BPM* y de integrar esta tecnología con los *legacy systems* que se utilizan para *planificar y controlar proyectos informáticos, gestionar documentación y todo tipo de contenidos, o bien para gestionar otras áreas de la organización como la económico-financiera, gestión de activos, relaciones con terceros (clientes, aprovisionamiento, activos, etc.)*.

El resultado serán procesos expresados con notaciones gráficas muy extendidas [Bonnet et al. 2014] como *BPMN* [OMG:BPMN 2013; Iso/Iec:19510 2013], que otorgan mayor comprensibilidad [Lu & Sadiq 2007] para el experto en software, o bien basados en otros *Meta-Modelos* muy extendidos en el sector como «*Software Engineering Meta-Model for Development Methodologies*» [Iso/Iec:24744 2007, 2014] o «*Software & Systems Process Engineering Meta-Model specification (SPEM 2.0)*» [Bendraou et al. 2007; OMG: SPEM 2008]. *BPMN* es esencialmente imperativo, y tiene cierta capacidad para expresar la *dimensión temporal*, aunque manifiesta cierta debilidad para expresar un abanico amplio de éstas, razón por la que diversos autores están proponiendo extensiones que recojan mejor esta *dimensión*. La superposición de reglas temporales mediante un enfoque complementario de especificación [Fahland 2009a, 2009b; Pichler et al. 2011; Reijers et al. 2013] evitará la sobrecarga de los modelos con más artefactos de carácter

---

<sup>1</sup> El término *legacy system* (sistema heredado) se asocia a sistemas de información basados en arquitecturas tecnológicas que pueden estar obsoletas. No obstante, estos sistemas suelen tener mucha funcionalidad y su sustitución es difícil y costosa.

imperativo que, de otro modo, harían perder claridad a la especificación de los procesos.

El enfoque de la tesis se contrasta con el caso “*AQUA-WS Project*”, que consistió en la modernización integral de la arquitectura de aplicaciones de la *Empresa Metropolitana de Abastecimiento y Saneamiento de Aguas de Sevilla S.A. (EMASESA<sup>2</sup>)*, basada en nuevo desarrollo web de distintos subsistemas, y soportada con la metodología «*Navigational Development Techniques (NDT)*» del grupo *IWT2*, que realizó el control de calidad del proyecto. Este caso ha servido para evaluar los resultados del trabajo de investigación, detectar limitaciones, e identificar trabajos futuros.

---

<sup>2</sup> <http://www.emasesa.com/>





# **PARTE I. INTRODUCCIÓN**



# Capítulo 1. CONTEXTO DE LA INVESTIGACIÓN

*“Lo último que uno sabe es por dónde empezar”*  
*Blaise Pascal (1623-1662)*

Las «organizaciones del sector del software (TI)» que compiten en la nueva economía global están sometidas a una acelerada evolución tecnológica, que provoca procesos de negocio singulares en comparación con los de otros sectores productivos; en el campo de gestión de procesos, el enfoque «*Business Process Management (BPM)*» es una ventaja competitiva que ninguna organización puede dejar de lado, y el sector del software no es una excepción. Los sistemas de gestión de procesos «*Business Process Management Systems (BPMS)*» coexisten con sistemas convencionales para *gestión de proyectos informáticos, gestión documental y gestión económica (legacy systems)*, entre otros, que en su capa de persistencia (*legacy database*) mantienen estados que son el resultado de la ejecución de procesos de negocio en la «TI». Los procesos pueden analizarse desde distintas dimensiones: *control del flujo, funcional, temporal, organizacional y casos*; en este trabajo se pone foco en la *dimensión temporal* y en el reflejo que queda de cualquier proceso en una *legacy database*. La motivación de este trabajo de tesis radica en ofrecer una propuesta que genere procesos de negocio de una «TI» a partir de *legacy databases*, para lo cual se desarrollará una propuesta de *ingeniería inversa* basada en el *paradigma dirigida por modelos (MDE)*.

En el capítulo se enmarca este trabajo de tesis en el contexto de la actividad del grupo de investigación *IWT2* y se resume la cronología de actividades realizadas. Por último, se describe la estructura de esta memoria. En el Capítulo 2 se identificará detalladamente el problema planteado, las hipótesis de trabajo y los objetivos de la investigación. En el resto de Capítulos se desarrollará la propuesta y se verificará, aplicándola a un caso de estudio en un proyecto I+D+i.

## 1.1 Organizaciones del negocio del software (TI)

El mundo está inmerso en una de las peores crisis económicas que se han vivido en décadas y la población mundial se acerca ya a los 7.400 millones de habitantes con una compleja situación *socio-económica* caracterizada por fuertes interdependencias.

Políticos y líderes empresariales tienen como reto reconducir la sociedad a nuevos caminos de equilibrio y crecimiento que permitan ofrecer oportunidades de negocio y crear empleo en el contexto de una *economía global* [Korten 1998; Dunning & Lundan 2008]. Como ejemplos: en 2008 la quiebra de *Lehman Brothers* [De Haas & van Horen 2012] generó rápidamente una crisis financiera mundial; cualquier día, en minutos, la apertura o cierre bajista o alcista del mercado financiero de un país puede afectar a países del otro lado del mundo; los movimientos del precio del petróleo pueden provocar importantes trastornos en el sector de la energía de países compradores, y extenderlo a otros sectores productivos con dependencia energética. Los últimos beneficios anuales de *Apple*<sup>3</sup>, 48.800 M€<sup>4</sup>, han superado records históricos (desde 2008 lo ostentaba la petrolera americana *Exxon*). Esa cifra supera el PIB de 110 de los 187 países de los que el *Fondo Monetario Internacional (FMI)* dispone de datos y es equivalente al valor de cotización en el *IBEX 35* de la entidad financiera *BBVA*. La mayor contribución a los beneficios se ha conseguido mediante ventas del *iPhone* en China.

Las «*Tecnologías de la Información y Comunicaciones (TIC)*» representan una clave innovadora para hacer frente a los retos a que se enfrenta el ser humano en las próximas décadas, tanto para las economías avanzadas como para las emergentes. El término «*Nueva Economía*» lo introduce [Alexander 1983] para referirse a los cambios *socio económicos* que se producen en las sociedades avanzadas de los 80's cuando crece el sector de servicios frente al sector de la manufactura. Hoy en día, hablar de *nueva economía* equivale a referirse al impacto de las *TICs* en todo tipo de relaciones personales, profesionales, particulares u oficiales entre individuos o cualquier tipo de organización. La liberalización del uso de *internet*, a mediados de los 90's por el *US Commerce Department*, ha acelerado más si cabe, este proceso, acuñándose nuevos términos como la «*Sociedad del Conocimiento*» o «*Sociedad de la Información*» y «*La Sociedad Red*» [Castells 1999]. En [Castells 2002] se analiza el fuerte impacto que el desarrollo de *internet* está teniendo en la sociedad. A lo largo de la historia, las transformaciones económicas ocurren cuando convergen nuevas tecnologías de comunicación con nuevos sistemas de energía (la aparición de la máquina de vapor, la electricidad, los motores de combustión o reacción, etc., son ejemplos de hitos históricos relevantes). La conjunción de las tecnologías de comunicaciones e *Internet* y las energías renovables, en el siglo XXI, está dando lugar a la denominada *Tercera Revolución Industrial* [Rifkin 2011].

En 1965, la *Ley de Moore* [Moore 1965], formulada por el cofundador de *Intel*, predijo que cada dos años se duplicaría el número de componentes (transistores) en los circuitos integrados. Esta Ley ha estado vigente durante cinco décadas y, aunque el mismo autor predice en [Chen 2015] que esta capacidad de reducción alcanzará sus límites físicos, por lo que la Ley dejará de cumplirse en la próxima década, lo cierto es que en estos cincuenta años el crecimiento de la capacidad de computación ha cumplido con las

---

<sup>3</sup> <http://www.apple.com/>

<sup>4</sup> [http://economia.elpais.com/economia/2015/10/27/actualidad/1445972060\\_243099.html](http://economia.elpais.com/economia/2015/10/27/actualidad/1445972060_243099.html)

expectativas de Moore, reduciendo drásticamente los costes y llevando una capacidad de computación insospechada hace unos pocos años hasta el pequeño teléfono celular que usan continuamente la mayoría de los miembros de cualquier familia. Como consecuencia, en los últimos años han madurado en el mercado soluciones de «Computación en la nube (Cloud Computing)» [Foster et al. 2008; Zissis & Lekkas 2012], en la que aparecen nuevos actores que ofrecen servicios virtualizados de distinto nivel [Doelitzscher et al. 2011]: i) «Infraestructure as a Service (IaaS)», ii) «Platform as a Service (PaaS)» y iii) «Software as a Service (SaaS)», que están permitiendo al público, profesionales, empresas y todo tipo de organizaciones públicas y privadas pasar de gestionar su propia infraestructura<sup>5</sup> en sus instalaciones físicas a poder contratarlas con estos proveedores de servicios al nivel que interese.

Las aplicaciones en *internet* han madurado y en el desarrollo de éstas se utiliza el concepto de segunda generación o *Web 2.0* [O'Reilly 2007] para crear servicios más avanzados y de mayor calidad. Estas soluciones, hoy, son viables gracias al aumento de velocidad y calidad del ancho de banda de las comunicaciones y a sus reducciones de costo. La aparición y rápida expansión de los *dispositivos inalámbricos conectados a internet* está suponiendo otro gran empuje para desarrollar nuevas aplicaciones [Castells 2011]. Las *redes sociales*, donde ya están representadas todas las actividades humanas, que incluyen relaciones personales, negocios, trabajo, cultura, comunicación, movimientos sociales y política, junto con el uso de *Smartphone* están introduciendo nuevos hábitos en el individuo [Castells 2013], llegando a generar nuevos usos y costumbres en la sociedad. Cualquier actor interesado (empresas de marketing y publicidad, administraciones, partidos políticos, ONGs, u organizaciones de cualquier otro sector) tiene en cuenta todo esto, dirigiendo sus acciones a las redes sociales y adaptando su imagen corporativa a dispositivos móviles.

En este nuevo escenario tecnológico, hoy en día, los contenidos multimedia e interactivos están al alcance de *Smartphone* o *Smart TV*; p. ej.: acabamos de ver como un proveedor de telecomunicaciones (Movistar) absorbe<sup>6</sup> a un proveedor de servicios de TV (Canal+ del Grupo PRISA). Existen iniciativas privadas de líderes mundiales, como: i) *IBM*<sup>7</sup> o *Hewlet Packard*<sup>8</sup> que son ejemplos de soluciones *IaaS* corporativas; ii) *Google*<sup>9</sup> o *Microsoft* con *Azure*<sup>10</sup> que ofrecen *PaaS*; iii) *SAP*<sup>11</sup> y *ORACLE*<sup>12</sup> que ofrecen sus soluciones *ERP* en la nube como *SaaS*; y iv) los líderes *Amazon*<sup>13</sup> y *eBay*<sup>14</sup> del comercio

<sup>5</sup> Hardware (servidores, hardware de comunicaciones, etc.), software (sistemas operativos, gestores de bases de datos, servidores web, plataformas y lenguajes de desarrollo, etc.) y aplicaciones (sistemas de gestión bajo licencia).

<sup>6</sup> [http://economia.elpais.com/economia/2015/07/08/actualidad/1436374391\\_989384.html](http://economia.elpais.com/economia/2015/07/08/actualidad/1436374391_989384.html)

<sup>7</sup> <http://www.ibm.com/cloud-computing/us/en/iaas.html>

<sup>8</sup> <http://www8.hp.com/us/en/cloud/cloudsystem-foundation.html>

<sup>9</sup> <http://www.google.com>

<sup>10</sup> <http://azure.microsoft.com/es-es/>

<sup>11</sup> <http://scn.sap.com/community/cloud>

<sup>12</sup> <https://www.oracle.com/cloud/saas.html>

<sup>13</sup> <https://www.amazon.com/gp/gw/ajax/s.html>

<sup>14</sup> <http://www.ebay.com/>

electrónico mundial.

En [Magoulas et al. 2007] se contemplan los conceptos «*e-Society*» y «*e-Community*»; una «*e-Society*» es una sociedad que está compuesta por un conjunto de «*e-Communities*» o *sociedades virtuales*. Las áreas de interés de estas comunidades son múltiples. «*e-Learning*» en el sector de la educación y «*e-Business*» en el desarrollo de negocio electrónico en la red han sido las primeras áreas en desarrollarse. El concepto se ha extendido a otras áreas [Monteiro et al. 2013], entre otras: «*e-Government*» para la gestión de relaciones entre administraciones y ciudadanos, «*e-Health*» relacionado con la gestión de información sobre pacientes y aplicaciones en tiempo real que se puedan desarrollar sobre dispositivos móviles «*m-Health*» para mantener información on-line sobre pacientes, y «*e-Democracy*» relacionado con los procesos de registro e identificación de ciudadanos con derecho a voto, control del ejercicio del derecho y cómputo de resultados.

En la «Comunidad Económica Europea (CEE)» la iniciativa «*Future Internet Public Private Partnership (FI-PPP)*» ha creado la plataforma tecnológica *FI-WARE* [Kaloxylou et al. 2012; Usländer et al. 2013] que está soportada por el concepto *Cloud Computing* en una red de amplia cobertura (entre ellas, está adherida la *RedIRIS*<sup>15</sup>, aunque colaboran otras a nivel europeo e iberoamericano). Más de quince países están adheridos a la iniciativa, con una dotación económica inicial del proyecto de más de 100M€. El objetivo de la CEE es crear una infraestructura robusta para facilitar el desarrollo de *e-Communities*. Se promueve el desarrollo de soluciones en diversas áreas: *Smart Cities*, *e-Health*, *Transportes*, *Energía y Medio Ambiente*, *Sector Agroalimentario*, *Medios y Contenidos*, *Fabricación y Logística*, *e-Learning* y *Servicios Sociales a la Comunidad*. Esta plataforma está dirigida a desarrolladores, gestores de contenidos y público privado e institucional, que podrán beneficiarse de esta iniciativa global frente a otros líderes del mercado como *Google*, *Microsoft*, *Amazon* o *eBay*, entre otras. La plataforma aporta un ecosistema o infraestructura común compuesta por nodos (*FI-LAB*) integrados por diversas tecnologías: *Cloud Computing*, *Gestión multimedia*, *Internet of Things (IOT)*, *Big Data* y *redes de comunicaciones* bajo los proyectos *FI-LINKS* y *FI-XIFI* [Alvarez et al. 2014]. Las soluciones se publican como *enablers* (*generic and specific enablers*) que pueden ejecutarse sobre máquinas virtuales en el *grid FI-WARE* y administrarse con el conjunto de herramientas *FI-OPS*.

El mundo se encuentra en los prolegómenos de lo que podría llamarse *Cuarta Revolución Industrial* o *Industria 4.0* [Kagermann et al. 2013], que corresponde a una nueva manera de organizar los medios de producción, en la que proliferarían un gran número de «*fábricas inteligentes (Smart Factories)*» con mayor adaptabilidad a las necesidades y a los procesos, y con asignaciones más eficaces de los recursos. Esta

---

<sup>15</sup> <https://www.rediris.es/proyectos/fi-ware/index.html.es>

revolución se apoyará en tecnologías emergentes como «*Internet of Things (IOT)*», «*sistemas ciberfísicos*», o «*cultura Hágalo Ud. Mismo (Maker)*».

La confluencia de todos estos factores, situación socio-política, economía global, complejidad y nuevas capacidades de las nuevas tecnologías nos afectan a todos, y las empresas orientadas al negocio del software no son una excepción. El sector de las tecnologías de la información y comunicaciones (TIC) es muy competitivo y ha de hacer frente a nuevos modelos colaborativos que buscan la eficiencia y disminución de costes, como: «*outsourcing*», «*offshoring*», «*nearshoring*» [Oshri et al. 2015], que, en cortos espacios de tiempo, pueden provocar localizaciones y deslocalizaciones de grandes y medianas empresas del software, pues ya no se necesita tener la producción localizada geográficamente cerca del cliente. La contratación de estos servicios puede involucrar a contratistas principales o subcontratistas de grandes proyectos (para gobiernos, sector bancario, de fabricación y utilities, etc.) que se extienden durante períodos de varios años.

Por otro lado, las empresas y profesionales del sector del software deben manejar una complejidad creciente, representada por: i) multitud de tecnologías, como sistemas operativos, gestores de bases de datos, servidores web, distintos lenguajes de programación; ii) normas o estándares internacionales, normas de facto, buenas prácticas y procedimientos para sus procesos productivos, amén de la gestión de sus propios procesos generales (recursos humanos, financieros, activos, etc.). Esto obliga a gestionar equipos multidisciplinares de personas (personal del cliente, del contratista principal, subcontratistas, proveedores de comunicaciones, hardware, software, etc.) en un ámbito nacional e internacional que necesitan protocolos muy dinámicos y eficientes de comunicación.

Una vez hemos situado el contexto general actual de una organización frente a las nuevas tecnologías, en el siguiente apartado, y específicamente para las «*organizaciones del software (TI)*», se introduce la motivación de este trabajo de tesis para facilitar la gestión de sus procesos productivos.

## 1.2 Motivación

Los esquemas organizativos clásicos están basados en modelos jerárquicos que compartimentan la estructura en áreas funcionales, como divisiones en líneas de negocio, líneas de productos, o geográficas, etc., lo cual lleva a una fragmentación de roles o competencias. A cada unidad organizativa o individuo se le asocian responsabilidades o competencias. La ejecución de tareas elementales se desarrolla en compartimentos estancos, de modo que un rol solo tiene visibilidad de su competencia y no existe una percepción del proceso global a través de distintas unidades o roles. Es habitual encontrar redundancia de tareas en distintas unidades, o peor aún, la ejecución de tareas que no son útiles para la organización, además de retrasos o bloqueos de actividades. Con este tipo de organización, los procesos no están descritos formalmente.

Los centros productivos existentes a comienzos del siglo XX [US GPO 1918] ya manejaban el concepto de *proceso* (Definición 1.1 [Weske 2012]) para gestionar sus operaciones, especificando *tareas, departamentos, entradas, salidas, recursos y su asignación, y la planificación de la ejecución del trabajo*, etc. No es hasta mediados de los 90's cuando se reactiva el uso del concepto de *proceso*, con la aparición del campo de la *modelación de procesos de negocio* [Rolstadås 1995].

«Un proceso de negocio consiste en un conjunto de actividades que se realizan en coordinación en un entorno organizativo y técnico. Estas actividades están relacionadas con un objetivo de negocio. Cada proceso de negocio es promulgado por una sola organización, pero puede interactuar con los procesos de negocio realizadas por otras organizaciones»

Definición. 1.1. Proceso de negocio

Un *modelo de procesos formal* permite compartir el conocimiento en toda la organización y facilita el adiestramiento de nuevo personal; también permite analizar la utilidad y oportunidad de cada actividad, medir eficiencias, tiempos muertos o bloqueos, parámetros de calidad, etc., tanto a nivel de detalle como para el proceso global. En la teoría de la organización «*Management*<sup>16</sup>» [Dean & Bowen 1994], enfoques como «*Gestión Total de la Calidad: Total Quality Management (TQM)*» [US Dod 1989; Dean & Bowen 1994; Dooley & Johnson 2001], y «*Mejora de los Procesos: Process Improvement*» [Dooley & Johnson 2001] se asocian a la mejora progresiva de calidad de los procesos existentes en la organización. La «*Reingeniería de procesos de negocio: Business Process Reengineering (BPR)*» (Definición 1.2 [Hammer & Champy 1993]) es un enfoque más ambicioso; *BPR* busca *desechar procesos obsoletos y/o instaurar nuevos procesos, más simples, rápidos, económicos y eficientes para la organización*, haciendo frente a los retos (se conoce como las tres *Cs*) que representan: i) las necesidades de los *Clientes*, ii) la *Competencia* en el mercado y iii) el riesgo de hacer *Cambios* drásticos en la organización. Para llevar a cabo un proyecto *BPR* es imprescindible contar con una plataforma *TIC*<sup>17</sup> robusta.

«La reingeniería de procesos es un análisis y rediseño radical de la concepción fundamental de los procesos de negocio para lograr mejoras relevantes en costos, calidad, servicio y rapidez»

Definición. 1.2. Business Process Reengineering (BPR)

La *reingeniería* supone una ventaja estratégica para la organización [Havey 2005; Trkman 201; Mohapatra 20120; von Rosing et al. 2014]. Inicialmente, la *reingeniería* se aplicaba a los procesos internos de cada organización, pero la evolución tecnológica

<sup>16</sup> El *Management* versa sobre la coordinación de esfuerzos de personas y equipos para lograr metas y objetivos mediante el uso de los recursos disponibles de manera eficiente y eficaz.

<sup>17</sup> Tecnologías de Información y Comunicaciones (TIC). Tecnologías de la Información (TI)



(*Internet, SOAP, Cloud Computing, etc.*) ha llevado a plantear *Procesos de Negocio Inter-Organizaciones* «*Inter-Organizational Business Processes (IOBP)*» [Attaran 2004; Cheikhrouhou 2013a, 2014b] y al rediseño de relaciones entre dos organizaciones o «*reingeniería X-engineering*» [Attaran 2004], donde los procesos cruzan las fronteras de una organización; p. ej.: para definir relaciones con proveedores, clientes, socios, etc. que se ejecutan en un entorno colaborativo.

Un concepto más amplio, que abarca a los anteriores, es «*Business Process Management (BPM)*» (Definición 1.3) [van der Aalst 2004; Netjes et al. 2006]:

«*Business Process Management (BPM)*» incluye aquellos métodos, técnicas y herramientas de soporte al ciclo de vida de los procesos de negocio, el cual a su vez incluye el diseño, aprobación, gestión y análisis de los procesos de negocio operacionales que involucran a personas, organizaciones, aplicaciones, documentos y cualquier otra fuente de información»

Definición. 1.3. Business Process Management (BPM)

*BPM* es considerado por muchos investigadores como una *ventaja estratégica* [Havey 2005; Trkman 201; Mohapatra 20120; von Rosing et al. 2014] para las organizaciones que utilizan este enfoque, pues contribuye a optimizar los procesos de las organizaciones, reduciendo tiempo y costes y aumentando la calidad de los servicios de la organización.

Desde la aparición de las *TIC*, a mediados de los 50's, el negocio del software ha venido padeciendo sensibles problemas, «*Crisis del software (Software crisis)*» [Dijkstra 1972; Sommerville & Kotonya 1998; Greenfield et al. 2004], que provocan serios inconvenientes en el desarrollo de sistemas informáticos, fundamentalmente por desviaciones en plazos y sobrecostes [Lauesen & Vinter 2001], que son bastante habituales y pueden provocar, en muchos casos, la cancelación de esos proyectos [Standish 2012; McNeil & Hanman 2013]. La reacción a esta crisis fue la aparición de la *ingeniería del software* (Definición 1.4) [Bauer 1972]:

«*La ingeniería del software es el establecimiento y uso de principios de ingeniería robustos, orientados a obtener económicamente software que sea fiable y funcione eficientemente sobre máquinas reales*»

Definición. 1.4. Ingeniería del Software

Acercarse a la problemática de las organizaciones orientadas al software requiere, en primer lugar, definir el concepto de «*Proceso software*» [Fuggetta 2000] (Definición 1.5.), entendido como *proceso de negocio en el sector del software*:

«El proceso software se define como un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos necesarios para concebir, desarrollar, desplegar y mantener un producto software»

Definición. 1.5. Proceso software

En *ingeniería del software* se contemplan distintos tipos de procesos [*Pressman 2010*]; entre ellos, pueden considerarse los grupos: i) *Definición*, que contempla las fases de captura de requisitos, análisis y diseño del sistema; ii) *Desarrollo*, que implica la generación de código y diferentes niveles de pruebas del mismo; y iii) *Mantenimiento*, que agrupa diferentes categorías de tareas para soportar la corrección de errores y evolución del producto para adaptarse a nuevos requisitos, a nuevas plataformas y para mejorar la estructura del propio código.

En [*Osterweil 1987*] se resalta la relevancia de los *procesos del software*: «*Software processes are software too*»; cualquier acción encaminada a conseguir software fiable y dentro de plazos y costes aceptables es considerada como parte del proceso del software. Habitualmente, estos procesos se organizan como *proyectos informáticos*, que hoy en día ya no se asocian sólo a la creación de software sino a todo lo que implica, tanto su creación como su adquisición y uso (p. ej.: *suministros y servicios informáticos*) y control (*control de la calidad*). Existen múltiples estándares para distintas categorías de procesos, entre otros: los *Iso/Iec* [*12207:2008; 24744:2007, 2014; 27000:2014; 29119:2014; 33001:2015; 9001:2015*], modelos de referencia como *SPEM* [*Bendraou et al. 2007; OMG: SPEM 2008*]; y buenas prácticas, como: *CMMI* [*Chrissis et al. 2011*], *ITIL* [*ITIL 2015; Iso/Iec 20000:2012; Clifford & van Bon 2008*], *PMBOK* [*PMI:PMBOK 2013*] y *PRINCE2* [*SOB 2009*], que proponen métodos y herramientas para, definir, desarrollar, mantener, usar, planificar y controlar plazos, costes y controlar la calidad de los productos generados en todo el ciclo de vida del software.

Hay que tener en cuenta la singularidad de los *procesos del software*, pues están evolucionando continuamente, con nuevos *ciclos de vida*, nuevas tecnologías, estándares y buenas prácticas que hay que integrar en estos procesos, que son más complejos e impredecibles [*Ruiz-González & Canfora 2004*] que los de otros sectores profesionales. Muchas de las actividades son difíciles de automatizar, pues tienen fuertes dependencias de personas, de la organización de equipos de trabajo y de protocolos de comunicación entre individuos y grupos. Para paliar estos inconvenientes, en los últimos años, el paradigma «*Ingeniería del Software Dirigida por Modelos: Model Driven Engineering (MDE)*» [*Kent 2002; Schmidt 2006*] se ha establecido como un enfoque ampliamente aceptado para el *desarrollo de software*, teniendo su mejor exponente en «*Model Driven Architecture (MDA)*» [*OMG:MDA; Kleppe et al. 2003*] (ver Sección 3.1.2) del organismo *OMG*.

Las organizaciones enfocadas al software gestionan su negocio con el mismo tipo de

modelos, sistemas de información, estándares y buenas prácticas que organizaciones de otros sectores productivos, definiendo sus procesos, ejecutándolos y llevando a cabo su diagnóstico. Hoy en día, la mayoría de estas organizaciones, como las de otros sectores, tienen en cuenta el enfoque *BPM* [van der Aalst 2004; Netjes et al. 2006] para apoyar sus operaciones internas y las que implican actividades de colaboración con otras organizaciones. En cada caso es necesario establecer modelos de estos procesos que evolucionen adaptados al negocio. Los principales estándares en gestión de procesos seguidos por las organizaciones del software tienen en cuenta el enfoque *BPM*; entre ellos, los más relevantes son: i) [Iso/Iec:24744 2007, 2014] «Software Engineering Meta-Model for Development Methodologies» y ii) «Software & Systems Process Engineering Meta-Model specification (SPEM 2.0)» [Bendraou et al. 2007; OMG: SPEM 2008]; en la Sección 3.2 se analizará con mayor profundidad la literatura relacionada con la gestión de procesos del software.

Tanto el enfoque *MDE* como *BPM* buscan reducir los inconvenientes detectados en los procesos del software [Dijkstra 1972; Sommerville & Kotonya 1998; Lauesen & Vinter 2001; Greenfield et al. 2004; Standish 2012] mediante la diferenciación de niveles de abstracción de *modelos* y *Meta-Modelos* [Seidewitz 2003; Fondemenet & Silaghi 2004] para conseguir independizar, en lo posible, los distintos *modelos* y los *artefactos* que los componen, buscando los beneficios que otorga la «interoperabilidad»; Veáanse la **Definición 1.6** del «Institute of Electrical and Electronics Engineers (IEEE)» [Geraci 1991], y la **Definición 1.7** de la «Comunidad Económica Europea (CEE)<sup>18</sup>» [Criado et al. 2010].

*«La habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada»*

Definición. 1.6. Interoperabilidad (IEEE)

*«La habilidad de organizaciones y sistemas dispares y diversos para interactuar con objetivos consensuados y comunes y con la finalidad de obtener beneficios mutuos. La interacción implica que las organizaciones involucradas compartan información y conocimiento a través de sus procesos de negocio, mediante el intercambio de datos entre sus respectivos sistemas de tecnología de la información y las comunicaciones»*

Definición. 1.7. Interoperabilidad (CEE)

Hasta hace pocos años la gestión de procesos no ha estado automatizada, realizándose manualmente o con un nivel de automatización muy bajo, sostenido con herramientas

<sup>18</sup> Este término es destacado en el ámbito de la Administración Electrónica Española. BOE núm. 25, de 29/1/2010, pp.8139-8156

ofimáticas y/o el intercambio de correos electrónicos. Esto se debe, en gran parte, a que la automatización de distintas áreas de la organización está soportada por sistemas de información desarrollados a medida o con productos de mercado que no soportan el concepto de proceso, sino de función que el usuario debe lanzar para que se ejecute. Entre estos sistemas, en las últimas décadas, las organizaciones del software han venido utilizando sistemas automatizados para i) Planificar y controlar proyectos, como ejemplo *MS Project* [Hansen & Hansen 2013; Stover 2007] y *RedMine* [Lang 2010]; ii) más recientemente, sistemas para la gestión documental que permiten almacenar, compartir y algunas funciones de gestión de versiones y circuitos de aprobación de documentos, por ejemplo: «*Alfresco*» [Shariff 2013] o «*MS Sharepoint*» [Smith & Bates 2007] y iii) sistemas, más o menos integrados, que permiten gestionar otras áreas de la organización, como recursos humanos, nóminas y seguros sociales, finanzas y contabilidad, fabricación y logística, etc.; como ejemplo los *ERPs SAP, Oracle Suite o Microsoft Business Solutions* [Hendricks et al. 2007]. La mayoría de estos sistemas podrían agruparse bajo la categoría de «*sistema heredado: legacy information system o legacy system*» [Bisbal et al. 1997, 1999]; un *legacy system*, frecuentemente, no contempla el concepto de proceso, e incluye sólo las funciones cuyo orden de ejecución queda en manos del usuario.

Un *legacy system* suele presentar tres niveles de abstracción o capas de la lógica [Eckerson 1995]: i) *Capa de presentación*, ii) *Capa de aplicación* y iii) *Capa de persistencia*. Un sistema informático es del tipo «*Process aware information system (PAIS)*» [Dumas et al. 2005] cuando contempla un nivel de abstracción adicional para los procesos, diferenciado de las aplicaciones y otorgando mayor nivel de independencia y de interoperabilidad de estos procesos con respecto a las aplicaciones internas de la organización o las de colaboradores externos. Una categoría especial de *PAIS* son los sistemas «*Business Process Management Systems (BPMS)*» [Dumas et al. 2013] (ver Sección 3.3.5), que cuentan con componentes para gestionar el *ciclo de mejora continuo BPM*<sup>19</sup> [van der Aalst 2004; Hill et al. 2006; Dumas et al. 2013]. La introducción de la tecnología *BPM* en una organización no supone sustituir o eliminar los *legacy systems*, sino integrar un *BPMS* con estos *legacy systems* y con los de otras organizaciones en un nuevo contexto global y tecnológicamente complejo. El *ciclo de mejora continuo BPM* propone la identificación, contraste y mejora de los procesos de negocio; mientras que un *BPMS* y, en general un sistema *PAIS*, cuenta con herramientas, como el *log de eventos de ejecución* [Reichert & Weber 2012] para automatizar, parcial o totalmente, la extracción de estos procesos mediante *reingeniería o mejora continua de los procesos* con técnicas como «*process mining*» [van der Aalst et al. 2007, 2012; van der Aalst 2011, 2013, 2014, 2015; Reichert & Weber 2012], que permiten contrastar *modelos de procesos* con las *instancias ejecutadas de sus modelos*, para diagnosticar el grado de cumplimiento de reglas del negocio y derivar nuevos modelos de procesos o mejorarlos. Pero no todos los *legacy systems* son de categoría *PAIS*, lo que significa que no existe *log de eventos*; por

---

<sup>19</sup> El ciclo BPM de van der Aalst es de los más simples y conocidos, aunque existen otros en la literatura científica.

ejemplo: sistemas para planificar y controlar proyectos, o para gestionar documentación y los sistemas desarrollados a medida para realizar otras funciones de gestión de la organización.

Llevar el enfoque *BPM* a una organización va íntimamente ligado al empleo intensivo de *TICs*. Los *expertos en el dominio del negocio* y *expertos en TI* colaboran estrechamente, pues *no es fácil extraer el conocimiento del dominio* [Sommerville & Kotonya 1998]; los primeros son los que cuentan con el conocimiento del problema, mientras que los *técnicos TI* ayudarán a automatizar la ejecución y monitorización de estos procesos. Aceptando que el enfoque *BPM* es también una *ventaja competitiva* [Havey 2005; Trkman 201; Mohapatra 20120; von Rosing et al. 2014] para el sector del *software*, una implantación *BPM* obliga a la especificación de *modelos de procesos*, lo que supone realizar entrevistas con el experto de negocio o manejar manuales organizativos que describan los procesos, pero que no siempre están actualizados; ésta es una fase crítica que representa un costo considerable, en sí, porque los procesos deben estar definidos fielmente para que sean útiles y por el coste que puede suponer cometer errores en su definición. Las organizaciones del *software* siguen utilizando *legacy system* no orientados a procesos, pero que contienen infinidad de datos (*legacy database*) que son el resultado de la ejecución de distintos procesos y esto sucede independientemente de que se utilice un sistema *PAIS* o un *legacy system*, aunque los primeros sí definen y dejan rastro de la ejecución de procesos.

Una «*TI*» que implanta *BPM* puede plantearse una serie de cuestiones (Figura 1.1), que motivan el trabajo de tesis:

- i) ¿Debe empezar definiendo manualmente sus modelos de procesos? o ¿Puede extraerse conocimiento de *legacy databases*? ¿Es posible automatizar esta extracción?
- ii) ¿Cuáles serían los *legacy systems* candidatos para iniciar esta extracción de conocimiento? ¿Qué dimensiones de los procesos pueden analizarse en una *legacy database*?
- iii) ¿Qué puede aportar el paradigma *MDE* para realizar ingeniería inversa de una *legacy database* y representar una aproximación a los procesos? ¿Qué lenguajes de representación de procesos interesan al experto en *software*?

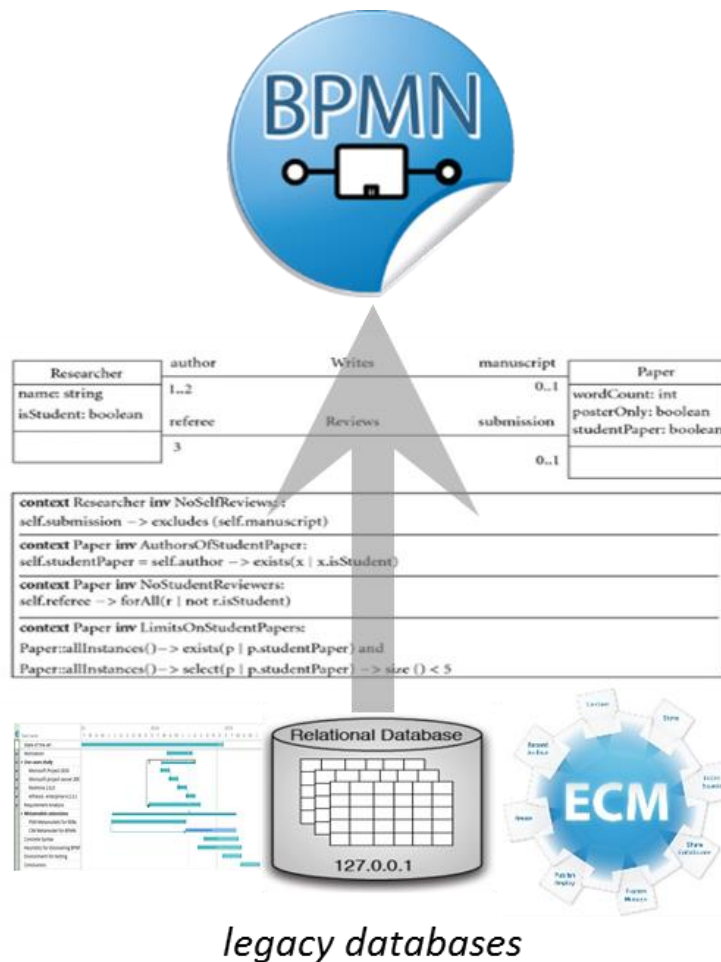


Figura. 1.1. Extracción de procesos de legacy systems

## 1.3 Desarrollo del trabajo de investigación

### 1.3.1 Contexto

Esta tesis se realiza dentro del Grupo de investigación «*Ingeniería Web y Testing Temprano (IWT2)*<sup>20</sup>» del *Departamento de Lenguajes y Sistemas Informáticos*<sup>21</sup> de la *Universidad de Sevilla*<sup>22</sup>, grupo referenciado en el Plan Andaluz de Investigación como PAIDI TIC021.

En el grupo *IWT2*, constituido en el año 2000, se han estado desarrollando un conjunto de tesis doctorales relacionadas con la implantación de procesos de negocio, basados en el marco tecnológico *NDTQ-Framework* [Ponce et al. 2013; Aragón 2014] que se ha desarrollado a raíz de la propuesta metodológica «*Navigational Development Techniques (NDT)*» [Escalona 2004; Escalona & Aragón 2008] (que se describirán en esta memoria de tesis; Sección 2.4). Los trabajos de tesis se han ido contrastando en

<sup>20</sup> <http://iwt2.org/>, referenciado como PAIDI TIC021 en el Plan Andaluz de Investigación.

<sup>21</sup> <http://www.lsi.us.es>

<sup>22</sup> <http://www.us.es>

proyectos de transferencia tecnológica (se han realizado, hasta la fecha, más de 150 proyectos) de estas contribuciones metodológicas con organizaciones del sector público y privado, a nivel nacional e internacional, como el estudio comparativo de herramientas de modelación en la Consejería de Cultura de la Junta de Andalucía [IWT2 2008] y *AQUA-WS Project* [Cutilla et al. 2012] en la Empresa Municipal de Abastecimiento de Aguas de Sevilla, y *CALIPSOneo* [Escalona et al. 2013; Salido et al. 2014] en el sector aeronáutico europeo, para el *Grupo Airbus*<sup>23</sup>. De todos estos proyectos realizados por el grupo IWT2, los relacionados con la gestión de procesos de negocio y/o el paradigma MDE, hasta la fecha suponen un volumen económico de 3,5 M€.

En particular, esta tesis está relacionada con otras tesis desarrolladas o en desarrollo por miembros del grupo de investigación. Aunque hay más trabajos en curso en el grupo, se destacan las siguientes, por estar relacionadas con procesos del software y emplear el *Meta-Modelo* de procesos *NDTQ-Framework*:

- i) García García, J.A. (2015): “Una propuesta para el uso del paradigma guiado por modelos (MDE) para la definición y ejecución de procesos de negocio”, IWT2.

Esta tesis define las bases teóricas y prácticas de un marco metodológico para llevar a cabo una gestión de procesos de software desde una perspectiva basada en modelos, proponiendo una serie de *Meta-Modelos* y transformaciones para resolver este problema. La propuesta ayuda a formalizar, sistematizar y automatizar la gestión de procesos, generando una versión ejecutable de dichos procesos desde la definición. En el marco de la tesis, se ha desarrollado la herramienta PLM4BS basada en *perfiles UML* [Alhir 2006; Giachetti et al 2008] y plugins desarrollados sobre *Sparx Systems Enterprise Architect (EA)* [Sparx:EA 2015], donde se plasma un protocolo automatizado para generar transformaciones QVT [OMG:QVT 2011] que llevan a procesos ejecutables.

- ii) García Borgoñón, L. (2015): “Un lenguaje para definir procesos del software integrados en entornos orientados a modelos”, IWT2.

Trata de proporcionar a las organizaciones del software un mecanismo que mejore la interoperabilidad entre los diferentes lenguajes de modelado de procesos de negocio, para que sean capaces de elegir el lenguaje más apropiado que satisfaga sus necesidades. Incluye varios aspectos:

- a) Un lenguaje de modelado de procesos de negocio flexible, ampliable y adaptable mínimo que se puede utilizar en cualquier contexto del negocio.
- b) Un conjunto de protocolos que permiten la transformación de modelado de procesos de un lenguaje origen a un lenguaje destino.

---

<sup>23</sup> <http://www.airbusgroup.com>

### 1.3.2 Metodología

En esta sección se describe la actividad (Véase Figura 1.2) y metodología utilizada en el desarrollo de los trabajos. La Figura 1.2 ilustra la cronología del trabajo de investigación llevado a cabo, destacando las fases, publicaciones y proyectos de investigación.

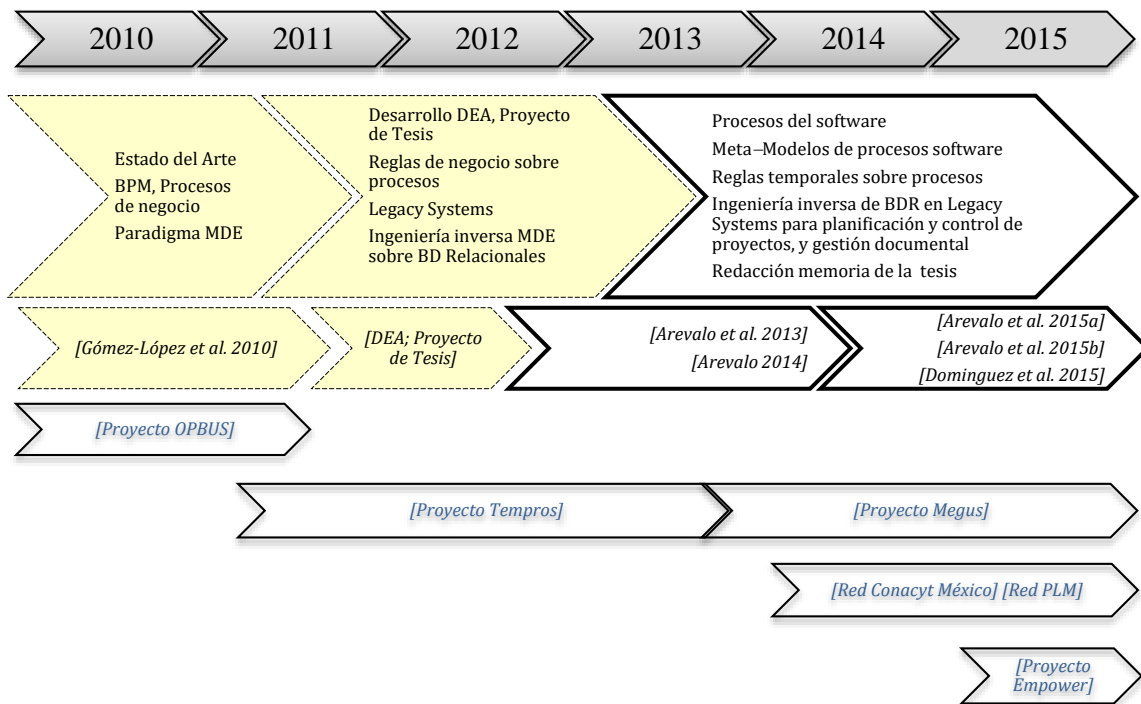


Figura. 1.2. Cronología del desarrollo de la tesis

Con anterioridad a 2010, el doctorando realizó la formación de tercer ciclo en el programa de doctorado del *Departamento de Lenguajes y Sistemas Informáticos: “Tecnología e Ingeniería del Software”*, adscribiendo el proyecto de tesis a la línea de investigación: “*Ingeniería del software*”.

A partir de **2010** se analiza el estudio del arte en relación a los paradigmas *MDE* y *BPM*, poniendo foco en la modelación y verificación de reglas sobre modelos de procesos [Gómez-López et al. 2010; Proyecto OPBUS].

Entre **2011** y **2013**, durante el periodo de investigación, se realiza el trabajo de investigación presentado en el *Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla* como requisito a cumplir para la presentación de la Tesis Doctoral: «*Diploma de Estudios Avanzados (DEA)*». Durante este período se analizan: i) Propuestas y enfoques para la especificación y verificación de reglas sobre procesos de negocio, ii) Estudio de la problemática de *legacy systems* [Bisbal et al. 1997, 1999; Stavru et al. 2013]; iii) Enfoques para la modernización de *legacy systems* basados en *MDE* y *ADM* [Arevalo et al. 2013; Proyecto Tempros], evaluándose, entre otros i) La plataforma «*Eclipse Modeling Framework (EMF)*» [Merks 2010; Eclipse:EMF 2015] y ii) trabajos como



*MARBLE* [Pérez-Castillo et al. 2009, 2011a, 2012a], basado en *ADM*, y la solución *Gra2Mol* [Izquierdo et al. 2008] y [Izquierdo & Molina 2009a, 2009b, 2014] desarrollada en Java sobre Eclipse.

Durante este período se trabajó con *EMF* enfocando el esfuerzo a la ingeniería inversa [Chikofsky & Cross 1990] de *legacy systems*, instalando, evaluando y manejando esquemas de sistemas reales sobre ORACLE; se han utilizado los subsistemas:

- i) Núcleo *EMF* que maneja la variante *Ecore* de la facilidad «*Essential Meta-object Facility (EMOF)*» [OMG:MOF 2011], generando modelos en este formato y conforme a su *Meta-Modelo*. Provee distintos editores para generar modelos e instancias que se pueden intercambiar en formato *.xmi* [OMG:XMI 2013].
- ii) *Xtext* [Eclipse:Xtext 2015] como framework para definir «*Lenguajes específicos de dominio o Domain Specific Languages (DSL)*» [van Deursen 1997, 2000]. *Xtext* utiliza «*ANTLR (ANother Tool for Language Recognition)*» [Parr & Fisher 2011; Antlr 2015] como base para crear reconocedores sintácticos, interpretes, compiladores en base a las descripciones gramaticales realizadas conforme a un *Meta-Modelo*.
- iii) *ATL* [Eclipse:ATL 2015; Jouault et al. 2008] que permite realizar transformaciones «*Modelo-to-Text (M2T)*».
- iv) *Modisco* [Eclipse:Modisco 2015; Bruneliere et al. 2010b] es el módulo que facilita la ingeniería inversa de *legacy systems*. Permite definir *Meta-Modelos Ecore* y extraer modelos con *discoverers* (Java, JSP y XML entre otros). La generación de nuevos *discoverers* es compleja y se requiere una gramática basada en *ANTLR*.

Sobre esta infraestructura se han realizado exploraciones a distintas bases de datos de proyectos de *I+D+i* [Cutilla et al. 2011; Escalona et al. 2013; Salido et al. 2014]. Sobre *EMF* también se ha instalado *Gra2Mol* [Izquierdo et al. 2008; Izquierdo & Molina 2009a, 2009b, 2014], como solución propietaria alternativa a *Modisco* que intentaba resolver inconvenientes de la plataforma *EMF* en el momento de su desarrollo. Se han manejado los *Meta-Modelos relacionales* existentes en la solución *Gra2Mol* para PL/SQL y *AtlanMod Zoos*<sup>24</sup>.

- v) *OCL* [OMG:OCL 2014; Iso/Iec 19507:2012]. *EMF* proporciona tres niveles para la especificación de restricciones *OCL*: a) «*Essential OCL*» para expresiones *OCL* básicas sobre elementos; b) «*Xtext OCLinEcore de OCL*» que permite incrustar fórmulas *OCL* en un modelo *Ecore*; y c) «*Complete OCL*», que permite cargar

<sup>24</sup> [Atlanmod 2015] ofrece *Meta-Modelos* para esquemas relacionales básicos: *CWMRelationalData 1.0*, *Relational 1.1*, *RelationalDBContent 1.0* y *RelationalDBSchema 1.0*. Estos *Meta-Modelos* están disponibles para su descarga libre; están expresados conforme a los *Meta-Modelos Kernel Meta Model* [Eclipse:KM3 2006] y *Ecore* para plataformas *Eclipse Modeling Framework* [Merks 2010; Eclipse:EMF 2015], pero se centran en las estructuras relacionales básicas y son simples.

<http://www.emn.fr/z-info/atlanmod/index.php/Zoos>

archivos con comandos *OCL* que se pueden validar contra un *Meta-Modelo Ecore*. *EMF* dispone de una consola *OCL* para visualizar los errores *OCL*.

En el capítulo del libro [Arevalo et al. 2013] se desarrolla una solución que aplica el paradigma *MDE* para obtener reglas de negocio definidas en *SBVR* [OMG:SBVR] desde una base de datos relacional en *ORACLE*, extrayendo términos del vocabulario y reglas que aparecen como restricciones [Türker & Gertz 2001] de tuplas o escritas como disparadores en el lenguaje procedural *PL/SQL* [Feuerstein & Pribyl 2005]. Se maneja como modelo intermedio el *Meta-Modelo* de *PRR* [OMG:PRR 2009] para especificar los disparadores como *ECA Rules* [Viana et al. 2007] a nivel *MDE PIM* y como *Meta-Modelo PSM* una extensión al *Meta-Modelo* relacional *PSM IMM* [OMG:IMM] (Ver Anexo III) para soportar la extracción de *ECA Rules* desde disparadores *PL/SQL* [Feuerstein & Pribyl 2005].

Entre 2014 y 2015 se ha profundizado en el estudio de i) Especificación de reglas temporales sobre procesos de negocio; ii) Procesos de negocio específicos del software, estándares [Bendraou et al. 2007; OMG: SPEM 2008; Iso/Iec:24744 2007, 2014] y buenas prácticas, así como en las relaciones con otros trabajos de tesis del grupo *IWT2* que están relacionados; iii) Ingeniería inversa de *legacy systems* [Bisbal et al. 1997, 1999; Stavru et al. 2013] estándar, utilizados por las organizaciones del software, como a) Planificación y control de proyectos [Hansen & Hansen 2013; Stover 2007; Lang 2010], b) Gestión documental o de contenidos [Shariff 2013; Smith & Bates 2007] y c) Gestión de procesos generales en organizaciones, *ERPs*, *CRMs*, *SCMs*, [Hendricks et al. 2007]. Se ha explorado la documentación de estos sistemas, sus bases de datos y *Meta-Modelos* que las soportan, en formatos *Ecore* y *KMM3* para *EMF*, y realizando ingeniería inversa con *Eclipse EMF* y *Sparx Systems Enterprise Architect (EA)* [Sparx:EA 2015], con objeto de extraer modelos y *Meta-Modelos*<sup>25</sup> (Ver Anexos II y III) de procesos relacionados con la dimensión temporal de un proceso de negocio, porque, el enfoque del trabajo de tesis es, utilizar como primera palanca, esta dimensión temporal para establecer heurísticas que mejoren la calidad del proceso de extracción de conocimiento desde un *legacy system* hasta modelos más cercanos al experto del dominio del negocio.

En relación a los proyectos [Proyecto Megus; Proyecto Conacyt México; Red PLM; Proyecto Empower], se han realizado las publicaciones [Arevalo 2014] y [Arevalo et al. 2015a] relacionadas con la ingeniería inversa de bases de datos relacionales para extraer procesos de negocio basadas en reglas temporales, y las publicaciones remitidas a Elsevier IST [Arevalo et al. 2015b], que propone una taxonomía de reglas y un *Meta-Modelo* para extender la especificación de reglas temporales sobre el *Meta-Modelo BPMN 2.0*, y a Springer y SQJO [Dominguez et al. 2015] que están en fase de revisión. El proyecto [Proyecto Empower] aplica enfoques de esta tesis para extraer procesos de

---

<sup>25</sup> Además de los *Meta-Modelos* de [Atlanmod 2015], se han extraído vistas de los *Meta-Modelos* de *OMG:IMM* [OMG:IMM; Elvesæter et al. 2006, 2010] y extensiones a estos realizadas por el autor.

negocio en el contexto de los procedimientos administrativos de diversos organismos de la Junta de Andalucía.

## 1.4 Estructura de la memoria

En este Capítulo se ha establecido el escenario y motivación del trabajo de tesis. El Capítulo 2 utiliza un ejemplo motivador para identificar los retos, hipótesis de trabajo y los objetivos que se persiguen. El siguiente Capítulo describe el estado del arte en relación a estos objetivos; en particular, enfoques, estándares y buenas prácticas como: el paradigma *MDE*, la gestión de procesos del software, el enfoque *BPM* y nuestra metodología *NDT*, y por último la modernización de *legacy systems*. El Capítulo 4 destaca la literatura científica seleccionada en relación con el contexto de la tesis: la especificación de reglas temporales sobre procesos, así como el tratamiento de otras perspectivas de los procesos, la gestión de procesos del software, y la modernización de *legacy systems*, ingeniería inversa de bases de datos relacionales, arqueología y minería de procesos. En el Capítulo 5 se lleva a cabo un análisis de algunos *legacy systems*, típicamente utilizados por las organizaciones del software; el estudio se realiza desde el punto de vista de su capacidad para soportar las reglas temporales de la taxonomía establecida, con objeto de seleccionar los mejores candidatos para extraer la dimensión temporal. El Capítulo 6 profundiza en la especificación de la dimensión temporal sobre procesos, resaltando la relevancia del estándar general *BPMN*, pero sacando también a la luz las limitaciones que ofrece para la expresión de reglas temporales que algunos autores están resolviendo con diversos enfoques: patrones, decoradores, etc. En el mismo capítulo se establece una taxonomía de reglas de tiempo sobre procesos y se propone un *Meta-Modelo* basado en *UML* y *OCL* que soporta dicha taxonomía de reglas; este *Meta-Modelo* se utilizará en los capítulos siguientes para soportar la propuesta del trabajo de tesis. El Capítulo 7 desarrolla la solución propuesta para generar modelos de procesos desde aplicaciones *legacy systems*, utilizando un *Meta-Modelo* que extiende al propuesto en el Capítulo 6 y maneja su semántica temporal así como la procedencia de *legacy systems*: se definen los *Meta-Modelos* necesarios para aplicar el enfoque de ingeniería inversa *MDE*, detallando las transformaciones a desarrollar entre plataformas de distintos niveles de abstracción. El Capítulo 8 expone la aplicación del enfoque de la tesis al proyecto de modernización de software *AQUA-WS Project*, donde el grupo de investigación *IWT2* colaboró en las directrices metodológicas y de control de calidad. Se extraen distintos procesos y se discuten los resultados y limitaciones del enfoque. El Capítulo 9 resume las contribuciones de la investigación así como el trabajo futuro que encajaría en la línea de trabajo desarrollada. A continuación figuran distintos Anexos: Referencias; Acrónimos más significativos; se incluyen también: Anexo II) *Meta-Modelos* de tareas para un conjunto de sistemas de información de mercado analizados (*legacy systems*): *MS Project*, *Alfresco* y *Redmine*; Anexo III) *Meta-Modelos* de *Bases de Datos Relacionales* propuestos por *OMG* (en *IMM*) y extensiones a estos

*Meta-Modelos* para soportar restricciones y disparadores *SQL*; Anexo IV) planificación del proyecto *AQUA-WS*; por último el Anexo V) Publicaciones del autor resume las publicaciones y proyectos.

## 1.5 Resumen y conclusiones

En este Capítulo se ha expuesto el contexto y motivación de este trabajo de tesis, identificando las características de los procesos de negocio de las «*organizaciones del software (TI)*» en el contexto tan competitivo que la economía global exige a todos los sectores. «*Business Process Management (BPM)*» es una ventaja estratégica para cualquier organización moderna, y las «*TI*» no son una excepción; sin duda, los expertos de negocio, incluyendo a los expertos en software, necesitan modelos de los procesos de su organización para implantar *BPM*. El *ciclo de mejora continuo BPM* propone la identificación, contraste y mejora de los procesos de negocio, existiendo herramientas en los «*Process aware information system (PAIS)*» que facilitan la automatización parcial o total de este ciclo mediante minería de procesos sobre *log de eventos de ejecución*; ahora bien, no todos los sistemas *legacy systems* cuentan con esta herramienta, lo que motiva el planteamiento de la línea de este trabajo de tesis, que propone usar las capacidades del paradigma *MDE* para plantear un enfoque de ingeniería inversa sobre *legacy systems*, utilizando la perspectiva temporal *BPM*. En la capa de persistencia (*legacy database*) de los *legacy systems* existen estructuras de datos y restricciones que representan estados íntimamente ligados con reglas temporales (p. ej., existen tareas, vencimientos, relaciones de precedencia, etc. en los sistemas para la planificación y gestión de proyectos, y en los circuitos de aprobación de documentos o en otros sistemas de gestión) que son el resultado de ejecutar procesos en la organización, aunque estos procesos no estén formalmente definidos ni automatizados. Se pretende generar una aproximación a los procesos mediante ingeniería inversa *MDE* de estas estructuras y reglas encontradas a un nivel bajo de abstracción (*legacy database*) y llevar esa representación a modelos más cercanos al nivel de abstracción (*modelos de especificación de procesos*) del experto en procesos del software.

## Capítulo 2. RETOS Y OBJETIVOS

*“No entiendes realmente algo a menos que seas capaz de explicárselo a tu abuela”*  
*Albert Einstein (1879-1955)*

Cualquier organización pueden obtener ventajas competitivas si mejora sus procesos. El software sigue manteniendo algunos de los problemas detectados hace más de cinco décadas, razón por la que este sector debe mantener un foco especial en la gestión de sus procesos, atendiendo a estándares, modelos de referencia, buenas prácticas, metodologías y normas internas. Nuevos enfoques como *MDE* y *BPM* son claves para que una organización sea competitiva y pueda mantenerse y tener crecimiento en el mercado.

La adopción del enfoque *BPM* exige la definición de procesos, su configuración y ejecución en un entorno tecnológico avanzado (hoy en día, mediante utilización de «*Business Process Management Systems (BPMS)*»), así como el diagnóstico de estos procesos. La recurrencia de este ciclo se conoce como *mejora continua o reingeniería de procesos (BPR)*. Un *BPMS* coexiste con otros sistemas informáticos «*legacy systems*», como los de *gestión de proyectos, gestión documental, gestión económica, relaciones con clientes y proveedores*, donde se ejecutan las funciones de negocio de la organización.

El foco de atención inicial de este trabajo de investigación es la definición de la *dimensión temporal de procesos* de una organización del software «*TI*» que quiere acometer un proyecto *BPM*. En este escenario, en primer lugar, se consideran los *legacy systems* utilizados para *planificar y controlar proyectos*<sup>26</sup>. Estos sistemas, entre otros, tienen artefactos (datos y reglas de negocio incluidas en *legacy databases*) que están íntimamente relacionados con los procesos de negocio: *tareas, recursos y grupos de recursos* que representan el estado de los trabajos de la organización. Se identificará el problema a resolver y establecerán las hipótesis y objetivos del trabajo de investigación, que consistirá en una propuesta de ingeniería inversa *MDE* para generar automáticamente aproximaciones a los procesos del software desde *legacy databases*.

---

<sup>26</sup> Posteriormente se puede extender el planteamiento a otros tipos de *legacy system*.

## 2.1 Identificación del problema

Las «TI» gestionan su negocio con sistemas de información que también aparecen en otros sectores productivos, aunque los procesos del software presentan características singulares [Ruiz-González & Canfora 2004]: i) están evolucionando continuamente para soportar nuevos ciclos de vida, nuevas tecnologías, estándares y buenas prácticas; ii) estos procesos son más complejos e impredecibles [Ruiz-González & Canfora 2004] que los de otros sectores, pues muchas de las actividades son difíciles de automatizar al tener fuerte dependencia de personas, de equipos de trabajo y de protocolos de comunicación entre individuos y grupos. En una sociedad globalizada [Korten 1998; Dunning & Lundan 2008] y muy competitiva, las «TI» deben hacer frente a retos que el mercado impone con nuevos modelos de colaboración como el *outsourcing*, *offshoring*, *nearshoring* [Oshri et al. 2015].

Las «TI» deben gestionar su negocio haciendo frente a la coexistencia de diversos sistemas de información que abordan las distintas áreas de la organización: *producción*, *relaciones con clientes y proveedores*, *gestión económica y financiera*, etc. Los procesos de negocio representan una capa de conocimiento que está por encima de estos sistemas, facilitando tanto las interacciones dentro de la organización como la colaboración con organizaciones externas (*procesos inter-organizacionales*) [Attaran 2004; Cheikhrouhou 2013a, 2014b].

Una «TI» debe gestionar el conocimiento adaptándose en cada momento a los requisitos de su entorno, por lo que debe tener en cuenta, entre otros, los siguientes aspectos:

- i) **Estándares.** Existen multitud de estándares que proponen: a) Normas de gestión de la calidad [Iso/Iec 9001:2015] y b) [Iso/Iec 33001:2015] para validación de procesos entre otras normas generales; y normas específicas para el software, como: a) Ciclo de vida del software [Iso/Iec 12207:2008], b) *Meta-Modelos de procesos* como [Iso/Iec 24744:2007, 2014] y *SPEM* [Bendraou et al. 2007; OMG: *SPEM 2008*], c) [Iso/Iec 27000:2014] para los procesos de seguridad, y d) [Iso/Iec 29119:2014] para pruebas.
- ii) **Modelos de referencia, buenas prácticas y metodologías.** Entre otros, cabe destacar a) Los modelos de madurez *CMMI* [Chrissis et al. 2011], b) *ITIL* [ITIL 2015; Iso/Iec 20000:2012; Clifford & van Bon 2008], c) «*Project Management Body of Knowledge (PMBOK)*» [PMI: *PMBOK 2013*] y «*Projects IN Controlled Environments 2 (PRINCE2)*» [SOB 2009] como propuestas para la gestión de proyectos.

Existen múltiples metodologías, entre las que cabe destacar «*Unified Software Development Process* o *Unified Process*» propuesta por los promotores de UML [Jakobson et al. 1999] y la implementación más conocida de este método genérico

«*Rational Unified Process (RUP)*» [Kruchten 2004]. Últimamente cabe destacar el auge de los *Métodos Ágiles* [Abrahamsson 2002; Boehm 2002] como *Scrum* [Schwaber 2004]. «*OpenUp*» [Arlow & Neustadt 2005] contempla el enfoque de métodos ágiles para *RUP* sobre plataforma «*Eclipse Process Framework*<sup>27</sup>». En proyectos que hemos desarrollado con administraciones públicas se ha empleado *MÉTRICA* [MAP 2005], basada en el ciclo de vida [Iso/Iec 12207:2008]. La metodología de desarrollo propio que utilizamos en el grupo de investigación *IWT*<sup>28</sup> es «*Navigational Development Techniques (NDT)*» [Escalona 2004; Escalona & Aragón 2008].

- iii) **Enfoques de gestión y tecnológicos.** «*Business Process Management (BPM)*» [van der Aalst 2004; Netjes et al. 2006] es un enfoque de gestión basado en las *TICs*, que permite especificar y diseñar modelos de procesos, los cuales pueden ejecutarse y dejar una traza de eventos en un *log de eventos* [Reichert & Weber 2012] de actividades donde se registran el tipo de actividad, su inicio y terminación; este log de eventos puede reutilizarse para diagnosticar los procesos a posteriori o en tiempo real, y además, puede facilitar la *reingeniería de procesos* (*BPR* [Hammer & Champy 1993; Mohapatra 2012]), para mejorar los procesos existentes, o bien, para derivar nuevos modelos mediante técnicas de «*process mining*» [van der Aalst et al. 2007, 2012; van der Aalst 2011, 2013, 2014, 2015; Reichert & Weber 2012].

*BPM* es un valor estratégico [Havey 2005; Trkman 201; Mohapatra 20120; von Rosing et al. 2014] para hacer frente a las características intrínsecas de los procesos del software que, al ser complejos e impredecibles [Ruiz-González & Canfora 2004], introducen factores de riesgo elevados en los proyectos informáticos. Si estos riesgos no se gestionan adecuadamente los proyectos fracasan con excesiva frecuencia [Standish 2012; McNeil & Hanman 2013].

La «*Ingeniería del Software Dirigida por Modelos: Model Driven Engineering (MDE)*» [Kent 2002; Schmidt 2006] está madurando como nuevo paradigma para contribuir a paliar los problemas ligados a los procesos del software [Dijkstra 1972; Sommerville & Kotonya 1998; Lauesen & Vinter 2001; Greenfield et al. 2004; Standish 2012].

Aunque hay que tener en cuenta todo lo anterior, cualquier «*TI*» debe definir sus procesos específicos, mantenerlos vigentes y llevar a cabo la mejora continua de los mismos. La utilización de un *BPMS* no exime del empleo de otro tipo de sistemas enfocados a la ejecución de funciones o transacciones, como son los *legacy systems*, en cuyo ámbito pueden encajar distintas categorías de sistemas, entre otros, dirigidos a: i) *Planificación y control de proyectos*, ii) *Gestión documental o de contenidos*, iii) *Gestión*

<sup>27</sup> <http://www.eclipse.org/epf/>

<sup>28</sup> <http://iwt2.org/>

de relaciones con clientes, proveedores y colaboradores, iv) *Gestión económico-financiera*. Estos sistemas deben integrarse con el *BPMS*, que albergará los modelos de procesos y facilitará su ejecución y diagnóstico.

El problema planteado es “*La extracción automática de procesos del software desde legacy systems, que ayuden a una organización a implantar BPM como impulso estratégico*”.

## 2.2 Retos

Las «*TI*» se enfrentan a distintos retos:

- i) Es *fundamental definir* y mantener vivos los *procesos del software* de una organización para ser competitiva en el mercado global. El enfoque *BPM* es estratégico para las «*TI*» que, con tecnología *BPMS*, debe aunar todo tipo de estándares, modelos de referencia y buenas prácticas para conseguir procesos del software de calidad.
- ii) Un proyecto *BPM* requiere una definición detallada de modelos de procesos, la cual está en manos del experto del dominio del problema (experto de negocio en software) que puede definirlos desde cero, o bien puede partir de los *legacy systems* existentes. Algunos *legacy systems* manejan procesos, como es el caso de los «*Process aware information system (PAIS)*» [Dumas et al. 2005], pero lo más habitual es que no soporten este concepto. Existen iniciativas de modernización de *legacy systems* como «*Architecture Driven Modernization (ADM)*» [Newcomb 2005; OMG:ADM 2005; Ulrich & Newcomb 2010] y enfoques de modernización del software como «*Model Driven Software Modernization (MDSM)*» [Favre 2004], ambas basadas en *MDE* y en procesos de ingeniería inversa [Chikofsky & Cross 1990; Favre 2004, 2010; Bruneliere et al. 2010a] que permiten extraer artefactos (estructuras de datos, restricciones) de niveles de abstracción bajos (código fuente, esquemas de bases de datos) para presentar otros artefactos de modelos de nivel mayor de abstracción (clases, actividades, flujos, reglas de negocio).

Ante estos retos: i) ¿Es posible ofrecer al experto en software una aproximación a los procesos de negocio que se están llevando a cabo en su organización?; ii) ¿Puede derivarse automáticamente ese conocimiento de los sistemas que están en funcionamiento (*legacy systems*)?; iii) ¿Qué tipo de *legacy systems* son mejores para abordar esta extracción?; iv) ¿Qué enfoques y tecnología puede generar estas aproximaciones de los modelos de procesos? y v) ¿Qué sistemas de representación de procesos interesan a los expertos en software?.

A continuación se expone un caso motivador de extracción de un proceso de negocio



desde «*Microsoft™ (MS) Project*<sup>29</sup>» [Hansen & Hansen 2013]. En la siguiente sección se definirán objetivos que detallan el enfoque del trabajo y, por último, se planteará una aproximación a la solución que se pondrá en esta tesis.

## 2.3 Un ejemplo motivador

Todas las «TI» llevan a cabo proyectos, donde uno de los riesgos principales de estos proyectos es una mala planificación, o peor aún, su inexistencia [Standish 2012; McNeil & Hanman 2013]. La planificación no es estática sino que suele estar viva durante todo el proyecto. El responsable o *Jefe de Proyecto* suele utilizar técnicas que le permiten gestionar planes que involucran tareas, recursos y grupos de recursos, asignaciones y reglas que establecen restricciones de tiempo y de recursos (personas, perfiles, costes u otros); las técnicas más conocidas son los «*Diagramas de Gantt*» [Gantt 1919; PMI:PMBOK 2013] y los grafos «*Project Evaluation and Review Techniques (PERT)*» [Malcolm et al. 1959; PMI:PMBOK 2013]. La duración mínima del proyecto se puede calcular resolviendo el grafo *PERT* con el método «*Critical Path Method (CPM)*» [Kelley & Walker 1959]. *MS Project* [Hansen & Hansen 2013] es un sistema informático para gestión de proyectos bajo licencia Microsoft™, utilizado frecuentemente en todo el mundo, entre otras, por las «TI».

El caso presentado es una planificación *MS Project* de un proyecto realizado por una «TI», representada mediante un «*Diagrama de Gantt*». El resultado buscado es un modelo de negocio correspondiente al proyecto especificado, en este caso, con el estándar «*Business Process Model and Notation (BPMN)*» [OMG:BPMN 2013; Iso/Iec:19510 2013]. Se verá, en esta memoria de tesis, que el sistema origen podría ser otro, incluso de otro tipo, y que, el sistema destino también podría ser otro sistema de representación de los procesos diferente a *BPMN*. El proceso de negocio generado se obtiene manejando sólo la *perspectiva temporal* que se observa en el plan de partida. El objetivo del planteamiento no es realizar un planteamiento formal, sino ilustrar la idea que se desarrollará en la tesis. En este documento se establecerán las hipótesis, objetivos y métodos para generar sistemáticamente procesos que representan las interacciones y reglas de negocio definidas en un proyecto.

### 2.3.1 Planificación de un proyecto como sistema origen

El *diagrama de Gantt* de la [Figura 2.1](#) representa el “*plan de un proyecto de una empresa consultora que realiza una reingeniería de procesos de negocio (BPR)* [Hammer & Champy 1993; Mohapatra 2012] en un cliente”. Las tareas contempladas son:

- 1) Reunión de Lanzamiento del proyecto.
- 2) Análisis de la situación actual.

---

<sup>29</sup> Existen otros productos para gestionar proyectos, tanto del mismo fabricante, como *MS Project Server* [Stover 2007], software libre como *RedMine* [Lang 2010] y otros.

5) Definir nuevo modelo de negocio.

9) Entrega del Modelo.

10) Aceptación del Modelo.

11) Facturación.

Las tareas (2) y (5) son en realidad grupos de tareas que se subdividen en:

3) Revisar documentación.

4) Realizar entrevistas.

6) Modelo básico.

7) Revisión.

8) Refinamiento.

Para definir el plan del proyecto es necesario especificar: a) *Restricciones de tiempo ligadas a cada tarea elemental* y b) *Dependencias temporales entre tareas*, expresadas como relaciones de precedencia en consonancia con las restricciones que aparecen en el «Álgebra de intervalos de Allen» [Allen 1983]. Las tareas de grupo (2) y (5) fijan sus restricciones de tiempo en función de la planificación de las tareas subordinadas y de sus interrelaciones con otras tareas del plan.



Figura. 2.1. Un ejemplo de planificación de actividades con MS Project

### 2.3.1.1 Restricciones temporales

En primer lugar hay que distinguir el tipo de *duración de cada tarea*, que puede ser *duración fija* o *estimada* (calculada con CPM) y puede especificarse (Figura 2.2) en distintas unidades (días, semanas, meses, etc.). Cuando es estimada, aparece con un signo de interrogación sobre el diagrama.

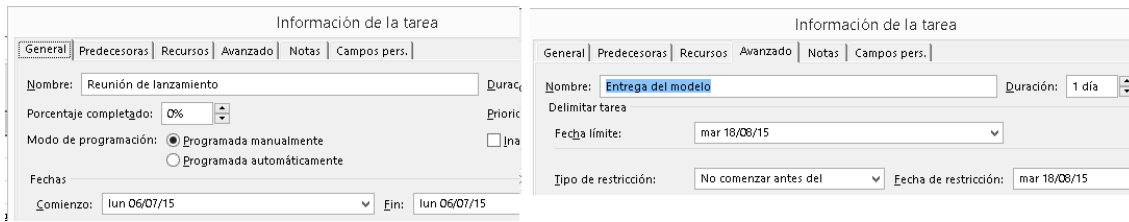


Figura. 2.2. Restricciones temporales sobre tareas en MS Project, p. ej.: MSON, SNET

También pueden especificarse restricciones temporales [Allen 1983] ligadas al inicio y terminación de cada tarea elemental, especificando si la programación es manual o automática (calculada mediante CPM). Por defecto, *MS Project* asocia restricciones «*As Soon As Possible (ASAP)*» a cada tarea; no se requiere la especificación de una fecha como restricción. No obstante, pueden especificarse restricciones específicas (programando manual o automáticamente y especificando en las pestañas generales y avanzadas).

### 2.3.1.2 Dependencias temporales

En la pestaña precedesoras (Figura 2.3) se especifican las dependencias temporales<sup>30</sup> (CC=*Start To Start*, CF=*Start To Finish*, FC=*Finish To Start* que es la opción por defecto y FF=*Finish To Finish*) [Allen 1983]. La columna *Pos* especifica el adelanto o retraso entre los dos eventos; en este caso, la tarea sucesora (*tarea 10: Aceptación*) comienza al terminar la predecesora (*tarea 9: Entrega del modelo*); se especifica una dependencia de tipo FS (en castellano FC).

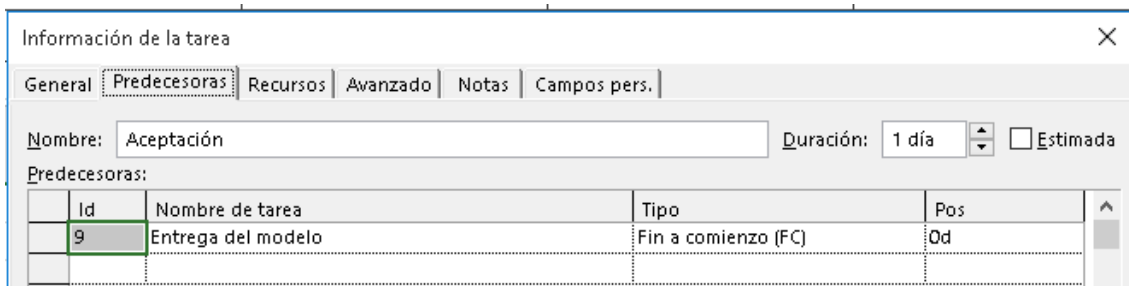


Figura. 2.3. Dependencias temporales sobre tareas en MS Project, p. ej.: FS

### 2.3.2 Proceso BPMN para el plan del proyecto

*MS Project* maneja una perspectiva temporal de tareas que permite ordenarlas en el tiempo y fijar restricciones para cada tarea y restricciones para las relaciones de precedencia. Se plantean las cuestiones:

- i) ¿Puede representarse el conocimiento del plan de proyecto de la Figura 2.1 como un proceso de negocio BPMN?
- ii) ¿Qué transformaciones hay que realizar para la obtención del proceso?

<sup>30</sup> La versión de *MS Project* usada es en castellano. Las dependencias temporales están traducidas y la equivalencia es CC=SS, CF=SF, FC=FS y FF=FF.

La primera equivalencia o patrón de transformación consiste en representar un “*Plan de Proyecto*” como un “*Proceso de Negocio BPMN*” (Figura 2.4):

A continuación hay que establecer patrones de mapeo para las tareas del proyecto. En *BPMN* una *tarea* es una *actividad elemental*, pero también son actividades: un *grupo de tareas*, un *subproyecto* o un *proyecto*. Por tanto, es posible:

- i) Mapear las tareas elementales del proyecto como *actividades BPMN*.
- ii) Considerar procesos *BPMN Ad-hoc* para representar *tareas de grupo* del proyecto, como es el caso de “*Análisis de la Situación Actual*”, modelado como subproceso expandido, que incluye las actividades elementales “*Revisar Documentación*” y “*Entrevistas*” que corren en paralelo. El subproceso “*Definir nuevo proceso de negocio*” se ha modelado como subproceso colapsado, cuyo detalle se muestra en la Figura 2.5.
- iii) Mapear *restricciones de tiempo* como *anotaciones en el diagrama (color azul)*. Así aparecen *Actividades* con “*Duración Fija*” y “*Duración Flexible*” (son aquellas donde la misma es estimada). Por defecto, sobre los eventos de inicio y terminación de una *Actividad*, la restricción *MS Project* es “*Tan Pronto Como Sea Posible*”; no se ponen leyendas en el diagrama *BPMN* para estos casos. Sí se ponen explícitamente para los casos: “*Entrevistas*” con una restricción “*Tan Tarde Como Sea Posible*” y “*Entrega del Modelo*” con una restricción “*Empezar No Antes De: 17/8/15*”. Las duraciones de los subprocesos “*Análisis de la situación actual*” y “*Definir nuevo proceso de negocio*” son calculadas mediante el método *CPM* [Kelley & Walker 1959].
- iv) Mapear las *dependencias temporales (color rojo)*. Estas dependencias (relaciones de precedencia en un diagrama «*PERT*» [Malcolm et al. 1959; PMI:PMBOK 2013]) se transcriben como flujo de control del proceso. Se especifican con arcos, puertas y leyendas en color rojo. Las dependencias “*Fin a Comienzo*” se resuelven con flujo de control secuencial, mientras que las dependencias “*Comienzo a Comienzo*” y “*Fin a Fin*” exigen ejecución en paralelo (puertas de bifurcación y sincronización en paralelo).

Así, puede obtenerse un modelo *BPMN* (Figura 2.4) que se corresponde con el plan del proyecto de partida (Figura 2.1). Este proceso refleja aspectos del *flujo de control* y de la *perspectiva temporal* que se ha derivado del plan del proyecto en *MS Project*.

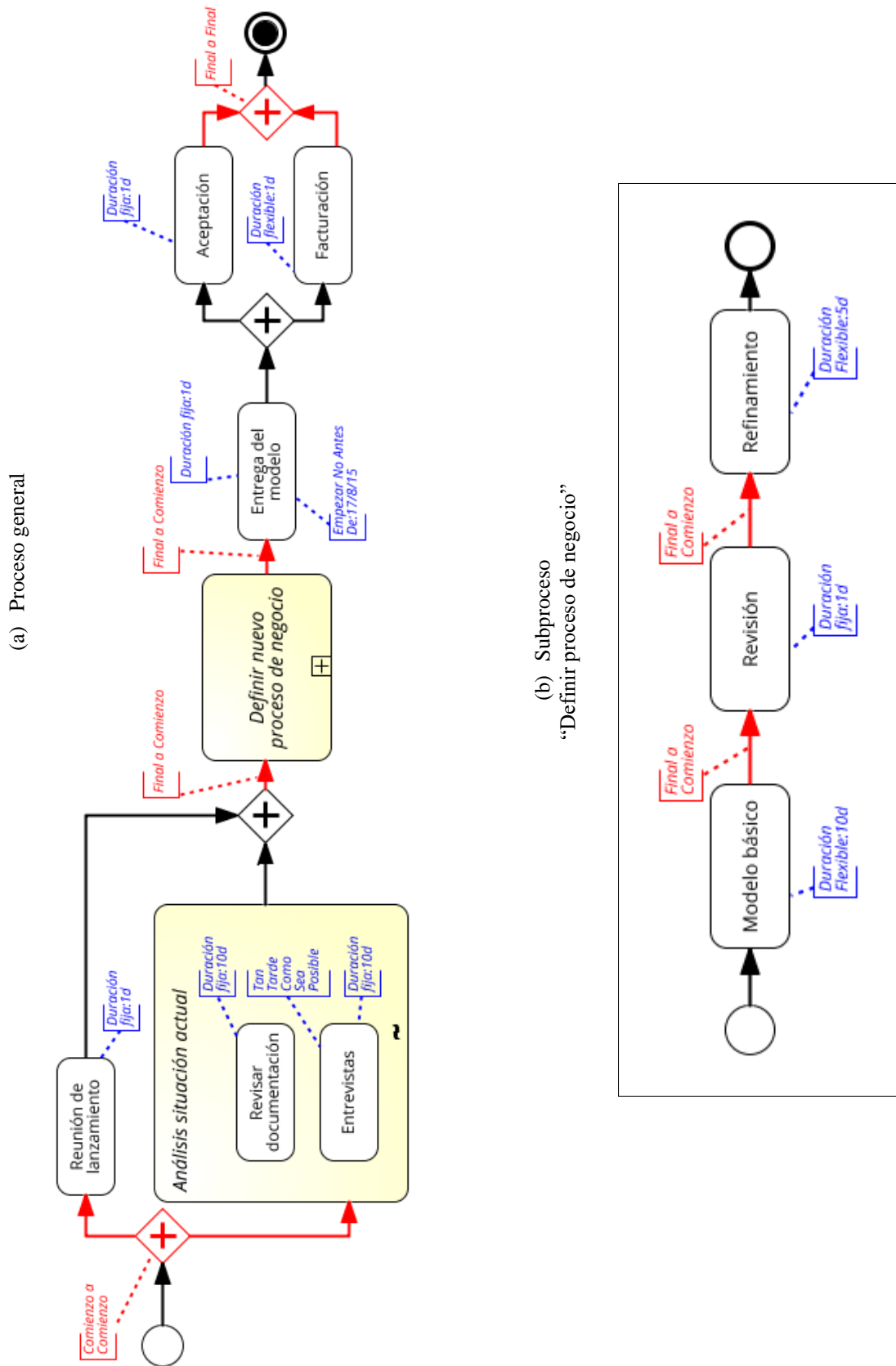


Figura. 2.4. Tareas del proyecto como actividades del proceso BPMN

### 2.3.3 Discusión

En la sección anterior se ha planteado un camino para derivar un *proceso de negocio BPMN* a partir de un *plan de proyecto* realizado con *MS Project*. Hay que reflexionar sobre algunos puntos que han permitido llegar a ese resultado:

- i) *MS Project* soporta el *Álgebra de Allen* [Allen 1983] y el método *Critical Path Method (CPM)* [Kelley & Walker 1959] que genera un grafo de duración mínima para un plan expresado con un grafo *PERT* [Malcolm et al. 1959; PMI:PMBOK 2013]. *BPMN* no soporta esta semántica, aunque sí tiene artefactos (*eventos de temporización y señales*) que permitirían generar construcciones o patrones, como se realiza en el trabajo [Flores & Sepúlveda 2011] para recoger la *dimensión temporal*. En el diagrama *BPMN* se han incluido anotaciones para todas las reglas detectadas, pero eso no supone una formalización de restricciones y dependencias temporales, por lo que se sugiere la cuestión: ¿De qué modo pueden formalizarse estas reglas sobre *BPMN* u otros lenguajes de especificación de procesos utilizados por los expertos del software?
- ii) *MS Project* también gestiona recursos y grupos de recursos, por tanto: ¿Podría enriquecerse la definición del proceso contemplando recursos, grupos de recursos y sus asignaciones?
- iii) Además de *MS Project*, ¿Qué otros sistemas de gestión de proyectos podrían utilizarse para extraer estas aproximaciones a los procesos? ¿Podrían utilizarse otros *legacy systems*, como sistemas de *gestión documental o de contenidos*, *ERPs*, *CRMs*, *SCMs* o bien *desarrollos a medida*?
- iv) ¿Podrían extraerse automáticamente estos procesos? ¿Qué enfoques tecnológicos podrían utilizarse?
- v) Se han establecido reglas de transformación que permiten ir de *un plan a un proceso*. Para un conjunto de proyectos se generaría un conjunto de procesos, pero ¿Sería posible obtener un *tipo o clase de procesos* que representase un *modelo de procesos genérico*?

A raíz del ejemplo analizado, estos retos planteados permitirán centrar los objetivos del trabajo de investigación.

## 2.4 Hipótesis y objetivos

El objetivo fundamental de esta investigación es “*Extraer aproximaciones a los procesos de las «TI» desde legacy systems* [Bisbal et al. 1997, 1999; Stavru et al. 2013] *utilizando la dimensión temporal*”. El objetivo de partida para el trabajo de investigación es llevar a cabo: **(ob.0) Un estudio del estado del arte** relacionado con:

- i) *Sistemas de información heredados (legacy systems)*. Se analizarán los problemas que plantean y los enfoques para su modernización, especialmente los relacionados con el

tratamiento de bases de datos (*legacy databases*).

ii) *Lenguajes para representación de procesos del software*. Se estudiarán los sistemas de representación de procesos más extendidos en el sector *TI*, contemplando tanto estándares generales como específicos para el ciclo de vida del software.

iii) *Especificación de reglas temporales sobre procesos*. Se estudiarán las fortalezas y debilidades de los sistemas del apartado anterior. Se seleccionará la bibliografía científica que contemple la modelación de la dimensión temporal sobre los procesos. El contraste de estos dos puntos permitirá establecer una taxonomía de reglas temporales que abarque la tipología de reglas de negocio manejadas en el sector *TI*. Por otro lado, permitirá detectar las necesidades de extensión de los modelos de procesos a utilizar.

En relación con los bloques anteriores se definen hipótesis de trabajo y, sobre ellas, los objetivos detallados a alcanzar.

#### 2.4.1 Legacy systems como sistemas origen

Esta sección está relacionada con los sistemas de información utilizados habitualmente por las «*TI*» que se plantean *BPM* como nuevo enfoque estratégico en su negocio. En los últimos años las «*TI*» han usado sistemas que, en mayor o menor medida, automatizan sus procesos de negocio, facilitando la adopción de metodologías, la definición y gestión de equipos de trabajo, personas, perfiles, recursos, productos entregables y la planificación y control de proyectos y tareas ligadas al software. Estos sistemas, bien como productos de mercado adaptables o como desarrollos específicos (a medida o propietarios) suelen estar soportados por bases de datos relacionales [*Codd 1970, 1992*] (*legacy databases*) que recogen estructuras de datos y, en muchos casos, tienen embebidas reglas de negocio [*Türker & Gertz 2001*] ligadas a estos procesos.

El esquema relacional de una base de datos (*legacy database*) se sitúa en el nivel de implementación de una aplicación. Está representado por tablas que definen tipos de datos y reglas intrínsecas o estructurales sobre valores de columnas en una fila, diversos tipos de claves (primarias, alternativas, ajenas). Estas definiciones son persistentes y están plasmadas en el catálogo de cada base de datos configurando la *Meta-base de datos*. Las versiones comerciales de gestores de bases de datos que soportan *disparadores* (*triggers*) permiten implementar reglas específicas codificadas, normalmente, como bloques de programación en lenguajes procedurales propietarios, que encapsulan código catalogado como artefactos ligados mediante eventos a las tablas. El referente ISO de lenguaje propietario procedural es *ISO SQL/PSM*<sup>31</sup> [*Eisenberg 1996; Melton & Simon 2002*] para la codificación de rutinas, funciones y disparadores en el esquema de la base de datos. Los productos comerciales basan sus lenguajes propietarios en variantes de *SQL/PSM*, aunque cada uno con una sintaxis concreta; así *PL/SQL* [*Feuerstein & Pribyl*

<sup>31</sup> El componente “Persistent Stored Modules” *SQL/PSM* es propuesto como lenguaje procedural para la especificación de funciones, procedimientos y *triggers*. Aparece por primera vez en el estándar *ISO SQL:1999* y se ha mantenido hasta la versión *ISO SQL:2011*.

2005] es el producto de ORACLE™, Transact-SQL [Colledge 2008; Ben-Gan 2012; Microsoft™:T-SQL 2014] el de SQL\*Server™, PL/pgSQL el de PostgreSQL™ y MySQL™ stored procedures [Harrison & Feuerstein 2008] el del popular MySQL[Dubois 2005].

Los *Meta-Modelos* de un esquema relacional se sitúan en el nivel de abstracción cercano a cada plataforma (*nivel PSM*). Se consideran dos tipos de *Meta-Modelos basados en sintaxis abstracta (ASTM)*: *genéricos (GASTM)* para las estructuras tabulares de cualquier base de datos relacional, y *específicos (SASTM)* para cada caso concreto de gestor de base de datos comercial, contemplando estructuras, restricciones y disparadores codificados en el lenguaje procedural propietario de cada producto (ver Anexo II donde figuran los *Meta-Modelos* de algunos *legacy systems* seleccionados para el estudio).

Se consideran las siguientes hipótesis y objetivos:

**(hp.1) Uso de legacy systems en «TI».** La mayoría de las «TI» utilizan frecuentemente sistemas de información automatizados (*legacy systems*) para i) *Planificación y gestión de proyectos*, p. ej.: «MS Project» [Hansen & Hansen 2013; Stover 2007]» o «RedMine» [Lang 2010]; ii) Gestores de contenido o documentales «Enterprise Content Management Systems (ECM)» como «Alfresco» [Shariff 2013] o «MS Sharepoint» [Smith & Bates 2007]; y iii) *Gestión general de recursos (ERPs)* [Hendricks et al. 2007]: procesos de negocio de compras, ventas, gestión financiera, gestión de activos, CRMs, SCMs [Hendricks et al. 2007], o bien en *desarrollos a medida*.

**(hp.2) Legacy database como estado de los procesos de negocio.** Los sistemas referenciados en (hp.1) suelen tener una *legacy database* relacional basada en SQL [Melton & Simon 2002], que hace persistentes estados que son el resultado de la ejecución de procesos de negocio [van der Aalst 2015]. En estas *legacy databases* aparecen tareas, recursos, grupos de recursos, restricciones temporales y asignaciones de recursos que están relacionados con las perspectivas de *control del flujo, información, temporal, organizacional, operacional, y de casos* [Jablonski 1996; Reichert & Weber 2012; van der Aalst et al. 2004, 2005, 2007, 2012] que pueden encontrarse en un proceso. Estos sistemas, en mayor o menor medida, contienen estructuras y reglas de negocio (*tablas, restricciones, disparadores*) que pueden, además, estar embebidas en los esquemas mediante código propietario [Feuerstein & Pribyl 2005].

*MS Project Server* no es un sistema de la categoría PAIS. Cada proyecto planificado está relacionado con instancias de uno o más procesos realizados por la «TI». La dimensión temporal de un proyecto (*visto como un proceso*) está formalmente definida con este sistema, de modo que la *legacy database* representa un conjunto de instancias de procesos, compuestos por tareas que se han planificado y que están en



ejecución o han terminado (lo cual puede ser visto como una sucesión de eventos que acontecen en el proceso de la «TI», independientemente de que el proceso esté automatizado o se ejecute manualmente mediante la invocación de sucesivas funciones por los usuarios del sistema).

Para utilizar estos *legacy systems* como fuente para extraer procesos de negocio se plantean los siguientes objetivos.

**(ob.1.) Establecer Meta–Modelos de tareas para legacy systems de uso habitual en «TI».** Se estudiarán los «*legacy systems* referenciados en (h1)» respecto a estructuras y reglas existentes en su capa de persistencia, relacionadas con las distintas perspectivas de los procesos, y en particular, las estructuras relacionadas con la *dimensión temporal*: tareas, agrupaciones de tareas, reglas y dependencias temporales entre ellas. El estudio se enfoca a representar *Meta–Modelos* de tareas existentes en cada *legacy system* (Ver Anexo II), que, a nivel tecnológico, están vinculados con los *Meta–Modelos* de los sistemas relacionales que los soportan (*GASTM*, *SASTM*; ver Anexo III).

Se prestará atención especial a las distintas bases de datos «*MS Project Server*» definidas sobre el gestor de base de datos *SQL\*Server* [*Colledge 2008*; *Ben-Gan 2012*; *Microsoft™:T-SQL 2014*], estudiando el soporte de la dimensión temporal de este *legacy system*.

#### 2.4.2 Reglas temporales sobre procesos

*BPM* contempla distintas perspectivas [*Jablonski 1996*; *Reichert & Weber 2012*; *van der Aalst et al. 2004, 2005, 2007, 2012*] para un proceso de negocio. Vamos a poner el foco en la *perspectiva temporal*, buscando artefactos ligados a la gestión de tareas en *legacy systems* que han venido sirviendo para *planificar y controlar proyectos informáticos*. Creemos que, con esta perspectiva, muy común en este tipo de sistemas, llegaremos a representar una aproximación de los procesos, extrayendo actividades, características del flujo de control y un conjunto de reglas o restricciones temporales que refinarán estos procesos. Con posterioridad, manteniendo el mismo esquema, podrían integrarse otras *perspectivas* para generar *procesos* más enriquecidos; por ejemplo, utilizando la *perspectiva de recursos o de casos* [*van der Aalst et al. 2005, 2007*] donde existen *Meta–Modelos* basados en *UML* [*Awad et al. 2009*; *Stroppi et al. 2012*] que servirían para extender nuestra propuesta inicial. Se consideran las siguientes hipótesis:

**(hp.3) Perspectiva temporal en los lenguajes de modelación de procesos.** Existen múltiples lenguajes para la modelación de procesos que contemplan la perspectiva temporal, recogiendo distinto tipo de restricciones y dependencias temporales entre actividades (Véase Capítulo 4: trabajo relacionado).

**(hp.4) Comprensibilidad de los lenguajes gráficos.** La notaciones gráficas para especificación de procesos aportan beneficios significativos en el contexto *BPM* [*Lu*

& Sadiq 2007] tanto para *expertos del dominio* como para *expertos en TIC*. *BPMN* [OMG:BPMN 2013; Iso/Iec:19510 2013] es uno de los estándares, basados en notación gráfica, más extendidos para la especificación de procesos [Bonnet et al. 2014]. *BPMN* contempla la perspectiva temporal pero algunos autores, entre ellos [Gagné & Trudel 2009a; Flores & Sepúlveda 2011; Watahiki et al. 2011; Cheikhrouhou et al. 2013b] advierten cierta debilidad para modelar reglas de tiempo frente a otras propuestas.

**(hp.5) Meta-Modelos y lenguajes declarativos para ganar interoperabilidad.** La *sintaxis abstracta* (uso de *Meta-Modelos*) y los *enfoques declarativos* [Fahland 2009a, 2009b; Pichler et al. 2011; Reijers et al. 2013] contribuyen a generar especificaciones con mejor nivel de *interoperabilidad* [Geraci 1991] en comparación con el empleo de la *sintaxis concreta de los lenguajes imperativos*.

En cuanto a la gestión de procesos específicos del software, los estándares con *robustos Meta-Modelos* de procesos, como *BPMN*, *SPEM 2.0* [Bendraou et al. 2007; OMG: SPEM 2008] o [Iso/Iec:24744 2007, 2014] están cobrando especial relevancia en el sector.

Se proponen los siguientes objetivos:

**(ob.2.) Establecer una taxonomía de reglas temporales.** Debe ser lo suficientemente amplia y representativa de las distintas posibilidades que pueden encontrarse sobre procesos de negocio en un abanico de lenguajes de especificación de procesos. Esta taxonomía debe servir para medir la capacidad de representación de distintos *Meta-Modelos* de procesos dentro del marco del trabajo de investigación.

**(ob.3.) Diseñar un Meta-Modelo de procesos que soporte la taxonomía de reglas temporales.** El *Meta-Modelo* debe contener el conjunto mínimo de artefactos para especificar reglas y dependencias temporales de la taxonomía (ob.2) con un enfoque declarativo [Fahland 2009a, 2009b; Pichler et al. 2011; Reijers et al. 2013]. Se utilizará *OCL* [OMG:OCL 2014; Iso/Iec 19507:2012] para formalizar las restricciones y dependencias temporales. Este *Meta-Modelo* será extensible para soportar la consecución de los objetivos de la propuesta de investigación.

### 2.4.3 Lenguajes para representación de procesos

Hay que obtener modelos inteligibles y de nivel cercano al experto en software, independientes de la tecnología (nivel *CIM*) que faciliten la comprensión, reestructuración y redefinición de los procesos y reglas de negocio. Pondremos el foco en el uso de *Meta-Modelos* de procesos utilizados por las «TI». El experto manejará un *Meta-Modelo* que permita representar las características esenciales de cualquier modelo de procesos del software:

**(hp.6) Meta–Modelos para especificación de procesos del software.** Se pueden emplear los estándares generales *BPMN* [OMG:BPMN 2013; Iso/Iec:19510 2013] o *XPDL* [WfMC:XPDL 2012], de extendido uso para la representación y portabilidad de procesos, o bien los estándares específicos para procesos del software: i) *SPEM 2.0* [Bendraou et al. 2007; OMG: SPEM 2008] e ii) [Iso/Iec:24744 2007, 2014], así como nuestro entorno metodológico *NDTQ-Framework* [Ponce et al. 2013; Aragón 2014] basado en la metodología *NDT* [Escalona 2004; Escalona & Aragón 2008].

Todos tienen en común la existencia de un *Meta–Modelo* de procesos, pero al igual que sucede con *BPMN*, sus correspondientes *Meta–Modelos* deben ser extendidos con la semántica temporal contemplada en el *Meta–Modelo* (ob.3).

Se propone:

**(ob.4.) Extensión del Meta–Modelo BPMN de procesos con reglas temporales.** En el alcance de este trabajo de investigación se selecciona *BPMN* por su popularidad y capacidad para el intercambio de diagramas (*interoperabilidad*). Se extenderá el *Meta–Modelo BPMN* de procesos con la semántica temporal del *Meta–Modelo* (ob.3).

#### 2.4.4 Extracción de procesos desde legacy databases

En el Capítulo 4, trabajo relacionado, existen múltiples referencias que aplican procesos de ingeniería inversa [Chikofsky & Cross 1990; Favre 2004, 2010; Bruneliere et al. 2010a], basados en *MDA* [OMG:MDA; Kleppe et al. 2003]: i) los enfoques específicos *MDSM* [Favre 2004] sobre los procesos del software y ii) *ADM* [Newcomb 2005; OMG:ADM 2005; Ulrich & Newcomb 2010] sobre *legacy systems*. Por otro lado, también se relacionan algunos que buscan levantar modelos de procesos [van der Aalst et al. 2007, 2012; van der Aalst 2011, 2013, 2014, 2015; Günther & van der Aalst 2007; Ingvaldsen & Gulla 2008; Reichert & Weber 2012, Verbeek et al. 2011; Kalenkova et al. 2014; Pérez-Castillo et al. 2009, 2011a], tanto desde *PAIS* como desde sistemas no orientados a procesos.

**(hp.7) Trazabilidad entre estados de legacy databases y eventos de procesos.** Las *legacy databases* representan estados que son consecuencia de la ejecución de procesos en una organización [van der Aalst 2015]. Estos estados recogen propiedades que están relacionadas con distintas perspectivas de los procesos. Es posible establecer una correspondencia entre estados de una *legacy database* y la representación con lenguajes específicos de definición de procesos de la perspectiva analizada.

**(ob.5.) Aplicar ingeniería inversa MDE para extraer procesos desde MS Project Server (nivel PSM) y llevarlos a BPMN (nivel CIM).**

El enfoque *MDE* ofrece distintos niveles de abstracción y mecanismos para simplificar la automatización del ciclo de vida del software, tanto para generación de

código (ingeniería directa, que permite transformar especificaciones de alto nivel de abstracción en código ejecutable) como para extraer conocimiento de un *legacy system* (la ingeniería inversa [Chikofsky & Cross 1990] parte de sistemas reales y niveles de abstracción bajos que representan al código, y es capaz de generar modelos de mayor nivel de abstracción que facilitan la comprensión del sistema). Partiendo de la capa de persistencia, representada habitualmente por *legacy databases* relacionales [Codd 1970, 1992], vamos a desarrollar una propuesta que extraiga artefactos de este nivel (estructuras y reglas temporales) y los transforme en otros artefactos de modelos de procesos al nivel de abstracción del experto de negocio.

MDA, entre otras características, ofrece niveles de abstracción para representar *modelos* y *Meta-Modelos* que describen diferentes plataformas (*PSM*, *PIM*, *CIM*), con objeto de facilitar un alto nivel de *interoperabilidad*, basada en el concepto «*Meta-Object Facility (MOF)*» [OMG:MOF 2011].

**(ob.5.1.) Definir un Meta-Modelo independiente de plataforma (PIM) que extienda al Meta-Modelo (ob.3) y soporte la extracción de procesos desde MS Project Server.**

Este *Meta-Modelo (PIM)* debe servir para representar cualquier instancia de un proyecto como un proceso con las reglas temporales formalizadas en el *Meta-Modelo (ob.3)*, pero también debe referenciar el origen (*legacy system*), pues *MS Project Server* sólo es un ejemplo. Se estudiarán las posibles colisiones que puedan aparecer si una actividad de un proceso está replicada en varios *legacy systems*, dando una solución con reglas adicionales sobre este *Meta-Modelo*.

**(ob.5.2.) Diseñar un algoritmo de transformación PSM→PIM.** Cada *proyecto* (procedente del *Meta-Modelo*, de nivel *PSM*, *MS Project Server*, **ob.1**) es una instancia de *proceso* en el *Meta-Modelo (ob.5.1)* de tareas de. Se diseñará un algoritmo detallado que mapee los artefactos de nivel *PSM* en artefactos de nivel *PIM*.

**(ob.5.3.) Diseñar un algoritmo de transformación PIM→CIM.** Se diseñará un algoritmo detallado que realice las transformaciones *PIM→CIM* en el caso de *BPMN*. Aunque el sistema destino podría ser otro (p. ej. *SPEM 2.0*, *NDTQ-Framework* o *Iso/Iec:24744*).

## 2.4.5 Verificación del enfoque en un caso de estudio

De los distintos proyectos de soporte metodológico desarrollados por el grupo de investigación *IWT2*<sup>32</sup>, se ha seleccionado el proyecto *AQUA-WS* [Cutilla et al. 2011], llevado a cabo en la *Empresa Metropolitana de Abastecimiento y Saneamiento de Aguas*

<sup>32</sup> <http://iwt2.org/>

de Sevilla S.A. (EMASESA<sup>33</sup>). En este proyecto, la función de los investigadores del grupo estuvo asociada al control de calidad de este proyecto de modernización Web de toda la arquitectura de sistemas del cliente.

**(hp.8) NDT como marco metodológico en AQUA-WS.** El marco metodológico de referencia fue *NDT* [*Escalona 2004; Escalona & Aragón 2008*].

**(hp.9) MS Project legacy database como sistema origen en AQUA-WS.** En este proyecto se utilizó *MS Project* y *MS Project Server 2007* sobre gestor de base de datos *SQL\*Server 2008*, para la planificación general y control detallado del desarrollo de subsistemas. Existen patrones de planificación (proyectos parametrizados) para las fases *NDT*.

Se plantean los siguientes objetivos específicos sobre este caso de estudio:

**(ob.6.) Analizar los resultados en referencia a los procesos y modelos de procesos obtenidos en el caso AQUA-WS.** Cada proyecto, o subconjunto de tareas, se trata como un caso de un proceso; ahora bien, la parametrización de patrones de proyecto para las fases *NDT* permite generar modelos de procesos, donde cada caso de aplicación de una fase *NDT* es una instancia del modelo de procesos. Se debe verificar la calidad de los procesos obtenidos así como las limitaciones y posibles ampliaciones del enfoque de la tesis.

---

<sup>33</sup> <http://www.emasesa.com/>

### 2.4.6 Resumen hipótesis vs objetivos de la investigación

La [Tabla 2.1](#) resume la trazabilidad entre hipótesis y objetivos que se han detallado en los apartados anteriores.

**Tabla. 2.1.** Hipótesis vs Objetivos

<div style="text-align: center;"> <b>Hipótesis</b> / <b>Objetivos</b> </div>	(ob.1.) Establecer Meta-Modelos de tareas para legacy systems de uso habitual en «TI»	(ob.2.) Establecer una taxonomía de reglas temporales	(ob.3.) Diseñar un Meta-Modelo de procesos que soporte la taxonomía de reglas temporales	(ob.4.) Extensión del Meta-Modelo BPMN de procesos con reglas temporales	(ob.5.) Aplicar ingeniería inversa MDE para extraer procesos desde MS Project Server (nivel FSM) y llevarlos a BPMN (nivel CIM).	(ob.6.) Analizar los resultados en referencia a los procesos y modelos de procesos obtenidos en el caso AQUA-WS
(hp.1) Uso de legacy systems en «TI».	✓					
(hp.2) Legacy database como estado de los procesos de negocio	✓					
(hp.3) Perspectiva temporal en los lenguajes de modelación de procesos		✓	✓			
(hp.4) Comprensibilidad de los lenguajes gráficos		✓	✓			
(hp.5) Meta-Modelos y lenguajes declarativos para ganar interoperabilidad		✓	✓			
(hp.6) Meta-Modelos para especificación de procesos del software				✓		
(hp.7) Trazabilidad entre estados de legacy databases y eventos de procesos					✓	
(hp.8) NDT como marco metodológico en AQUA-WS						✓
(hp.9) MS Project legacy database como sistema origen en AQUA-WS						✓

## 2.5 Arquitectura de la solución

La [Figura 2.6](#) es la arquitectura de la solución de este trabajo de investigación, que ilustra las hipótesis de trabajo y objetivos que se han propuesto.

La base de partida son *legacy systems* de uso habitual en las «TI». El resultado está formado por procesos expresados en lenguajes familiares al experto en el negocio de software. El enfoque está basado en ingeniería inversa *MDE* para la extracción de la *perspectiva temporal* subyacente en la *legacy database* de cada sistema origen.

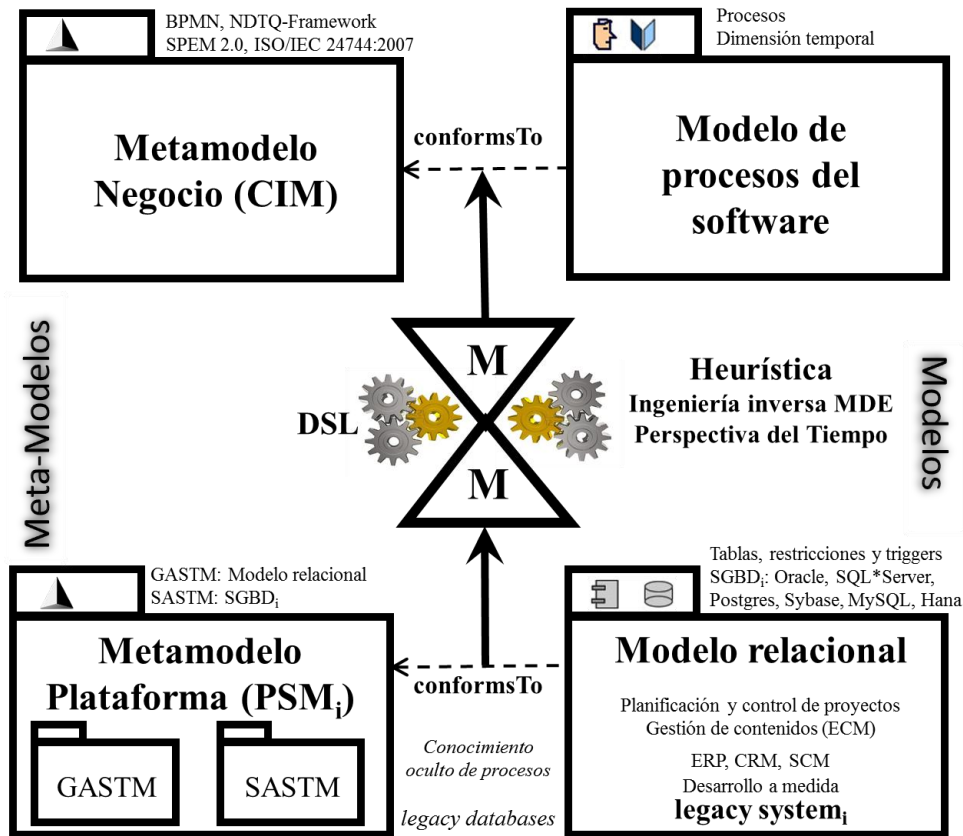


Figura. 2.5. Arquitectura MDE para extraer procesos desde legacy databases

Para extraer la *dimensión temporal* desde *legacy systems* es necesario analizar cada tipo de sistema y reconocer dicha dimensión. Los sistemas de *planificación y gestión de proyectos* y los *gestores de contenidos (ECMs)* son utilizados por todas las «TI», al margen de existir también otros como los *ERPs*, *CRMs* y *SCMs*, y por último los *desarrollos a medida*. La capa más estable de estos sistemas es su base de datos (*legacy database*). Aunque los *legacy systems* no estén orientados a procesos, almacenarán estados que son el resultado de la ejecución de estos en la «TI» y es posible hacer una reinterpretación [van der Aalst 2015] que permita derivar procesos. El foco inicial se va a poner en sistemas que representan bien la dimensión temporal, como *MS Project*. Para estos sistemas es necesario estudiar: i) La representación de tareas, restricciones y dimensiones temporales entre ellas; ii) La plataforma que sustenta su *legacy database* (p. ej.: *MS SQL\*Server*). Esto pasa por establecer *Meta-Modelos* para cada plataforma o *legacy database* (*GASTM*, *SASTM*).

Los lenguajes de representación de procesos que utiliza el experto en software suelen estar basados en estándares: generales como *BPMN* [OMG:BPMN 2013; Iso/Iec:19510 2013] o *XPDL* [WfMC:XPDL 2012], o bien en estándares específicos como *SPEM 2.0* [Bendraou et al. 2007; OMG: SPEM 2008] o [Iso/Iec:24744 2007, 2014], o bien en entornos metodológicos específicos, siendo para nosotros el más cercano, el nuestro: *NDTQ-Framework* [Ponce et al. 2013; Aragón 2014] basado en la metodología *NDT*

[Escalona 2004; Escalona & Aragón 2008]. Todos ellos tienen en común la existencia de *Meta-Modelos* de procesos que necesitan ser reforzados para representar adecuadamente la dimensión temporal, razón por la cual se hace necesario establecer una clasificación o taxonomía de reglas temporales y un *Meta-Modelo PIM* que soporte el proceso de extracción automática o ingeniería inversa mediante *MDE*.

## 2.6 Resumen y conclusiones

Las «*TI*» manejan procesos de negocio complejos, cambiantes e íntimamente ligados a personas, equipos y protocolos de comunicación entre ellos, incluyendo actividades que no siempre es posible automatizar. Los *legacy systems* evolucionan continuamente para soportar nuevos estándares, tecnologías y mejores prácticas. El enfoque *BPM* es un valor estratégico indiscutible para todo tipo de organización, y también lo es para las «*TI*». Este enfoque propone un ciclo de mejora continua de los procesos, donde es necesario modelarlos, ponerlos en ejecución, y evaluar el grado de cumplimiento o conformidad de estos respecto a los modelos definidos, para mejorarlos y eliminar inconvenientes que perjudican a la organización. Mientras que los *PAIS* proveen herramientas automáticas que ayudan a la mejora continua, en general, otros *legacy systems* carecen de ellas. No obstante, cualquier organización deja trazas de sus procesos, como estados persistentes en *legacy databases*.

En este Capítulo se utiliza un ejemplo motivador, basado en la planificación de un proyecto con el sistema *MS Project*. El caso representa un *proyecto* compuesto por *tareas*, y una representación alternativa del mismo con la notación *BPMN*, asociando un *proceso al proyecto* y *actividades a las tareas*, además de trasladar un conjunto de restricciones y dependencias temporales del proyecto al proceso, que condicionan el flujo de control del mismo y asociando anotaciones que recogen las restricciones de la dimensión temporal asociada al proyecto. Este caso de transformación parte del estado del proyecto (almacenado en la *legacy database* de *MS Project*), utiliza el análisis de la dimensión temporal y usa ciertos patrones para acondicionar el flujo de control del proceso con *BPMN*.

Tras el ejemplo, se han definido hipótesis de partida y objetivos que se persiguen con este trabajo de investigación, agrupados en: i) Tratamiento de los sistemas de información de origen (*legacy systems* comunes en las «*TI*»); ii) Estudio de la dimensión temporal de los procesos; iii) Elección de lenguajes para la representación de procesos en notaciones cercanas al experto; iv) Métodos basados en ingeniería inversa *MDE* para transformar automáticamente estados de *legacy databases* en aproximaciones a los procesos del software, contemplando aspectos del control del flujo y de la dimensión temporal; v) Por último se selecciona un caso de estudio, suficientemente significativo en cuanto a nivel de complejidad, *AQUA-WS Project*, que es un proyecto de modernización Web en la Empresa Municipal de Abastecimiento de Aguas del Excmo. Ayuntamiento de Sevilla (*EMASESA*), donde se aplican los principios metodológicos de *NDT*, y se han utilizado



sus procesos en las distintas fases de concepción, análisis, diseño, construcción e implantación de distintos subsistemas software. El caso debe servir para contrastar el grado de cumplimiento de los objetivos definidos y las limitaciones del enfoque propuesto en el trabajo de investigación.



## **PARTE II. ANTECEDENTES**



# Capítulo 3. ESTADO DEL ARTE

*“La verdadera ciencia enseña, sobre todo, a dudar y a ser ignorante”  
Ernest Rutherford (1871-1937)*

Las organizaciones del software «TI» se desenvuelven en una sociedad globalizada muy compleja y competitiva. Como en otros sectores, necesitan del enfoque *BPM* para gestionar su negocio y reducir los problemas detectados en los 70's por Dijkstra en la denominada *crisis del software*, algunas de cuyas causas siguen estando aún vigentes. En este Capítulo se analizan estándares y buenas prácticas en relación con los objetivos que se han planteado para la tesis en el Capítulo 2. Se revisa el paradigma dirigido por modelos *MDE* y la conocida iniciativa *MDA* de la *OMG*. Se estudia el enfoque *BPM*, desde el punto de vista de su ciclo de vida de mejora continua y la arquitectura de sistemas y estándares que desarrollan este paradigma. Específicamente, para los procesos del software, se han analizado lenguajes, estándares, tendencias y mejores prácticas. Se dedica un apartado a describir las características de nuestra metodología «*Navigational Development Techniques (NDT)*» y *NDTQ-Framework* fundamentado en *NDT*, *MDE*, *UML* y *OCL*, que propone procesos para distintas fases del ciclo de vida del software. Por último se han analizado los problemas de los *legacy systems* y el enfoque de modernización *ADM* propuesto por el *OMG*.

## 3.1 Ingeniería del software dirigida por modelos

### 3.1.1 Problemas del desarrollo tradicional de software

Hace décadas que *Dijkstra* identificó la problemática del desarrollo de software, acuñando el término «*Crisis del software*» [*Dijkstra 1972; Pressman 2010*]; aunque han transcurrido más de cuarenta años, algunos de los problemas siguen aún estando vigentes:

- i) Baja calidad del producto en general. El software no satisface los requisitos y la funcionalidad alcanzada no es adecuada. La captura de requisitos del sistema es uno de los puntos críticos de un sistema, pues, por diversas razones, no es fácil que, utilizando lenguaje natural, un *experto en TI* obtenga fielmente el conocimiento que posee el *experto en el dominio* [*Sommerville & Kotonya 1998*]. Esto supone una fuente de errores importantes, que se propagan y multiplican a otros niveles, generando, entre otros inconvenientes, retrasos y costos económicos [*Lauesen & Vinter 2001*]. La detección en etapas posteriores obliga a redefinir los

requisitos y a corregir los artefactos dependientes en los niveles afectados del sistema.

- ii) Desviaciones significativas en la planificación, tanto en plazos como económica. Costes desorbitados de mantenimiento con el tiempo.

En [Greenfield et al. 2004] se apuntan posibles causas de estos inconvenientes:

- i) Deficiencias en la gestión del conocimiento. No se maneja bien el *Know how* ni el código generado en proyectos anteriores.
- ii) Desarrollo monolítico con componentes muy interdependientes.
- iii) Utilización de lenguajes de bajo nivel que, ofreciendo flexibilidad y eficiencia en la ejecución, provocan baja productividad por la cantidad de errores que el programador debe subsanar.
- iv) Aunque se ha avanzado en estandarización y modelos de madurez del software, los procesos no son tan estables como otros procesos de la ingeniería (construcción, fabricación, etc.).
- v) La elevada demanda de los usuarios provoca el planteamiento de proyectos en los plazos muy reducidos.

En relación a estos problemas, sus causas y la repercusión en proyectos del software, la empresa consultora *The Standish Group International, Inc.* elabora el informe «*The Standish Group Chaos*» [Standish 2012] que evalúa grandes proyectos software en el mundo para medir, entre otras variables: a) Proyectos que acaban con éxito (*Successful*), cumpliendo plazo y costes, b) Proyectos que terminan con desviaciones en plazo o costes (*Challenged*) y c) Proyectos cancelados (*Failed*). La Figura 3.1 muestra el informe global de 2012, donde sólo el 39% de los proyectos acaban según lo presupuestado y planificado y el 18% son cancelados.

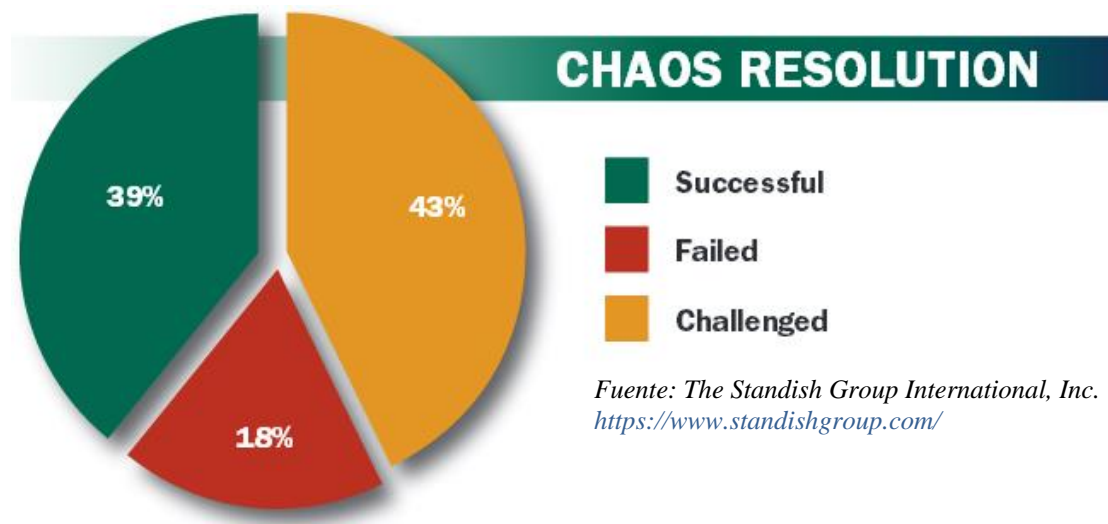
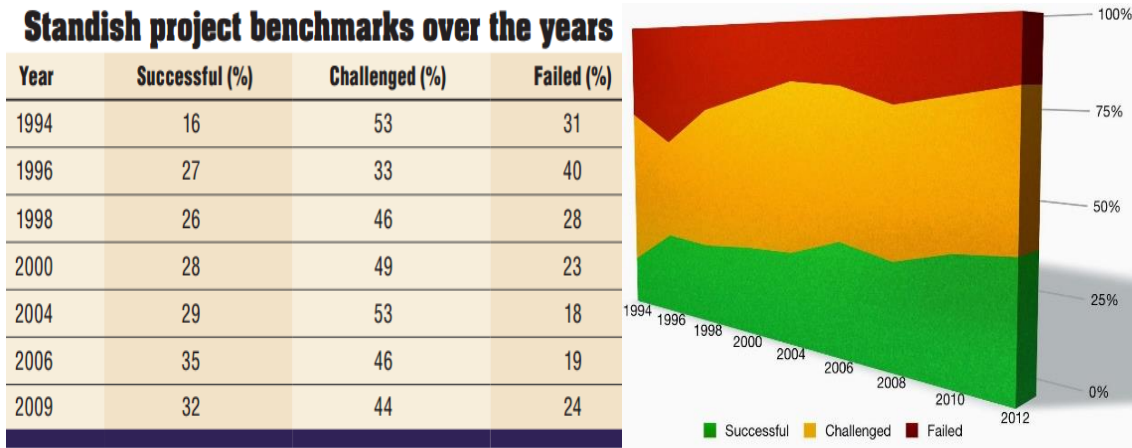


Figura. 3.1. Informe Chaos 2012



Fuente: The Standish Group International, Inc.  
<https://www.standishgroup.com/>

Figura. 3.2. Informe Chaos: serie temporal 1994-2012

El informe *Chaos* [Standish 2012] comienza a elaborarse en 1994 (tabulado en la Figura 3.2) y, aunque se observa que la tasa de éxitos ha ido mejorando, pues sólo en el informe de 2006 la tasa de proyectos finalizados con éxito llegó hasta el 35%, mientras que el valor más bajo de proyectos cancelados nunca ha bajado del 18% y llegó al 40% en el informe de 1996. La consultora desglosa estos datos y analiza una serie de factores o causas, que están muy en la línea de los esgrimidos por los autores anteriores [Dijkstra 1972; Sommerville & Kotonya 1998; Greenfield et al. 2004]. Aunque existen autores que discuten los métodos estadísticos y la exactitud de los resultados obtenidos por Standish, como [Moløkken & Jørgensen 2003; Eveleens & Verhoef 2009], la discrepancia suele estar en el segundo grupo de proyectos que tienen desviaciones en plazos y costes, lo que significa que es aceptado que la tasa de proyectos que finalizan con éxito es demasiado pequeña y la de proyectos cancelados sigue siendo considerable.

[Ruiz-González & Canfora 2004] identifican al software como un *producto con un elevado nivel de complejidad e impredecibilidad*. Observan las propiedades de los procesos específicos del software comparándolos con otros tipos de procesos; destacando:

- i) Los procesos software son complejos debido a que se ven fuertemente condicionados por multitud de circunstancias *impredecibles* y por muchos *equipos de trabajo*. Esta característica hace que los procesos software sean completamente diferentes a procesos de producción de otros sectores productivos.
- ii) Las fases de diseño y producción no están claramente diferenciadas, como sucede en otros procesos industriales. Frecuentemente, surgen nuevos requisitos durante la fase de desarrollo del producto.
- iii) Es difícil presupuestar y planificar los procesos software con fiabilidad y certeza para garantizar una calidad suficiente del producto.
- iv) Los procesos software tienen fuertes dependencias de los esquemas organizativos, de los protocolos de comunicación, coordinación y cooperación de equipos de

trabajo compuestos por usuarios y personas con diversos roles de distintas compañías. A esto hay que añadir la complejidad que supone manejar escenarios o entornos heterogéneos de desarrollo, con diversas tecnologías y versiones.

- v) La gestión de los procesos software está en continua y constante evolución debido a que frecuentemente incorpora diferentes ciclos de vida, estándares y buenas prácticas, que manejan distintas versiones de entregables para llegar al producto software definitivo.

El desarrollo de software también ha evolucionado según los lenguajes de programación predominantes en cada momento y el nivel de abstracción de dichos lenguajes.

- i) **Primera generación o código máquina.** La complejidad de los desarrollos era muy elevada, ya que el bajo nivel del lenguaje obligaba a realizar mucho código para implementar algoritmos y dificultaba la realización de pruebas.
- ii) **Segunda generación o lenguaje ensamblador.** Elevan el nivel de abstracción con respecto al código máquina, introduciendo mejoras a nivel cualitativo y productivo.
- iii) **Tercera generación o lenguajes procedurales.** Mejoran aún más la productividad pues se acercan más al pensamiento humano, sin embargo, suponen una pérdida de rendimiento.
- iv) **Cuarta generación o lenguajes orientados a objetos.** Introducen un mayor nivel de abstracción, manejando clases y sus instancias, y apareciendo términos como la encapsulación, el polimorfismo y la herencia. De nuevo, aumenta la productividad a costa de la eficiencia en ejecución.
- v) **Quinta generación o lenguajes orientados a aspectos** [Elrad et al. 2001; Filman et al. 2004; Reina et al. 2004, Soule 2010]. El objetivo principal es la separación de funcionalidades dentro del código de la aplicación «Programación Orientada a Aspectos (AOP)», que conduce a una mejor definición y localización de conceptos. Es más fácil mantener las aplicaciones y el código es más reutilizable [Soule 2010]. Existen todavía discrepancias entre autores sobre el establecimiento de esta categoría, aunque, hoy, supone el mayor nivel de abstracción para la clasificación de lenguajes y el mayor nivel de productividad, aunque el concepto se ha extendido a todo el ciclo del software, acuñándose términos como «Desarrollo del Software Orientado a Aspectos (AOSD)», «Análisis Orientado a Aspectos (AOA)» y «Diseño Orientados a Aspectos (AOD)».

En paralelo a los avances en la capacidad de los lenguajes ha crecido la complejidad de las plataformas de desarrollo, evolucionando mucho más rápido que los primeros; entre otras, «Java Enterprise Edition» (J2EE) [Rubinger et al. 2014] y «Microsoft .NET Framework» [Millas 2013] están compuestas por miles de clases y métodos con un complejo intrincado de dependencias que pueden provocar efectos secundarios inesperados. Estas plataformas demandan un esfuerzo considerable a los desarrolladores,



más aún, cuando aparecen en escena nuevas versiones y se consideran proyectos de migración o *evolución tecnológica* de los sistemas. En estos casos se debe mantener el mismo *modelo conceptual del negocio o del dominio del problema*, independientemente de la plataforma.

Teniendo en cuenta el *nivel de abstracción*, las cinco categorías anteriores podrían reagruparse en *lenguajes de bajo nivel*, *nivel intermedio* y *lenguajes de alto nivel*. Un aumento del nivel de abstracción genera mejoras en la productividad y reduce el rendimiento de ejecución de las aplicaciones. [Selic 2008] propone dos *tipos de complejidad* en el desarrollo de *software*: a) *la complejidad esencial*, que no se puede evitar y es inherente al dominio del problema, y b) *la complejidad arbitraria*, que está asociada a los métodos y tecnología empleada. En esta sección se expondrán los dos enfoques: *MDE* y *BPM*, que intentan reducir la *complejidad arbitraria*.

### 3.1.2 Model-Driven Engineering (MDE)

En [Seidewitz 2003] se ofrecen definiciones de los conceptos de *Modelo* (3.1.) y *Meta-Modelo* (3.2):

«Modelo es un conjunto de declaraciones sobre un sistema en estudio»

Definición. 3.1. Modelo

En este sentido, declaración significa algún tipo de expresión que puede ser considerada como verdadera o falsa en el contexto del sistema en estudio. Un *modelo* es una *abstracción del mundo real*, que normalmente es un tipo de descripción; su significado tiene dos aspectos: i) la relación del modelo con lo que se está especificando y ii) con otros modelos derivables de él; considerar cuidadosamente ambos aspectos puede ayudar a: i) entender cómo utilizar modelos para razonar acerca de los sistemas que construimos y ii) cómo utilizar *Meta-Modelos* para diseñar lenguajes que puedan expresar modelos.

Una interpretación de un modelo establece una correspondencia entre los elementos del modelo y los elementos del sistema en estudio, de tal manera que se puede determinar el valor de verdad de las declaraciones con cierto nivel de precisión.

Un *Meta-Modelo* contiene declaraciones sobre cómo pueden expresarse modelos válidos en el lenguaje de especificación de los modelos. Existen conceptos relacionados en otros campos, entre otros: *Meta-lenguaje*<sup>34</sup> [Jakobson 1980] para hablar sobre reglas en los lenguajes, *Meta-matemáticas*<sup>35</sup> [Hunter 1971] y *Meta-teoría* [Ritzer 1991; Wallis

<sup>34</sup> Bertrand Russell (1872–1970) establece la teoría de los niveles de los lenguajes, y Alfred Tarski (1902–1983) diferencia los conceptos *lenguaje objeto* y *Meta-lenguaje*.

<sup>35</sup> En el campo de la lógica matemática moderna existen contribuciones de autores como Gottfried Leibniz (1646–1716), Gottlob Frege (1848–1925), David Hilberth (1862–1943), Bertrand Russell y Kurt Gödel (1906–1978), que han

2010] para analizar teorías matemáticas o cualquier tipo de teoría:

«*Meta-Modelo es una especificación para una clase de sistemas en estudio, donde cada elemento de la clase es un modelo válido expresado en un cierto lenguaje*»

Definición. 3.2. Meta-Modelo

- i) Un *Meta-lenguaje* es un lenguaje usado para describir un *lenguaje objeto*; p. ej. un tratado sobre *gramática en castellano* es un *Meta-Lenguaje* y también la notación «*Backus Normal Form or Backus-Naur Form (BNF)*» para describir la sintaxis de los lenguajes de programación [Chomsky 1956].
- ii) La *aritmética, geometría euclídea, la estática, cinemática o dinámica* son ejemplos de teorías que tienen por objeto el estudio de propiedades de los objetos que pretenden describir: i) números, operaciones en la aritmética; puntos, rectas, planos en la *geometría*; ii) el espacio, tiempo, gravedad, movimiento, masa, la fuerza, etc. en la *física newtoniana*. Una *Meta-Teoría* tiene como objeto el estudio de las propiedades de la *teoría* en cuestión. Así, por ejemplo, la *Meta-aritmética* no versará sobre los números, sino sobre las propiedades de la *teoría de números*. Los teoremas de la *Meta-Aritmética* serán por tanto *Meta-Teoremas o proposiciones* relativas a la *consistencia, axiomatizabilidad, decidibilidad, existencia de modelos, etc. de la teoría de números*. [Wallis 2010] es un estudio comparativo de *Meta-teorías* en distintos campos de conocimiento.

En el campo del software, en [Selic 2003] se proponen las siguientes características para un *modelo del software*:

- i) **Bajo coste.** Su coste de elaboración debe ser muy inferior al del desarrollo del sistema real.
- ii) **Precisión.** Para resultar útiles, deben captar una representación lo más completa posible del sistema.
- iii) **Comprensión.** Deben ser fácilmente comprensibles por las personas involucradas en su construcción y aceptación.

En [Kurtev et al. 2002] se define el concepto de «espacio tecnológico» como «Forma de definir tecnologías a un alto nivel de abstracción»; un espacio tecnológico es un contexto de trabajo que tiene asociado un conjunto de conceptos, métodos, técnicas y herramientas. Algunos de estos espacios tecnológicos son: i) Las gramáticas de los lenguajes de programación (*Grammarware*), ii) Bases de datos, iii) Ontologías y iv) Los Modelos (*Modelware*).

El enfoque «*Model-Driven Software Engineering*» o «*Model-Driven Engineering (MDE)*» [Kent 2002; Schmidt 2006] se ha establecido como un enfoque ampliamente

---

desarrollado distintas *Meta-Teorías* para discutir, entre otros aspectos, hipótesis, axiomas, métodos, corrección, completitud, etc.

aceptado para el desarrollo de software. *MDE* está relacionado con el término Modelware de [Kurtev et al. 2002]. El principio básico de *MDE* es «Everything is a model» [Bézivin & Kurtev 2005] y el aspecto clave es el manejo de diversos *modelos* situados en distintos niveles de abstracción donde es posible automatizar ciertas transformaciones entre ellos. *MDE* contribuye a reducir la *complejidad arbitraria*, y el objetivo básico es la utilización de *modelos* en el *proceso de desarrollo de software* [Seidewitz 2003].

En [Völter et al. 2006] se recogen los conceptos esenciales de *MDE* mediante un diagrama de clases (Figura 3.3). Se comentan a continuación estos conceptos:

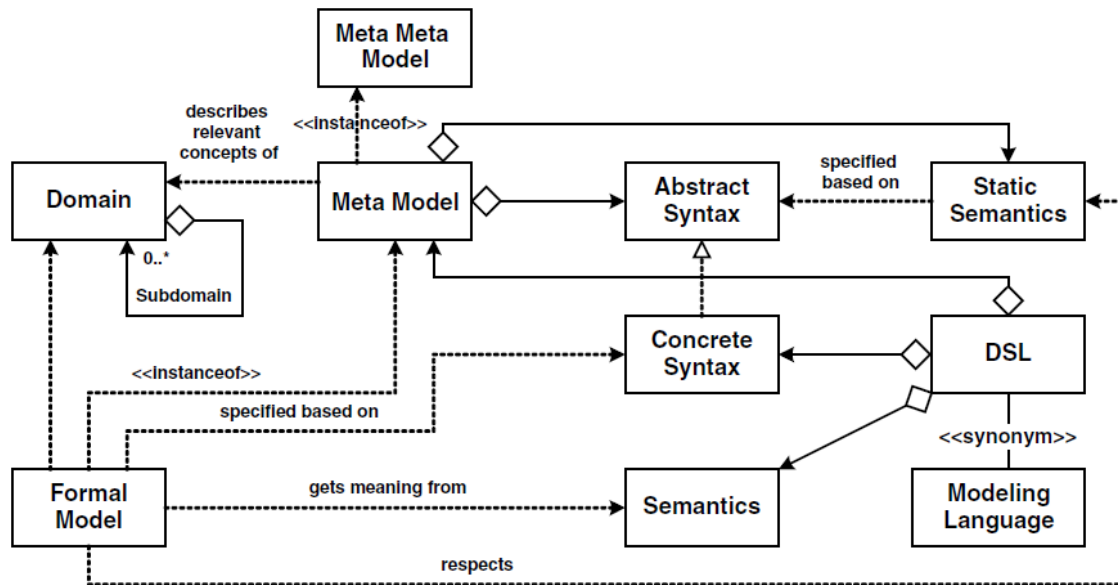


Figura. 3.3. Conceptos fundamentales MDE

- i) El *dominio* delimita el campo de conocimiento. Se distinguen: a) *dominios profesionales* que hacen referencia al carácter conceptual del problema, y b) *dominios tecnológicos*, en relación a la tecnología del software.
- ii) El *Meta-Modelo* describe formalmente los conceptos importantes del *dominio del problema*. Un *Meta-Modelo* recoge la estructura y restricciones que permiten definir cualquier modelo de ese dominio. Es un concepto esencial en *MDE*. La *reutilización*, *interoperabilidad* y *portabilidad* de los *Meta-Modelos* dependen de la existencia de otro *Meta-Modelo* en un nivel superior de abstracción: el *Meta-Meta-Modelo*. Este será el encargado de describir de forma única los conceptos que sirven para representar *Meta-Modelos* de cualquier dominio.
- iii) En un *Meta-Modelo* existe una *sintaxis abstracta* y una *sintaxis concreta*; la *sintaxis abstracta* captura la semántica de los *artefactos* a nivel conceptual y la *sintaxis concreta* se enfoca a la representación de dichos conceptos. Así, con la misma *sintaxis abstracta* de un *Meta-Modelo* podrían coexistir varias *sintaxis concretas* para representar los mismos conceptos. Trabajar con *Meta-Modelos*

supone situarse a mayor nivel de abstracción que con *modelos* y que utilizar una *notación o lenguaje concreto* (Figura 3.4) [Alanen et al. 2003].

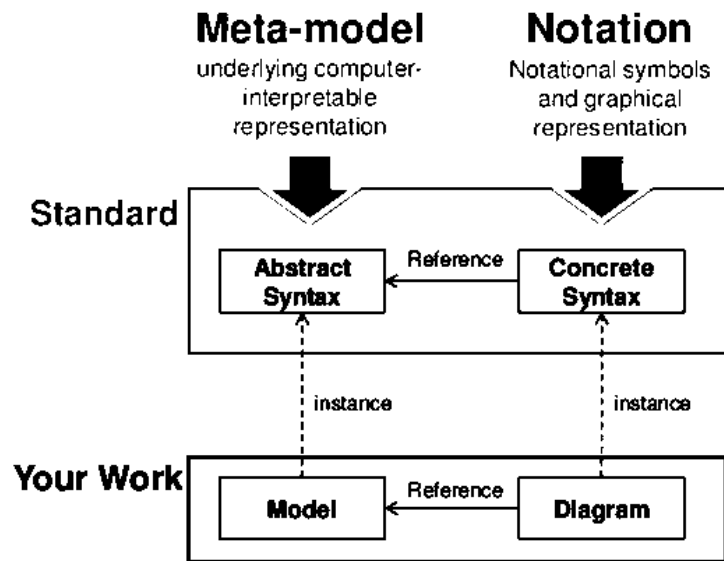


Figura. 3.4. Meta-Modelo: sintaxis abstracta vs sintaxis concreta

- iv) La *semántica abstracta* del *Meta-Modelo* está basada en la *sintaxis abstracta* de éste y comprende los aspectos de verificación de modelos a nivel semántico [Giachetti et al. 2008].
- v) Los «*Lenguajes específicos de dominio o Domain Specific Languages (DSL)*» [van Deursen 1997, 2000] permiten expresar conceptos de un dominio específico. Están formados por *Meta-Modelos*, una o varias *sintaxis concretas* y una herramienta para soportar su uso.
- vi) Un modelo formal sustenta la transformación de artefactos entre distintos niveles de abstracción; son instancias de los *Meta-Modelos*, se representan con una sintaxis concreta y deben respetar la semántica del *Meta-Modelo*. Los conceptos del lenguaje se mapean sobre los conceptos del dominio que se está modelando, eliminando interpretaciones equivocadas.

Según [Kleppe et al. 2003] las principales diferencias del enfoque *MDE* frente al enfoque tradicional son: a) la *productividad*, b) la *portabilidad*, c) la *interoperabilidad* y d) el *mantenimiento y documentación* del sistema.

- i) **Productividad.** Al existir distintos niveles de abstracción y artefactos distribuidos en cada nivel, la modificación de un artefacto concreto implica modificar todos los artefactos correlacionados en otros niveles. *MDE* provee mecanismos para que cualquier cambio que se realice en un artefacto tenga un efecto inmediato y directo en los modelos interrelacionados, ya que los artefactos dependientes se pueden generar automáticamente.

- ii) **Portabilidad.** Continuamente aparecen nuevas tecnologías o versiones de productos que obligan a realizar proyectos de migración de los sistemas, provocando costes económicos y contingencias que hay que controlar en las organizaciones. En *MDE* se contemplan modelos independientes de la plataforma por lo que las modificaciones se pueden concentrar en los modelos específicos de la plataforma.
- iii) **Interoperabilidad.** Los sistemas informáticos se integran de diversas maneras entre ellos. Existen multitud de productos, plataformas, versiones, etc. Conseguir un buen grado de interoperabilidad no es sencillo. *MDE* provee facilidades para crear puentes de comunicación (*bridges*) para ayudar en este sentido.
- iv) **Mantenimiento y documentación.** Durante el proceso de desarrollo se pone más interés en los requisitos, modelación, la codificación y pruebas que en la documentación generada por el coste que supone mantener manualmente cualquier cambio en todos los niveles del sistema, sin embargo esta es fundamental cuando el sistema entra en mantenimiento. A través del uso de *MDE* es posible generar todo el sistema a partir de los modelos, por lo que la documentación de alto nivel estaría contenida en los modelos de mayor nivel de abstracción, reduciendo la documentación que se genera manualmente en los niveles inferiores.

El desarrollo de software bajo el paradigma *MDE* propone crear modelos con alto nivel de abstracción que representan los aspectos relevantes del sistema en las primeras etapas del desarrollo, sin incluir aspectos tecnológicos y, a medida que el proceso avanza, generar nuevos modelos con nivel mayor de detalle y cada vez más cercanos a la implementación final del sistema. Surgen dos necesidades principales [*Fondemenet & Silaghi 2004*]:

- i) Disponer de un conjunto de elementos comunes para que todos los modelos se desarrollen de la misma forma, utilizando para ello el concepto de *Meta-Modelo*.
- ii) Definir mecanismos que hagan posible la obtención de unos modelos a partir de otros de manera sistemática y con posibilidad de automatizar total o parcialmente el proceso.

Una *transformación* entre dos *modelos* representa una relación entre dos *sintaxis abstractas* o *Meta-Modelos* y se define mediante un conjunto de relaciones entre los elementos del correspondiente *Meta-Modelo* [*Thiry & Thirion 2009*], de tal manera que cuando es ejecutada proporciona un *modelo destino* conforme al *Meta-Modelo destino* partiendo de un *modelo origen* conforme a su *Meta-Modelo*.

«*Model-driven Software Development (MDSD)*» [*Völter et al. 2006; Liddle 2011; García et al. 2013*], también denominado «*Model-driven Development (MDD)*» [*Mellor*

*et al. 2003; Selic 2003; Elvesæter et al. 2006*], es un enfoque *MDE* para abordar el ciclo de vida del software, donde los artefactos software son generados en sucesivas transformaciones de *modelos*. Un *modelo* es una descripción de estos artefactos desde una perspectiva determinada a un nivel de abstracción, de modo que las características de interés pueden percibirse claramente. *MDD* va mejorando su nivel de aceptación apoyándose en la facilidad de automatización que puede ofrecer en el proceso de desarrollo de software [*Deltombe et al. 2012*]. Estas técnicas de transformación de modelos, además de utilizarlas en el proceso directo de desarrollo de software, también pueden emplearse en procesos de *ingeniería inversa* [*Chikofsky & Cross 1990; Favre 2004, 2010; Bruneliere et al. 2010a*]. En [*Elvesæter et al. 2006, 2010*] se trata en profundidad los aspectos sobre *interoperabilidad* en *MDD*. Existen diversas iniciativas *MDD*, entre otras: «*Software Factory*») [*Greenfield et al. 2004*], «*Model-integrated Computing*») [*Sprinkle 2004*], «*Domain Specific Languages (DSL)*») [*van Deursen 1997, 2000*] y «*OMG Model Driven Architecture (MDA)*») [*OMG:MDA; Kleppe et al. 2003*], que se analiza a continuación.

El mejor y más conocido exponente *MDE* es el estándar *MDA*. El objetivo fundamental de *MDA* es la *interoperabilidad*. *MDA* se sustenta [*Völter et al. 2006*] sobre otros estándares del grupo: i) «*Unified Modeling Language (UML)*») [*OMG:UML 2011*], ii) «*Meta-Object Facility (MOF)*») [*OMG:MOF 2011*], iii) «*OMG Common Warehouse Metamodel (CWM)*») [*OMG:CWM 2003; Bruneliere et al. 2010a*], iv) «*XML Metadata Interchange (XMI)*») [*OMG:XMI 2013*] y v) «*Query/View/Transformation (QVT)*») [*OMG:QVT 2011*].

- i) *UML* es un lenguaje de modelado gráfico que permite modelar el ciclo de vida de una aplicación. Algunos de sus artefactos son: clases, interfaces, casos de uso, diagramas de actividad, etc. que se pueden combinar en distintos tipos de diagramas. Los artefactos y diagramas *UML* pueden ser intercambiados entre herramientas del ciclo de vida utilizando *XMI* [*Ruiz et al. 2003*].
- ii) *MOF* es un ejemplo de un *Meta-Meta-Modelo*. Define los elementos esenciales que se utilizan para construir *modelos de sistemas*, proporcionando un modelo común para los *Meta-Modelos* de *CWM* y *UML*. El mayor valor de *MOF* es que permite *interoperar* entre *Meta-Modelos* sobre diversos dominios. Una implementación *MOF* debe proporcionar herramientas para la autorización y la publicación de metadatos contra un repositorio *MOF*.
- iii) *CWM* permite modelar, manipular e intercambiar respectivamente almacenes de datos en un formato estándar. Usa *UML*, *MOF* y *XMI* [*Ruiz et al. 2003*] para intercambiar metadatos.
- iv) *XMI* es un estándar que permite expresar en *XML* cualquier modelo o *Meta-Modelo* definido en *MOF*; mediante *serialización o disposición en forma de flujo de datos (o meta-datos)* ofrece un formato adecuado para intercambiar

entre diferentes herramientas.

- v) *QVT* define el modo en que se llevan a cabo transformaciones entre modelos definidos bajo *MOF*. Consta de tres partes: a) Un lenguaje para crear vistas de un modelo; b) Un lenguaje para realizar consultas sobre modelos y c) Un lenguaje para escribir definiciones de transformaciones. *QVT* soporta las transformaciones «*Model-to-Model (M2M)*», aunque también existe otro tipo de transformaciones en *MDA* como «*MOF Model to Text Transformation Language (MOFM2T)*» [OMG:MOFM2T 2008] para la realización de transformaciones «*Modelo-to-Text (M2T)*».

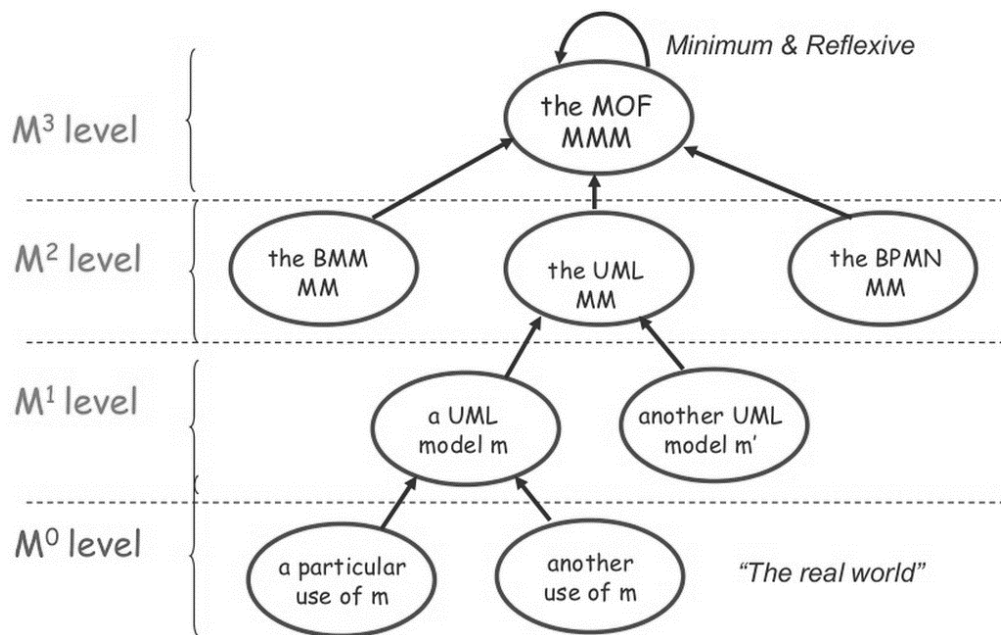


Figura. 3.5. MOF y niveles de abstracción

En la Figura 3.5 se ilustra el uso de la facilidad *MOF* para manejar niveles de abstracción *MDE*; se contemplan cuatro niveles:

- i) **Nivel M0.** Corresponde al mundo real. En él se pueden observar estados o casos concretos que obedecen a un modelo del nivel M1.
- ii) **Nivel M1.** En este nivel se sitúan los modelos con una sintaxis concreta, donde los casos del nivel inferior respetan las reglas del modelo.
- iii) **Nivel M2.** Aquí, un *Meta-Modelo* tiene estructuras que permiten definir cualquier modelo que cumpla unas características semánticas.
- iv) **Nivel M3.** El último nivel, el *Meta-Meta-Modelo* facilita estructuras semánticas mínimas para definir cualquier *Meta-Modelo*. *MOF* es este *Meta-Meta-Modelo* que tiene capacidad reflexiva para definirse a sí mismo como *Meta-Modelo*.

MDA también observa distintos niveles de abstracción para representar *modelos* y *Meta-Modelos* [van der Straeten et al. 2009; Brambilla et al. 2012] que se asocian a plataformas: i) «*Platform Specific Meta-Model (PSM)*», ii) «*Platform Independent Meta-Model (PIM)*» y iii) «*Computer Independent Meta-Model CIM*» y, así, facilitar un alto nivel de *interoperabilidad*, basada en el concepto *MOF*.

### 3.2 Gestión de procesos del software

En [Fuggetta 2000] (Definición 1.5.) se define el concepto de «*Proceso software*». Existen distintos enfoques en materia de propuestas para el apoyo al proceso de software:

- i) Las primeras aportaciones, aparecidas en los 90's:
  - a) Basadas en reglas como «*Lenguajes de modelación de procesos software (SPMLs)*» [Kaiser et al. 1990].
  - b) Las Redes de Petri [Bandinelli et al. 1993].
  - c) Y otras, basadas en lenguajes de programación [Conradi et al. 1992; Sutton et al. 1995].

Todos estos lenguajes están enfocados a la ejecutabilidad, de ahí, su formalidad, poca flexibilidad y complejidad, por lo que no han propiciado su adopción en la industria del software.

- ii) Como alternativa han emergido otras propuestas basadas en buenas prácticas y estándares:
  - a) El estándar para la modelación de procesos *BPMN 2.0* [OMG:BPMN 2013; Iso/Iec:19510 2013] debido a su simplicidad y popularidad, así como la posibilidad de ejecutar los procesos modelados [Bonnet et al. 2014].
  - b) Existen enfoques de diversos autores basados en *UML 2.0* [OMG:UML 2011] y en la propuesta «*SPEM, Software & Systems Process Engineering Metamodel specification 2.0*» [Bendraou et al. 2007; OMG: SPEM 2008] que provee un lenguaje para la especificación de metodologías de desarrollo de software
  - c) Así como la propuesta del estándar «*Software Engineering Meta-Model for Development Methodologies*» [Iso/Iec:24744 2007, 2014].

### 3.3 Gestión de procesos de negocio (BPM)

#### 3.3.1 Conceptos

En [van der Aalst 2004; Netjes et al. 2006] se define el concepto «*Business Process Management (BPM)*» (Definición 1.3) y [Weske 2012] define el concepto de «*Proceso de negocio*» (Definición 1.1):

En «*Process Mining Manifesto*» [van der Aalst et al. 2012] se definen las distintas perspectivas que caracterizan a la gestión de procesos de negocio: a) *Perspectiva de control*, b) *Perspectiva organizacional o de los recursos*, c) *Perspectiva del tiempo* y d)



*Perspectiva de casos* [van der Aalst et al. 2005, 2007]; la *perspectiva de control del flujo lógico* es la más típica pero no la única, pues también tienen relevancia los aspectos ligados a la estructura de la organización, los recursos, roles, y los flujos de información, así como los aspectos ligados al tiempo (tipos de reglas de negocio temporales) y la especificación de reglas de negocio que pueden ligarse a casos concretos o agrupaciones de estos. En [Reichert & Weber 2012] se consideran las siguientes perspectivas de un sistema basado en procesos: i) *Operacional o de servicios de las aplicaciones*; ii) *Organizacional, para definir unidades de negocio, actores y sus roles*; iii) *Temporal para definir reglas de tiempo*; iv) *Funcional, que contempla actividades elementales o agrupaciones de estas en unidades mayores como procesos y subprocesos que tienen capacidad de realizar una acción*; v) *Comportamiento, asociada a la dinámica del proceso o descripción de flujo de control*; y vi) *Información, asociada a los objetos de datos y flujos de información*.

[Reichert & Weber 2012] definen los conceptos de *instancia de un proceso* y *modelo de un proceso* (Definición 3.3) como una *clase o tipo de instancias de procesos* cuyas *trazas de ejecución* (Definición 3.4) quedan reflejadas en un registro de la actividad o *log de eventos* del sistema.

Sea  $S$  un modelo de proceso, entonces una instancia  $I=(S, \sigma_I)$  queda definida por  $S$  y su traza de ejecución  $\sigma_I$  de  $S$  en el log de eventos. Cada traza  $\sigma_I$  en el log de eventos está relacionada con la ejecución de actividades (inicio y terminación de cada actividad) o la evaluación de condiciones de transición de  $S$  relativas a una instancia.

Definición. 3.3. Instancia de un proceso

Sea  $S$  un modelo de proceso e  $I_S$  el conjunto de todas las instancias de procesos que ejecutan  $S$ , entonces:

La traza de ejecución  $\sigma_I$  de la instancia de proceso  $I \in I_S$  viene dada por:

$I = \langle e_1, e_i, \dots, e_n \rangle$  donde el orden de  $e_i$  en  $I$  refleja el orden temporal en que los eventos relativos a la ejecución de actividades o evaluación de condiciones de transición ocurren en  $S$ .

El conjunto de todas las trazas que capturan todas las instancias  $\{ \forall I \in I_S \}$  se denota como el log de ejecución.

Definición. 3.4. Traza de ejecución de un proceso

### 3.3.2 Ciclo de vida BPM

En la Figura 3.6. se presenta el Ciclo *BPM de mejora continua de van der Aalst* [van der Aalst 2004], similar a otros propuestos, p. ej. en [Hill et al. 2006] y [Dumas et al. 2013], donde se contemplan diversas fases en el ciclo de vida de implantación de una solución *BPM*: i) Diseño, ii) Configuración, iii) Ejecución y iv) Diagnóstico.

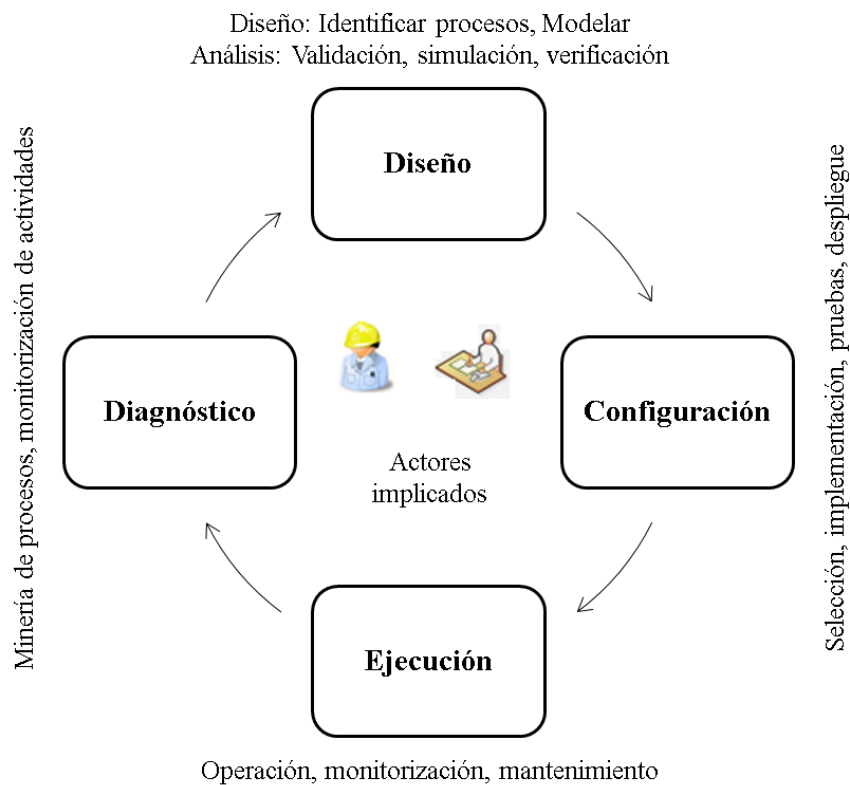


Figura. 3.6. Ciclo de mejora continuo BPM

- i) En la fase de *Diseño* [Ko 2009] se identifican los procesos, se modelan, analizan, simulan y verifican por expertos en el dominio del problema, analistas de negocio y usuarios. Es necesario contar con lenguajes y entornos para la definición formal, simulación y verificación de estos procesos.
- ii) Las otras fases requieren herramientas para desplegar los procesos y conectarlos con el nivel de aplicación (*Configuración*) y ejecutar estos procesos en un entorno automatizado, dejando un registro de la actividad (*log de actividades*) que permitirá monitorizar y diagnosticar la ejecución de estos procesos e iniciar una mejora mediante *redefinición* o *reingeniería de procesos* [Hammer & Champy 1993; Mohapatra 2012].

La *reingeniería de procesos*, como etapa dentro de la fase de *Diagnóstico del ciclo de mejora continua*, se ha desarrollado en torno a los sistemas automatizados que gestionan procesos (*PAIS*) (ver sección 3.3.5), apareciendo el campo de «*process mining*» [van der Aalst et al. 2007, 2012; van der Aalst 2011, 2013, 2014, 2015; Reichert & Weber 2012]. Se distinguen tres tipos de actividades en relación al «*process mining*» (Figura 3.7) [van der Aalst et al. 2011]: i) *Descubrimiento de procesos (discovery)*, mediante algoritmos, se pueden inferir modelos de procesos analizando un conjunto de trazas del log de eventos; ii) *Conformidad (conformance)*, se compara el modelo de procesos con las trazas existentes para detectar incumplimientos o desviaciones en la ejecución de los

modelos de procesos; y iii) *Mejora (enhancement) del proceso*, que compara el modelo con las trazas reales para proponer mejoras a los modelos. Existen herramientas automatizadas, como «*ProM*» [Verbeek et al. 2011; Kalenkova et al. 2014] para llevar a cabo este proceso a partir del *log de eventos* [Reichert & Weber 2012] de los sistemas que mantienen este artefacto como reflejo de la actividad de ejecución de los procesos que acontecen en el sistema.

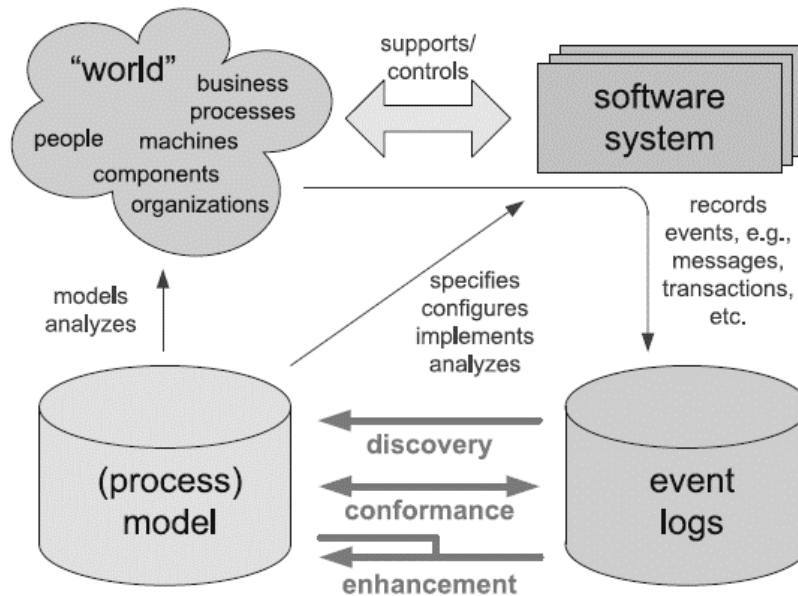


Figura. 3.7. Minería de procesos (process mining)

### 3.3.3 Beneficios del enfoque BPM

El enfoque *BPM* también contribuye a reducir la *complejidad arbitraria* [Selic 2008], ya que evita que todo lo asociado con la definición, configuración, ejecución y diagnóstico de un proceso dependa del código de las aplicaciones informáticas. Entre el nivel del proceso y el nivel de aplicación existirá solo el puente necesario.

*BPM* aporta beneficios [Havey 2005; Trkman 201; Mohapatra 20120; von Rosing et al. 2014] a las organizaciones que utilizan este enfoque:

- i) **Formalización de procesos.** En muchos escenarios los procesos sólo los conocen sus responsables, dificultando su seguimiento por otros actores. *BPM* obliga a la modelación mediante un lenguaje formal, así se facilita la comunicación y la posibilidad de modificar el proceso.
- ii) **Automatización de procesos.** Cada proceso se compone de actividades pertenecientes a un proceso. La duración de las actividades y el flujo de control condicionan la duración del proceso global. La reducción de tiempos muertos entre actividades optimiza la duración del proceso.
- iii) **Incremento de la productividad.** En escenarios reales *BPM* reduce el tiempo de ejecución y disminuye la necesidad de intervención manual, mejorando así la

productividad y calidad de los procesos.

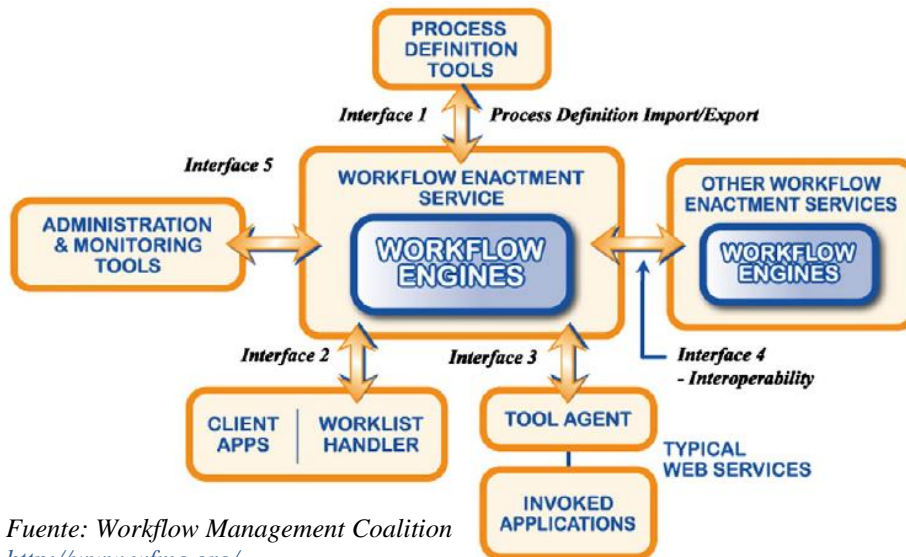
En [Malinova & Mendling 2013] se realiza un trabajo de campo, entrevistando a expertos *BPM* de varias compañías. El estudio concluye que las razones para adoptar *BPM* son tres principalmente: i) comprender y asimilar el conocimiento intrínseco de los procesos de negocio («*Know-how*» de la organización); ii) conocer el desempeño de sus trabajadores en la ejecución de los procesos; iii) controlar y medir los procesos.

Las organizaciones que no contemplan el enfoque *BPM* están desaprovechando una ventaja competitiva [Havey 2005; Trkman 201; Mohapatra 20120; von Rosing et al. 2014] esencial. Las «*TI*» no son una excepción, por lo que este enfoque es cada vez más común en este negocio. Por otro lado el enfoque *BPM* se va proponiendo por distintos organismos como buenas prácticas o estándares: «*Project Management Institute (PMI)*» con «*Project Management Body of Knowledge (PMBOK)*» [PMI:PMBOK 2013]; «*Stationery Office Books (SOB)*» con «*Projects IN Controlled Environments 2 (PRINCE2)*» [SOB 2009]; o estándares como «*Capability Maturity Model Integration (CMMI)*» [Chrissis et al. 2011], en la que define modelos de madurez para la mejora y evaluación de procesos o «*Information Technology Infrastructure Library (ITIL)*» [ITIL 2015; Iso/Iec 20000:2012; Clifford & van Bon 2008]; y la organización *ISO* con algunas de sus normas, como por ejemplo la norma «*Quality management systems, Requirements*» [Iso/Iec 9001:2008] y el estándar *BPMN* [OMG:BPMN 2013; Iso/Iec:19510 2013].

Ahora bien, conseguir todos los beneficios de implantación de *BPM* en una «*TI*» no es tarea fácil por las características intrínsecas de esta tecnología y por la complejidad e impredecibilidad de los procesos [Ruiz-González & Canfora 2004] en el sector del software.

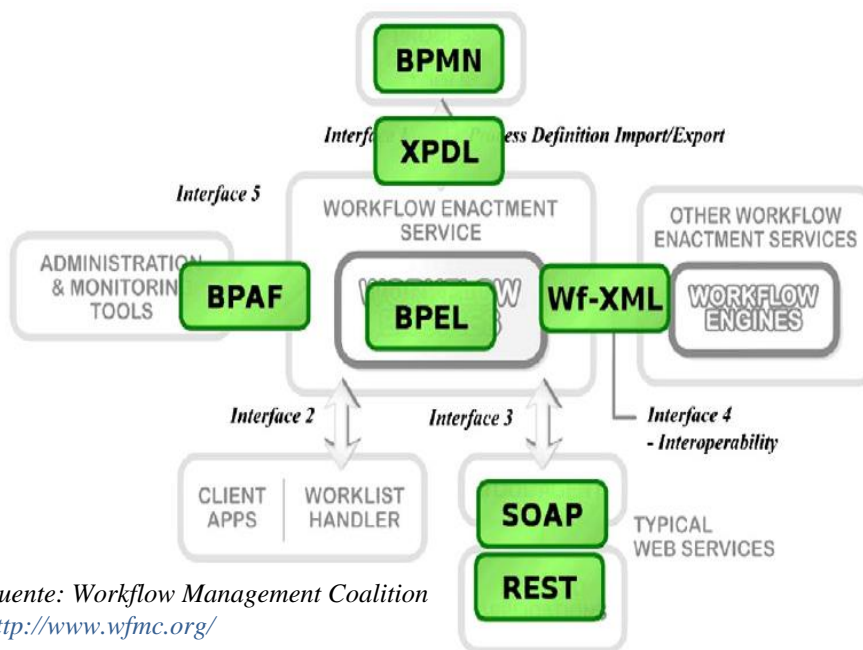
### 3.3.4 Estándares *BPM*

El organismo «*Workflow Management Coalition (WfMC)*» define una arquitectura de referencia para los sistemas de gestión de procesos de negocio (Figura 3.8). Diferencia distintos componentes para la gestión del ciclo de vida de los procesos [van der Aalst 2004]: i) *Herramientas para la definición de procesos*, ii) *Herramientas para la activación y control de actividades de los procesos en ejecución*, iii) *Herramientas para la integración con la capa de aplicaciones y con otros sistemas de ejecución de procesos* y iv) *Herramientas para la administración y monitorización de los procesos*. Entre los distintos componentes se definen interfaces que permiten la portabilidad de artefactos ligados a los procesos que existen en cada nivel.



Fuente: Workflow Management Coalition  
<http://www.wfmc.org/>

Figura. 3.8. Arquitectura de referencia para la gestión de procesos



Fuente: Workflow Management Coalition  
<http://www.wfmc.org/>

Figura. 3.9. Principales estándares para la gestión de procesos

La Figura 3.9 muestra el mapa de estándares para los componentes enumerados. Entre ellos, por el enfoque de la tesis a la especificación de procesos, cabe destacar «*Business Process Model and Notation (BPMN)*» [OMG:BPMN 2013] para la definición de procesos de negocio, «*XML Process Definition Language (XPDL)*» [Shapiro 2010; WfMC:XPDL 2012] para el intercambio de metadatos (esquemas e instancias de procesos) y *BPEL* («*Web Services Business Process Execution Language (WS-BPEL)*» [OASIS:BPEL 2007], («*WS-BPEL Extension for People BPEL4People*»

[*OASIS:BPML4People 2010*]) como estándar para la ejecución de procesos.

[*Ko et al. 2009*] coinciden en la selección de estándares anteriores pero añaden alguno más (Figura 3.10). Así, para la especificación de procesos, considera también los «*Diagramas de actividad UML (UML AD)*» incluidos en el estándar [*OMG:UML 2011*]» como alternativa a la especificación gráfica, y «*Semantics of Business Vocabulary and Rules (SBVR)*» [*OMG:SBVR*]» como lenguaje natural declarativo de reglas de negocio. Aunque *UML AD* tienen mucha similitud con los diagramas *BPMN*, su capacidad expresiva es inferior a los diagramas *BPMN* [*Eloranta et al. 2006*].

En cuanto a los estándares de intercambio de metadatos de procesos, además de *XPDL*, se considera la propuesta «*Business Process definition Meta-Model (BPDM)*» [*Harmon 2004; OMG:BPDM 2008*]; ambas se analizan a continuación.

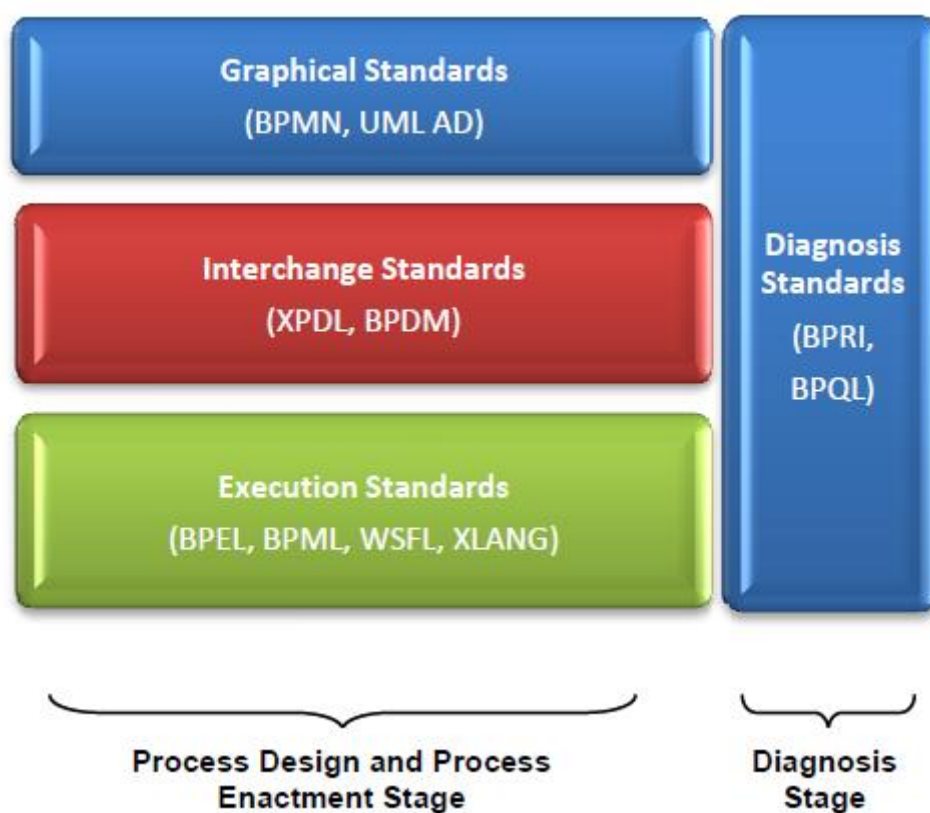


Figura. 3.10. Principales estándares para la gestión de procesos

El foco del trabajo de tesis se pone en la *especificación de procesos del software*. Por esta razón se describen a continuación los estándares *BPDM*, *BPMN* y *XPDL* que otorgan tanto facilidades para la *definición* como para la *portabilidad* o *interoperabilidad* de procesos.

#### 3.3.4.1 Business Process Definition Meta-Model (BPDM)

OMG propone *BPDM* [*Harmon 2004; OMG:BPDM 2008*] como primer *Meta-Modelo* del mercado para la definición y ejecución de procesos. Dentro de *MDA* es

una plataforma de nivel de abstracción *PIM* que puede involucrar personas o solo sistemas automatizados. El objetivo esencial es la *interoperabilidad de procesos* (Figura 3.11), ofreciendo capacidades para facilitar el intercambio de procesos entre distintos entornos. Los objetivos específicos buscados son:

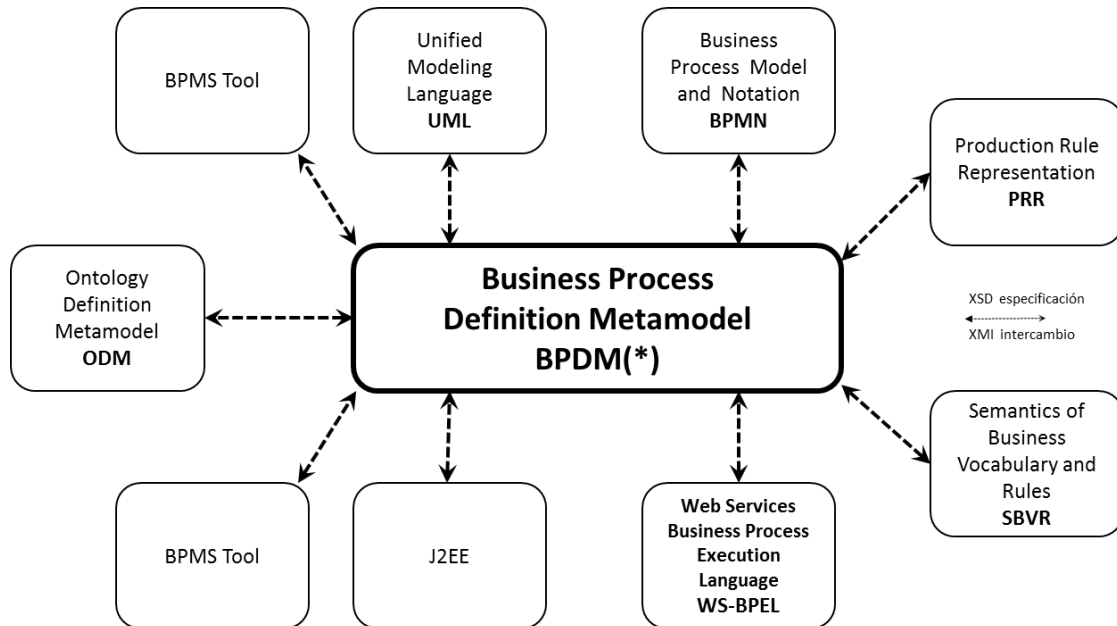


Figura. 3.11. BPDM

- i) Unificar la comunicación entre distintos lenguajes de definición de procesos de negocio, tanto de carácter gráfico como de carácter textual.
- ii) Definir un *Meta-Modelo* que complemente a *UML* para ganar consistencia y completitud en la definición de procesos.
- iii) Usar coreografías en la comunicación entre procesos, favoreciendo la colaboración a través de servicios web.

Desde la aparición de *BPMN 2.0 (2011)*, con un robusto *Meta-Modelo* y creciente popularidad, *BPDM* ha tenido baja aceptación, ya que pocos fabricantes de *BPMS* ofrecen facilidades de *interoperabilidad* de procesos bajo este estándar [Cabot 2010]. Se revisa en el apartado siguiente la contribución de *BPMN 2.0* a la definición, ejecución y portabilidad de procesos.

### 3.3.4.2 Business Process Model and Notation (BPMN)

Sobre las *notaciones gráficas* para la definición de procesos, en [Lu & Sadiq 2007] se concluye que: i) permiten representaciones más simples de la mayoría de los procesos y con mayor abstracción, lo que reduce la complejidad y facilita la verificación; ii) las *notaciones textuales*, aunque son más precisas y flexibles, requieren un conocimiento técnico que no se puede presuponer a usuarios expertos del dominio.

*BPMN [OMG:BPMN 2013]* es un estándar *OMG* con una *lenguaje gráfico* que tiene una elevada cantidad de artefactos para definir procesos de negocio. Los esenciales son:

procesos, actividades, compuertas, flujo de control, elementos de información y flujos de información. Existen muchos más que ofrecen características más avanzadas para modelar procesos con distintos tipos de diagramas: coreografías, colaboración y conversaciones. La versión *OMG BPMN 2.0.1* es también el estándar [*Iso/Iec:19510 2013*] y cubre los siguientes aspectos de los procesos [*White & Bock 2011*]:

- i) Una notación gráfica para la especificación de procesos, comprensible por usuarios, analistas y expertos en la gestión y monitorización de procesos de negocios.
- ii) Soporte de la notación con un *Meta-Modelo* interno capaz de soportar la semántica necesaria para interrelacionar y ejecutar procesos.
- iii) Formatos de intercambio estándar para la transferencia de procesos e interacción de modelos. «*BPMN Model Interchange Working Group (BPMN MIWG)*» se encarga de los aspectos de intercambio de modelos *BPMN*. Se ofrecen capacidades «*BPMN Diagram Intechange (BPMN DI)*» con ficheros *XML* para la serialización de procesos: a) Intercambio de esquemas (.xsd format) y b) Metadatos para el intercambio de instancias de procesos (.xmi format [*OMG:XMI 2013; Ruiz et al. 2003*]).

*BPMN 2.0* crea un puente entre la definición e implementación de procesos. Desde su aparición ha ido cobrando creciente popularidad en la comunidad científica [*Bonnet et al. 2014*] como tecnología preferida por la mayoría de expertos en la definición de modelos de procesos, porque ofrece un lenguaje gráfico, sencillo y como un soporte para llegar a especificaciones ejecutables de procesos. *BPMN* provee con mayor facilidad [*Lu & Sadiq 2007*] para la intercomunicación entre expertos del negocio (usuarios y consultores) y *expertos en TIC*.

*BPMN* es principalmente un «*lenguaje imperativo*»; esto significa que muestra exactamente cómo tiene que ejecutarse el proceso. Por el contrario, un «*lenguaje declarativo*» solo propone las características esenciales que restringen la ejecución de actividades dentro de un proceso de negocio. De este modo, los enfoques imperativos están más cercanos a la producción, mientras que los enfoques de especificación están más próximos a usuarios y a los expertos en negocio [*Fahland 2009a, 2009b; Pichler et al. 2011; Reijers et al. 2013*] facilitando la comprensión de la especificación y el mantenimiento de las reglas de negocio.

Por otro lado, existen autores que argumentan excesiva complejidad en *BPMN 2.0*, por el elevado número de artefactos y el coste de instrucción en su empleo a un usuario no experto [*Wahl & Sindre 2005; Recker 2010*]. Por otro lado en [*Zur Muehlen & Recker 2008, 2013; White 2008; Chinosi & Trombetta 2012*] se argumenta que, en muchos casos, sólo se emplea una cantidad reducida de artefactos para definir los procesos. [*Moreno-Montes de Oca et al. 2015*] es un estudio sistemático de la literatura cuya conclusión es mejorar la calidad de las distintas perspectivas de los procesos en los modelos reales; en



[Mendling et al. 2010] se proponen guías para plantear modelos con buena comprensibilidad, entre las que está la utilización de un conjunto reducido de símbolos.

### 3.3.4.3 Xml Process Definition Language (XPDL)

WfMC propone XPDL [Shapiro 2010; WfMC:XPDL 2012] (Figura 3.12) como un ecosistema para facilitar la *interoperabilidad* de procesos (aspectos de especificación y ejecución) entre distintas herramientas [Hornung et al. 2006]. XPDL ofrece un *Meta-Modelo* de procesos y un formato de intercambio XML (ficheros que capturan metadatos con XSD y datos).

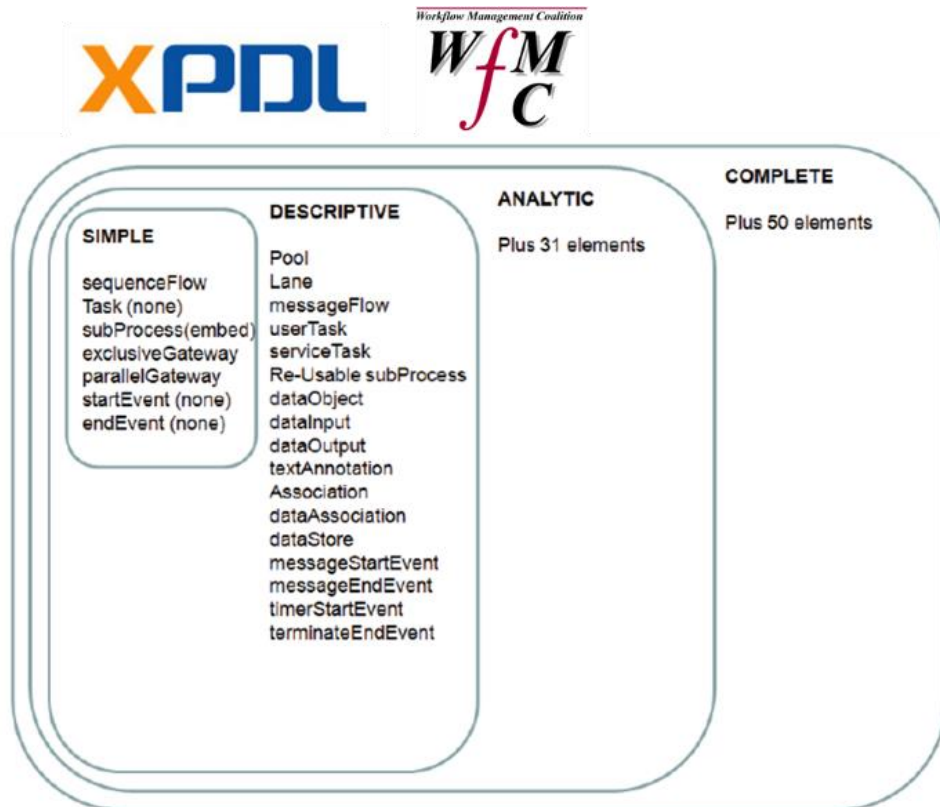


Figura. 3.12. Capas de XPDL 2.2

Desde la versión 2.2 (versión actual) se soportan gran parte de las características de *BPMN 2.0*, ofreciendo distintas capas de conformidad de la especificación: Simple, Descriptiva, Analítica y Total. Hasta la versión 3.0, en desarrollo, no se cubrirán todos los aspectos de *BPMN 2.0* [Shapiro 2010]. Generar formatos de intercambio XPDL da portabilidad a las herramientas que soporten esta facilidad, pero también garantiza la conformidad *BPMN 2.0* soportada por el estándar.

### 3.3.4.4 Evolución histórica

La Figura 3.13 muestra la evolución de los principales estándares para la definición y ejecución de procesos [Chinosi & Trombetta 2012] que destaca a los distintos organismos del escenario de estándares de procesos: a) *OMG* en las propuestas para la

especificación gráfica con *BPMN*, que también ha pasado a soportar la *interoperabilidad* en detracción de *BPDM*, b) *WfMC* con *XPDL* para *interoperabilidad* de procesos y c) *OASIS* con *BPEL* y *BPEL4People* lidera el campo de estándares para la ejecución de procesos. Estos estándares son generales, aunque también aplicables a los procesos del software.

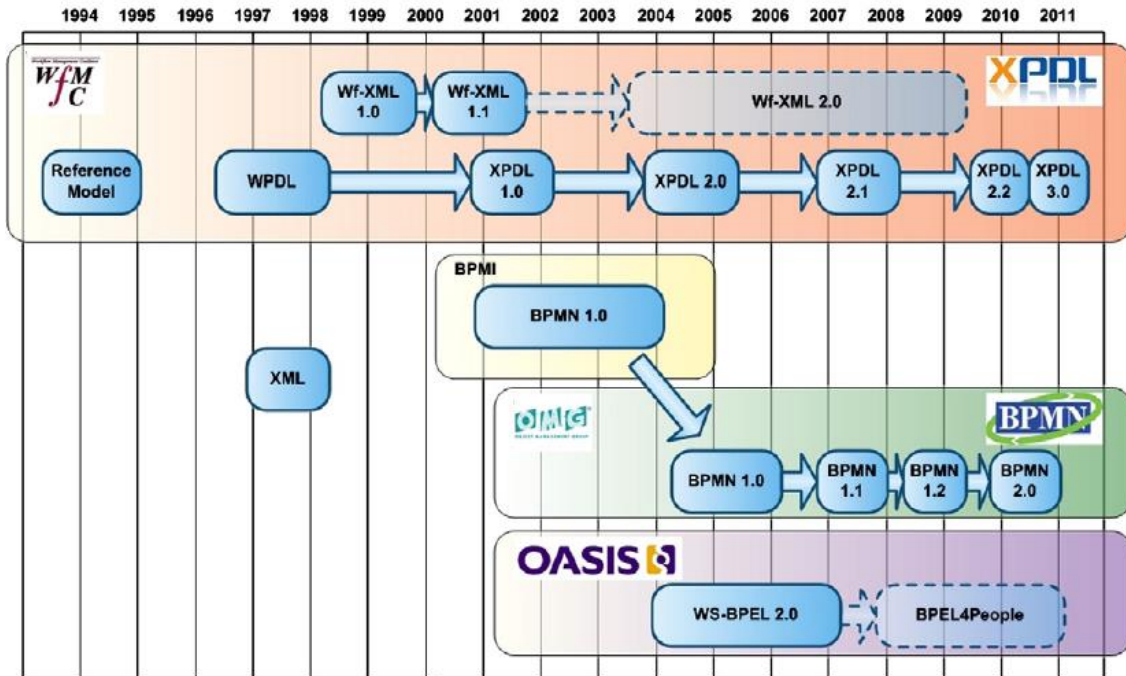


Figura. 3.13. Evolución histórica de estándares BPM

### 3.3.5 Sistemas de gestión de procesos (PAIS)

En [Eckerson 1995] se define la arquitectura de tres capas de un sistema informático, estructurando su lógica en: i) *Capa de presentación*, ii) *Capa de aplicación* y iii) *Capa de persistencia*. Con esta arquitectura se pretende separar la lógica del sistema en tres niveles, con objeto de alcanzar buenos niveles de escalabilidad, eficiencia y rendimiento, a la vez que independizando los artefactos del software para reducir errores y mejorar la mantenibilidad del sistema. No obstante, en esta arquitectura no se propone nada respecto a la gestión de flujos de trabajo o procesos.

En [Dumas et al. 2005] se introduce el concepto de «*Process aware information system (PAIS)*» (Definición 3.5) como evolución de los sistemas anteriores (en tres capas) a sistemas orientados a procesos.

«Un sistema software que gestiona y ejecuta los procesos operativos que involucran a personas, aplicaciones, y/o fuentes de información sobre la base de modelos de procesos»

Definición. 3.5. Process Aware Information System (PAIS)

[Dumas et al. 2005] propone la arquitectura de cuatro capas (Figura 3.14), donde se introduce la capa de procesos respecto a la arquitectura de tres capas. Esta nueva capa de procesos debe otorgar cierto nivel de independencia de los procesos respecto a las capas de presentación, aplicación y persistencia, aunque debe estar totalmente integrada y ofrecer un alto nivel de *interoperabilidad* [Milanovic 2009; Elvesæter et al. 2006, 2010] con otros sistemas de información.

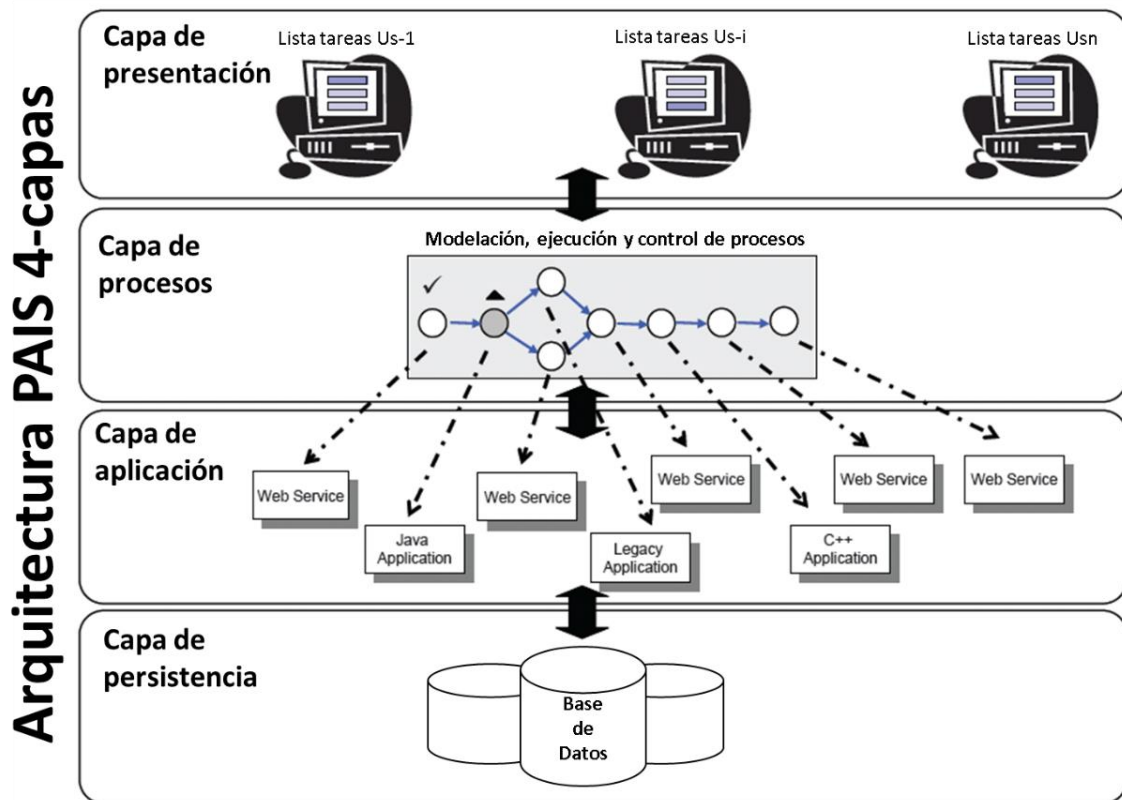


Figura. 3.14. Arquitectura PAIS de cuatro capas

En la capa de aplicación pueden aparecer sistemas de información clásicos que se integran vía *APIs* (escritas en diversos lenguajes de programación) o *Web Services*. La capa de persistencia suele estar representada por *Bases de Datos* (donde todavía sigue dominando el modelo relacional [Codd 1970, 1992] y el estándar *SQL* [Melton & Simon 2002], aunque en los últimos años están emergiendo soluciones *No-SQL* [Burtica et al. 2012] y la aplicación de nuevas arquitecturas como *Big Data* [Zikopoulos & Eaton 2011]).

Existen diversos tipos de sistemas de información automatizados dentro de la categoría *PAIS*, entre otros [Hendricks et al. 2007]: algunos «*Enterprise Resource Planning (ERP)*», «*Customer Relationship Management (CRM)*» y «*Supply Chain Management (SCM)*» ofrecen la capacidad de definir sus procesos, pero en los últimos años han cobrado gran protagonismo los denominados «*Business Process Management Systems (BPMS)*» [Dumas et al. 2013] que abarcan el ciclo de mejora continua de los procesos, integrando a los anteriores sistemas con cualquier tipo de *legacy system*

genéricos (de mercado o desarrollados a medida), soportando incluso las actividades realizadas manualmente.

La característica esencial de un *BPMS* es que cuenta con una definición explícita de los procesos de negocio plasmada como un modelo de procesos. Un *BPMS* permite la modelación de cualquier tipo de procesos a excepción de los sistemas de mercado anteriores que cuentan con variaciones de sus procesos específicos. Un *BPMS* es una herramienta estratégica para la gestión de todo el ciclo de vida *BPM* y facilita la *interoperabilidad* de estos modelos de procesos con cualquier tipo de aplicación, sistema (*ERP*, *CRM*, *SCM*, *desarrollos a medida*) e incluso con otros *BPMS*. La *Figura 3.15* muestra la arquitectura de un *BPMS* según el «*Workflow Management Coalition Reference Model*» [Dumas et al. 2013] para estos sistemas de automatización de procesos. Se diferencian *subsistemas* para la a) *Especificación de modelos de procesos* [Reichert & Weber 2012] que quedan almacenados en un *repositorio o catálogo de procesos como modelos*, b) *Ejecución de procesos* que generará *instancias de procesos* [Reichert & Weber 2012] conforme a su correspondiente modelo y en colaboración con el nivel de aplicación, almacenando estas *instancias* en un *log de eventos* y c) El *subsistema de administración y monitorización* que permite analizar el log de eventos [Reichert & Weber 2012], aplicando métricas y facilitando la *reingeniería de procesos* [Hammer & Champy 1993; Mohapatra 2012]. El subsistema de ejecución debe facilitar también la *interoperabilidad* con otros *BPMS*.

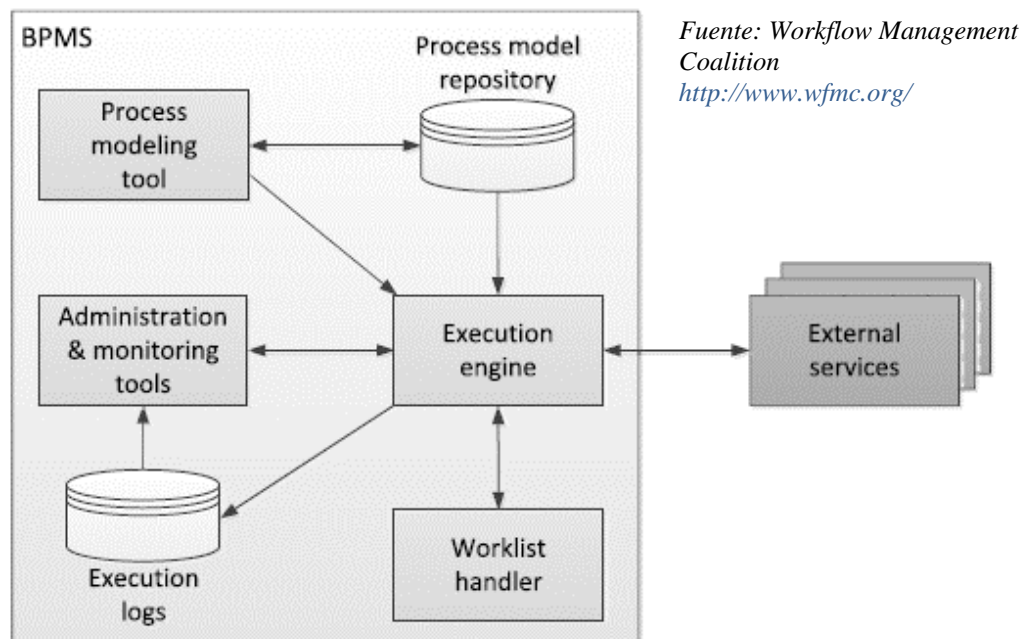


Figura. 3.15. Arquitectura de un BPMS según WfMC

Para la implantación de un sistema automatizado de gestión de procesos (*BPMS*) es necesario que un experto en negocio especifique los modelos con la herramienta de modelación, generando estos modelos sobre el repositorio. Posteriormente se genera

código ejecutable para estos procesos que puede tener interfaces con el nivel de aplicación o con otras herramientas *BPMS*; este segundo paso puede realizarse a través de lenguajes estándar o propietarios, con *APIs* o con *Web Services*. Los modelos iniciales de procesos exigen un trabajo preciso al experto de negocio. En este trabajo de tesis se va a proponer un enfoque que genere automáticamente aproximaciones a los modelos de procesos, partiendo de los estados que se almacenan en algunos tipos de *legacy systems*, como los sistemas para la planificación y gestión de proyectos.

### 3.4 Navigational Development Techniques (NDT)

La mayor parte de los proyectos realizados por el grupo de investigación *IWT2* son de carácter metodológico, basados en «*Navigational Development Techniques (NDT)*» [Escalona 2004; Escalona & Aragón 2008] y en el marco tecnológico *NDTQ-Framework* [Ponce et al. 2013; Aragón 2014] que está basado en modelos de madurez y estándares ISO, en el paradigma *MDE* y el enfoque *BPM*. En el grupo *IWT2* se han desarrollado más de 150 proyectos, donde se ha contrastado la metodología *NDT* y el entorno *NDTQ-Framework* que están en continua evolución, incorporando tendencias tecnológicas y nuevos enfoques. En [Valderas & Pelechano 2011] se analizan y comparan distintas metodologías para la ingeniería de requerimientos aplicada al desarrollo de aplicaciones web bajo el enfoque *MDD*; entre ellas figura *NDT*.

*NDT* utiliza el paradigma *MDE* y consta de un conjunto de *Meta-Modelos* que, inicialmente, ponen el foco en las fases de i) *Ingeniería de Requisitos* y ii) *Análisis*, prestando especial atención en la trazabilidad de transformaciones de artefactos. La Figura 3.16 esquematiza el proceso de transformación para generar los modelos de la fase de Análisis desde la fase de Requisitos.

La fase de requisitos consta de tres actividades: i) *Captura*, en la que se utilizan diferentes técnicas de captura de requisitos tales como entrevistas o «*brainstorming*» para averiguar los objetivos y requisitos del sistema; ii) *Definición*, actividad en la que se formaliza la información capturada a partir de las distintas entrevistas y reuniones con el cliente; y iii) *Validación*, en la que se utilizan de técnicas de evaluación y conciliación de requisitos.

Al concluir la fase de *Requisitos* se generan automáticamente (estereotipo «*QVTTransformations*») los modelos básicos de la fase de *Análisis* aplicando diferentes reglas *QVT* de transformación. *NDT* contempla cuatro modelos en la fase de análisis: i) *Modelo de navegación*, en el que se refleja un esquema de navegación para usuarios; ii) *Modelo de contenido*, que representa la estructura estática del sistema; iii) *Modelo de procesos*, que representa la estructura funcional; y iv) *Modelo de interfaz abstracta*, en el que se representa un conjunto de prototipos de la interfaz del sistema. A partir de estos modelos básicos, el *analista TI* puede completarlos y enriquecerlos para llegar al modelo definitivo de análisis. Este paso no es automático y está representado con el estereotipo

«NDTSupport».

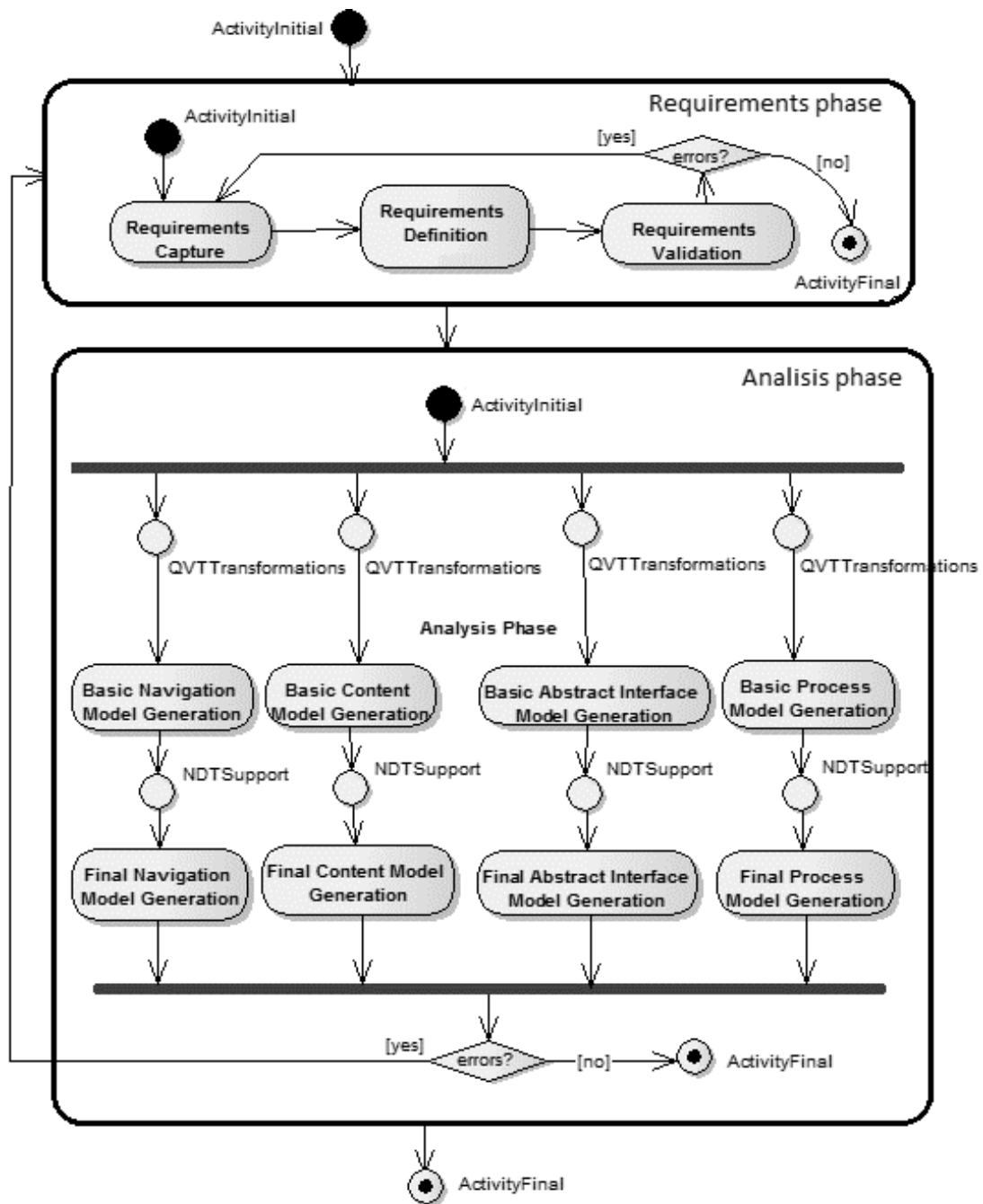


Figura. 3.16. Generación NDT de la fase de Análisis desde la fase de Requisitos

NDT se ha ido extendiendo hasta soportar el ciclo de vida completo del software: i) Estudio de Viabilidad, ii) Ingeniería de Requisitos, iii) Análisis, iv) Diseño, v) Implementación, vi) Pruebas, y vii) Mantenimiento. Los *Meta-Modelos NDT* se formalizan mediante *perfiles UML* [Alhir 2006; Giachetti et al 2008], con el aporte de «Object Constraint Language (OCL)» [OMG:OCL 2014; Iso/Iec 19507:2012] para la

definición de las restricciones sobre los conceptos de los *Meta-Modelos*, y el soporte de la herramienta, de *Sparx Systems Enterprise Architect (EA)* [*Sparx:EA 2015*] como herramienta de modelado UML [*OMG:UML 2011*]. Las transformaciones se formalizan mediante el lenguaje QVT [*OMG:QVT 2011*].

Sobre *EA* se han realizado un conjunto de personalizaciones agrupadas en el entorno *NDT-Suite* [*Garcia-Garcia et al. 2012a, 2012b, 2014; Armario et al. 2012*], un paquete de herramientas Java cuyas herramientas son:

- i) **NDT-Profile**. Proporciona un proyecto base en *EA* que implementa cada uno de los *Meta-Modelos* de *NDT* utilizando los mecanismos de extensión de *UML* junto con la funcionalidad «*Model Driven Generation (MDG)*».
- ii) **NDT-Quality**. Cubre dos objetivos:
  - a) Garantizar la aplicación de las reglas de la metodología *NDT*: 1) en cada una de las fases del ciclo de vida software; para ello, se comprueba la consistencia de todas las restricciones *OCL* definidas en los diferentes *Meta-Modelos* de *NDT*; 2) Controlando la completitud en las definiciones de elementos, la correcta identificación de artefactos siguiendo los patrones lingüísticos establecidos por la metodología, definiciones incorrectas de restricciones de artefactos, la existencia de ciclos en diagramas de actividades, etc.
  - b) Garantizar la trazabilidad entre artefactos cuando se utilizan las reglas de transformación entre modelos definidas por la metodología. Este control es necesario porque *NDT* permite al *analista TI* llevar a cabo modificaciones manuales para enriquecer y completar sus modelos.
- iii) **NDT-Driver**. Esta herramienta implementa las transformaciones *QVT*.
- iv) **NDT-Prototype**. Permite generar un conjunto de prototipos en formato *XHTML* a partir de los modelos de navegación descritos.
- v) **NDT-Merge**. Se encarga de analizar y detectar inconsistencias sintácticas y semánticas en diferentes visiones del mismo catálogo de requisitos.
- vi) **NDT-Report**. Esta herramienta permite generar un entregable de la documentación contenida en un proyecto especificado con *NDT-Profile*.
- vii) **NDT-Glossary**. Implementa mecanismos *MDE* para generar un glosario terminológico a partir del catálogo de requisitos de un proyecto especificado con *NDT-Profile*.
- viii) **NDT-Counter**. Facilita la incorporación de una estimación del esfuerzo o coste necesario para desarrollar un proyecto utilizando la técnica de puntos de casos de uso [*Karner 1993*] sobre el catálogo de requisitos del proyecto definido con *NDT-Profile*.

*NDT* ha evolucionado recientemente para proporcionar un marco de trabajo que motive el uso de nuevos enfoques, normas y paradigmas para el desarrollo de software de calidad. Todo ello desde una perspectiva basada en la *definición de procesos software*. A

día de hoy, *NDT* cubre y define un conjunto de procesos categorizados en seis grupos:

- i) Procesos de *Desarrollo de Software*, basados en el ciclo de vida software que define la propia metodología *NDT*.
- ii) Procesos de *Mantenimiento de Software*, basados en *ITIL* [ITIL 2015] y los modelos de madurez de *CMMI* [Chrissis et al. 2011].
- iii) Procesos de *Pruebas Software*, basados en la reciente norma [Iso/Iec 29119:2014].
- iv) Procesos de *Calidad del Software*, basados en la norma [Iso/Iec 9001:2008] y en los modelos de madurez de *CMMI* [Chrissis et al. 2011].
- v) Procesos de *Gestión de Proyecto Software*, basados en la guía de dirección de proyectos *PMBOK* [PMI:PMBOK 2013] y en los modelos de madurez de *CMMI*.
- vi) Procesos de *Seguridad*, basados en la norma [Iso/Iec 27000:2014].

Todos estos procesos están definidos formalmente con la herramienta *NDTQ-Framework* [Ponce et al. 2013; Aragón 2014] y se han estado aplicando en entornos reales. *NDTQ-Framework* se sustenta sobre un *Meta-Modelo básico de procesos* (Figura 3.17) compatible con la norma [Iso/Iec:24744 2007, 2014], la cual proporciona una guía para la descripción de *procesos del software*. Se contemplan las *Meta-clases*: i) «*Process*» que representa un conjunto de acciones ordenadas que deben ser ejecutadas para obtener un determinado objetivo o resultado de negocio; ii) cada una de estas acciones se representa mediante la *Meta-clase* «*Activity*» que contempla una asociación recursiva para segmentar la actividad en un conjunto de sub-actividades.

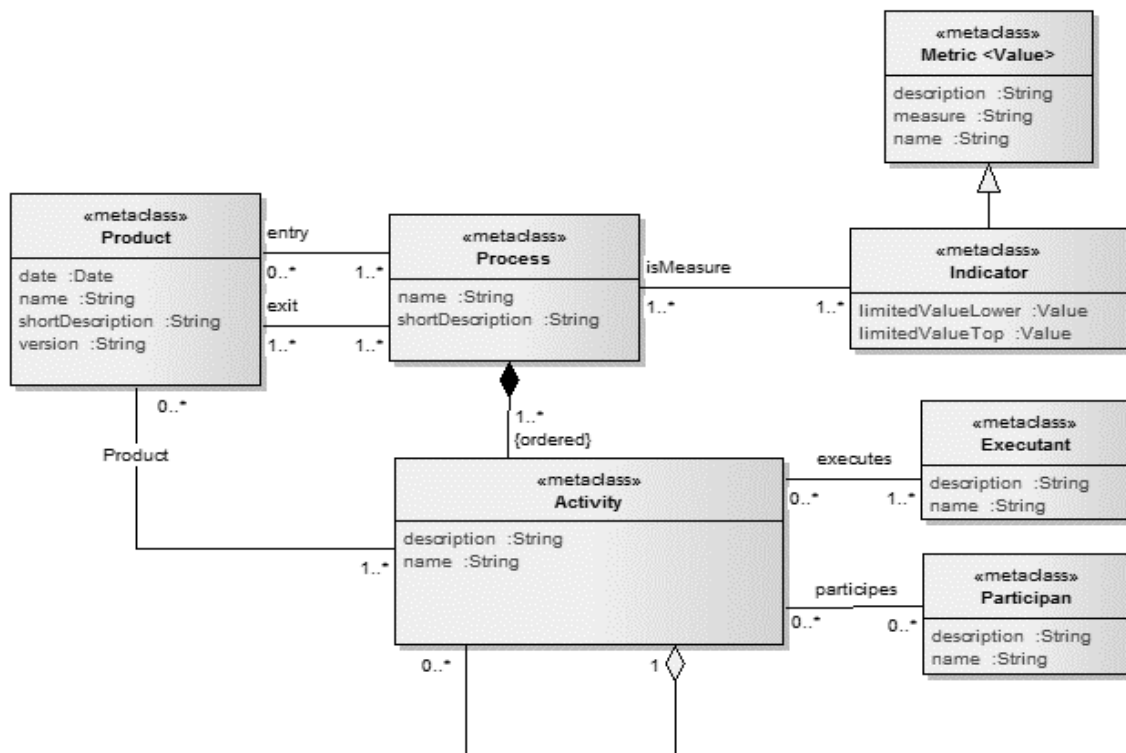


Figura. 3.17. Meta-Modelo de procesos NDTQ-Framework



iii) las clases «*Executant*» y «*Participant*» diferencian al responsable de la actividad de otro actor que participa en la actividad; iv) la *Meta-clase* «*Product*» representa los productos obtenidos como resultado de ejecutar el proceso; v) la *Meta-clase* «*Metric*» se contempla en el *Meta-Modelo* para definir métricas que permitan medir el desempeño del proceso y la *Meta-clase* «*Indicator*» para medir una métrica concreta contra un rango posible de valores «*limitedValueTop*» y «*limitedValueLower*» de la *Meta-clase* «*Indicator*».

NDT se ha utilizado con éxito en diferentes proyectos *I+D+i* a lo largo de la última década. Entre ellos, se pueden mencionar los siguientes: el proyecto *Mosaico* llevado a cabo en el seno de la Consejería de Cultura y Deporte de la Junta de Andalucía [Escalona et al. 2008]; el proyecto *AQUA-WS* [Cutilla et al. 2011] llevado a cabo en EMASESA (Empresa Metropolitana de Abastecimiento y Saneamiento de Aguas de Sevilla S.A.); y más recientemente el proyecto de *CALIPSONeo* [Escalona et al. 2013; Salido et al. 2014] llevado a cabo de forma conjunta con *El Grupo Airbus* (<http://www.airbusgroup.com>).

En [García-Borgoñón et al. 2013] se plantea una propuesta de lenguaje de especificación de procesos del software, lo que ha dado lugar al *Meta-Modelo InRoma* (Figura 3.18), que extiende NDTQ-Framework en el tratamiento de productos, métricas y agentes involucrados en la gestión y ejecución de los procesos del software.

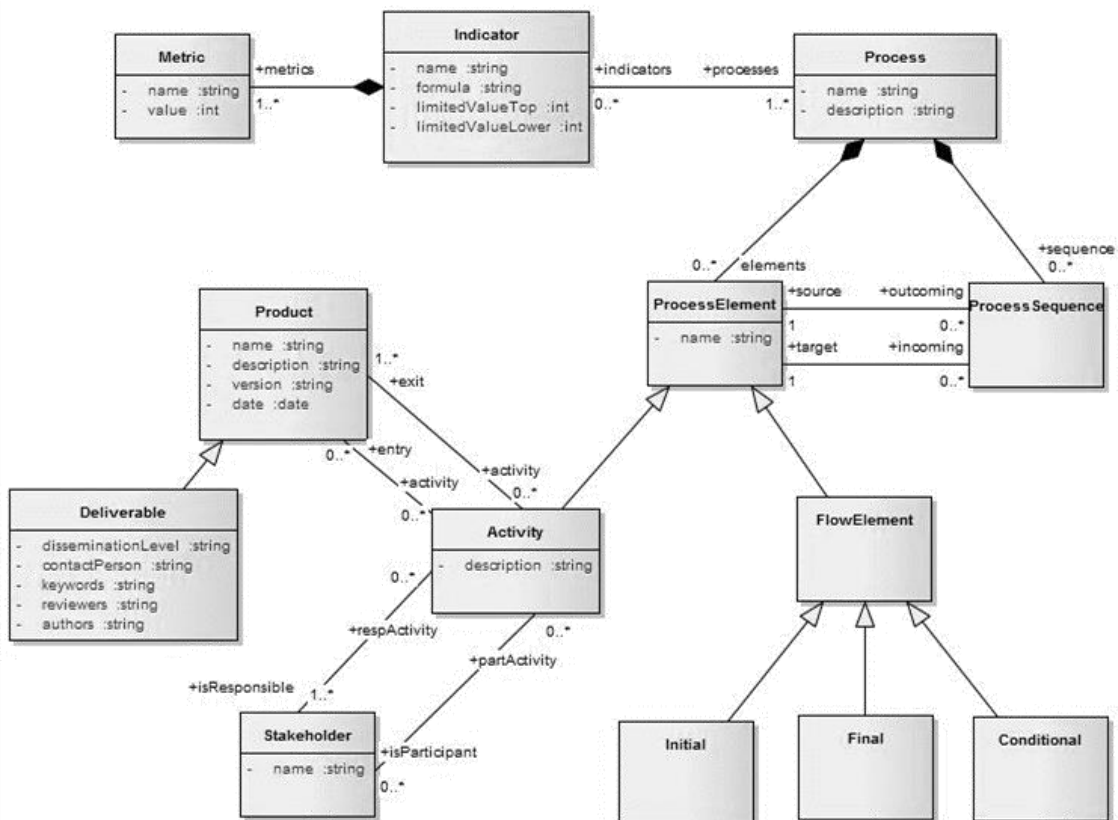


Figura. 3.18. Meta-Modelo InRoma

## 3.5 Modernización de legacy systems

### 3.5.1 Conceptos y problemas en legacy systems

[Bisbal et al. 1997, 1999] identifican la problemática de los *legacy systems*:

- i) Algunas organizaciones cuentan con complejos sistemas, compuestos por centenares de miles de líneas de código, con una vida superior a los diez años, pudiendo llegar incluso a los veinte. Estos sistemas suelen representar el corazón del negocio de la organización y no es fácil sustituirlos por un sistema nuevo sin causar un fuerte impacto en la organización.
- ii) Estos sistemas, creados por diversos equipos de trabajo, sometidos a mantenimiento correctivo y perfectivo a lo largo de años, suelen contar con escasa documentación y de baja calidad. La arquitectura de los sistemas está basada en la tecnología existente en el momento de la concepción, que distará bastante de la tecnología actual.
- iii) El mantenimiento de estos sistemas se vuelve tedioso y muy costoso.
- iv) El desconocimiento de interfaces claras con estos sistemas hace muy difícil su integración con nuevos sistemas.
- v) Es necesario encontrar nuevos enfoques para modernizar estos sistemas, que basándose en tecnología actual, permitan adaptarlos en tiempo y costo aceptables.

En [Ulrich 2002; Seacord et al. 2005; Heuvel et al. 2006; Heckel et al. 2008] se analiza el proceso de modernización de *legacy systems* así como estrategias y procedimientos para llevar a cabo un proyecto con éxito.

En [Stavru et al. 2013] se realiza un estudio del estado del arte sobre modernización de *legacy systems* basada en *MDE*. Se identifican dos tipos de retos: i) Relacionados con *MDD* y ii) Relacionados con la modernización del software (*SM*). A su vez, se establecen dos subcategorías: i) *Retos organizacionales (O)* y ii) *Retos técnicos (T)*. La [Tabla 3.1](#) resume la clasificación de estos retos:

**Tabla. 3.1.** Retos de los sistemas legacy systems

Cat	SC	Reto	
MDD	O	Ausencia de modelos de procesos	
		Adquisición de competencias y experiencia	
		Reestructuración del equipo de desarrollo	
		Reestructuración de ciclos de vida del software	
		Dependencia de modelos de alto nivel	
			Herramientas inmaduras, ausencia de entornos de desarrollo integrados con la plataforma
	T	Gestión de modelos	
		Transformación de modelos	
		Diseño de los lenguajes de modelación	
		Calidad de los modelos	
Generación artesanal de código			
SM	O	Definición del contexto del negocio	
		Falta de compromiso empresarial	
		Resistencia al cambio	
		Adquisición de competencias y experiencia	
		Riesgo incrementado por nuevas tecnologías	
			Falta de conocimiento detallado de las reglas de negocio
	T	Extracción de conocimiento de negocio y técnico del legacy system	
		Garantizar la equivalencia de comportamiento entre los sistemas de origen y destino	
		Coexistencia del sistema origen y destino	

En [Favre 2004] se sientan los fundamentos de la *modernización de software dirigida por modelos* «Model Driven Software Modernization (MDSM)», basado en el paradigma MDE. Un proceso de modernización de software implica cambios sustanciales en un sistema informático existente (*sistema origen*); no se trata de un proceso de mantenimiento del software [Pressman 2010]. Las técnicas empleadas son el *metamodelado* y las *transformaciones entre modelos*.

La reingeniería de un sistema es un tipo de modernización disciplinado, cuyo fin es mejorar la calidad del sistema [Tilley & Smith 1995; Müller et al. 2000], manteniendo el valor estratégico del sistema existente. Un proceso de modernización consta de tres etapas: i) Ingeniería inversa, ii) Reestructuración y iii) Generación o ingeniería directa.

- i) La *ingeniería inversa* se inicia con la inyección de modelos a partir de artefactos (como *código o un esquema relacional*) del *sistema origen*.
- ii) La *reestructuración* consiste en una sucesión de transformaciones *modelo-a-modelo (M2M)* que pueden elevar el nivel de abstracción de cada modelo obtenido en función de la semántica del *Meta-Modelo* correspondiente. Cuando el nivel de abstracción llega al experto de negocio, éste puede incorporar nuevo conocimiento a los modelos generados, enriqueciendo su valor para la fase de *generación*.
- iii) La *generación o ingeniería directa* es otra cadena de transformaciones *M2M* que van disminuyendo el nivel de abstracción, pudiendo llegar hasta obtener código final en el *sistema destino*.

Una *migración* es un *caso especial de reingeniería* donde sólo se cambia la *plataforma tecnológica*, manteniendo la funcionalidad del sistema.

A continuación se describe la iniciativa *OMG ADM* para soportar *MDSM*.

### 3.5.2 Arquitectura ADM para modernización de sistemas

«*Architecture Driven Modernization (ADM)*» [Newcomb 2005; *OMG:ADM 2005*; Ulrich & Newcomb 2010] es el estándar propuesto por el *OMG* para la *modernización de legacy systems*. Propone un proceso y *Meta-Modelos* para llevar a cabo esta acción. Los *Meta-Modelos* son «*Abstract Syntax Tree Meta-Model (ASTM)*» [OMG:ASTM 2011] y «*Knowledge Discovery Model (KDM)*» [Iso/Iec 19506:2009]. ASTM se utiliza en las primeras etapas para obtener árboles de sintaxis abstracta, bien con las características gramaticales genéricas «*Generic Abstract Syntax Tree Meta-Model (GASTM)*» o específicas «*Specific Abstract Syntax Tree Meta-Model (SASTM)*» del código del *legacy system* que se inspecciona.

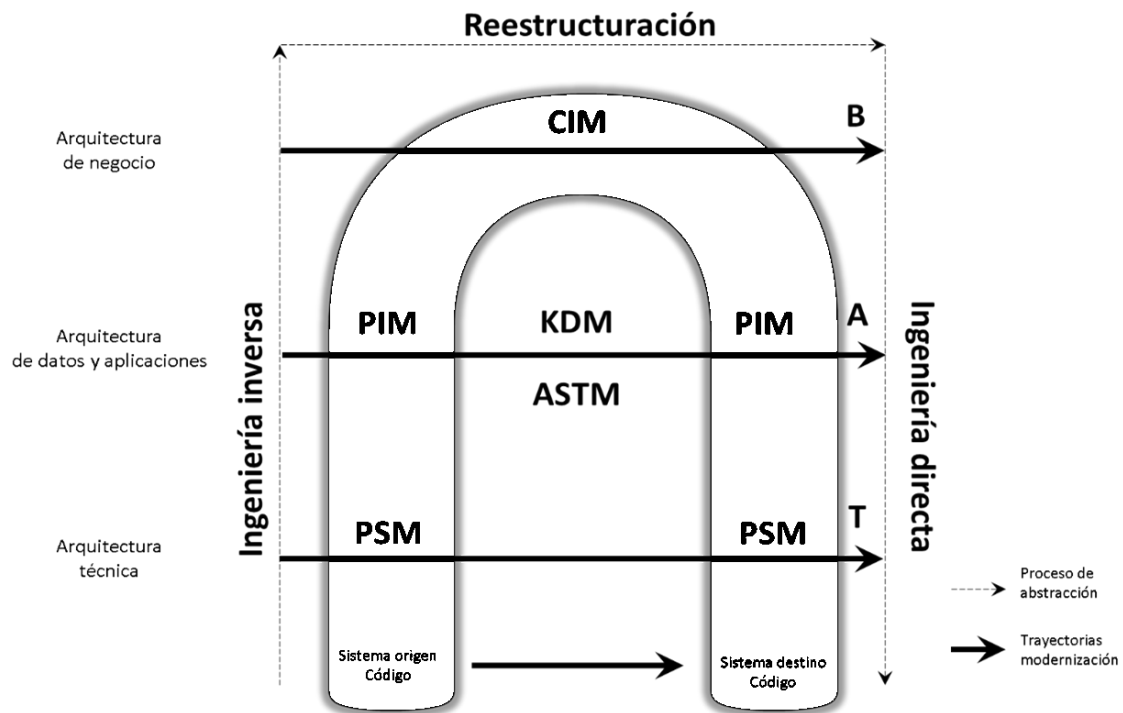


Figura. 3.19. Modelo de herradura ADM «Horseshoe Model»

El proceso de modernización de *legacy systems* que propone *ADM* está caracterizado por un «*ciclo en herradura (Horseshoe Model)*» (Figura 3.19.), donde se propone un camino ascendente para el proceso de «*Ingeniería inversa*» [Chikofsky & Cross 1990] y un camino descendente para la «*Ingeniería directa*». La «*Ingeniería inversa*» parte de los artefactos del sistema origen (código y esquemas de BD) y aplica transformaciones para ir derivando modelos de mayor nivel de abstracción (*PSM*, *PIM* y *CIM*) en la plataforma que elige el ingeniero de sistemas. A nivel *CIM*, el experto en negocio cuenta con modelos

independientes de la tecnología dónde puede analizar la especificación y puede modificar o añadir nuevos artefactos al modelo (*proceso de Reestructuración*). La «*Ingeniería directa*» es el proceso de obtención automática de código desde la especificación de más alto nivel. *ASTM* se utiliza en las capas o niveles próximos al código (*PSM*) y *KDM* para resolver diversas transformaciones entre las capas *PSM* y las *PIM* y *CIM*. Las trayectorias de modernización pueden plantear objetivos de menor o mayor calado en cuanto a su impacto en la organización; así, pueden plantearse:

- i) «*Trayectorias de modernización tecnológica (T)*», donde se mantiene la funcionalidad del sistema y se cambian sólo componentes tecnológicos (lenguajes, infraestructura software y hardware).
- ii) «*Trayectorias de modernización de la arquitectura de aplicaciones y datos (A)*», donde se reestructuran la capa de persistencia y el código de las aplicaciones para facilitar el mantenimiento y beneficiarse de una arquitectura de datos más moderna pero manteniendo la funcionalidad del sistema.
- iii) «*Trayectorias de modernización de la arquitectura de negocio (B)*». Esta es la más ambiciosa, pues afecta a la manera de ver los procesos de negocio e incluso a readaptarlos o reestructurarlos (reingeniería de procesos de negocio [*Hammer & Champy 1993; Mohapatra 2012*]); obliga a regenerar la arquitectura de datos y aplicaciones y a generar nuevo código para todos los nuevos artefactos o los modificados.

### 3.6 Resumen y conclusiones

En el Capítulo se han revisado tecnologías, estándares, tendencias y buenas prácticas relacionadas con los objetivos del trabajo de tesis.

En primer lugar se han planteado las características de los procesos software cuyos problemas inherentes, aunque se detectaron hace ya más de 40 años (crisis del software enunciada por *Dijkstra*), todavía siguen estando vigentes en muchos proyectos informáticos, llevándolos incluso a su cancelación<sup>36</sup>. Los lenguajes de programación, con el tiempo, han ido elevando su nivel de abstracción, acercándose cada vez más al nivel cognitivo del programador, facilitándole la tarea, evitando o reduciendo algunos errores y, en general aunque aisladamente, mejorando la productividad; pero no es la única solución, puesto que la tecnología, en paralelo, se ha ido complicando con multitud de sistemas operativos, gestores de bases de datos, lenguajes, entornos de desarrollo, etc. Los paradigmas *MDE* y *BPM* ofrecen nuevos niveles de abstracción que se acercan al experto para que colabore activamente en la especificación de modelos más cercanos al dominio del negocio.

Se han revisado los conceptos *MDE*, donde el artefacto fundamental es el

---

<sup>36</sup> informe Chaos de la consultora Standish Group Inc [*Standish 2012*].

*Meta-Modelo* para soportar distintos niveles de abstracción y plataformas que ayudan a conseguir buenos niveles de interoperabilidad. *MDA* es la iniciativa del *OMG* más popular que soporta *MDE*. Gira en torno al concepto de *Meta-Meta-Modelo MOF* y la utilización de otros *Meta-Modelos*, como: *UML*, *CWM*, *IMM* y *QVT*.

*BPM* también otorga independencia e interoperabilidad cuando se sustenta sobre sistemas automatizados orientados a procesos (*PAIS* y especialmente *BPMS*), que independizan la capa de procesos de la capa de aplicación. Se revisa la arquitectura de un *BPMS* propuesta por el *WfMC* y los estándares esenciales en gestión de procesos, fundamentalmente para la definición, ejecución y portabilidad o intercambio de procesos, revisando las propuestas de *OMG*, *WfMC* y *OASIS*. Se analizan los principales estándares vinculados al ciclo de vida *BPM*, donde destacan *BPMN*, *XPDL* y *BPEL* para la definición, intercambio y ejecución de procesos.

En el grupo de investigación *IWT2*, los proyectos y colaboraciones con distintas empresas y organismos se desarrollan bajo el paraguas de la metodología *NDT*. Se revisan las características de esta metodología y se describe el entorno *NDTQ-Framework* que soporta *NDT*, su *Meta-Modelo* de procesos y un conjunto de herramientas, *NDT-Suite*, que soportan *NDTQ-Framework* con distintos plugins desarrollados sobre *Sparx Systems Enterprise Architect (EA)* que implementan distintas transformaciones *QVT* y reglas de validación.

Por último, ya que se van a proponer enfoques de ingeniería inversa desde *legacy databases*, se detalla la contribución de *OMG ADM* como arquitectura de modernización de *legacy systems*, que aporta *Meta-Modelos* para describir sintaxis abstracta (*ASTM: GASTM* y *SASTM*) así como el *Meta-Modelo KDM* para extraer conocimiento de estos sistemas. *ADM* promueve distintos caminos de modernización desde un sistema origen a un sistema destino, con un ciclo ascendente inicial de ingeniería inversa, que va incrementando el nivel de abstracción según el camino elegido y un camino descendente o de ingeniería directa que lleva un modelo del sistema origen al sistema destino. Durante todo el proceso, los modelos se describen conforme a *Meta-Modelos* de referencia y las transformaciones se realizan en *QVT* utilizando las facilidades *MOF*.

# Capítulo 4. TRABAJO RELACIONADO

*“Aquel que duda y no investiga, se torna no sólo infeliz, sino también injusto”  
Blaise Pascal (1623-1662)*

**E**n este capítulo se analizará la literatura científica específica de distintos investigadores que desarrollan el estado del arte del Capítulo anterior. Se consideran distintos bloques: i) Publicaciones relacionadas con la definición y verificación de reglas temporales sobre procesos de negocio; ii) Literatura sobre procesos del software; iii) Modernización de *legacy systems*, que a su vez describe trabajos sobre a) La problemática de los *legacy systems* y como abordar su tratamiento, b) La ingeniería dirigida por modelos (*MDE*), c) La arquitectura *ADM* que el organismo *OMG* propone para la modernización de *legacy systems* y d) Ingeniería inversa de bases de datos relacionales. Por último: iv) Se han analizado trabajos sobre otras perspectivas de los procesos, sobre todo, *la organizacional* y *la de casos*, ya que sirven como referencia al enfoque de la tesis y permitirán extenderlo en un futuro.

## 4.1 Introducción

La dimensión temporal de los procesos es la base sobre la que se sustentan las propuestas de este trabajo de tesis; con su especificación a distintos niveles de abstracción, van a cobrar sentido transformaciones basadas en *MDE* que llevarán desde sistemas de bajo nivel a sistemas al nivel de conocimiento del experto de negocio. La bibliografía está llena de trabajos de investigación concernientes a la dimensión temporal sobre procesos de negocio. Hay que distinguir los enfoques que proponen la especificación de esta dimensión en los procesos de negocio de aquellos que van más allá y, basándose en la especificación, proponen métodos para verificar el cumplimiento de las reglas establecidas. Aquí, a su vez, caben dos orientaciones: i) Métodos de verificación que analizan o monitorizan trazas de procesos que ya se han producido y ii) Métodos que son capaces de analizar el proceso en tiempo de ejecución.

A continuación se dedica una sección al análisis de bibliografía dedicada a los procesos de gestión del software, contemplando estándares y buenas prácticas. Para la selección de referencias se han tenido en cuenta las recomendaciones que permiten hacer un «Estudio sistemático de la literatura (*SLR*)» [*Kitchenham et al. 2009*].

Utilizar un enfoque de ingeniería inversa *MDE* sobre *legacy systems* obliga a revisar aportaciones de la comunidad científica relacionados con: i) Procesos de modernización del software, en particular, los basados en la propuesta *OMG ADM*, así como otros alternativos que no utilizan *ADM*; ii) Hay que contemplar trabajos que focalizan en la ingeniería inversa desde bases de datos relacionales, ya que va a ser el primer artefacto de dónde se pretende extraer la dimensión temporal que permitirá ofrecer aproximaciones a los procesos de la organización.

La última sección se dedica a trabajos que, de algún modo, contemplan otras dimensiones BPM, como la organizacional o la de casos; hay que tener en cuenta este tipo de contribuciones por la orientación que dan a la captura de esta dimensión y porque estas dimensiones pueden ser utilizadas en ampliaciones o trabajos futuros relacionados con esta tesis.

## 4.2 Reglas temporales sobre procesos

Los trabajos pueden clasificarse en i) Autores que sólo trabajan en la especificación o modelación del tiempo y ii) Autores que además proponen métodos de verificación de reglas temporales sobre procesos.

- i) En el primer grupo, cabe destacar:
  - a) Quienes utilizan «*Time Petri Nets*», por ejemplo: [Huai et al. 2010; Makni et al. 2010, 2011].
  - b) Otros prefieren como herramienta «*Timed Automata*», tales como [Watahiki et al. 2011; Cheikhrouhou et al. 2014a, 2014b].
  - c) [Du et al. 2011] proponen «*Time Workflow Nets*».
  - d) [Kazhamiakin et al. 2006] usan «*Web Service Timed State Transition Systems*».
  - e) [Kallel et al. 2009] con «*XTUS-Automata*».
  - f) [Wong & Gibbons 2009] proponen «*Communicating Sequential Processes (CSP)*».
  - g) [Eder & Tahamtan 2008a, 2008b] usan «*Timed Activity Graphs*».
  - h) En [Barba et al. 2012] se propone un enfoque para generar planes que optimizan el tiempo global de ejecución de un proceso en base a una función objetivo que tiene en cuenta modelos declarativos de restricciones temporales sobre procesos. Los planes generados se pueden utilizar para diferentes propósitos: suministrar planes de ejecución para usuarios; facilitar la detección temprana de situaciones críticas y predecir el tiempo de ejecución de actividades.

[Barba et al. 2013] realizan una propuesta a los usuarios para minimizar el tiempo de ejecución global de un proceso sobre un *PAIS* donde exista flexibilidad para este objetivo. Es un enfoque basado en «*programación con*



*restricciones (constraint programming)*» [Rossi et al. 2006], que considera el flujo de control y la asignación de recursos con capacidad para regenerar nuevos planes de ejecución del proceso.

- i) En [Gómez-López et al. 2014b] se propone también un enfoque declarativo de reglas basadas en la «programación con restricciones» [Rossi et al. 2006] sobre un proceso. Muchos de los métodos habituales de verificación de reglas están basados en el diagnóstico sobre trazas de procesos en el log de eventos, lo que significa que la regla ya se ha violado. En el trabajo se propone un método para pronosticar en tiempo de ejecución la violación de las reglas declaradas y asegurar que se obtienen trazas robustas en el log de eventos.
- j) Por último, un grupo de autores coinciden en considerar *BPMN* como un pilar para desarrollar extensiones temporales, tales como [Gagné & Trudel 2009a; Flores & Sepúlveda 2011; Watahiki et al. 2011; Cheikhrouhou et al. 2013b].
- ii) En la segunda categoría, en relación a la verificación de reglas temporales, los investigadores tratan de verificar cuellos de botella, situaciones de abrazo mortal o bucles infinitos, tareas muertas, duraciones mínimas y máximas de procesos y la consistencia temporal entre otros aspectos. Pueden distinguirse las siguientes contribuciones:
  - a) [Watahiki et al. 2011; Du et al. 2011; Cheikhrouhou et al. 2014a, 2014b] trabajan con el entorno integrado «UPPAAL» para llevar a cabo la validación y verificación de sistemas en tiempo real modelados como redes de autómatas temporizados, extendidos con tipos de datos (enteros acotados y arrays entre otros).
  - b) [Eder & Tahamtan 2008a, 2008b] usan algoritmos.
  - c) [Kazhamiakin et al. 2006; Wong & Gibbons 2009; Huai et al. 2010; Makni et al. 2010, 2011], manejan otras técnicas diferentes a las anteriores.
  - d) [Cheikhrouhou et al. 2013a] es un buen estudio comparativo de la literatura sobre aspectos de especificación y verificación de reglas temporales sobre procesos.

En este trabajo de tesis se pone el foco en la especificación de reglas temporales sobre *Meta-Modelos* de procesos, pues se va a proponer un enfoque *MDE* basado en la perspectiva temporal *BPM* para extraer modelos de procesos de *legacy systems*; por esta razón se discutirán con más detalle las propuestas relacionadas con la expresión de reglas de tiempo sobre el estándar *BPMN*, donde todos los autores identifican cierta debilidad de *BPMN* para especificar algunos tipos de reglas temporales.

[Flores & Sepúlveda 2011] trabajan sobre *BPMN* 1.2. Parten de las restricciones de tiempo utilizadas en *Diagramas de Gantt* [Gantt 1919; PMI:PMBOK 2013] y *PERT* [Malcolm et al. 1959; PMI:PMBOK 2013] para la planificación y control de proyectos con *MS Project* [Hansen & Hansen 2013; Stover 2007]. Proponen usar patrones o construcciones *BPMN* para modelar cada clase de restricción de tiempo sobre una

actividad o bien las restricciones asociadas a las relaciones de precedencia entre dos actividades. Esta propuesta se basa en la aplicación de artefactos *BPMN* estándar, aunque la versión *BPMN 1.2* todavía no soportaba la característica «*Eventos intermedios no interruptores del flujo de control (non-interrupting-intermediate events)*» y esto obliga a los autores a manejar eventos de lanzamiento y captura de señales que acaban sobrecargando los patrones propuestos por dicha limitación de la versión del estándar.

Existe una contribución de referencia en cuanto a la especificación de la perspectiva temporal con *BPM*, denominada «*Time-BPMN*» [Gagné & Trudel 2009a] que se basa en trabajos que sustentan dicha propuesta: [Gagné & Trudel 2008] describe una formalización de reglas de tiempo sobre procesos y [Gagné & Trudel 2009b] formaliza la semántica temporal que se encuentra en *MS Project*, utilizando el «*Álgebra de intervalos de Allen*» [Allen 1983]. En estos trabajos, los autores proponen aplicar la semántica de los eventos de inicio y terminación de actividades (que está poco desarrollada en el estándar), tomando control de los mismos, así como estableciendo dependencias entre estos eventos cuando existen relaciones de precedencia temporales entre ellas (añaden este tipo de asociación y las tipifican como distintas reglas temporales). También relajan el concepto de duración de una actividad, admitiendo *duración fija* o *flexible* de la misma. Recomiendan nuevos tipos de decoradores (eventos incrustados en las actividades y asociaciones para las dependencias temporales) que alteran la apariencia de un diagrama *BPMN*, ofreciendo una especificación simple y clara, pero sin embargo, fuera de la cobertura actual del estándar [OMG:*BPMN 2013*; *Iso/Iec:19510 2013*], lo cual puede dificultar la transformación de estos modelos en modelos ejecutables.

[Cheikhrouhou et al. 2013b] es una propuesta que extiende la propuesta *Time-BPMN* anterior, basándose también en el empleo de nuevos decoradores *BPMN*. Incluyen las siguientes categorías de reglas: i) *Restricciones temporales intra-actividad*, ii) *Restricciones temporales inter-actividades*, iii) *Restricciones temporales inter-procesos*, iv) *Restricciones temporales correlacionadas con recursos/datos*, que se comentan a continuación:

- i) **Restricciones temporales intra-actividad.** Recogen restricciones temporales (*TC*) asociadas con una actividad, tales como: a) Duración, b) *TC* sobre la cardinalidad, c) Comienzo y Fin y d) La restricción de ausencia intra-actividad.
- ii) **Restricciones temporales inter-actividad.** Estas agrupan las restricciones que cruzan la frontera de una actividad, pero siempre dentro del mismo proceso y en la misma organización, tales como a) Dependencias temporales (*TD*) y b) La restricción de ausencia inter-actividades.
- iii) **Restricciones temporales inter-procesos.** Esto puede ocurrir en los casos de colaboraciones o «*Procesos de Negocio Inter-Organizaciones (Inter-Organizational Business Processes (IOBP) entre dos organizaciones*»

[Attaran 2004; Cheikhrouhou 2013a, 2014b]. Se definen dos subcategorías de restricciones en función del tipo de artefacto que las soporta: a) Datos temporales intercambiados y b) Vencimientos o fechas tope de mensajes intercambiados.

- iv) **Restricciones de tiempo correlacionadas con recursos/datos.** Conciene a grupos de recursos o recursos individuales que tienen otorgada autorización para ejecutar actividades dentro de ciertos intervalos de tiempo donde pueden llevar a cabo su trabajo. Estas correlaciones pueden definir duraciones de algunas actividades en función de determinados tipos de datos que son relevantes en cada actividad.

[Lanz et al. 2014] contiene un estudio más reciente donde se analiza la perspectiva temporal sobre PAIS, estableciendo categorías y patrones de reglas que permiten realizar un análisis sistemático de la literatura para analizar PAIS, propuestas académicas, estándares y *legacy systems*, entre los cuales está *Microsoft Project*. Se contemplan las siguientes categorías de reglas y patrones temporales:

- i) **Duraciones y lapsos de tiempos**, que contempla los patrones: *TP1: Lاپso de tiempo entre dos actividades*, *TP2: duraciones*, *TP3: Lاپso de tiempo entre dos eventos arbitrarios*;
- ii) **Restricciones sobre tiempos de ejecución**, que incluye: *TP4: Fechas fijas sobre elementos*, *TP5: Restricción de horarios sobre elementos*, *TP6: Restricciones basadas en el tiempo* y *TP7: Períodos de validez*.
- iii) **Variabilidad**, *TP8: Variabilidad dependiente del tiempo*;
- iv) **Elementos del proceso recurrentes**, que incluye: *TP9: Elementos cíclicos* y *TP10: Periodicidad*.

En [Arevalo 2015b] se realiza una propuesta para extensión de *BPMN 2.0* con reglas temporales, sustentada por un *Meta-Modelo* de procesos que extiende al de *BPMN* y un conjunto de fórmulas *OCL* que soportan una taxonomía de reglas.

### 4.3 Procesos de negocio del software

En [García-Borgoñón et al. 2014b] se ha llevado a cabo un estudio comparativo (*Systematic Literature Review (SLR) sobre lenguajes de modelación de procesos del software*). Una de las conclusiones es la evolución de las organizaciones del software hacia la integración del paradigma *BPM* [van der Aalst 2004; Netjes et al. 2006] para gestionar sus procesos de negocio, utilizando estándares genéricos para la modelación como *BPMN* [OMG:BPMN 2013] y específicos como *SPEM* [Bendraou et al. 2007; OMG: SPEM 2008] e *Iso/Iec:24744* [Iso/Iec:24744 2007, 2014], amparándose en la sintaxis abstracta de sus *Meta-Modelos*.

Por otro lado, dentro del grupo de investigación *IWT2*, se han realizado diversos

proyectos de carácter metodológico [Cutilla et al. 2012; Escalona et al. 2013; Salido et al. 2014], utilizando el paradigma MDE y con enfoque BPM con instituciones públicas y privadas bajo el paraguas de la metodología «*Navigational Development Technique (NDT)*» [Escalona 2004; Escalona & Aragón 2008] que se ha acompañado del desarrollo del framework *NDTQ-Framework* [Ponce et al. 2013; Aragón 2014] y del conjunto de herramientas *NDT-Suite* [García-García et al. 2012a, 2012b, 2014; Armario et al. 2012]. *NDTQ-Framework* está basado en *Meta-Modelos*; entre ellos, el *Meta-Modelo* de procesos es compatible con el *Meta-Modelo* de procesos propuesto en la norma [Iso/Iec:24744 2007, 2014] y ha sido utilizado, añadiendo extensiones, en trabajos como [García-Borgoñón et al. 2014a] y en la tesis [García-García 2015], así como en otros trabajos de tesis en curso dentro del grupo IWT2. En la Sección 2.4 se describe con más detalle *NDT*.

#### 4.4 Modernización de legacy systems

Modernizar un *legacy system* implica procesos de ingeniería inversa e ingeniería directa. La ingeniería inversa se define en [Chikofsky & Cross 1990]: “*consiste en analizar un sistema origen, identificando sus componentes e interrelaciones y creando una representación del mismo en un sistema destino a un nivel mayor de abstracción*”. A partir de estas representaciones, materializadas como modelos, puede abordarse la ingeniería directa para llegar al sistema destino, con transformaciones más o menos automatizadas.

En primer lugar se revisan un grupo de aportaciones que utilizan el paradigma MDE y la arquitectura MDA:

- i) La plataforma *MOSkitt* [Pelechano 2012; García et al. 2013] es un framework basado en *Eclipse EMF* [Merks 2010; Eclipse:EMF 2015]. *MOSkitt* es utilizado en el ámbito de la *Comunidad Valenciana*. Permite tanto ingeniería inversa como ingeniería directa.
- ii) En [Bézivin & Kurtev 2005] se utiliza el concepto de «espacio tecnológico» [Kurtev et al. 2002] como «forma de definir tecnologías a un alto nivel de abstracción», considerando «Grammarware» y «Modelware» como espacios tecnológicos para tratar las gramáticas de los lenguajes y los modelos. En [Wimmer & Kramler 2005] se trata la integración entre estos dos espacios o, dicho de otro modo, se proveen mecanismos para integrar el espacio «Grammarware» con el «Modelware». «MOF Model to Text Transformation Language (MOFM2T)» [OMG:MOFM2T 2008] es la propuesta OMG para este tipo de transformaciones (M2T), mientras que (QVT) [OMG:QVT 2011] lo es para las transformaciones M2M.

En torno a los espacios *Grammarware* y *Modelware*, se destacan un grupo de trabajos:

- a) En [Izquierdo et al. 2008] y [Izquierdo & Molina 2009a, 2009b, 2014] se propone la tecnología «Gra2Mol» como puente para soportar la extracción de modelos desde el código de aplicaciones (*Java*, *Cobol*, etc.). En [Izquierdo et al. 2012], «Api2Mol» permite obtener *Meta-Modelos* desde una *API* en *Java*.
- b) En [Sánchez Ramón et al. 2013] se realiza una valoración de los procesos de modernización de *legacy systems* basados en *MDE*. Algunas de sus conclusiones son: 1) Los *Meta-Modelos estándar ASTM* y *KDM* ofrecen interoperabilidad pero son complejos y difícil de integrar; 2) *KDM* no ofrece suficiente nivel de detalla y, en función del lenguaje origen, hay que utilizar extensiones (*GASTM*, *SASTM*); 3) Las herramientas disponibles para la modernización y las soluciones basadas en estos *Meta-Modelos* son poco escalables.

[Sánchez Ramón et al. 2014] proponen la obtención de modelos desde interfaces gráficas de usuario de *legacy systems*.

- c) En [Bermúdez et al. 2014] se propone un enfoque *MDE* para la interoperabilidad de herramientas de modelado de datos con *DB-Main* a través de *APIs*, *XML* y formatos propietarios. «*Model-driven Interoperability (MDI)*» es el área que se ocupa de esta aplicación de *MDE*; este trabajo se enmarca en el área *MDI* para la interoperabilidad de herramientas basadas en modelos [Fabro et al. 2006; Bruneliere et al. 2010a].

Dentro del enfoque *MDE* hay que distinguir trabajos que utilizan específicamente el estándar “*Architecture Driven Modernization (ADM)*” [Newcomb 2005; OMG:ADM 2005]:

- i) [Ulrich & Newcomb 2010] exponen casos de estudio de modernización de *legacy systems* basados en *ADM* y en [Stavru et al. 2013] se analizan los retos a los que se enfrentan las organizaciones en los procesos de modernización de *legacy systems*.
- ii) En [Pérez-Castillo et al. 2011b] se propone la aplicación del estándar *KDM* [Iso/Iec 19506:2009] para el proceso de modernización de *legacy systems*.
- iii) [Cuadrado et al. 2012] definen parametrizaciones para las transformaciones en procesos de modernización *ADM* con trayectorias horizontales (en el mismo nivel de abstracción).
- iv) En [Deltombe et al. 2012] se analiza la aplicación de *KDM* y *ASTM* como puentes para el proceso de modernización.
- v) [Estevez et al. 2013], del grupo de investigación de la organización *Open Canarias*<sup>37</sup>, describen una experiencia de modernización en el sector bancario, basada en *ADM*.

Por el carácter del enfoque que se propondrá en la tesis, se han revisado trabajos

---

<sup>37</sup> <https://www.opencanarias.com/>

específicos que realizan procesos de *ingeniería inversa sobre bases de datos relacionales*. Los siguientes trabajos cubren la extracción de modelos desde esquemas relacionales, centrándose fundamentalmente en las características estructurales (tipos, claves, etc.):

- i) En [Pérez et al. 2003] se propone un esquema para la extracción de modelos UML desde esquemas relacionales (*legacy databases* [Bisbal et al. 1997,1999]). Este trabajo es previo a la aparición y aplicación de MDE.
- ii) [Boronat et al. 2005] utilizan reescritura de fórmulas lógicas como soporte en el proceso de extracción de esquemas desde *legacy systems*.
- iii) [Paradauskas & Laurikaitis 2006] trabajan con *legacy databases*; aplican heurísticas muy elaboradas para inferir inicialmente clases básicas UML y en pasos posteriores rastrean el esquema origen para enriquecer el esquema conceptual destino con información sobre claves y restricciones embebidas en las sentencias de creación de las tablas.
- iv) En [Reus et al. 2006] se lleva a cabo un proceso de ingeniería inversa de una base de datos ORACLE para una compañía de Seguros, partiendo de código PL/SQL [Feuerstein & Pribyl 2005] y utilizando ASTM.
- v) En [Manzón & Trujillo 2007] se trata el diseño de *datawarehouses* desde bases de datos relacionales, aplicando un enfoque MDE para llevar a cabo la ingeniería inversa.
- vi) [Alalfi et al. 2008] proponen «SQL2XMI» para realizar ingeniería inversa de bases de datos relacionales [Codd 1970, 1992] a esquemas del modelo entidad/relación de Chen [Chen 1976], obteniendo formatos «XML Metadata Interchange (XMI)» [OMG:XMI 2013; Ruiz et al., 2003] como resultado.
- vii) En [Bermúdez et al. 2012] se define un framework para la modernización de datos.
- viii) En [Díaz et al. 2013] se propone «Schemol» para la extracción de *Meta-Modelos* desde esquemas relacionales. Es una aplicación sobre la base «Grammarware» y «Modelware» de trabajos anteriores como «Gra2Mol» y «Api2Mol».
- ix) En [Pérez-Castillo et al. 2012b, 2012c] se proponen heurísticas, basadas en extensiones de ADM, para la extracción de modelos desde esquemas relacionales.
- x) En trabajos como [Cosentino 2013, Cosentino & Martínez 2013] y [Arevalo et al. 2013], además de los aspectos estructurales se propone la extracción de reglas de negocio sustentadas en restricciones encapsuladas en el esquema de la BD. [Arevalo et al. 2013] utilizan un *Meta-Modelo PIM* basado en «UML Production Rule Representation (PRR)» [OMG:PRR 2009] para modelar *ECA Rules* [Viana et al. 2007] y recoger las restricciones [Türker & Gertz 2001] escritas en columnas o en código de disparadores sobre lenguajes procedurales como «PL/SQL» [Feuerstein & Pribyl 2005], «Transact\*SQL» [Colledge 2008; Ben-Gan 2012; Microsoft™:T-SQL 2014], «PL/pgSQL» [PL/pgSQL 2014] o «mySQL Stored Procedures» [Harrison & Feuerstein 2008] como variantes del lenguaje estándar «SQL Persistent Storage Modules SQL/PSM» [Eisenberg 1996; Melton

& Simon 2002]. En [Arevalo 2014; Arevalo et al. 2013, 2015a, 2015b] se proponen enfoques MDE para extraer reglas de negocio desde *BD relacionales*.

- xi) En [Gómez-López et al. 2014a] se proponen técnicas para sincronizar los estados de la *base de datos relacional (legacy database* [Bisbal et al. 1997,1999; Pérez et al. 2003]) de un sistema informático con «*procesos de negocio centrados en actividades (Activity-centric business process)*» [Kumaran et al. 2008], utilizando un *lenguaje de dominio específico* [van Deursen 1997, 2000] y reglas de mapeo a una *base de datos Objeto-Relacional*.

## 4.5 Otras perspectivas BPM: Organizacional y Casos

En «*Process Mining Manifesto*» [van der Aalst et al. 2012] se referencian las perspectivas que entran en juego en procesos de *reingeniería de procesos de negocio* [Hammer & Champy 1993; Mohapatra 2012]: i) *Flujo de control*, ii) *Organizacional o de los Recursos*, iii) *Temporal* y iv) *Casos*, aunque han sido tratadas en otros trabajos como [Jablonski 1996; Reichert & Weber 2012; van der Aalst et al. 2004, 2005, 2007, 2012].

El foco del trabajo de tesis se va a realizar sobre la *perspectiva temporal*, hay que tener en cuenta algunas contribuciones que servirán para extender en el futuro las propuestas que se realizarán en esta investigación.

- i) En [Awad et al. 2009; Stroppi et al. 2012] se considera la perspectiva *Organizacional* o de los *Recursos* y la de *Casos*, proponiendo un *Meta-Modelo* basado en *BPMN* que captura estas dimensiones para establecer invariantes *OCL* que modelan hasta veintiocho patrones *Workflow Resource Patterns (WRPs)* [Gamma et al. 1995; Tran et al. 2007]. Su *Meta-Modelo* tiene el mismo objetivo que el que se propondrá en la sección 7.2. para formular la dimensión temporal, por lo que la fusión de nuestra solución con este enfoque generaría modelos de negocio enriquecidos con las *perspectivas Temporal* y la *Organizacional*.
- ii) [Cabanillas et al. 2011a, 2011b, 2014, 2015a, 2015b; Del Rio Ortega et al. 2013a] son trabajos relacionados con la especificación de la *perspectiva Organizacional* de los procesos, poniendo el foco en la asignación de *Recursos Humanos* a actividades. [Cabanillas et al. 2011a y 2015b] utilizan «*Resource Assignment Language (RAL)*» para esta especificación. En la segunda referencia se propone la notación *RALph*. [Cabanillas et al. 2011b] es una propuesta para la especificación y asignación de recursos sobre *BPMN 2.0*.
- iii) En [Del Rio Ortega et al. 2010] se define el concepto «*Process Performance Indicator (PPI)*» como una variante del concepto «*Indicador de rendimiento (KPI)*» [Fitz-Gibbon 1990] para añadir *métricas* a los procesos de negocio. Los *PPIs* se calculan durante la ejecución del proceso mientras que los *KPIs* sirven para medir el cumplimiento de objetivos de la organización. En [Del Rio Ortega

*et al.* 2012] usan *plantillas lingüísticas o patrones* [Gamma *et al.* 1995; Tran *et al.* 2007] para definir textualmente *PPIs*, introduciendo la notación y el *Meta-Modelo PPINOT*. En [Del Rio Ortega *et al.* 2013b] se propone el análisis de *PPIs* utilizando el *Meta-Modelo PPINOT*. [Del Rio Ortega *et al.* 2013a], ya citado en el párrafo anterior, es un trabajo donde se asocian *PPIs* a la *perspectiva de recursos* de los procesos de negocio y se especifican los *PPIs* a partir del archivo de log generado durante la ejecución del *BPMS* de código abierto *Activiti*<sup>38</sup> [Rademakers 2012]. En [Cruz *et al.* 2014] se aplica la notación *PPINOT* a un proceso de negocio definido sobre el servidor de correo corporativo de la *Universidad de Sevilla*.

## 4.6 Minería de procesos

La *minería de procesos* («*process mining*») intenta descubrir, monitorizar y mejorar los procesos existentes mediante la extracción de conocimiento de los *diarios de eventos* (*event logs*) de los sistemas de información.

En [van der Aalst *et al.* 2005] se resalta la relevancia de la *perspectiva de casos* de los procesos para sustentar nuevos enfoques para realizar «*process mining*». En [van der Aalst *et al.* 2007, 2012; van der Aalst 2011, 2013, 2014] se desarrolla este enfoque y se proponen distintos algoritmos para descubrir y verificar modelos de procesos desde el *log de eventos* cuya existencia es habitual en los *BPMS* [Dumas *et al.* 2013]. El algoritmo básico,  *$\alpha$ -algorithm* [van der Aalst 2011], rastrea el *log de eventos*, buscando patrones de flujo de control (ejecución en paralelo, secuencial; puertas inclusivas, exclusivas, etc.) sobre múltiples instancias que puedan llevar a un modelo genérico del proceso. Para procesos complejos y *logs* de gran tamaño se emplean algoritmos que particionan el *log* y utilizan *lógica inductiva* para resolver el descubrimiento de procesos con mayor eficiencia [van der Aalst 2013].

Existen distintos sistemas que realizan «*process mining*», como:

- i) Los sistemas comerciales [van der Aalst 2015] *Disco* (Fluxicon), *Perceptive Process Mining* (Perceptive Software, before Futura Reflect and BPMone by Pallas Athena), *ARIS Process Performance Manager* (Software AG), *Celonis Process Mining* (Celonis GmbH), *ProcessAnalyzer* (QPR), *Interstage Process Discovery* (Fujitsu), *Discovery Analyst* (StereoLOGIC), and *XMAnalyzer* (XMPPro).
- ii) El sistema de código abierto *ProM*<sup>39</sup> que trabaja con el estándar *XES* para *log de eventos* [Verbeek *et al.* 2011; Kalenkova *et al.* 2014; van der Aalst 2015].

Ahora bien, no todos los sistemas orientados a procesos (*PAIS*) [Dumas *et al.* 2005]

<sup>38</sup> <http://activiti.org/>

<sup>39</sup> <http://www.promtools.org/doku.php>



cuentan con *logs de eventos* disponibles para «*process mining*», aunque sí mantienen diarios de la actividad en el sistema. Esto lo han aprovechado autores como:

- i) [Ingvaldsen & Gulla 2008] que trabajan con el *ERP SAP* para construir un log de eventos desde los diarios de transacciones de datos.
- ii) [Günther & van der Aalst 2007] proveen un framework genérico para obtener un log de eventos desde distintos tipos de *PAIS*.

Existen autores como [Pérez-Castillo et al. 2009, 2011a] que trabajan sobre sistemas no orientados a procesos para obtener modelos de procesos. Definen el framework (*Modernization Approach for Recovering Business processes from Legacy Systems* «*MARBLE*»), basado en mejoras de *ADM* para extraer una base de procesos de negocio dentro de su concepto de «*Arqueología de Procesos de Negocio*». Utilizan el camino de ingeniería inversa *ADM* para llegar a propuestas de modelos de procesos basados en *BPMN*. En [Pérez-Castillo et al. 2012a] se aplica *MARBLE* a una familia de casos.

Por último [van der Aalst & Nikolov 2007; van der Aalst 2015] son otros ejemplos de tentativas que parten de sistemas origen no orientados a procesos:

- i) [van der Aalst & Nikolov 2007] es un trabajo que describe *EEmailAnalyzer*, que es un plugin para extraer trazas de procesos de una corriente de correos electrónicos con objeto de alimentar al framework *ProM*. Los autores establecen como hipótesis que los correos originarios estén generados por un sistema automatizado, de modo que contengan datos que permiten clasificarlos (es posible asociarlos a una tarea codificada). De este modo, un conjunto de correos son reutilizados para reconstruir un log de eventos que se utiliza en *ProM*.
- ii) En [van der Aalst 2015] se propone un enfoque que establece una base conceptual para sistemas de información genéricos que se construyen sobre una base de datos. La *base de datos* del sistema refleja el *estado actual de un conjunto de procesos*, mientras que los *cambios* en la misma se producen *mediante eventos*. El autor propone modelos de objetos para representar procesos, *modelos de clases* para representar *modelos de procesos* y un *modelo de eventos* para representar las modificaciones a la base de datos. Sobre esta conceptualización se propone un método para generar un log de eventos que puede alimentar a los métodos genéricos de «*process mining*». El autor concluye que es necesario sustentar este enfoque sobre gestores de bases de datos específicos.

## 4.7 Resumen y conclusiones

En este Capítulo se ha revisado la literatura relacionada con los objetivos de la tesis. El foco del trabajo de investigación es la dimensión temporal de los procesos. En primer lugar se han analizado los lenguajes para la especificación y verificación de reglas temporales sobre procesos. A continuación se han extraído un conjunto de referencias

relevantes en torno a lenguajes, estándares y buenas prácticas para la definición de procesos ligados al software, resaltando las que utilizan *Meta-Modelos*.

El proceso de modernización del software comprende el tratamiento de *legacy systems*, los cuales no son fáciles de mantener ni de sustituir. La mayoría de los métodos, actualmente, están basados en el paradigma *MDE* y contemplan caminos de modernización como: i) La ingeniería inversa que va de niveles bajos de abstracción que empiezan con las plataformas tecnológicas origen del proceso: código y bases de datos, y ii) Ingeniería directa, que permite transformar modelos de alto nivel de abstracción en modelos de menor nivel, hasta llegar al código del nuevo sistema destino. Una posición destacada en este proceso de modernización está ocupada por *OMG ADM* y distintas contribuciones que han extendido este estándar. Las *legacy databases* van a ser el foco de este trabajo de investigación; por esta razón se han revisado un conjunto de trabajos que realizan ingeniería inversa desde esquemas de bases de datos relacionales. Se han analizado también trabajos relacionados con conceptos como la *arqueología de procesos* y «*process mining*».

Los procesos de ingeniería inversa de *legacy databases* suelen capturar entidades y algunas reglas de negocio de carácter estructural, siendo más difícil derivar reglas que estén implementadas como algoritmos más complejos en el código de las aplicaciones. Por otro lado, la minería de procesos lleva a derivar modelos de procesos basándose en heurísticas que analizan múltiples instancias de éstos. Estos modelos de procesos presentan sólo la perspectiva de control del flujo.

El enfoque que se va a desarrollar para derivar procesos propone buscar otras perspectivas de los procesos en las *legacy databases*, poniendo inicialmente el foco en la perspectiva temporal mediante el estudio específico de *legacy systems* de uso frecuente en el sector *TI* y su caracterización de la dimensión temporal. Aunque la perspectiva temporal sea el foco principal de la investigación, se han seleccionado un conjunto de trabajos que tratan otras dimensiones de los procesos, como *la organizacional* y *la de casos*, porque podrían enriquecer los procesos obtenidos en trabajos futuros.

# Capítulo 5. LEGACY SYSTEMS EN EL SECTOR DE LAS TI

*"Es de importancia para quien desee alcanzar una certeza en su investigación, el saber dudar a tiempo"*

*Aristóteles (384-322 A.C.)*

Las «organizaciones del software (TI)» usan distintos tipos de *legacy systems* que, habitualmente, no están diseñados teniendo en cuenta el enfoque *BPM* (no son *PAIS*), aunque, de un modo u otro, contienen artefactos en su *legacy database* (estructuras de datos y reglas) que representan estados de ejecución relacionados con las distintas *perspectivas de los procesos que han sucedido en la organización*. Se han seleccionado un grupo de *legacy systems*, habituales en las «TI», para extraer sus esquemas (*legacy databases*, ver Anexo II) y realizar un estudio comparativo del soporte de cada una de las perspectivas de los procesos, en particular, de la dimensión temporal. Las conclusiones del análisis servirán para seleccionar los sistemas candidatos a ser utilizados en el enfoque de ingeniería inversa *MDE* propuesto en la tesis, que transformará artefactos de *legacy databases* en procesos definidos con lenguajes cercanos al experto en software.

## 5.1 Sistemas de gestión en organizaciones orientadas al software

Cualquier organización dedicada al negocio del software ha utilizado sistemas de información para apoyar el desarrollo de su actividad. Así es común que, entre sus actividades realice planificación y control de sus proyectos; Esto implica estructurar y ordenar las actividades del proyecto, prever fechas de inicio y terminación, estimar duraciones, así como hacer un seguimiento de la ejecución de esas actividades para medir las desviaciones y ajustar y acordar la planificación con los actores implicados. Por otro lado, esa planificación también puede incluir la asignación de recursos (humanos, económicos, etc.), grupos de recursos, etc. La generación, recepción, aprobación, control de versiones y envío de documentos es también un conjunto de tareas muy comunes. Tampoco hay que olvidar que una organización orientada al software, como cualquier otra, necesita sistemas de gestión horizontales (no específicos del software) para llevar a cabo sus relaciones con proveedores, bancos, clientes y empleados, además de las obligaciones contables y fiscales. Desde la década de los 90's ha crecido una tendencia a

utilizar aplicaciones de mercado (fabricadas por distintos niveles de proveedores especializados en software estándar) para adaptarlas a las necesidades de cada organización.

En diversos puntos de esta memoria de tesis, se ha comentado la tendencia de los últimos años a utilizar enfoques *BPM* como ventaja competitiva. Según su ciclo de vida [van der Aalst 2004; Hill et al. 2006; Dumas et al. 2013], utilizar esta tecnología implica comenzar por identificar y definir sus procesos de negocio que habrán de integrarse con los sistemas existentes. Cabe en este momento preguntarse si el experto de negocio debe partir de cero para hacer esta definición mediante un proceso manual o bien puede recabar automáticamente características de estos procesos de los sistemas existentes. En este trabajo de tesis se ha desarrollado el estudio de la dimensión temporal de los procesos y se pretende utilizar la misma para extraerla de determinados tipos de sistemas mediante un enfoque de ingeniería inversa *MDE*.

Se han analizado sistemas de información de amplia cobertura en el mercado y utilizados por las organizaciones del software, llevando a cabo las siguientes tareas:

- i) Análisis de documentación de productos.
- ii) Instalación del software o búsqueda de esquemas *XSD*.
- iii) Importación de esquemas (*ODBC* o *XSD*) a *Enterprise Architect (EA)* [Sparx:EA 2015].
- iv) Extracción de vistas con reglas temporales sobre actividades que soporta cada producto (Ver Anexo II).

### 5.1.1 Sistemas de planificación y control de proyectos

Estos sistemas permiten crear proyectos, compuestos por actividades y subproyectos. Los proyectos constan de un conjunto de tareas que pueden planificarse, asignándole gran parte de las reglas temporales que se han estudiado en la taxonomía de reglas (Figura 6.1), como restricciones sobre tareas o bien como dependencias temporales entre las mismas basadas en «Álgebra de intervalos de Allen» [Allen 1983]. También facilitan la medición del grado de avance o terminación de cada tarea, el establecimiento de líneas de base para realizar comparaciones, la definición de recursos, grupos de recursos y la asignación de tareas a recursos. En cualquier momento de un proyecto activo, pueden realizarse planificaciones o re-planificaciones del proyecto que tienen en cuenta las actividades finalizadas, las activas, las no iniciadas y el nivel de asignación de recursos, tipo de restricciones o dependencias, etc., aplicándose para el cálculo de duraciones flexibles o fechas de terminación flexibles el método «*Critical Path Method (CPM)*» [Kelley & Walker 1959].

Los sistemas más sencillos son herramientas unipersonales, como *MS Project* [Hansen & Hansen 2013] (Anexo II.a), pero en organizaciones más complejas es necesario contar con sistemas colaborativos, soportados por BD relacionales y tecnología web, como *MS Project Server* [Stover 2007] (Anexo II.b) que permite la integración de

MS Project sobre un SGBD SQL\*Server™ [Colledge 2008; Ben-Gan 2012; Microsoft™:T-SQL 2014] y RedMine™ [Lang 2010] (Anexo II.d), sobre mySQL™ [Harrison & Feuerstein 2008] y servidores web Apache™ y Tomcat™. Redmine también facilita la gestión de peticiones de usuarios o incidencias (se usa comúnmente en Centros de Atención a Usuarios).

Estos sistemas tratan la *dimensión temporal* y la *dimensión organizacional* (recursos) que son ejes fundamentales planteados en el «*Process Mining Manifesto*» [van der Aalst et al. 2012]. De hecho, existen trabajos, como [Flores & Sepúlveda 2011; Gagné & Trudel 2009b], que tratan de establecer equivalencias entre la representación de tareas en estos sistemas y el mundo de los procesos de negocio.

En los Anexos II.a, II.b y II.d se han extraído vistas ERD de las bases de datos de cada sistema, buscando las entidades relacionadas con los procesos, tareas y la especificación de la dimensión temporal.

### 5.1.2 Sistemas de gestión documental (ECMs)

Los sistemas de gestión documental o gestores de contenido «*Enterprise Content Management Systems (ECMs)*» suelen configurarse como un entorno colaborativo, hoy basado en tecnología Web y en potentes bases de datos relacionales o documentales que permiten expresar la vista organizacional y gestionar la publicación, intercambio, aprobación y gestión de versiones de documentos. Suelen integrar un cierto nivel de gestión simplificada de flujos de trabajo «*Workflows*» o bien la integración con gestores de procesos. Luego, sin duda, también incluyen comportamiento común con la perspectiva del tiempo y organizacional *BPM*. A esta categoría de sistemas pertenecen «*Alfresco™*» [Shariff 2013] o «*MS Sharepoint™*» [Smith & Bates 2007]. *Alfresco* utiliza como SGBD *PostgreSQL™* y *MS Sharepoint SQL\*Server™* [Colledge 2008; Ben-Gan 2012; Microsoft™:T-SQL 2014]. Se ha extraído una vista *ERD* del esquema relacional de *Alfresco* (ver Anexo II.c). *Alfresco* maneja una serie de workflows predefinidos (secuencial, paralelo) para los circuitos de aprobación de documentos, pero permite la integración con *Activity* para diseñar flujos más complejos.

### 5.1.3 ERPs, CRMs, SCMs y desarrollos a medida

Estos sistemas [Hendricks et al. 2007] integran diversos módulos o aplicaciones para la gestión del negocio en un núcleo estándar, sobre el que se superponen capas para localizaciones nacionales con los aspectos normativos o legales de cada país. También suelen incluir facilidades para parametrizar y/o adaptar sus interfaces o generar código que se integre en el estándar. De este modo pueden llegar a satisfacer gran parte de los requisitos de las organizaciones y hacen más fácil la expansión a otras líneas de negocio y/o geográfica. Los *ERPs* suelen contar con módulos contables, de recursos humanos, almacenes, compras, ventas, gestión de la producción, etc. Los *CRMs* son módulos especializados en la gestión de relaciones con clientes (Marketing, fichas de clientes, ofertas, contratos, ventas, facturación, cobro y gestión de incidencias). Los *SCMs*

involucran todos los procesos logísticos, internos o vinculados a colaboradores externos para gestionar la cadena de suministro. Aunque cada vez es menos común, no hay que olvidar el desarrollo de aplicaciones específicas a medida del cliente, en casos en que las adaptaciones del software estándar son muy costosas, o no existe funcionalidad suficiente en el estándar, o bien en nuevas líneas de negocio que no están suficientemente bien abordadas por esos paquetes.

Aunque estos sistemas no tengan modelos de procesos transparentes a usuarios y expertos de negocio, se pueden encontrar dimensiones *BPM* en la capa de persistencia de cada uno. Ahora bien, al igual que en casos anteriores, sería necesario extraer las vistas de estos procesos vinculadas a la dimensión *BPM* que se pretenda contrastar.

## 5.2 Perspectivas de procesos vs legacy systems

Tras el estudio de los diversos tipos de sistemas utilizados por las organizaciones del software, se observa que cada tipo de sistema soporta en mayor o menor medida las perspectivas *BPM*: *temporal*, *organizacional* y *casos*; se excluyen otras perspectivas, como la de *flujo de control*, *información* y *operacional* que suelen ser específicas de los PAIS [Dumas et al. 2005]. La [Tabla 5.1](#) puede ilustrar la existencia de estructuras que soportan reglas de negocio asociadas a cada perspectiva *BPM* según el tipo de *legacy system*:

**Tabla. 5.1.** Perspectivas BPM soportadas por tipo de legacy system

Tipo de legacy system	Perspectiva BPM		
	Tiempo	Organizacional (Recursos, información)	Casos
i) Sistemas de planificación y control de proyectos <i>p. ej.: MS Project, RedMine</i>	▲	△	⊗
ii) Enterprise Content Management Systems (ECMs) <i>p. ej. Alfresco, SharePoint</i>	△	▲	△
iii) ERPs, CRMs, SCMs (workflows de sus procesos)	△	△	△
iv) Desarrollos a medida	△	△	△

▲ Bien soportado, △ Soportado con limitaciones, ⊗ No soportado.

- i) Los sistemas de planificación de proyectos suelen recoger bien la dimensión temporal y una aproximación al tratamiento de la vista organizacional con recursos y grupos de recursos, pero no están enfocados para establecer jerarquías, departamentos, empresas, holdings, etc. La especificación de reglas para casos es manual.
- ii) Los *ECMs* suelen manejar bien la perspectiva organizacional y contienen algunas reglas temporales básicas, así como la agrupación y parametrización del tratamiento de casos que obedecen a patrones de gestión, pero bajo una

- especificación muy simple de reglas.
- iii) El grupo de *ERPs*, *CRMs*, *SCMs* soportan reglas temporales (vencimientos, fechas tope o límite), facilitan la definición de la estructura organizacional (con modelos robustos que permiten definir holdings, empresas, departamentos, roles, responsabilidades, etc.) y también el tratamiento de casos mediante reglas de parametrización. Los modelos de datos de los sistemas de mayor alcance internacional son excesivamente complejos.
  - iv) Los desarrollos a medida pueden soportar cualquier perspectiva, pero normalmente los analistas enfocan más los modelos a la funcionalidad que a tratar de manera estándar las distintas perspectivas BPM de un proceso. El otro inconveniente importante estriba en el nivel de actualización de la documentación para poder extraer artefactos ligados a los modelos de procesos.

### 5.3 Dimensión temporal en un legacy system

En relación al análisis específico de la dimensión temporal, las Tablas 5.2 y 5.3 muestran la comparación de estos sistemas respecto a la taxonomía de reglas (Figura 6.1).

**Tabla. 5.2.** Reglas temporales soportadas por tipo de legacy system

Tipo de legacy system	Tipo de regla temporal														
	Duración	Cardinalidad	Ausencia	Inflexible	Flexible										
	FIXD	FLEXD	CARD	ABS-U	ABS-C	MSON	MFON	FALAP	FASAP	FNET	FNLT	SALAP	SASAP	SNET	SNLT
i) Sistemas de planificación y control de proyectos <i>p. ej.: MS Project, RedMine</i>	▲	△	⊗	⊗	⊗	▲	▲	▲	▲	▲	▲	▲	▲	▲	▲
ii) Enterprise Content Management Systems (ECMs) <i>p. ej. Alfresco, SharePoint</i>	▲	△	⊗	⊗	⊗	▲	▲	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
iii) ERPs, CRMs, SCMs (workflows de sus procesos)	▲	△	⊗	⊗	⊗	▲	▲	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗

▲ Bien soportado, △ Soportado con limitaciones, ⊗ No soportado.

Los sistemas de planificación y control son los que mejor soportan la dimensión temporal, mientras que el resto de los sistemas soportan mejor otras perspectivas (Tabla 6.1). Por esta razón son los primeros en ser elegidos para aplicar un enfoque de ingeniería inversa *MDE* que extraiga modelos de procesos, aunque, posteriormente pueden utilizarse otros sistemas y añadir también otras perspectivas que enriquezcan los resultados.

**Tabla. 5.3.** Dependencias temporales soportadas por tipo de legacy system

Tipo de legacy system	Dependencia Temporal					
	FF	FS	SF	SS	ABS-U	ABS-C
i) Sistemas de planificación y control de proyectos <i>p. ej.: MS Project, RedMine</i>	▲	▲	▲	▲	⊙	⊙
ii) Enterprise Content Management Systems (ECMs) <i>p. ej.: Alfresco, SharePoint</i>	⊙	▲	⊙	▲	⊙	⊙
iii) ERPs, CRMs, SCMs (workflows de sus procesos)	⊙	▲	⊙	▲	⊙	⊙

▲ Bien soportado, Δ Soportado con limitaciones, ⊙ No soportado.

## 5.4 Resumen y conclusiones

En este Capítulo se han analizado diversos tipos de legacy systems habituales en las «TI»: planificación y control de proyectos, gestores documentales o de contenido (ECMs), ERPs, CRMs, SCMs y, por último, desarrollos a medida. Se ha estudiado en primer lugar el soporte de las distintas perspectivas de los procesos; en segundo lugar, utilizando la taxonomía de reglas (Figura 6.1), se han contrastado los tipos de reglas que soportan estos legacy systems. La conclusión es que los legacy systems que mejor soportan la perspectiva temporal son los sistemas para la planificación y control de proyectos (p. ej.: MS Project o Redmine) y, en segundo lugar, los ECMs. Este resultado, el extendido uso en el mercado y nuestra experiencia en proyectos invita a seleccionar en primer lugar MS Project como legacy system para aplicar el enfoque de ingeniería inversa MDE que se desarrolla en el trabajo de tesis.



## **PARTE III. SOLUCIÓN PROPUESTA**



# Capítulo 6. REGLAS TEMPORALES SOBRE PROCESOS

*“Por modelo, se entiende una construcción matemática que, con la adición de ciertas interpretaciones verbales, describe los fenómenos observados. La justificación de tal construcción matemática es sólo y precisamente que se espera que funcione”*

*John von Neumann (1903-1957)*

El foco del trabajo de tesis es la utilización de la *perspectiva temporal* para obtener procesos del software desde *legacy databases*. Respecto a estándares, buenas prácticas y trabajos de investigación sobre procesos del software, un grupo de ellos tienen en común estar basados en robustos *Meta-Modelos*, como el popular *BPMN* para todo tipo de procesos y *SPEM 2.0* e *Iso/Iec:24744* para los procesos específicos del software. No obstante, algunos investigadores coinciden en apreciar cierta debilidad en ellos para especificar reglas temporales, razón por la cual existen otros lenguajes y también propuestas para extender la dimensión temporal. Se va a desarrollar una propuesta que ahonda en la relevancia de la dimensión temporal, plantea una taxonomía de reglas y establece un *Meta-Modelo* mínimo de procesos mediante el lenguaje de especificación *OCL*, que se utilizará como base para realizar distintas extensiones necesarias en la investigación.

## 6.1 Perspectiva temporal en los procesos de negocio

El tiempo está presente en cualquier faceta de la vida del ser humano. El campo de los procesos de negocio está influenciado por una amplia gama de limitaciones temporales que provienen de prescripciones jurídicas o normativas y también de normas de gestión internas y prácticas en el mercado. En las relaciones entre organizaciones, los plazos de entrega y las demoras que puedan producirse influyen directamente en aspectos de la contratación: caducidades, penalizaciones, seguros, devoluciones, etc. El tiempo está íntimamente ligado con la calidad del servicio: i) responder a tiempo (entregar una oferta, entregar el bien o servicio en plazo, atender reclamaciones pronto, preavisar al cliente) significa aumentar la calidad del servicio y, por tanto, aumentar el grado de satisfacción del cliente; ii) responder tarde o la falta de respuesta puede provocar: caducidades o el desecho del producto, incurrir en aumento de costos (almacenaje,

transporte, seguros, etc.) y llegar a romper la relación comercial.

En las relaciones con la Administración, entre Administraciones o entre Administración y ciudadanos, cualquier procedimiento administrativo (*proceso*) está compuesto por trámites (*actividades*) con plazos o fechas de finalización y flujos de decisión condicionados por las acciones de las partes. Cumplir con los plazos implica ejecutar los flujos predefinidos; transgredir estos plazos supone entrar en flujos de excepción que pueden implicar: inspecciones, sanciones y perjuicios mayores, ya que la Administración tiene relaciones con el ciudadano en múltiples servicios públicos: Educación, Sanidad, Tributación, Tráfico, Procesos Judiciales, etc.

Si hablamos del campo específico del negocio del software, encontramos que la forma típica de organizar el trabajo es mediante proyectos. Si el proyecto implica al cliente con un proveedor o proveedores software, la formalización de estas relaciones obliga a formular contratos, con hitos temporales que, en la realidad, son interdependientes y difíciles de gestionar y controlar, exigiendo métodos y herramientas automatizadas para la gestión de proyectos. Entre todas estas actividades surge también una variedad de reglas temporales que pueden implicar penalizaciones o retrasos indeseables en los proyectos [Lauesen & Vinter 2001], con aplicación de penalizaciones contractuales y, que en último caso pueden llevar a cancelar [Standish 2012] los proyectos.

Cualquier proceso de negocio, compuesto por actividades y flujos de control lleva ya implícita la dimensión temporal. Los caminos de este flujo condicionarán tiempos de espera de actividades, duraciones y la duración del proceso global. Pero pueden existir reglas más complejas. El *manifiesto BPM* [van der Aalst et al. 2012] contempla la *perspectiva temporal*, entre otras como la *perspectiva de control*, la *organizacional* y la *de casos* [van der Aalst et al. 2005, 2007]. Existen múltiples enfoques para modelar reglas temporales sobre procesos (ver Sección 3.1); en [Cheikhrouhou et al. 2013a] se realiza una comparación sobre distintas aproximaciones para especificar la dimensión temporal y aplicar métodos de verificación de este tipo de reglas.

Obviamente, es fundamental tener en cuenta esta dimensión, y definir la tipología de reglas que se considerarán en la propuesta de este trabajo de tesis (Sección 5.2 siguiente).

## 6.2 Taxonomía de reglas temporales

El concepto de regla de negocio está bien definido en [Ross 2013; Baisley 2005], aunque hay múltiples clasificaciones de reglas; en [Arévalo et al. 2014] se analizan dos clasificaciones de reglas propuestas por a) [Jablonski & Bussler 1996], entre las cuales aparecen las reglas de tiempo y b) [Wagner 2003, 2005] que ofrece una subdivisión más cercana a la tecnología.

Nos centraremos en reglas temporales sobre *actividades* pertenecientes a *procesos de negocio* (Una *actividad* puede ser una *tarea simple*, un *proceso* o un *subproceso*.) que

se ejecutan *en la misma organización*, por lo que quedan excluidas algunas reglas como las ligadas a procesos «IOBP» [Attaran 2004; Cheikhrouhou 2013a, 2014b] que podrían ser tratadas en futuros trabajos. La Figura 6.1 [Arevalo et al. 2015b] presenta una taxonomía de reglas con la que se trabajará en adelante.

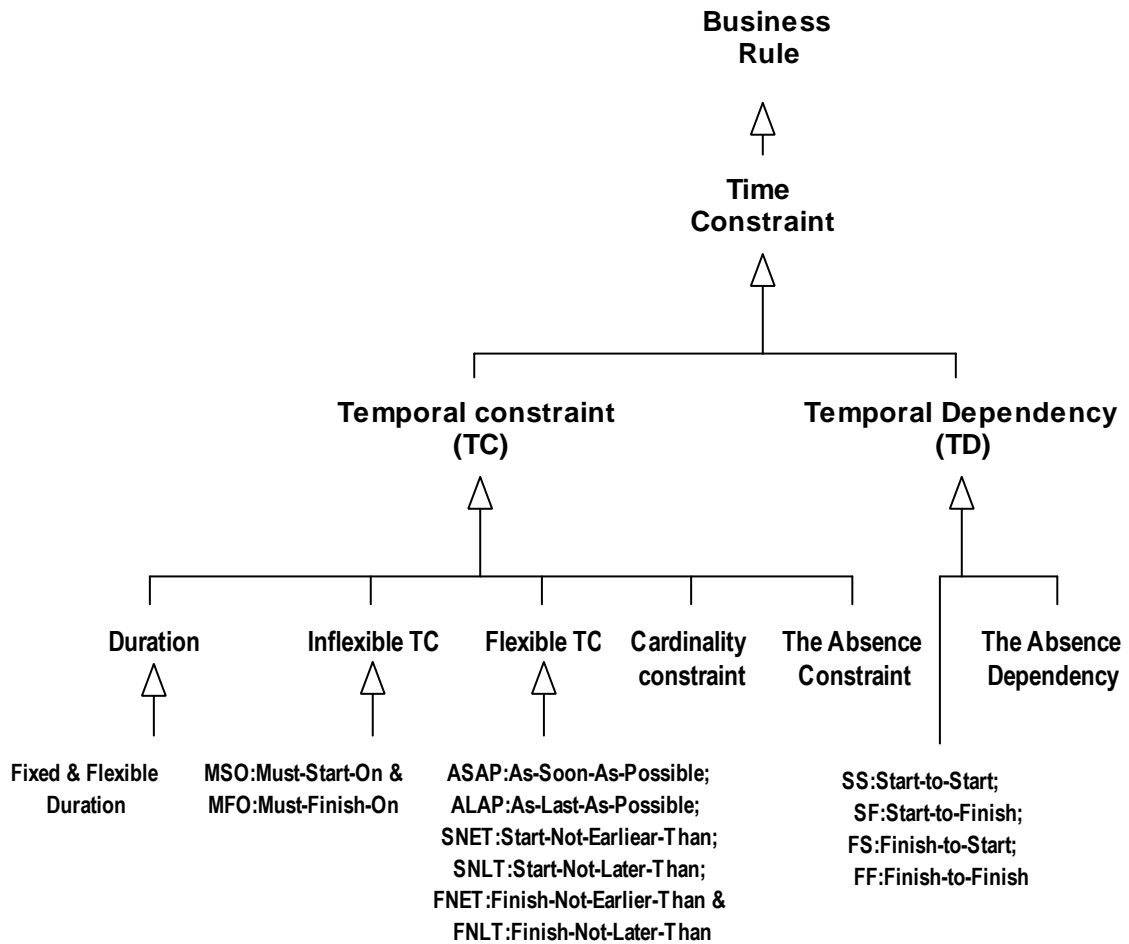


Figura. 6.1. Taxonomía de reglas temporales

Pueden distinguirse: *i) Restricciones temporales sobre actividades* y *ii) Dependencias temporales* que restringen las relaciones entre dos *actividades*. Se describen estas reglas en los siguientes apartados.

### 6.2.1 Restricciones temporales

Una restricción temporal expresa reglas temporales sobre una actividad individual; pueden ser de los siguientes tipos:

- i) **Duración.** Cada actividad  $A$  perteneciente a un proceso  $P$  tiene un evento de comienzo  $s(A)$  y otro de finalización  $e(A)$ . La duración de una actividad ( $dur(A) \geq 0$ ) puede ser fija o bien variable dentro de un intervalo (entre una duración mínima y máxima). Cuando la duración es fija, se satisface la ecuación (expresión 6.1), mientras que en el caso de duración flexible debe ser reformulada

en términos de la ecuación (expresión 6.2):

$$[e(A) \geq s(A)] \Rightarrow dur(A) = e(A) - s(A)$$

Expresión. 6.1. Duración de una actividad perteneciente a un proceso

$$[e(A) \geq s(A)] \Rightarrow \min Dur(A) \leq (e(A) - s(A)) \leq \max Dur(A)$$

Expresión. 6.2. Duración flexible de una actividad de un proceso

ii) **Restricciones de tiempo inflexibles.** Fijan el inicio o terminación de la actividad:

a) *Must Start On (MSON):*  $s(A) = mson$

b) *Must Finish On (MFON):*  $e(A) = mfon$

iii) **Restricciones de tiempo flexibles.** Dejan abierto el inicio o terminación de la actividad; así aparecen: *As Soon As Possible (ASAP)* que se desdobla en *Start As Soon As Possible SASAP* y *Finish As Soon As Possible (FASAP)*, y por otro lado *As Last As Possible (ALAP)* que se desdobla en *Start As Last As Possible (SALAP)* y *Finish As Last As Possible (FALAP)*. Dentro de esta categoría también pueden encuadrarse restricciones que utilizan una fecha de referencia para delimitar bien el inicio o terminación: *Not Earlier Than (NET)* que incluye *Start Not Earlier Than (SNET)* y *Finish Not Earlier Than (FNET)* y *(NLT)* subdividida en *Start Not Later Than (SNLT)* y *Finish Not Later Than (FNLT)*.

Estas restricciones están íntimamente relacionadas con el método «*Critical Path Method (CPM)*» [Kelley & Walker 1959], que es un algoritmo matemático que calcula la duración mínima de un grafo de actividades, donde cada actividad es un nodo y los arcos representan sus relaciones de precedencia, orientados de la relación predecesora a la actividad sucesora que está condicionada por la anterior con un tipo de interrelación determinada. *CPM* define un subgrafo compuesto por las actividades críticas denominado el «*camino crítico*». Las actividades que pertenecen al camino crítico no tienen ninguna «*holgura*» en sus eventos de inicio y terminación sin alterar la duración mínima de todo el grafo. Por el contrario, el resto de actividades tienen cierta *holgura* en sus eventos de inicio o terminación, de este modo un gestor o planificador de tareas que se base en el algoritmo *CPM* tiene capacidad para mover los eventos de estas actividades, realizándose un análisis de sensibilidad hasta que se calcula la duración mínima de todo el grafo que, como se ha expresado, coincide con la duración del camino crítico.

Por tanto, si  $P\{A_i\}$  es un proceso compuesto por las actividades  $A_i$ , entonces a  $P$  le corresponde un grafo  $G$  y debe satisfacerse la ecuación siguiente (expresión 6.3):

$$G \equiv \{A_i\}; G' \equiv \{A'_i\}; G' \subseteq G;$$

$$dur(P) \geq \min(dur(G)) = dur(G') = \sum_{A'_i \in G'} dur(A'_i)$$

Expresión. 6.3. Restricciones temporales de inicio y terminación flexibles

Cada actividad tiene una duración  $dur(A_i)$  y  $G'$  representa el subgrafo de actividades críticas  $A'_i$  con una duración  $dur(G')$  como la duración mínima para completar ese subgrafo y la del grafo completo. En los casos *NET* y *NLT* el usuario o modelador pueden fijar una fecha de referencia como cota (fecha *NET* o fecha *NLT*), mientras que en los casos *ASAP* y *ALAP* no existe esta cota. En ambos, el planificador *CPM* tiene capacidad para mover los eventos de inicio y terminación de las actividades con holgura para buscar el camino o crítico de mínima duración. La **Figura 6.2** muestra los rangos de variación para las cotas inferiores y superiores que *CPM* maneja para mover los eventos de inicio  $s(A)$  y terminación  $e(A)$  de una actividad del proyecto  $P$ , cuya duración es  $dur(P)=(e(P)-s(P))$ , considerando  $[e_o, s_o]$  como el intervalo de fechas para establecer el correspondiente evento, y estando fijado para los casos *NET* y *NLT* y libre entre las cotas  $[min(e(A), max(e(A))]$ , de modo que la duración del camino crítico no se altere. La **Tabla 6.1** especifica el método de fijación de cotas y de los eventos.

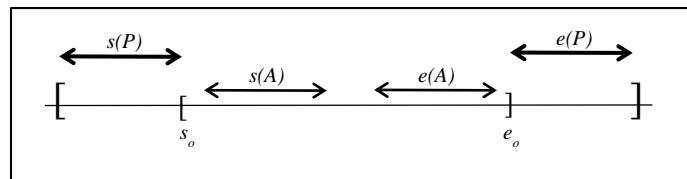


Figura. 6.2. Análisis de sensibilidad de eventos al aplicar CPM

**Tabla. 6.1.** Cotas de eventos para restricciones de inicio y terminación flexibles

Cotas de eventos	Método de cálculo	TipoRestricción
Cotas inferiores y superiores de eventos	Fijadas por Modelador y calculadas por CPM	<i>SNET</i> ( $s_o = s_{net}$ ) $\wedge$ ( $e_o \leftarrow CPM$ )
		<i>SNLT</i> ( $s_o \leftarrow CPM$ ) $\wedge$ ( $e_o = s_{nlt}$ )
		<i>FNET</i> ( $s_o = f_{net}$ ) $\wedge$ ( $e_o \leftarrow CPM$ )
		<i>FNLT</i> ( $s_o \leftarrow CPM$ ) $\wedge$ ( $e_o = f_{nlt}$ )
	Calculadas por CPM	<i>SASAP</i> ( $s_o = \min(s(A))$ ) $\wedge$ ( $e_o \leftarrow CPM$ )
		<i>SALAP</i> ( $s_o \leftarrow CPM$ ) $\wedge$ ( $e_o = \max(s(A))$ )
		<i>FASAP</i> ( $s_o = \min(e(A))$ ) $\wedge$ ( $e_o \leftarrow CPM$ )
		<i>FALAP</i> ( $s_o \leftarrow CPM$ ) $\wedge$ ( $e_o = \max(e(A))$ )

En lo concerniente a *NET* y *NLT* es el modelador el que establece estas cotas, así  $[(s_o = s_{net})$  y  $(s_o = f_{net})]$  o  $[(e_o = s_{nlt})$  y  $(e_o = f_{nlt})]$  indica que se han fijado las cotas inferiores o superiores para los eventos de inicio o terminación según corresponda. Las anotaciones  $(e_o \leftarrow CPM)$  o  $(s_o \leftarrow CPM)$  indican que es el algoritmo quien mueve estas cotas en las iteraciones; en los casos de restricciones flexibles *ASAP* y *ALAP*, el planificador relaja el valor de los dos eventos: inicio y finalización dentro de sus holguras.

- iv) **Restricción de cardinalidad.** Limita el número de iteraciones de una actividad en bucle en relación a su duración. Si  $N$  es el número de veces que una actividad  $A\{A_i\}$  se ejecuta, siendo  $\{A_i\}$  el conjunto de instancias de ejecución de la actividad  $A$ . Se restringe el tamaño del bucle que puede ejecutarse según la [expresión 6.4](#):

$$e(A) - s(A) = dur(A) = \sum_{i=1}^N (e(A_i) - s(A_i))$$

$$\min Dur(A) \leq \left( \sum_{i=1}^N (e(A_i) - s(A_i)) \right) \leq \max Dur(A)$$

Expresión. 6.4. Restricción temporal de cardinalidad

- v) **Restricción de ausencia.** Establece que una actividad no puede ser ejecutada, lo que significa que se evita la ejecución de dicha actividad, bien en permanentemente (*incondicional*:  $ABS-U$ ; [Expresión 6.5a](#)) o bien habilitando su ejecución sólo fuera de un intervalo de tiempo  $[t_1, t_2]$  (*condicionada*:  $ABS-C$ ; [Expresión 6.5b](#)), lo que equivale a saltar o pasar por alto la ejecución de dicha actividad dentro de dicho intervalo:

$$(a) : \forall t \Rightarrow A \text{ is in Absence}$$

$$(b) : \forall t \in [t_1, t_2] \Rightarrow A \text{ is in Absence}$$

Expresión. 6.5. Restricción de ausencia de una actividad

### 6.2.2 Dependencias temporales

Las dependencias temporales suponen relaciones de precedencia entre una actividad predecesora y una actividad sucesora, especificándose el tipo de relación; así:

- i) **Dependencia de ausencia.** Establece que la actividad sucesora no puede ser ejecutada si la predecesora está activa o ha sido ejecutada. Al igual que en la ausencia ligada a una actividad del bloque anterior, esta restricción puede ser permanente, o bien ligada a un intervalo de tiempo  $[t_1, t_2]$  donde la sucesora puede tomar control para realizar su trabajo y fuera de este intervalo está desactivada.

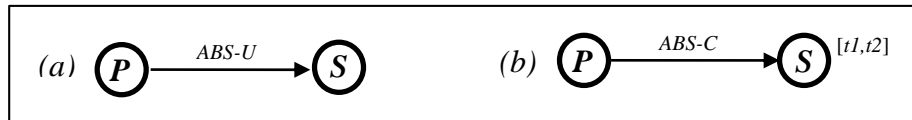


Figura. 6.3. Dependencia de ausencia entre actividades

Si  $P$  y  $S$  son dos actividades,  $P$  la predecesora y  $S$  la sucesora ([Figura 6.3](#) y [Expresión 6.6](#)), las dependencias (*incondicional*:  $ABS-U$ ; [Expresión 6.6a](#)) y (*condicionada*:  $ABS-C$ ; [Expresión 6.6b](#)) expresan dependencias de ausencia para  $S$ :



$$\begin{aligned}
 (a) & ABS - U : \forall t \wedge ((s(P) \neq null) \vee (e(S) \neq null)) \Rightarrow A \text{ is in Absence} \\
 (b) & ABS - C : \forall t \in [t_1, t_2] \wedge ((s(P) \neq null) \vee (e(S) \neq null)) \Rightarrow A \text{ is in Absence}
 \end{aligned}$$

Expresión. 6.6. Dependencia de ausencia entre actividades

ii) **Start to Start (SS), Start to Finish (SF), Finish to Start (FS) y Finish to Finish (FF)**. Son dependencias temporales del «Álgebra de intervalos de Allen» [Allen 1983], estableciendo restricciones entre los eventos de inicio y terminación de la actividad sucesora (S) y sus predecesoras (P) (Figuras 6.4a, b, c y d), teniendo en cuenta un lapso de tiempo  $\Delta t$  que puede significar un adelanto ( $\Delta t < 0$ ) o retraso ( $\Delta t > 0$ ) entre los eventos implicados (Expresiones 6.7a, b, c y d).

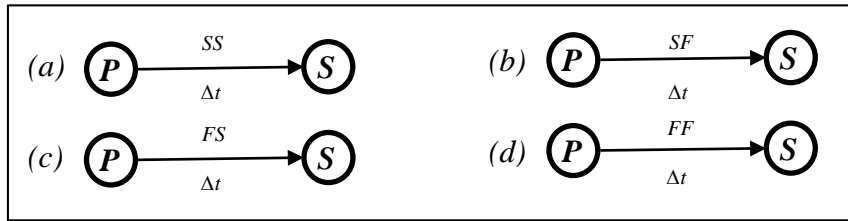


Figura. 6.4. Dependencias temporales SS, SF, FS, FF

$$\begin{aligned}
 (a) & SS : s(S) \geq s(P) + \Delta t & (b) & SF : e(S) \geq s(P) + \Delta t \\
 (c) & FS : s(S) \geq e(P) + \Delta t & (d) & FF : e(S) \geq e(P) + \Delta t
 \end{aligned}$$

Expresión. 6.7. Dependencias temporales SS, SF, FS, FF

### 6.3 Enfoques para la extensión de BPMN 2.0 con reglas temporales

Aunque *BPMN 2.0* contempla la perspectiva temporal, diversos autores (ver Capítulo 4) coinciden en que este estándar muestra algunas debilidades para contemplar determinadas reglas (taxonomía de la Figura 6.1) temporales. A continuación se plantea una sección que revisa capacidades y limitaciones del estándar respecto a esta perspectiva y, en las siguientes, se analizan dos alternativas que proponen enfoques para dotarlo con mayor capacidad expresiva, basados en la utilización de nuevos decoradores que ofrecen semántica adicional y mediante patrones o construcciones [Gamma et al. 1995; Tran et al. 2007] a base de artefactos *BPMN*.

#### 6.3.1 Capacidades y limitaciones de BPMN 2.0 para representar la perspectiva temporal

*BPMN 2.0* permite definir la dimensión temporal. En este sentido, un proceso está compuesto de artefactos, donde la actividad es el principal de ellos, representando un estado y un trabajo a realizar en un tiempo finito. Existen flujos de control, flujos de información y eventos que dependen del espacio temporal, induciendo órdenes o relaciones de precedencia entre artefactos (actividades, objetos o almacenes de datos, etc.). Muchos investigadores están de acuerdo en que, a pesar de estas capacidades,

*BPMN* manifiesta ciertas debilidades para expresar la dimensión temporal (ver Capítulo 4). Es importante hacer notar que *BPMN* no soporta todas las clases de reglas temporales incluidas en la taxonomía de reglas (Figura 6.1).

En lo concerniente al análisis del ciclo de vida de una actividad en *BPMN*, la Figura 6.5 muestra una máquina de estados *UML* [OMG:UML 2011] simplificada y extraída de la versión original incluida en el estándar [OMG:BPMN 2013]. En [Weske 2012] se profundiza en el análisis de este ciclo de vida, contemplando más estados y transacciones que no están en el ciclo contemplado en *BPMN*; así, aparecen: a) “Desactivada” y las transiciones correspondientes con el estado “Lista”, b) “Suspendida” y transiciones entre el estado “Activa” y c) “Ignorada” para no ejecutar la actividad cuando le llega el *token* del flujo de control, evitando la ejecución sin hacer el trabajo correspondiente.

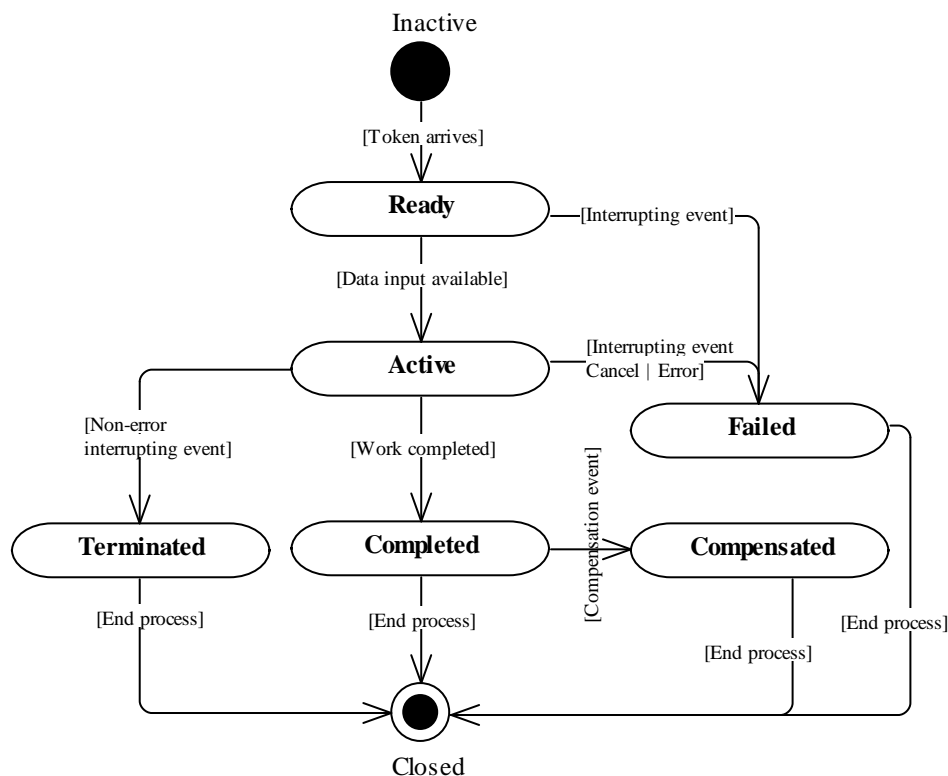


Figura. 6.5. Ciclo de vida de una actividad BPMN

En [Svatos 2012] se muestra que los lenguajes de modelación de procesos contemporáneos, entre los que se incluye *BPMN*, sólo cubren parcialmente el ciclo de vida general de una actividad perteneciente a un proceso de negocio. Esta podría ser una de las razones principales de la debilidad del estándar a la hora de especificar el tiempo. Otra razón de peso es que las relaciones de precedencia que se consideran en el «Álgebra de intervalos de Allen» [Allen 1983] tampoco están soportadas directamente. Además, tampoco existen mecanismos simples para capturar el evento de comienzo de la actividad y controlar el trabajo a realizar después del mismo. No obstante, *BPMN 2.0* introduce el concepto de «eventos sin-interrupción (*non-interrupting-events*)» ligados a actividades,

que permiten modelar nuevos flujos de control que parten de la actividad mientras aún está activa, ayudando así a diseñar nuevos patrones [Gamma et al. 1995; Tran et al. 2007] o construcciones.

El estándar *BPMN* [OMG:BPMN 2013; Iso/Iec:19510 2013] se ha convertido en los últimos años en una de las tecnologías preferida por la mayoría de expertos [Bonnet et al. 2014] para la definición de modelos de procesos, porque ofrece un lenguaje sencillo y como un soporte para llegar a especificaciones ejecutables de procesos. La notación gráfica [Lu & Sadiq 2007] de *BPMN* otorga a los expertos del dominio (consultores y usuarios finales) mayor facilidad para la intercomunicación con expertos en TIC.

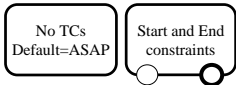
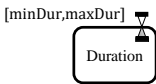

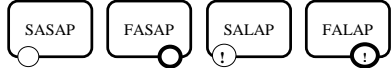

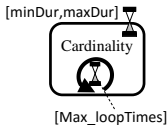
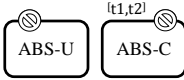
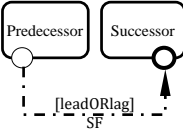
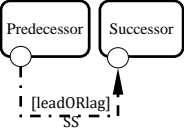
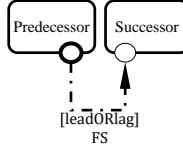
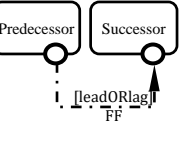
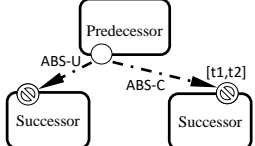
*BPMN* es principalmente un «lenguaje imperativo», mientras que los «lenguajes declarativos», como *SBVR* y *OCL*, solo proponen las características esenciales que restringen la ejecución de actividades dentro de un proceso de negocio, facilitando la comprensión de la especificación y el mantenimiento de las reglas de negocio [Fahland 2009a, 2009b; Pichler et al. 2011; Reijers et al. 2013]. Por estas razones, desde el punto de vista de la especificación de reglas, creemos que es mejor superponer una especificación de reglas de tiempo que utilizar patrones [Gamma et al. 1995; Tran et al. 2007] o construcciones *BPMN* para expresar estas reglas como agrupaciones de artefactos, ya que esta orientación puede sobrecargar en exceso los modelos y/o hacerlos muy dependientes y difícil de mantener ante cambios en las reglas.

A continuación se dedican dos secciones a comparar estos enfoques.

### 6.3.2 Extensiones a *BPMN* mediante decoradores

Los trabajos *Time-BPMN* de [Gagné & Trudel 2009a] y los de [Cheikhrouhou et al. 2013b] proponen nuevos decoradores para plasmar sobre *BPMN 2.0* la semántica de reglas temporales que no soporta directamente el estándar. La segunda propuesta es un poco más completa, al contemplar duración flexible, restricciones de cardinalidad y las de ausencia sobre una actividad o en dependencias entre actividades. La Tabla 6.2 ilustra la aplicación de estos decoradores y compara las dos propuestas:

**Tabla. 6.2.** Extensión temporal a BPMN 2.0 mediante nuevos decoradores

Tipo	Regla		T <sup>a</sup>	C <sup>b</sup>
TC, TD		(a) TC y TD sobre eventos inicio y fin	✓	✓
TC		(b) Duración flexible		✓
TC		(c) Inicio y terminación inflexibles (fijos)	✓	✓
TC		(d) Inicio y terminación flexibles (ASAP, ALAP)	✓	✓
TC		(NET, NLT)	✓	✓
TC		(e) Restricción de cardinalidad		✓
TC	<p>(f) Ausencia incondicional (ABS-U)</p> 	(g) Ausencia condicionada (ABS-C)		✓
TD	<p>(h) Start to Finish</p>  <p>(i) Start to Start</p> 		✓	✓
TD	<p>(j) Finish to Start</p>  <p>(k) Finish to Finish</p> 		✓	✓
TD	<p>(l) Ausencia incondicional (ABS-U)</p> 	(m) Ausencia condicionada (ABS-C)		✓

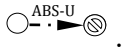
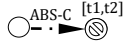
<sup>a</sup> Time-BPMN [Gagné & Trudel 2009a]

<sup>b</sup> [Cheikhrouhou et al. 2013b]

- a) **TC y TD sobre eventos de inicio y fin.** Se especifica una actividad sin restricciones ni dependencias y los nuevos decoradores para asociar restricciones ligadas a los eventos de inicio ( $\circ$  círculo con trazo fino) y terminación de la actividad ( $\bullet$  círculo con trazo grueso).
- b) **TC (Duración flexible).** El decorador reloj de arena ( $\mathbb{V}$ ) denota la duración flexible de la actividad en el intervalo  $[minDur, maxDur]$ .
- c) **TC (Inicio y terminación inflexibles).** Permite la especificación de inicio ( $\odot$  *MSON*) y terminación fija ( $\ominus$  *MFON*). Se debe asociar el instante de referencia (vencimiento) a cada evento.
- d) **TC (Inicio y terminación flexibles).** Permite la especificación de los cuatro tipos de restricciones *ASAP* ( $\circ$  *SASAP*,  $\bullet$  *FASAP*) y *ALAP* ( $\odot$  *SALAP*,  $\ominus$  *FALAP*) que maneja el planificador de tareas.

**TC (NET, NLT).** Aunque estos inicios y terminaciones también son flexibles, se establece una cota inferior o superior para cada evento; así, pueden especificarse las restricciones *NET* ( $\bullet$  *SNET*,  $\bullet$  *FNET*) y *NLT* ( $\bullet$  *SNLT*,  $\bullet$  *FNLT*). Sobre el evento en cuestión se especifica la cota.

- e) **TC (Restricción de cardinalidad).** Se expresa el máximo número de bucles  $\mathbb{V} \dots [Max\_loopTimes]$  de ejecución de una actividad cuya duración total es flexible  $[minDur, maxDur] \mathbb{V}$ .
- f) **TC (Ausencia incondicional ABS-U).** La actividad está permanentemente inactiva ( $\odot$ ).
- g) **TC (Ausencia condicional ABS-C).** La actividad está inactiva en el intervalo  $[t1, t2]$  ( $\odot$ ).
- h) **TD (Start to Finish).** Es una dependencia entre el evento de inicio de una actividad precesora y el evento de finalización de la sucesora  $\circ \xrightarrow[\text{SF}]{[leadORlag]} \bullet$ . Se puede especificar un adelanto o retraso entre estos eventos (*leadORlag*).
- i) **TD (Start to Start).** Es una dependencia entre los dos eventos de inicio de la precesora y sucesora  $\circ \xrightarrow[\text{SS}]{[leadORlag]} \circ$ .
- j) **TD (Finish to Start).** Marca la dependencia entre el fin de la precesora e inicio de la sucesora  $\bullet \xrightarrow[\text{FS}]{[leadORlag]} \circ$ .
- k) **TD (Finish to Finish).** Dependencia entre los dos eventos de finalización  $\bullet \xrightarrow[\text{FF}]{[leadORlag]} \bullet$ .

- l) **TD (Ausencia incondicional, ABS-U)**. Si la predecesora se ha activado, la sucesora debe estar inactiva .
- m) **TD (Ausencia condicionada en un intervalo, ABS-C)**. Si se activó la predecesora, la sucesora está inactiva en el intervalo de tiempo especificado .

La especificación de reglas temporales mediante este enfoque es clara y elegante, pero por contra utiliza una semántica que no está todavía soportada por *BPMN 2.0*.

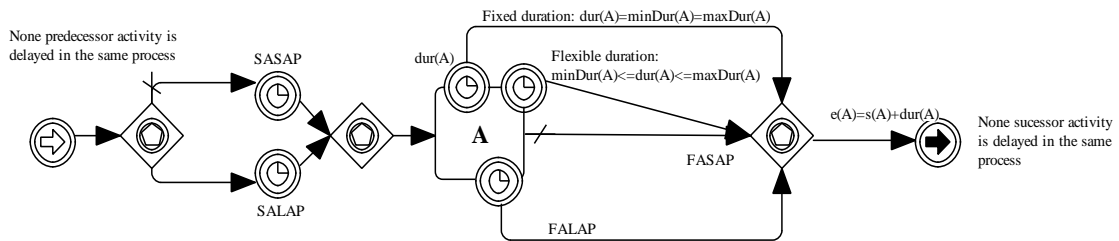
### 6.3.3 Patrones *BPMN* para representación de reglas temporales

En [Flores & Sepúlveda 2011] se realiza una propuesta que utiliza patrones [Gamma et al. 1995; Tran et al. 2007] o construcciones *BPMN* para especificar las distintas restricciones temporales que soporta *MS Project* [Hansen & Hansen 2013; Stover 2007] bajo la perspectiva de restricciones contempladas en «Álgebra de intervalos de Allen» [Allen 1983]. Se contemplan: a) *Restricciones temporales sobre una actividad* y b) *Dependencias temporales entre dos actividades*. Proponen una construcción específica para tipo de restricción, considerando exclusivamente artefactos del estándar *BPMN 1.2* (en el cual se basan), es decir: actividades, eventos, flujo de control y eventos de señal. La ventaja de la propuesta estriba en su plena adaptación al estándar, pero por otro lado hay que hacer mención que los modelos obtenidos si se aplican estas construcciones o patrones estarían sobrecargados. El empleo de señales puede deberse a la limitación de la versión *BPMN 1.2* que no soportaba los «Eventos intermedios no interruptores del flujo de control (*non-interrupting-intermediate events*)» que aparecen en la versión *BPMN 2.0*. En esta sección se realiza como ejercicio el establecimiento de patrones similares a las propuestas de [Flores & Sepúlveda 2011] (que sólo representa las restricciones del Álgebra de Allen [Allen 1983]) pero utilizando *non-interrupting-intermediate events* para analizar el tipo de construcciones o patrones obtenidos. Las Tablas 6.3 y 6.4 muestran una propuesta de estos patrones para las restricciones y dependencias temporales, utilizando *BPMN 2.0* y la taxonomía de reglas de referencia (Figura 6.1).

*BPMN 2.0* aporta nuevos artefactos que le otorgan una mejora en la expresividad frente a la versión *BPMN 1.2* utilizada en [Flores & Sepúlveda 2011]. Se ha realizado el ejercicio de elaboración de estas construcciones sobre todas las reglas de la taxonomía (Figura 6.1) para compararlo con las construcciones de estos autores, y se observa que algunos patrones para moderar las reglas de tiempo siguen todavía muy sobrecargado de artefactos; es decir, si se utilizasen estos patrones para representar reglas de tiempo en los procesos de negocio, los procesos quedarían poco claros a ojos del experto. No obstante, la ventaja del enfoque estriba en que respeta el estándar al no aportar nueva semántica. A continuación se comentan las restricciones temporales de la Tabla 6.3:

**Tabla. 6.3.** Construcciones BPMN 2.0 para representar Restricciones de Tiempo (TC)

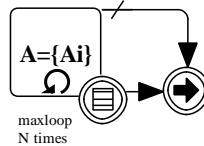
(a) Comienzo y terminación flexible (*ASAP, ALAP*). Duración fija y flexible.



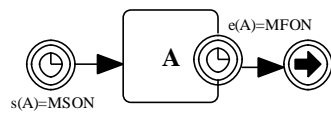
(b) Restricción de cardinalidad.

$$e(A) - s(A) = dur(A) = \sum_{i=1}^N (e(A_i) - s(A_i))$$

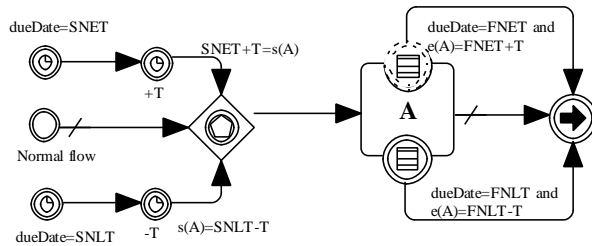
$$min Dur(A) \leq \left( \sum_{i=1}^N (e(A_i) - s(A_i)) \right) \leq max Dur(A)$$



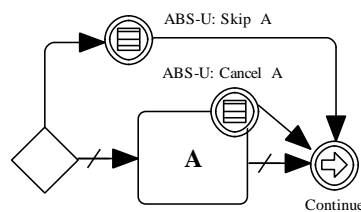
(c) Comienzo y terminación fijos (inflexibles).



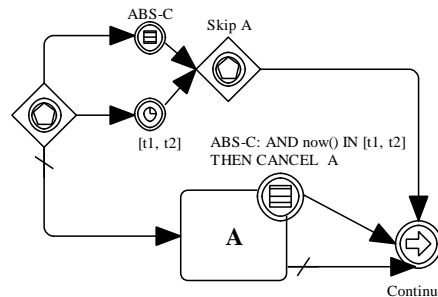
(d) Comienzo y terminación flexible (*NET, NLT*)



(e) Restricción de ausencia incondicional (*ABS-U*)



(f) Restricción de ausencia condicionada en un intervalo (*ABS-C*)



a) **Comienzo y terminación flexible (*ASAP, ALAP*)**. Los flujos normales de ejecución (flujos por defecto marcados como  $\rightarrow$ ) son comienzo (*SASAP*) y finalización (*FASAP*). La introducción de las variantes *ALAP* (*SALAP, FALAP*) se puede realizar con un evento intermedio (timer  $\odot$ ) que dispare el inicio de la actividad. En el caso *FALAP*, el evento es incrustado y provoca la finalización de la actividad al alcanzarse el evento de tiempo. Por otro lado, la duración (*fija o flexible*) puede modelarse también con eventos intermedios incrustados que disparen la finalización de la tarea aunque no se haya completado todo el trabajo. En los casos *ASAP* y *ALAP* no deben introducirse retrasos en el proceso global que contiene a la actividad.

b) **Cardinalidad (*CARD*)**. La actividad puede finalizar al realizar todo su trabajo o bien al alcanzar el máximo número de iteraciones en los límites de duración de la actividad. Se puede modelar con el evento condicional incrustado  $\odot$  <sup>maxloop N times</sup>.

- c) **Comienzo y terminación fijos (inflexibles).** Los casos **MSON** y **MFON** pueden modelarse con timers (⊙) que disparen el inicio y finalización de la actividad.
- d) **Comienzo y terminación flexibles (NET, NLT).** En cuanto al evento de inicio, si se especifican fechas tope, la actividad se podrá iniciar no antes de **SNET** ( $s(A)=snet+T$ ), o bien no después de **SNLT** ( $s(A)=SNLT-T$ ), siendo en ambos casos ( $T \geq 0$ ) y modelándose con timers (⊙). Para especificar la terminación se introducen eventos condicionales ligados a las fechas tope **FNET** ( $e(A)=FNET+T$ ) y **FNLT** ( $e(A)=FNLT-T$ ). En el primer caso, al alcanzar la fecha de referencia FNET no cancela la actividad, pues puede transcurrir un lapso posterior de tiempo. En el segundo caso, **FNLT** sí dispara la cancelación de la actividad.
- e) **Ausencia incondicional (ABS-U).** Si la actividad no se ha iniciado, equivale a saltar (skip) su ejecución; si se ha iniciado, corresponde cancelar la actividad y seguir el flujo del proceso.
- f) **Ausencia condicionada a un intervalo (ABS-C [t<sub>1</sub>, t<sub>2</sub>]).** En este caso hay que saltar la actividad o inhabilitarla sólo dentro del lapso de tiempo especificado.

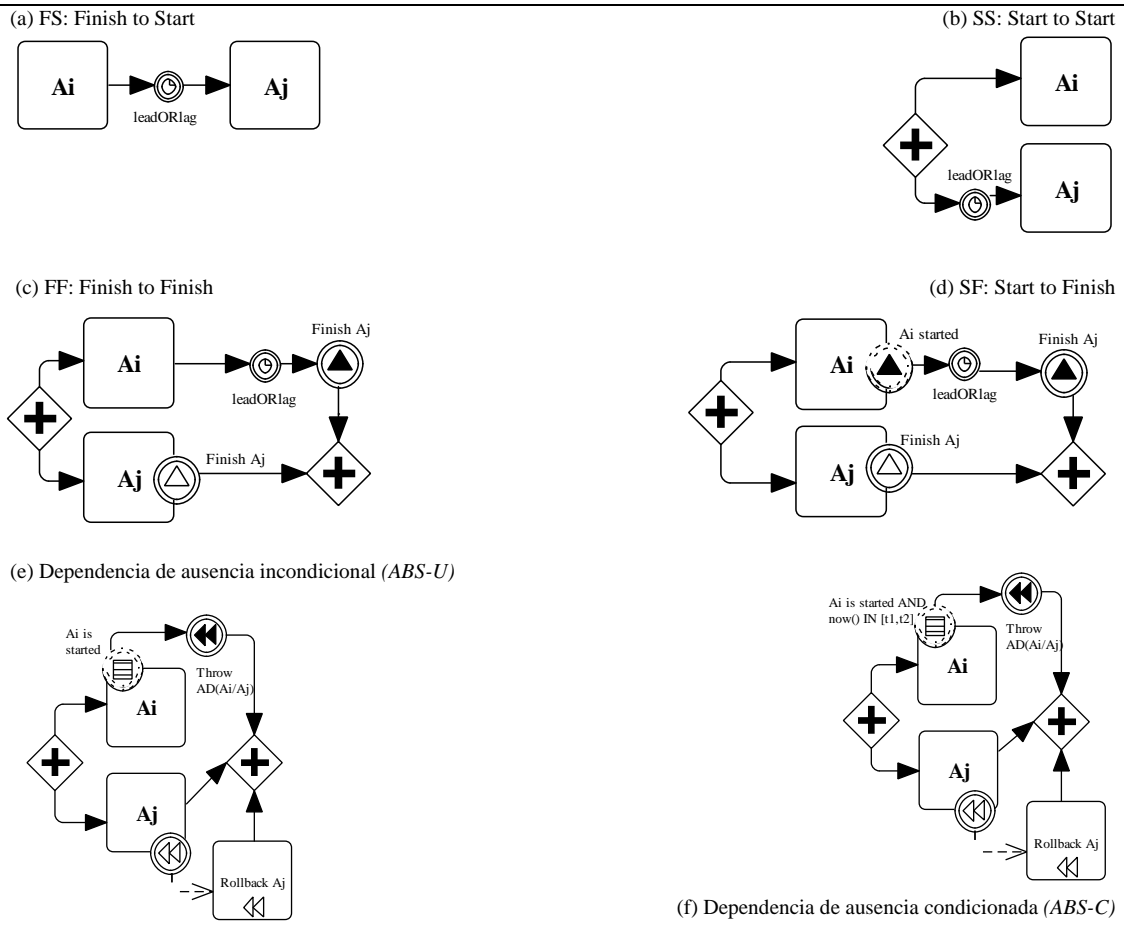
La **Tabla 6.4** muestra propuestas de construcciones para las dependencias temporales:





- a) **Finish to Start (FS).** En este caso simplemente hay que considerar un lapso de tiempo (**leadORlag** ⊙) entre la finalización de la predecesora y el inicio de la sucesora. El flujo de control es secuencial.
- b) **Start to Start (SS).** El flujo de ejecución de la predecesora y sucesora es en paralelo, pero puede existir un lapso de tiempo (**leadORlag** ⊙) entre ambos inicios.
- c) **Finish to Finish (FF).** El flujo de ejecución también es en paralelo, aunque no se exige la sincronización de los inicios. Tras la finalización de la predecesora se captura con un evento el transcurso del lapso de tiempo (**leadORlag** ⊙), y tras él se emite una señal (⊙<sup>Finish Aj</sup>) que la sucesora puede capturar (⊙<sup>Finish Aj</sup>) para interrumpir su ejecución, con lo que se sincronizan ambas finalizaciones.
- d) **Start to Finish (SF).** En este caso hay que capturar el inicio de la predecesora con la señal (⊙<sup>Ai started</sup>) que no interrumpe la actividad. Tras este evento y transcurrido el lapso de tiempo (**leadORlag** ⊙) se emite la señal (⊙<sup>Finish Aj</sup>) que la sucesora puede capturar (⊙<sup>Finish Aj</sup>) para ser interrumpida, sincronizando así el inicio y terminación de ambas.



e) **Dependencia de ausencia incondicional (ABS-U).** Si la actividad predecesora

**Tabla. 6.4.** Construcciones BPMN 2.0 para representar Dependencias Temporales (TD)



se activa (  ), entonces la sucesora no debe llevar a cabo su trabajo. Se modela mediante el lanzamiento de un evento de compensación (  ) que captura la sucesora (  ), entrando en compensación y deshaciendo su trabajo (  ).

f) **Dependencia de ausencia condicionada en un intervalo (ABS-C [t<sub>1</sub>, t<sub>2</sub>]).** El esquema es el anterior pero añadiendo como condición que el instante considerado esté en el intervalo

Se han analizado las dos propuestas: a) *Declarativa*, mediante *extensión del estándar con decoradores*; y b) *Imperativa*, con utilización de *patrones BPMN* que se construyen utilizando exclusivamente artefactos existentes en el estándar. Pretendemos captar las ventajas de ambas propuestas: *enfoque declarativo* [Fahland 2009a, 2009b; Pichler et al. 2011; Reijers et al. 2013] para especificar la dimensión temporal y, además, respetar el soporte del estándar actual, lo cual otorgará simplicidad a los modelos y la posibilidad de implementarlos con esas características temporales. Se propondrá un *enfoque*

*declarativo*, basado en *UML* y *OCL* (Sección 6.5), pero antes se estudiarán los modelos de procesos de los estándares más utilizados en el intercambio de diagramas *BPM: XPDL* y *BPMN*.

## 6.4 Meta–Modelos estándar para representación de procesos

En párrafos anteriores se ha justificado la decisión de optar por enfoques declarativos en vez de imperativos para expresar la dimensión temporal. Los estándares actuales y de mayor aceptación para captar la esencia de los procesos [*OMG:BPMN 2013*], y específicamente, la de los procesos del software [*Iso/Iec:24744 2007, 2014*] y *SPEM 2.0* [*Bendraou et al. 2007; OMG: SPEM 2008*] están soportados por robustos *Meta–Modelos* de procesos. La utilización de la *sintaxis abstracta* de los *Meta–Modelos* otorgará mayor grado de *interoperabilidad* a las propuestas que se realicen, buscándose procesos de extracción más generalistas y no centrados en una sintaxis concreta.

Por otro lado, en los últimos años los *analistas TI* y *expertos de negocio* han venido utilizando frecuentemente «*UML Activity Diagrams (AD)*» [*OMG:UML 2011*] y *diagramas BPMN 2.0* [*OMG:BPMN 2013*] para la modelación de flujos de actividades de procesos. *BPMN 2.0* y *XPDL* [*WfMC:XPDL 2012*] también ofrecen características para el intercambio de diagramas.

En esta sección se va a realizar un análisis de las vistas *UML* relacionadas con *actividades* y *procesos* dentro de estos *Meta–Modelos*, incluyendo además el *Meta–Modelo XPDL 2.2*. En el último subapartado se comparan los *Meta–Modelos* con objeto de obtener un conjunto mínimo de artefactos y proponer un modelo *UML* común de procesos sobre el que se podrá modelar la dimensión temporal.

### 6.4.1 Meta–Modelo UML Activity Diagrams

La *Figura 6.6* es una vista o diagrama de clases de *UML AD* [*OMG:UML 2011*], donde las clases principales son *Activity*, *ActivityNode* y *ActivityEdge* y permiten modelar flujo de control y flujo de información.

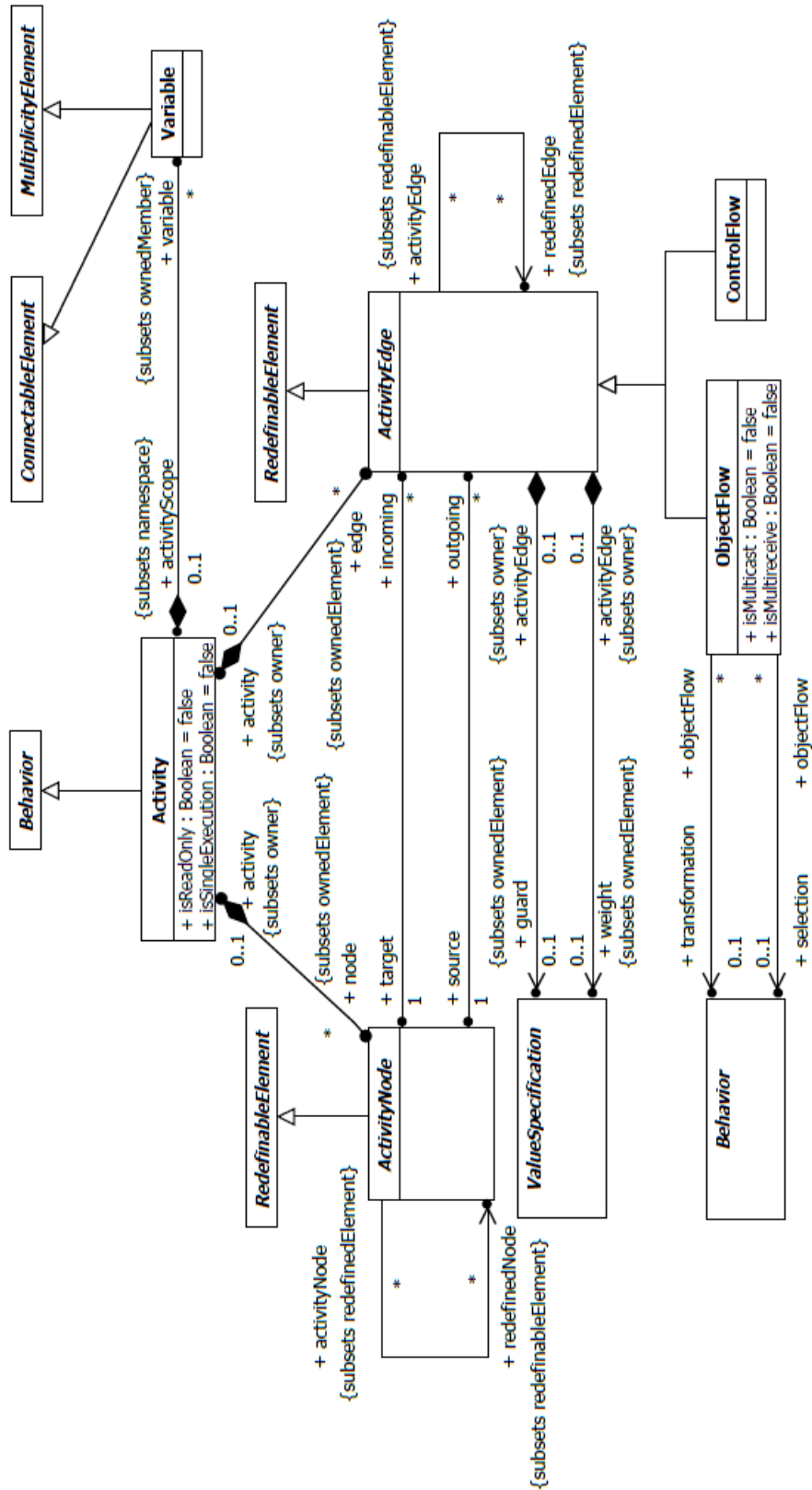


Figura. 6.6. Sintaxis abstracta «UML Activity Diagrams (AD)»

La Figura 6.7 representa la vista del *Meta-Modelo UML* [OMG:UML 2011] que está destinada a soportar el intercambio de distintos tipos de diagramas de este estándar.

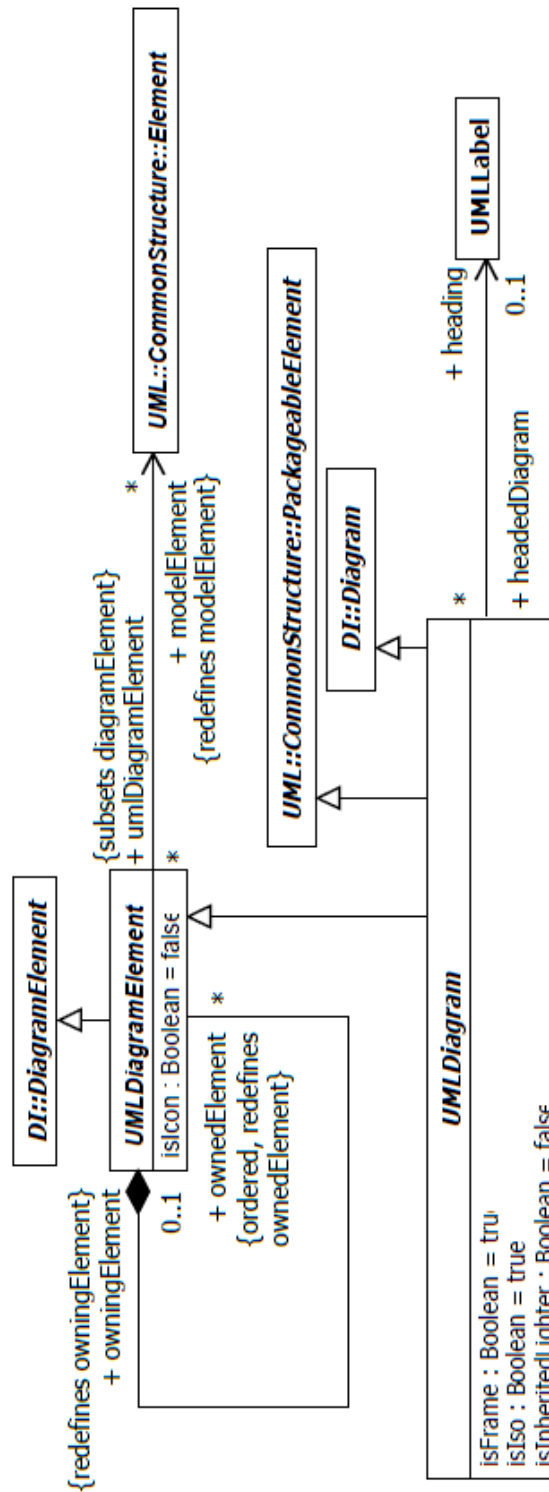


Figura. 6.7. Meta-Modelo para intercambio de diagramas UML

### 6.4.2 Meta-Modelo BPMN 2.0

La Figura 6.8 es una vista del *Meta-Modelo* de procesos BPMN 2.0 [OMG:BPMN 2013], extraídos del paquete *Semantic*. Las clases esenciales son *activity* que es un subtipo de *tActivity* (*Activity* → *tActivity*); *process* que es un subtipo de *tProcess* y *flowElement*, *subProcess* como subtipo de *tActivity* y *flowElement*. Una actividad elemental es una tarea: subclase *tTask*. Las clases *laneSet*, *tLaneSet* sirven para representar el concepto de *pool* BPMN y *lane* y *tLane* para las subdivisiones de cada *pool*.

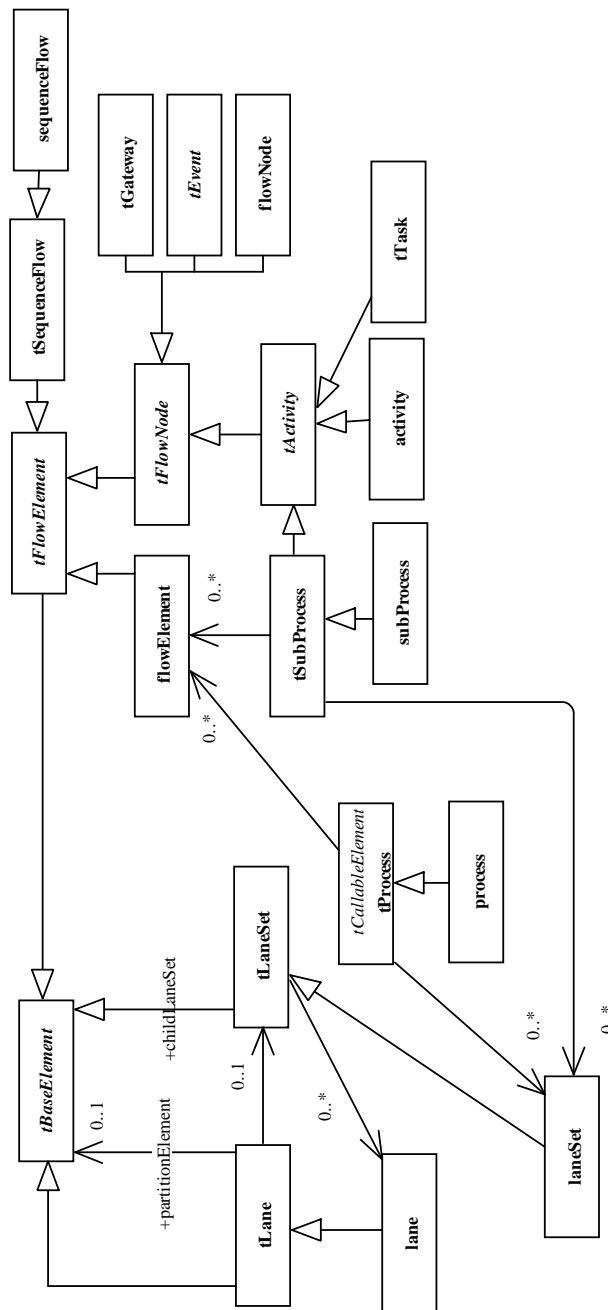


Figura. 6.8. Meta-Modelo de procesos y actividades BPMN 2.0

La Figura 6.9 muestra la vista del *Meta-Modelo BPMN 2.0* [OMG:BPMN 2013] que facilita el intercambio de diagramas en formato XML (.xsd y .xmi).

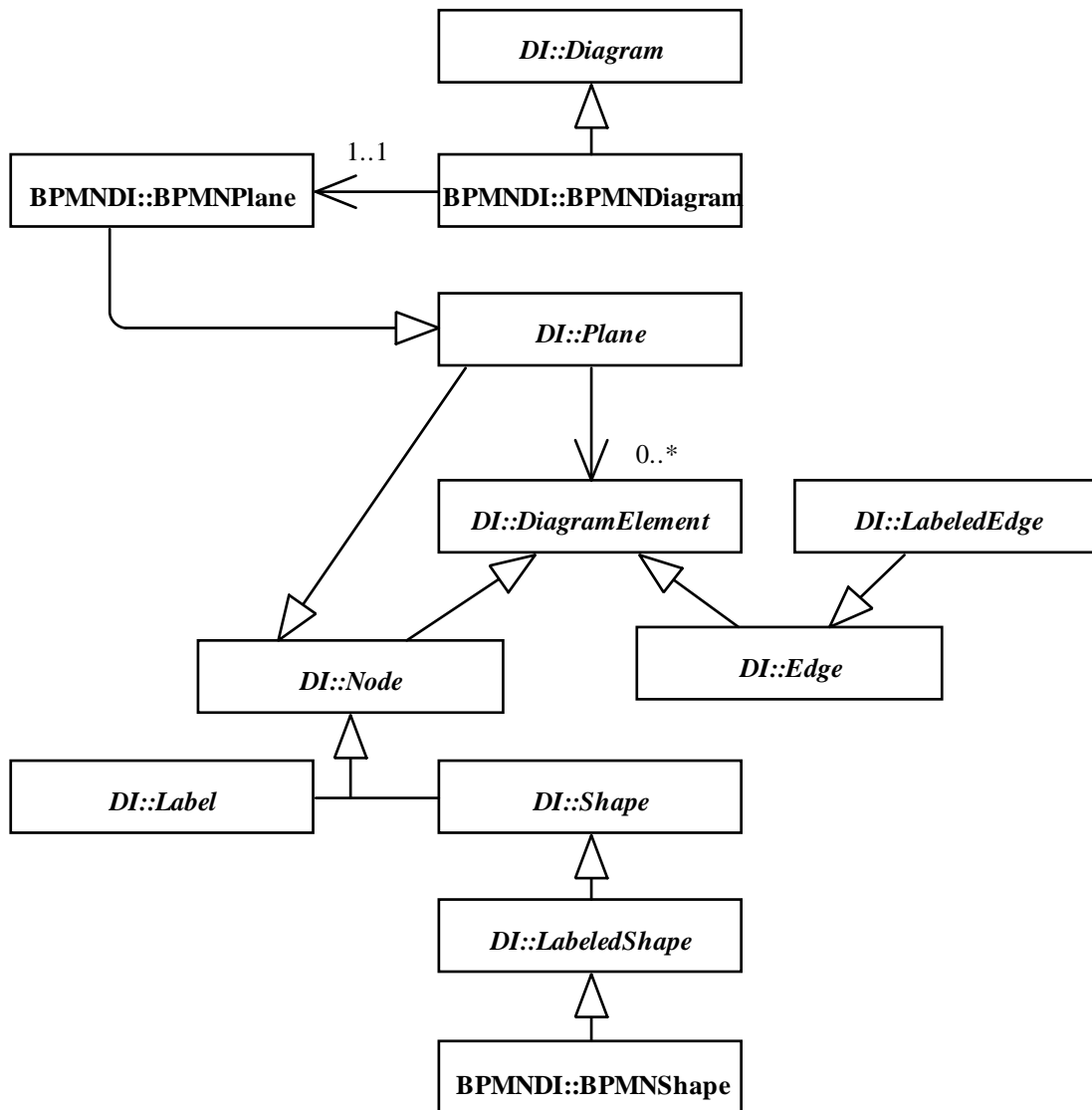


Figura. 6.9. Meta-Modelo para intercambio de diagramas BPMN 2.0

### 6.4.3 Meta-Modelo XPDL 2.2

XPDL es un estándar de *WfMC* orientado a intercambiar diagramas y metadatos de procesos en formato XML. La Figura 6.10 muestra su *Meta-Modelo* de procesos [WfMC:XPDL 2012]. Un proceso está representado por la clase *WorkflowProcess*, que es un subtipo de *ProcessType*. Cada *WorkflowProcess* está compuesto por un conjunto de actividades: *Activity* o bien por actividades agrupadas: *ActivitySet*. Una actividad (*Activity*) puede ser una tarea elemental (*Task*) o bien un bloque de actividades (*BlockActivity*), que a su vez es un subtipo de *ActivitySet*. Las clases *Businessprocessdiagram*, *Pool* y *Lane* complementan el *Meta-Modelo*, situando

procesos y actividades en su correspondiente contenedor.

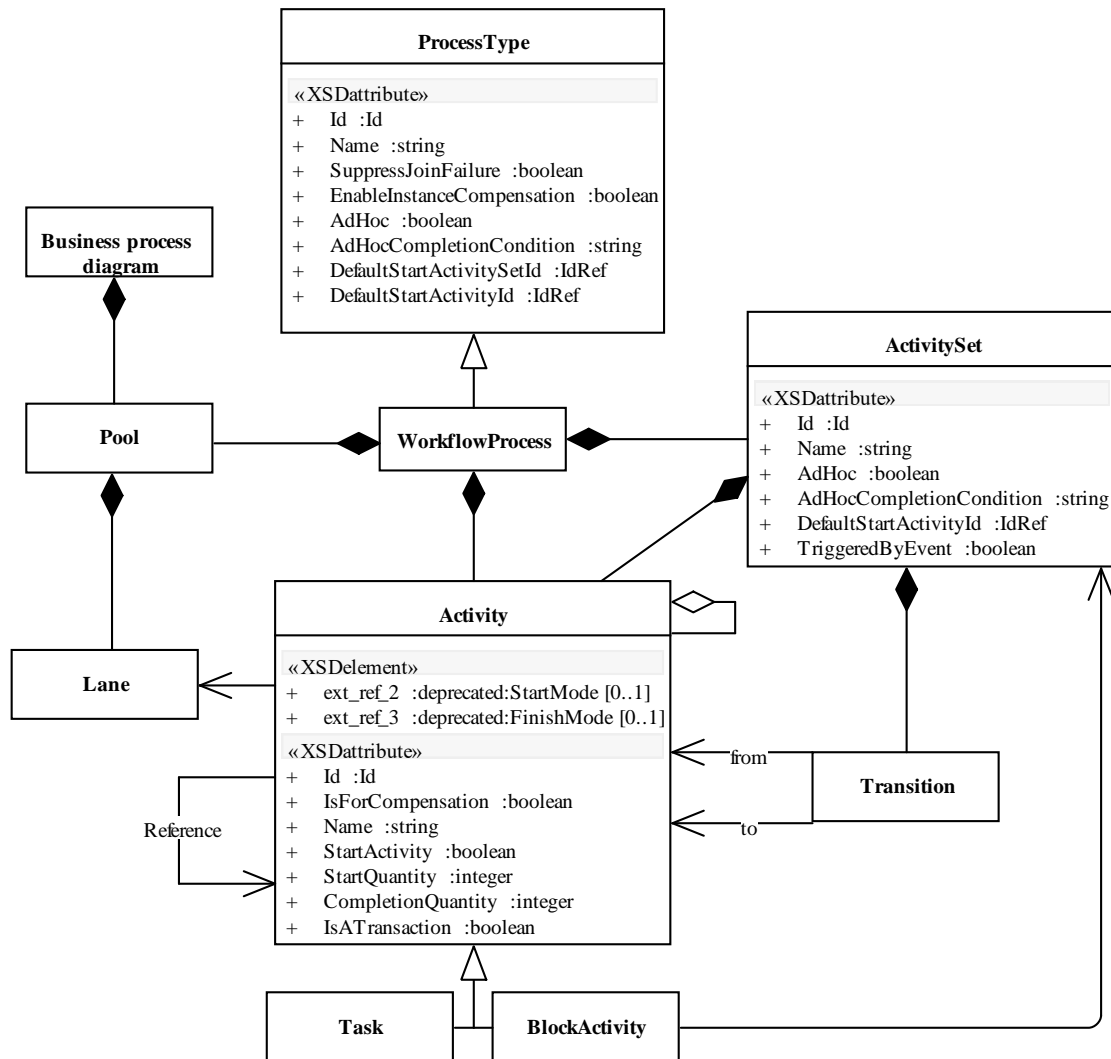


Figura. 6.10. Meta-Modelo de procesos y actividades XPD 2.2

#### 6.4.4 Comparación de Meta-Modelos de procesos

La [Tabla 6.5](#) compara los artefactos esenciales de los modelos de procesos de *UML AD*, *BPMN 2.0* y *XPD 2.2* mostrados en secciones anteriores. Vamos a necesitar un *Meta-Modelo* con una estructura mínima para representar procesos, sobre el que poder especificar la dimensión temporal y poderlo extender posteriormente para su utilización en procesos de ingeniería inversa de sistemas (Capítulo 7). Este *Meta-Modelo* lo denominamos *PIM UML Pivote (PIM-up)*; se sitúan sus clases en la última columna para los conceptos: i) Proceso, ii) Subproceso o grupo de actividades, iii) Actividad, iv) Tarea, v) Piscina o grupo de calles y vi) Calle.

**Tabla. 6.5.** Comparación de Meta-Modelos de procesos: clases y asociaciones

Concepto	Meta-Modelos			
	UML AD	BPMN 2.0	XPDL 2.2	PIM-up
Proceso	∅	process→tprocess→flowElement flowElement→tFlowElement	WorkflowProcess→processType	Process
Subproceso, grupo de actividades	∅	subprocess→tSubProcess→flowElement tSubProcess→tActivity flowElement→tFlowElement	BlockActivity→Activity BlockActivity→ActivitySet	Subprocess→Process Subprocess→Activity
Actividad	Activity	Activity→tActivity tActivity→tFlowNode→tFlowElement	Activity	Activity
Tarea	Activity	tTask→tActivity	Task→Activity	Task→Activity
Piscina o grupo de calles	SwinLane	laneSet→tLaneSet	Pool	Pool
Calle	SwinLane	lane→tLane	Lane	Lane

## 6.5 Un Meta-Modelo de procesos con reglas temporales (PIM-up)

Se propone un enfoque *declarativo* [Fahland 2009a, 2009b; Pichler et al. 2011; Reijers et al. 2013] para expresar *reglas temporales sobre procesos*, basado en un *Meta-Modelo UML pivote y fórmulas OCL (PIM-up)* para especificar este tipo de restricciones. Las estructuras esenciales son las capturadas en la [Tabla 5.5](#) anterior; este *Meta-Modelo (PIM-up)* otorgará mayor autonomía frente a la elección de alguno de los *Meta-Modelos* de los estándares de procesos de la sección anterior. Se utilizará posteriormente para sustentar el enfoque de ingeniería inversa [Chikofsky & Cross 1990] basada en *MDE*; además de ofrecer una representación más clara de la dimensión temporal, ayudará en la resolución de algunos problemas, como colisiones en la integración de artefactos (Capítulo 7).

La utilización de un *Meta-Modelo* permitirá, con pocas adaptaciones, aprovechar las facilidades de lenguajes gráficos que lo soporten, ya que, dentro del estándar, las herramientas suelen permitir adaptaciones del interfaz, por ejemplo con *estereotipos y/o personalizaciones* de las figuras ligadas a los artefactos *BPMN*, de modo que se facilite la percepción visual de esta nueva semántica temporal, y asimilarla a propuestas como [Gagné & Trudel 2009a; Cheikhrouhou et al. 2013b] donde la semántica subyace en los nuevos decoradores aportados.

### 6.5.1 Meta-Modelo UML de procesos

#### 6.5.1.1 Diagrama de clases

El estudio de los *Meta-Modelos* de los estándares estudiados en las secciones anteriores permite extraer un conjunto mínimo de clases que pueden representarse en el diagrama *UML* de clases de la [Figura 6.11](#) [Arevalo et al. 2015b]. Estas clases se agrupan dentro de un paquete «*Semantic*».

La clase principal es *Activity*, cuyos subtipos pueden ser *Task* y *Subprocess*, que a su



vez es un subtipo de *Process*.

Un *Proceso (Process)* se compone de un conjunto de *actividades (+activities)*; tiene los atributos nombre (*name:char*), instante de inicio (*start:date*) y de finalización (*end:date*), y los eventos calculados por *CPM*: (*startCPM:date*), (*endCPM:date*) y la *duración mínima (minDur:interval)* del proceso para el camino crítico del proceso. El atributo (*end*) se activa cuando el proceso ha terminado. Un subproceso puede ser de tipo *Adhoc (isAdhoc:boolean)*.

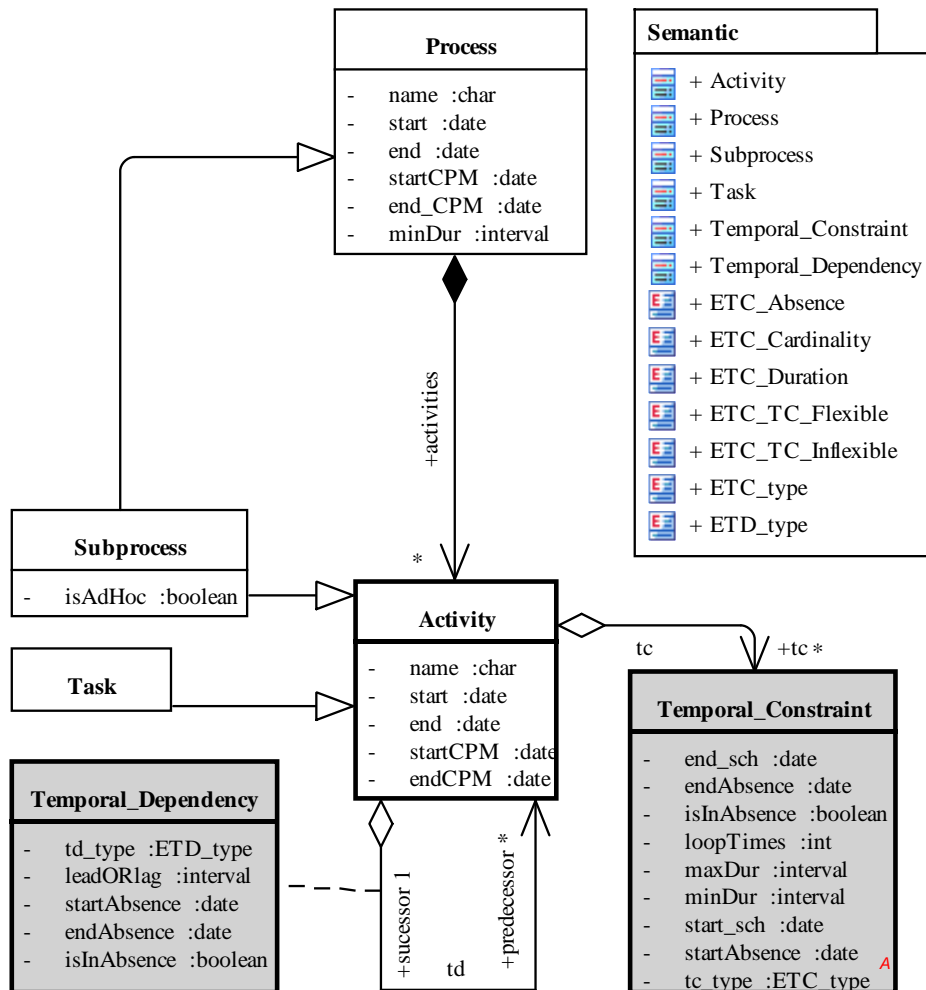


Figura. 6.11. Meta-Modelo PIM-up

La clase *Activity* posee los atributos nombre (*name:char*) y los eventos de inicio (*start:date*), de finalización (*end:date*) y los eventos calculados por *CPM*: (*startCPM:date*) y (*endCPM:date*). Una actividad puede tener asociado un conjunto de restricciones temporales (*Temporal\_Constraint*). Para la clase *Temporal\_Constraint* figuran los siguientes grupos de propiedades:

- i) *start\_sch:date* y *end\_sch:date* denota un intervalo de fechas de referencia que se van a instanciar para restricciones que fijan la fecha (*MSON*, *MFON*) o

dejan los eventos de inicio o terminación flexible (*NET, NLT*).

- ii) *isInAbsence:boolean* especifica la restricción de ausencia (*ABS-U*), y *startAbsence:date*, *endAbsence:date* definen el intervalo de tiempo de la restricción de ausencia condicionada (*ABS-C*).
- iii) *minDur:interval* y *maxDur:interval* es el intervalo de tiempo para especificar duración fija (*FIXD*) o flexible (*FLEXD*) de ejecución de la actividad.
- iv) *loopTimes:int* es un entero que limita el número máximo de instancias de una actividad que se ejecuta en bucle (*Restricción de cardinalidad: CARD*); en este caso *minDur:interval* y *maxDur:interval* es el intervalo de tiempo de ejecución de cada instancia, puesto que las propiedades *Activity.start* y *Activity.end* marcan el intervalo de ejecución de todas las instancias.
- v) *tc\_Type:ETC\_Type* es el tipo de restricción, según el enumerado *ETC\_Type* que se desglosa jerárquicamente en la [Figura 6.12 \[Arevalo et al. 2015b\]](#).

Para la clase *Temporal\_Dependency* se instancian los atributos:

- i) *isInAbsence:boolean* especifica la dependencia de ausencia (*ABS-U*), y *startAbsence:date*, *endAbsence:date* definen el intervalo de tiempo de la dependencia de ausencia condicionada (*ABS-C*).
- ii) *leadORlag:interval* es el adelanto o retraso entre los eventos de la *actividad sucesora* y *predecesora* según el tipo de dependencia (*SF, SS, FS, FF*).
- iii) *td\_type* es el *tipo de dependencia*, según la enumeración *ETD\_type* de la [Figura 6.12](#).

### 6.5.1.2 Enumeraciones

La [Figura 6.12 \[Arevalo et al. 2015b\]](#) representa una jerarquía de enumeraciones para tipificar las restricciones y dependencias temporales.

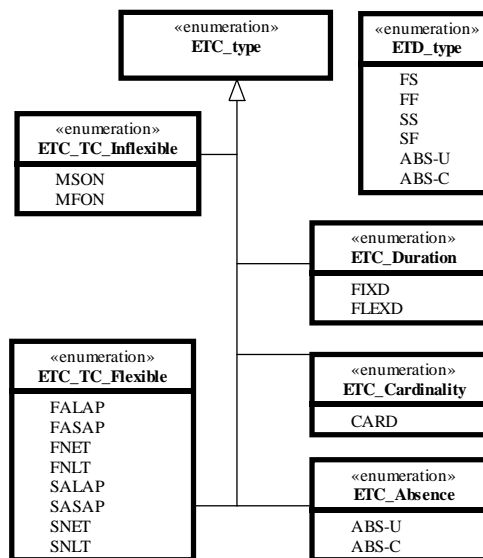


Figura. 6.12. Enumeraciones del Meta-Modelo PIM-up

### 6.5.1.3 Extensión del ciclo de vida de una actividad bajo CPM

En la Figura 6.5 se ha expuesto una máquina de estados para ilustrar el ciclo de vida de las actividades de un proceso. En la Figura 6.13 se introduce un estado intermedio, asociado a las acciones o iteraciones del método *CPM*: estado «*Scheduled*» entre el estado «*Inactive*» y el estado *Closed*». También se especifican las transiciones en función de los estados que deben tener los atributos del *Meta-Modelo*. Hay que tener en cuenta que un *proceso* es un tipo de *actividad* que puede aparecer en otro *proceso*, con lo que la máquina de estados es válida para una *actividad* y también para un *proceso*.

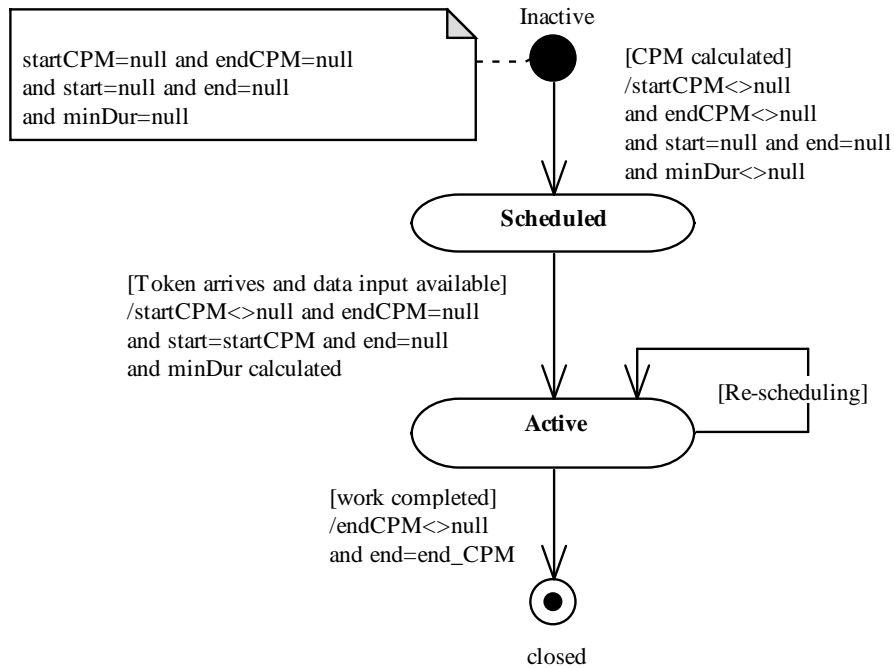


Figura. 6.13. Ciclo de vida de proceso y actividades (cálculo CPM)

En el estado «*Inactive*», todos los atributos implicados no están instanciados: *startCPM*, *endCPM*, *minDur* (si es un proceso) *start* y *end*.

Cuando se lleva a cabo una planificación mediante el método *CPM*, se instancian los atributos *startCPM*, *endCPM* y *minDur* (si es un proceso, con la duración del camino crítico del grafo). Los atributos *start* y *end* no instanciados representan que la tarea aún no ha recibido el token para activarse.

Al llegar el token a la *actividad* y quedar disponibles los datos de entrada, la *actividad* puede llevar a cabo su trabajo «*Active*». Se instancia entonces el atributo *start* (inicio real de la tarea) y se igualan (*startCPM=start*). Desde el estado «*Active*» se puede realizar una nueva planificación que afecte al proceso y su conjunto de actividades o bien que la actividad termine su trabajo, pasando a estado «*Closed*», en cuyo momento se instancia *end* y se igualan (*endCPM=end*).

Esta máquina de estado será de gran utilidad en la siguiente sección, que formula las

restricciones temporales mediante el lenguaje *OCL* [OMG:OCL 2014], pues permitirá formular las reglas temporales independientemente del estado, ofreciendo así fórmulas *OCL* más compactas y sencillas.

### 6.5.2 Definición de la dimensión temporal con OCL

*OCL* es un «lenguaje declarativo Side-effect free» [Demuth et al. 1999, 2001; Cabot et al. 2012; OMG:OCL 2014]. *Declarativo* significa que está exento de construcciones imperativas, especificándose sólo fórmulas que expresan cada restricción. «Side-effect free» significa que se restringe el estado pero no incluye primitivas que modifiquen un estado inicial.

En primer lugar se establecen fórmulas que representan como restricciones *OCL* las transiciones de la máquina de estados del apartado anterior (Figura 6.13). A continuación se formulan las distintas *restricciones de tiempo y dependencias temporales* de la taxonomía de reglas contemplada en la Figura 6.1. De este modo, las expresiones de las reglas temporales serán más compactas y no obligarán a sobrecargar cada sentencia con condiciones excesivas (asociadas a distinguir en qué estado está la actividad).

#### 6.5.2.1 Restricciones sobre el ciclo de vida de procesos y actividades

La Figura 6.14 representa la formulación *OCL* de las transiciones de la máquina de estados anterior (Figura 6.13), distinguiendo los invariantes de *procesos y actividades*.

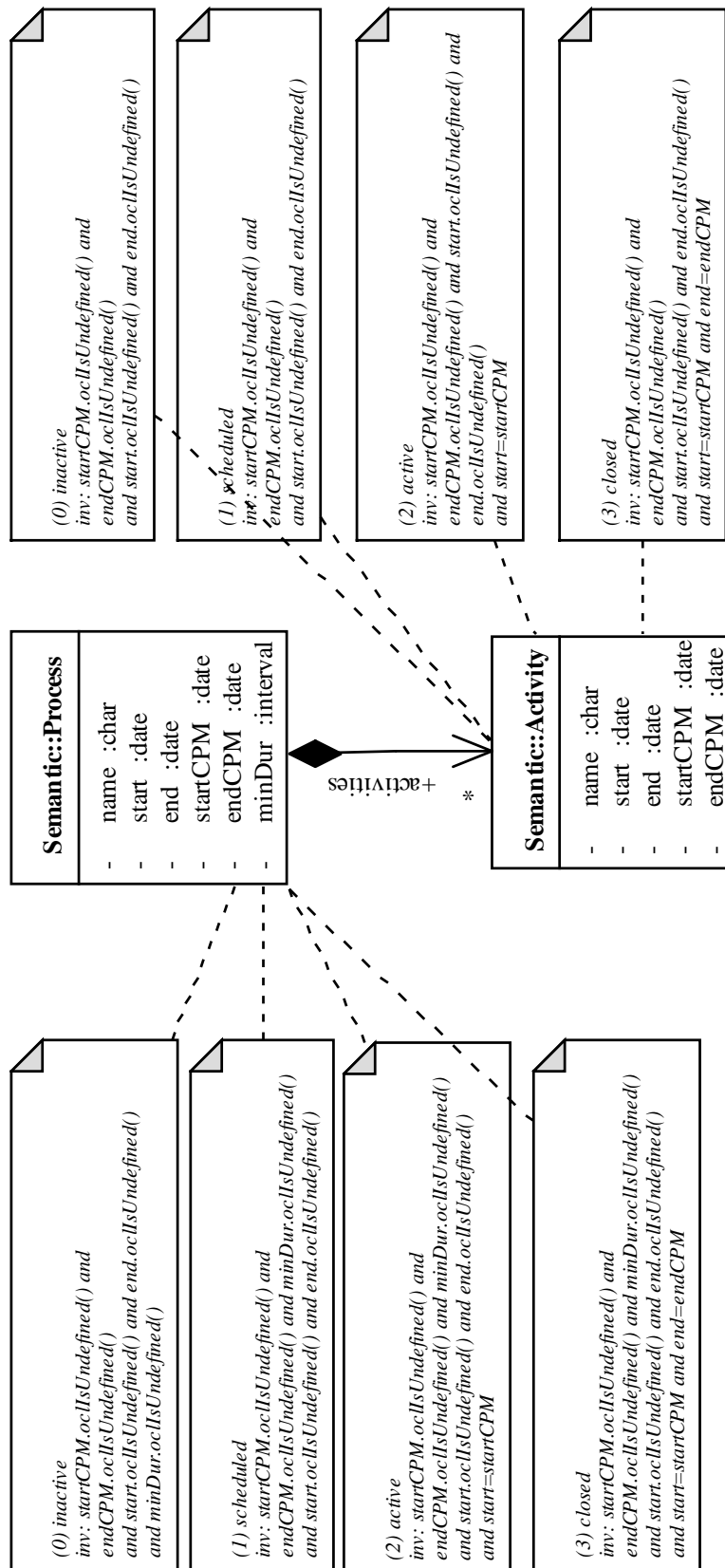


Figura. 6.14. Restricciones OCL sobre el ciclo de vida de procesos y actividades MM PIM-up

6.5.2.2 Restricciones temporales sobre una actividad

Las restricciones de tiempo se establecen sobre las clases *Activity* y *Temporal\_Constraint* del Meta-Modelo (Figuras 6.15, 6.16 y Tabla 6.6) [Arevalo et al. 2015b].

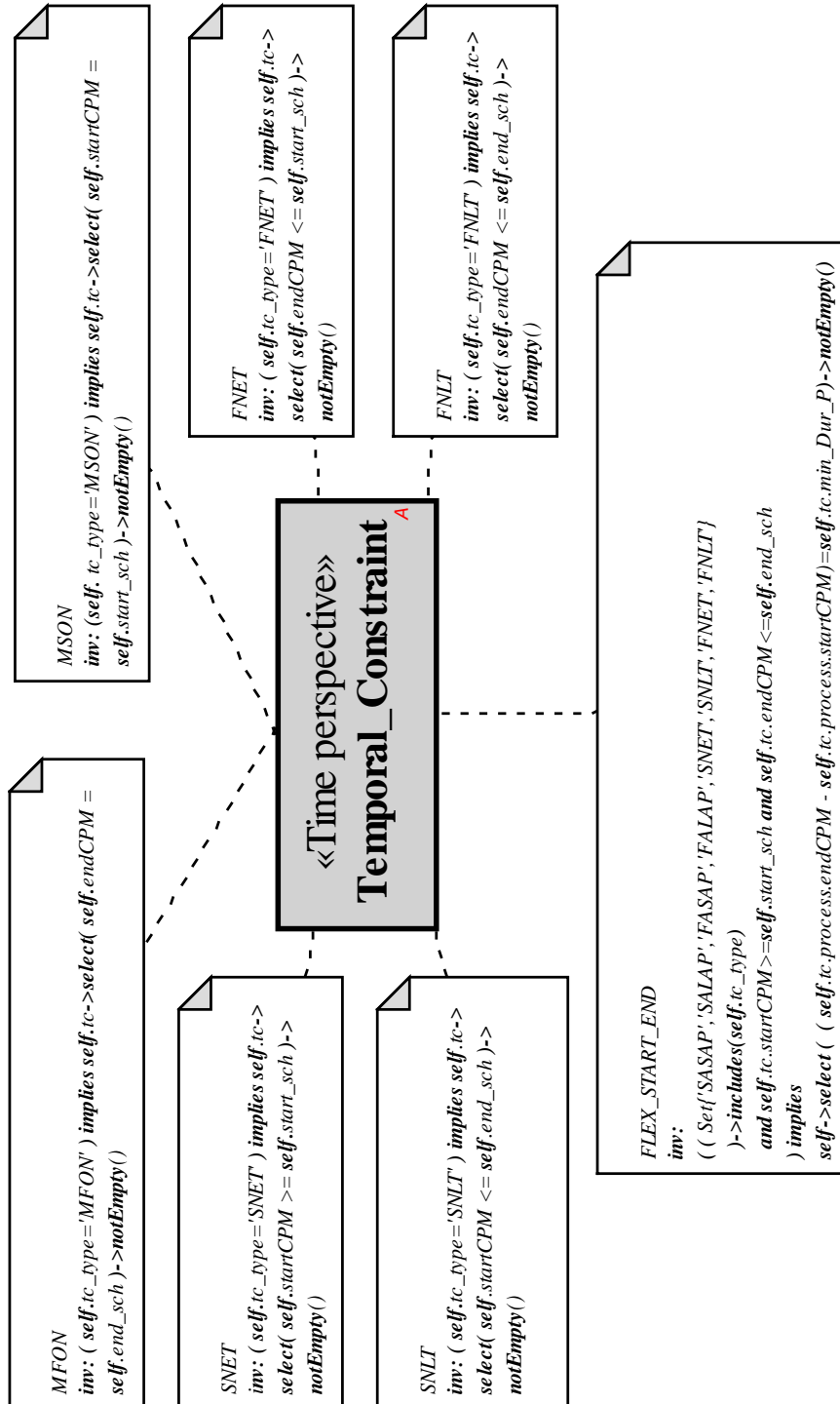


Figura. 6.15. Restricciones de tiempo OCL (a)

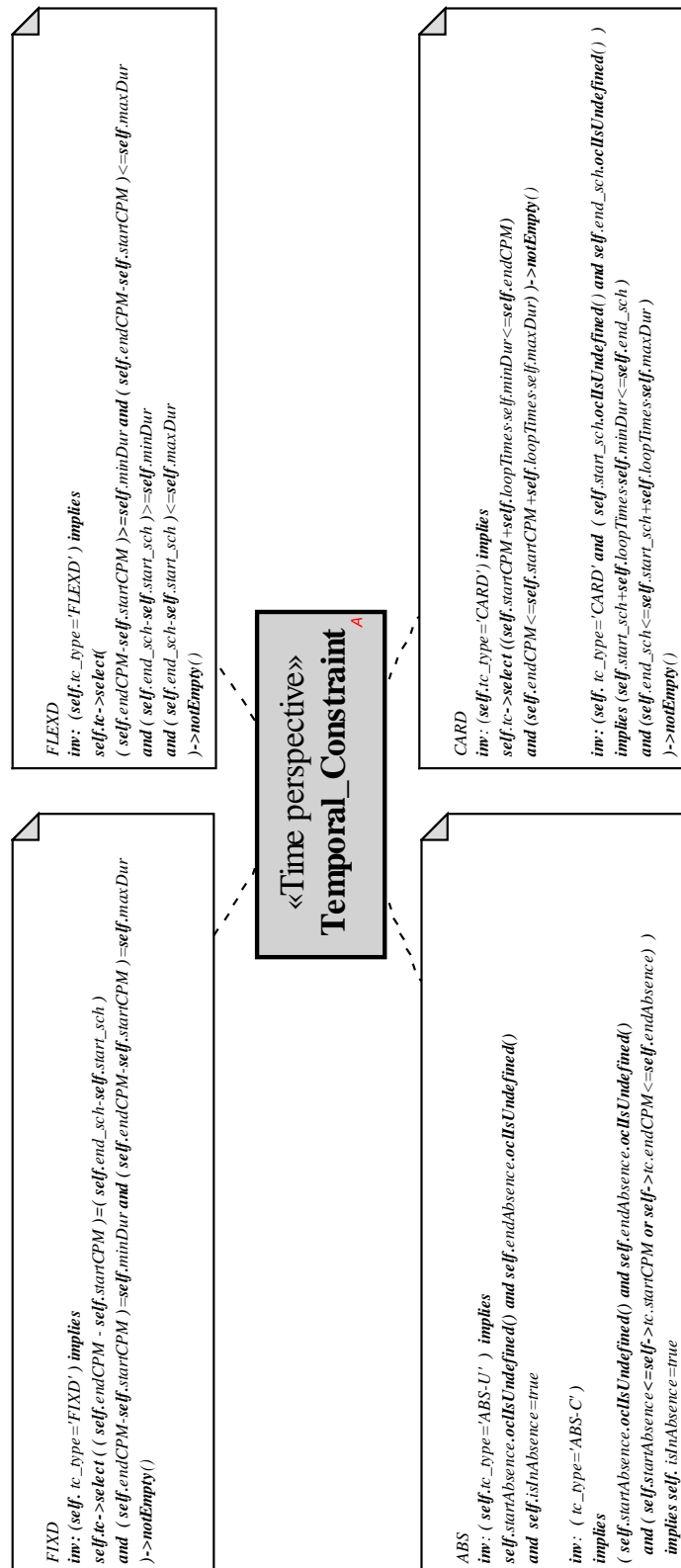


Figura. 6.16. Restricciones de tiempo OCL (b)

**Duración Fija ('FIXD')**. Este invariante expresa que la duración de la *actividad* coincide con la duración planificada, siendo ésta: ( $minDur=maxDur$ ), que debe coincidir con el intervalo de ejecución de dicha actividad ([expresión 6.1](#)).

**Duración Flexible ('FLEXD')**. La expresión es similar a la anterior, pero relajando la duración de la *actividad* a un intervalo ([expresión 6.2](#)).

**Inicio y terminación flexibles (FLEX Start End)**. Este invariante agrupa a un conjunto de restricciones; esto se expresa con la condición *OCL*.

*Set{ 'SASAP', 'SALAP', 'FASAP', 'FALAP', 'SNET', 'SNLT', 'FNET', 'FNLT' } → includes(tc\_type)*

Hay que distinguir el grupo  $\{ASAP, ALAP\}$  del grupo  $\{NET, NLT\}$ , pues en el primero, *CPM*, en las iteraciones de cálculo del camino crítico, puede alterar libremente los eventos de inicio de las actividades que tienen holgura (las que no afectan a la duración mínima del proceso), mientras que en el segundo grupo, el modelador puede poner cota a alguno de los eventos. Esto ya se ha expresado en la [Figura 6.2](#) y [Tabla 6.1](#); sustituyendo en esa expresión los parámetros: ( $s_o=start\_sch$  y  $e_o=end\_sch$ ). Con *start\_sch* y *end\_sch* se fijan las cotas en cada caso ([Figura 6.2](#)). El atributo *process.minDur* es la duración del camino crítico calculada mediante *CPM*, luego el invariante expresa que modificando los eventos de las restricciones de inicio y terminación flexible, la duración total de proceso se conserva igual a *process.minDur*.

**Cardinalidad ('CARD')**. *minDur* y *maxDur* son las duraciones máximas de ejecución de cada instancia de la actividad en bucle. El bucle no puede ejecutarse más de *loopTimes* veces dentro del intervalo de ejecución global de la tarea (en magnitudes reales o sobre el plan de ejecución).

**Comienzo (MSON) o terminación fijas (MFON)**. El comienzo de la actividad debe suceder en el instante planificado (*MSON*) o el evento de finalización de la actividad está determinado por la planificación (*MFON*).

**SNET, SNLT, FNET, FNLT**. En este caso los invariantes expresan restricciones sobre las fechas tope que actúan como cotas para los eventos de inicio o terminación de la actividad. Este grupo de restricciones también están involucradas en el invariante previo de inicio y terminación flexible, así que no deben alterar la duración mínima del proceso.

**ABS (ABS-U, ABS-C)**. El atributo *isInAbsence* se refiere al estado de ausencia de una *actividad*. Si el estado es permanente (*isInAbsence=true*) la ausencia es incondicional (*ABS-U*) y la actividad se salta cuando le llega el flujo de control, pero si la actividad figura en este estado sólo durante un intervalo de tiempo, la ausencia es condicionada (*ABS-C*) en el intervalo [*startAbsence, endAbsence*].



**Tabla. 6.6.** Restricciones temporales OCL

# tr	Especificación OCL de la restricción temporal
<b>FIXD</b>	<p><u>Context Temporal constraint inv:</u>  <i>(self.tc_type='FIXD')</i> implies  <i>self.tc</i>→select ( ( <i>self.endCPM</i> - <i>self.startCPM</i> )=( <i>self.end_sch</i>-<i>self.start_sch</i> )                      and ( <i>self.endCPM</i>-<i>self.startCPM</i> )=<i>self.minDur</i> and ( <i>self.endCPM</i>-<i>self.startCPM</i> )=<i>self.maxDur</i> )→notEmpty()</p>
<b>FLEXD</b>	<p><u>Context Temporal constraint inv:</u>  <i>( self.tc_type='FLEXD' )</i> implies  <i>self.tc</i>→select( ( <i>self.endCPM</i>- <i>self.startCPM</i> )&gt;=<i>self.minDur</i> and ( <i>self.endCPM</i>- <i>self.startCPM</i> )&lt;=<i>self.maxDur</i>                      and ( <i>self.end_sch</i>-<i>self.start_sch</i> )&gt;=<i>self.minDur</i>                      and ( <i>self.end_sch</i>-<i>self.start_sch</i> )&lt;=<i>self.maxDur</i> )→notEmpty()</p>
<b>FLEX</b> <b>Start_</b> <b>End</b>	<p><u>Context Temporal constraint inv:</u>                      ( ( <i>Set</i>/'SASAP','SALAP','FASAP','FALAP','SNET','SNLT','FNET','FNLT' ) )→includes( <i>self.tc_type</i> )                      and <i>self.tc.startCPM</i>&gt;= <i>self.start_sch</i> and <i>self.tc.endCPM</i>&lt;= <i>self.end_sch</i> ) implies  <i>self</i>→select ( ( <i>self.tc.process.endCPM</i> - <i>self.tc.process.startCPM</i>)=<i>self.tc.min_Dur_P</i>)→notEmpty()</p>
<b>CARD</b>	<p><u>Context Temporal constraint inv:</u>  <i>( self.tc_type='CARD' )</i> implies  <i>self.tc</i>→select ( ( <i>self.startCPM</i>+<i>self.loopTimes</i>-<i>self.minDur</i>&lt;=<i>endCPM</i>)                      and ( <i>self.endCPM</i>&lt;= <i>self.startCPM</i>+<i>self.loopTimes</i> - <i>self.maxDur</i> ) )→notEmpty()  <u>Context Temporal constraint inv:</u>  <i>inv:</i> ( <i>self.tc_type='CARD'</i> and ( <i>self.start_sch</i>→notEmpty() and <i>self.end_sch</i>→notEmpty() ) )                      implies ( <i>self.start_sch</i>+<i>self.loopTimes</i>-<i>self.minDur</i>&lt;=<i>self.end_sch</i> )                      and ( <i>self.end_sch</i>&lt;=<i>self.start_sch</i>+<i>self.loopTimes</i>-<i>self.maxDur</i> ) )→notEmpty()</p>
<b>MSON</b>	<p><u>Context Temporal constraint inv:</u>  <i>( self.tc_type='MSON' )</i> implies <i>self.tc</i>→select( <i>self.startCPM</i> = <i>self.start_sch</i> )→notEmpty()</p>
<b>MFON</b>	<p><u>Context Temporal constraint inv:</u>  <i>( self.tc_type='MFON' )</i> implies <i>self.tc</i>→select( <i>self.endCPM</i> = <i>self.end_sch</i> )→notEmpty()</p>
<b>SNET</b>	<p><u>Context Temporal constraint inv:</u>  <i>( self.tc_type='SNET' )</i> implies <i>self.tc</i>→select( <i>self.startCPM</i> &gt;= <i>self.start_sch</i> )→notEmpty()</p>
<b>SNLT</b>	<p><u>Context Temporal constraint inv:</u>  <i>( self.tc_type='SNLT' )</i> implies <i>self.tc</i>→select( <i>self.startCPM</i> &lt;= <i>self.end_sch</i> )→notEmpty()</p>
<b>FNET</b>	<p><u>Context Temporal constraint inv:</u>  <i>( self.tc_type='FNET' )</i> implies <i>self.tc</i>→select( <i>self.endCPM</i> &lt;= <i>self.start_sch</i> )→notEmpty()</p>
<b>FNLT</b>	<p><u>Context Temporal constraint inv:</u>  <i>( self.tc_type='FNLT' )</i> implies <i>self.tc</i>→select( <i>self.endCPM</i> &lt;= <i>self.end_sch</i> )→notEmpty()</p>
<b>ABS</b>	<p><u>Context Temporal constraint inv:</u>  <i>( self.tc_type='ABS-U' )</i> implies  <i>self.startAbsence.oclIsUndefined()</i> and <i>self.endAbsence.oclIsUndefined()</i>                      and <i>self.isInAbsence</i>=true</p> <p><u>Context Temporal constraint inv:</u>  <i>( self.tc_type='ABS-C' )</i> implies                      ( <i>self.startAbsence.oclIsUndefined()</i> and <i>self.endAbsence.oclIsUndefined()</i>                      and ( <i>self.startAbsence</i>&lt;=<i>self</i>→<i>tc.startCPM</i> or <i>self</i>→<i>tc.endCPM</i>&lt;=<i>self.endAbsence</i> ) )                      implies <i>self.isInAbsence</i>=true</p>

6.5.2.3 Dependencias temporales entre actividades

Las restricciones de tiempo se establecen sobre las clases *Activity* y *Temporal\_Dependency* del *Meta-Modelo* (Figura 6.17 y Tabla 6.7) [Arevalo et al. 2015b]. La clase *Temporal\_Dependency* tiene dos roles para pares de actividades: *predecessor* (P) y *successor* (S); sobre estos roles se establecen los invariantes que se expresan a continuación.

**FS, FF, SF, SS.** Todas estas restricciones se refieren a las precedencias entre eventos. Los invariantes expresan que la sincronización puede involucrar cierto adelanto o retraso (*leadORlag*) entre los eventos que asocian. La navegación sobre los roles *predecessor* y *successor* permite escribir expresiones compactas para estas reglas.

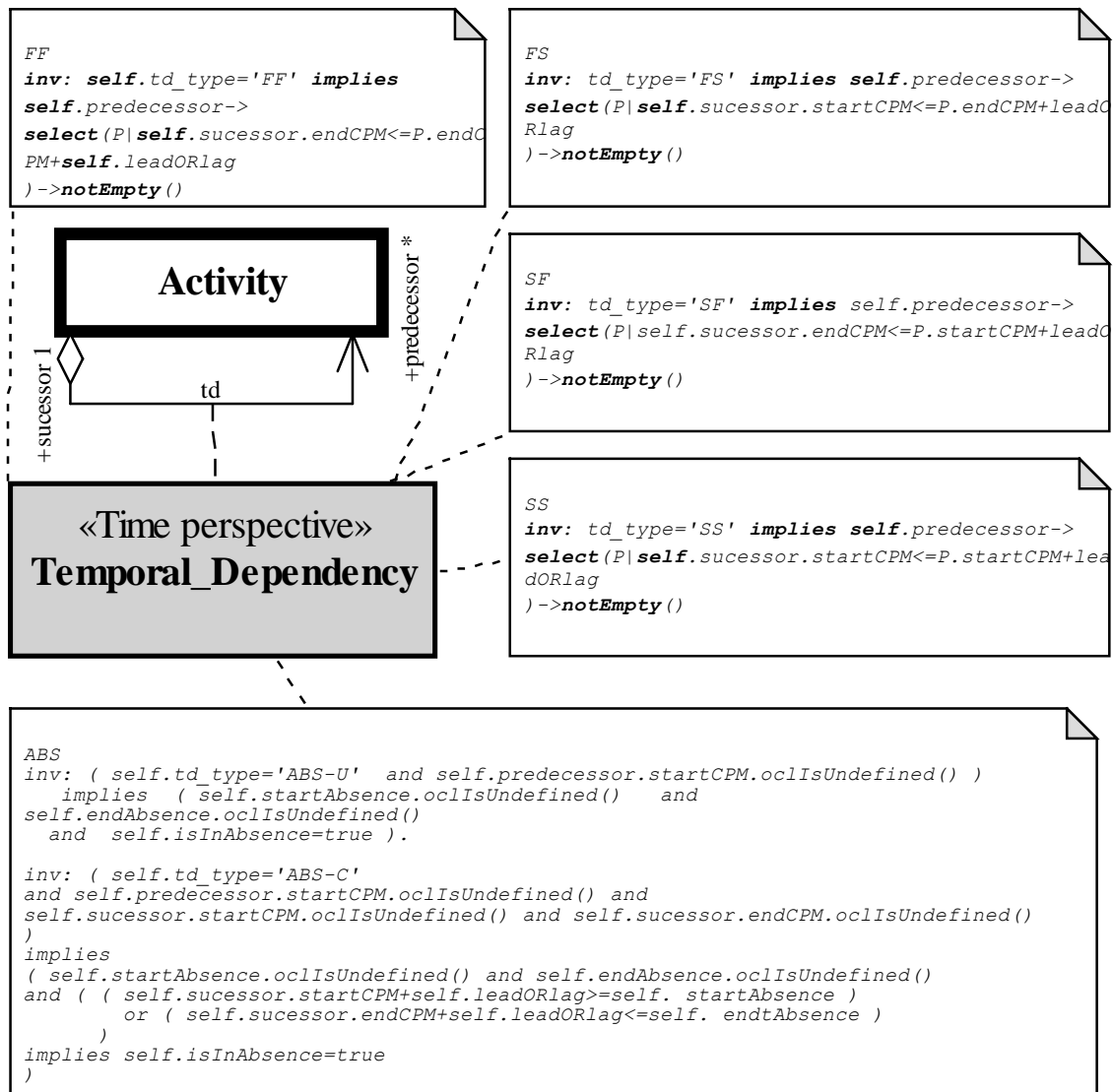


Figura. 6.17. Dependencias temporales OCL

**Tabla. 6.7.** Dependencias temporales OCL

# tr	OCL rule specification
FS	<u>Context Temporal Dependency inv:</u> self.td_type='FS' implies self.predecessor→select( P  self.successor.startCPM<=P.endCPM+leadORlag )→notEmpty()
FF	<u>Context Temporal Dependency inv:</u> self.td_type='FF' implies self.predecessor→select( P  self.successor.endCPM<=P.endCPM+leadORlag )→notEmpty()
SF	<u>Context Temporal Dependency inv:</u> self.td_type='SF' implies self.predecessor→select( P  self.successor.endCPM<=P.startCPM+leadORlag )→notEmpty()
SS	<u>Context Temporal Dependency inv:</u> self.td_type='SS' implies self.predecessor→select( P  self.successor.startCPM<=P.startCPM+leadORlag )→notEmpty()
ABS	<u>Context Temporal Dependency inv:</u> ( self.td_type='ABS-U' and self.predecessor.startCPM.oclIsUndefined() ) implies ( self.startAbsence.oclIsUndefined() and self.endAbsence.oclIsUndefined() and self.isInAbsence=true ).  <u>Context Temporal Dependency inv:</u> ( self.td_type='ABS-C' and self.predecessor.startCPM.oclIsUndefined() and self.sucessor.startCPM.oclIsUndefined() and self.sucessor.endCPM.oclIsUndefined() ) ) implies ( self.startAbsence.oclIsUndefined() and self.endAbsence.oclIsUndefined() and ( ( self.sucessor.startCPM+self.leadORlag>=self.startAbsence ) or ( self.sucessor.endCPM+self.leadORlag<=self.endtAbsence ) ) ) implies self.isInAbsence=true )

**ABS (ABS-U, ABS-C).** La ejecución de la actividad *predecesora* causa que la *sucesora* entre en estado de ausencia (*ABS-U*) incondicional; dicho de otro modo, esta actividad no recibe flujo de control cuando llega el token. La segunda versión del invariante es una ausencia condicionada (*ABS-C*); en este caso, la sucesora sólo puede recibir control dentro de un intervalo de tiempo [*startAbsence*, *endAbsence*]. En ambos casos puede especificarse un adelanto o retraso entre los eventos dependientes (*leadORlag*).

## 6.6 Resumen y conclusiones

En este Capítulo se ha profundizado en la especificación de reglas temporales sobre *procesos* propuestas por distintos autores sobre lenguajes para la especificación de procesos. El análisis de sus distintas reglas permite ofrecer un primer resultado que es un taxonomía de *reglas* clasificadas en restricciones *intra-actividad* y *dependencias inter-actividades*. Esta taxonomía permite analizar debilidades en la representación de la dimensión temporal de estándares como *BPMN*, reconocida por algunos autores (otros lenguajes de procesos, como *UML AD* o *XPDL*, o los específicos del software como *SPEM*, *Iso/Iec 24744* y *NDTQ-Framework* manifiestan el mismo tipo de debilidades).

Se han analizado enfoques de autores que proponen alternativas basadas en patrones [*Flores & Sepúlveda 2011*] y en extensiones con decoradores sobre *BPMN 2.0* [*Gagné &*

*Trudel 2009a; Cheikhrouhou et al. 2013b*]. La propuesta basada en patrones respeta el estándar pero sobrecarga bastante los modelos obtenidos, mientras que la segunda propuesta ofrece mayor elegancia y simplicidad, pero se sale del alcance de la versión actual *BPMN 2.0* para representar la perspectiva temporal. Se han estudiado los *Meta-Modelos* de procesos de *UML Activity Diagrams*, de *BPMN 2.0* y de *XPDL 2.2.*, lo cual ha servido para extraer un conjunto mínimo de artefactos que nos han permitido proponer un segundo resultado: el *Meta-Modelo UML* (denominado *PIM-up*) que tiene en cuenta la taxonomía de reglas anterior, formalizado las reglas temporales con el lenguaje de especificación *OCL*.

El *Meta-Modelo PIM-up* (nivel de abstracción *PIM*) formaliza la especificación de la dimensión temporal y no sobrecarga el flujo de control principal de un proceso. Este *Meta-Modelo* es la base para desarrollar distintas extensiones que resolverán los requisitos planteados a lo largo del trabajo de investigación.

# Capítulo 7. EXTRACCIÓN DE PROCESOS DESDE LEGACY DATABASES

*"Descifrar lo que está delante de nuestros ojos requiere una lucha constante"*  
*George Orwell (1903-1950)*

Ha quedado argumentado que *BPM* es una ventaja competitiva para todo tipo de organizaciones y el sector del software no es una excepción. Los *legacy systems* coexisten con nuevos sistemas especializados para la gestión de procesos (*BPMS*). Implantar *BPM* supone manejar procesos y modelos de procesos en un ciclo de mejora continua, donde es necesario extraer y mantener estos modelos. En el Capítulo 5 se ha estudiado la relación entre dimensiones de los procesos y *legacy systems*, observando que los de *planificación y control de proyectos* resultan ser los que *mejor incluyen la perspectiva temporal*; *MS Project* es un buen ejemplo de *legacy system* para extraer dicha perspectiva y representarla como procesos de negocio. En este Capítulo se desarrolla una propuesta basada en ingeniería inversa *MDE* que analiza la *legacy database* de *MS Project Server*, y obtiene aproximaciones a los procesos de negocio en un lenguaje cercano al experto en software, como *BPMN*, sin menoscabo de la extensión del enfoque a otros *legacy systems* y sobre otros *lenguajes de especificación de procesos*.

## 7.1 Enfoque MDE para la extracción de procesos

«*Process Mining Manifesto*» [van der Aalst et al. 2012] trata los aspectos de «*process mining*» [van der Aalst et al. 2007, 2012; van der Aalst 2011, 2013, 2014, 2015; Reichert & Weber 2012]. Contempla distintas perspectivas para procesos de negocio: *flujo de control, datos o flujo de información, tiempo, recursos y casos*. En la tesis se propone utilizar un enfoque dirigido por modelos para extraer, inicialmente, la *dimensión temporal* de cada *legacy system*, aunque estos sistemas, no siempre, estén orientados a procesos. La tipología de *legacy systems* utilizados por una organización del sector del software puede ser diversa (ver Capítulo 5). La *dimensión temporal* está presente en cualquier proceso de negocio y, por tanto, también en estos sistemas; es común contemplar para cualquier actividad (tareas y/o subprocesos) fechas de comienzo, vencimientos o fechas tope, así como dependencias temporales entre actividades que, de entrada, condicionan el flujo de control de las mismas en un proceso de negocio. Estas reglas pueden aparecer en sistemas de información de diversas categorías: a) *planificación y control de proyectos*

[Hansen & Hansen 2013; Stover 2007; Lang 2010] (este tipo es el que *mejor recoge la dimensión temporal*; ver Tabla 6.1); b) en los circuitos de aprobación de documentos contemplados en un ECM [Shariff 2013; Smith & Bates 2007] o en la gestión de pedidos, facturas, pagos, etc. de un ERP, CRM o SCM [Hendricks et al. 2007], y, por último, también en cualquier desarrollo a medida.

Existe trabajo relacionado (ver Capítulo 4) con la modernización de *legacy systems*, la ingeniería inversa [Chikofsky & Cross 1990] de bases de datos relacionales y la extracción de procesos de negocio desde *legacy systems*, pero la mayoría se centra en la transformación de tablas en clases y la representación de reglas de negocio simples, no ahondando en la obtención de estas perspectivas, ya que los sistemas origen no contienen modelos de procesos. [Boronat et al. 2005] es un precedente interesante de ingeniería inversa de bases de datos relacionales, dónde se reescriben fórmulas lógicas en dos niveles de abstracción distintos. En este trabajo de tesis se pone el foco en la dimensión temporal de cada *legacy system*, con objeto de elevar dicha semántica al nivel de abstracción inteligible por el experto en software. Se utilizará como base el *Meta-Modelo PIM-up* (Figura 6.11), que soporta las reglas temporales de la taxonomía de la Figura 6.1, extendiéndolo para alcanzar los requisitos planteados.

La arquitectura MDA [OMG:MDA; Kleppe et al. 2003], dentro del paradigma MDE [Kent 2002; Schmidt 2006] ofrece la facilidad «*Meta-Object Facility (MOF)*» [OMG:MOF 2011] para soportar los procesos dirigidos por modelos. Se consideran distintos niveles de abstracción para representar modelos y *Meta-Modelos* y, así, facilitar un alto nivel de *interoperabilidad*, basada en el concepto *MOF*.

La Figura 7.1 muestra en los niveles más bajos de abstracción los *casos o estados* del mundo real (nivel M<sub>0</sub>) o de los sistemas. Estos casos pueden estar representados mediante el código de las aplicaciones o los estados de una base de datos. A un nivel mayor de abstracción (nivel M<sub>1</sub>) figura el concepto de *modelo*; p. ej.: modelos basados en UML o en BPMN.

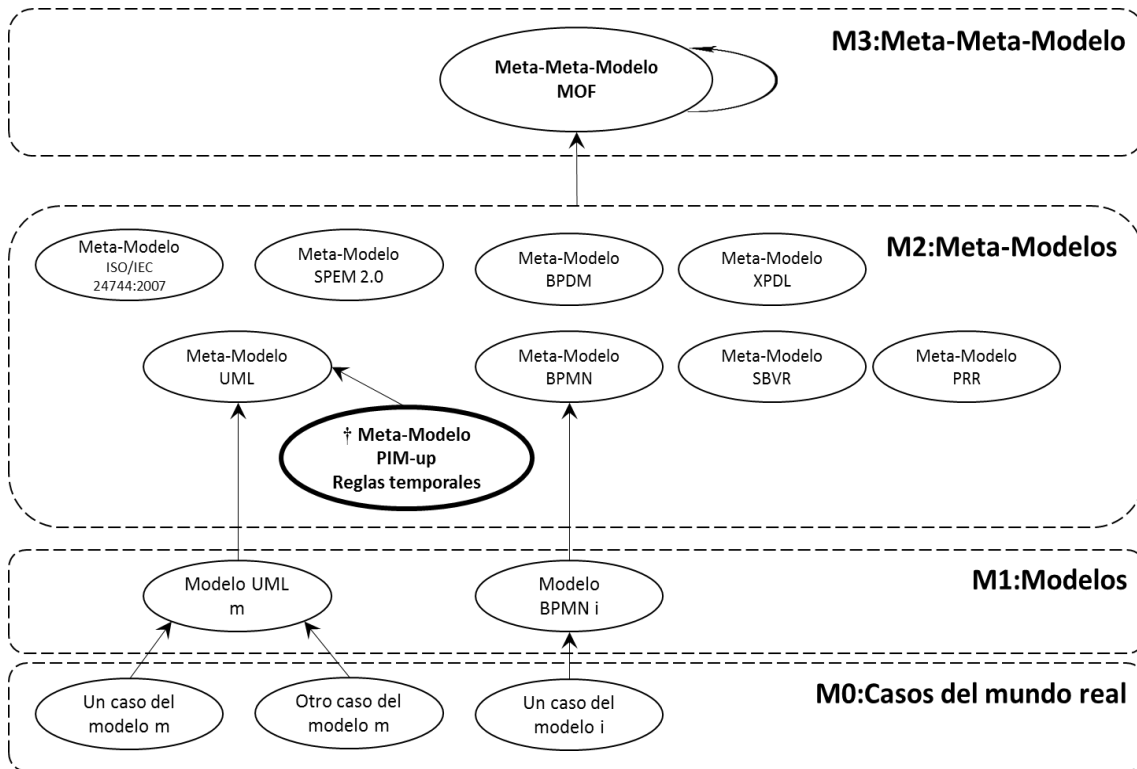


Figura. 7.1. MOF e ingeniería inversa aplicado al enfoque

En el tercer nivel (nivel M<sub>2</sub>) se sitúa el concepto de *Meta-Modelo* para expresar la *sintaxis abstracta* de cualquier modelo especificado mediante una *sintaxis concreta*; en este nivel pueden existir, a su vez, distintos niveles de abstracción, en función de si el *Meta-Modelo* está más cercano a la plataforma real, al *técnico TI* o al *experto de negocio* [van der Straeten et al. 2009; Brambilla et al. 2012], apareciendo así distintas plataformas: «*Platform Specific Meta-Model (PSM)*», «*Platform Independent Meta-Model PIM*» y «*Computer Independent Meta-Model CIM*» (Figura 7.1). En este nivel se sitúa el *Meta-Modelo PIM-up* propuesto en el Capítulo 6 para contemplar la dimensión temporal de los procesos, que se extenderá en este Capítulo. Por último (nivel M<sub>3</sub>), el concepto de *Meta-Meta-Modelo* que permite definir cualquier *Meta-Modelo*.

La Figura 7.2 ilustra la aplicación del concepto *MOF* al proceso de ingeniería inversa que se va a proponer basado en *MDE*. A nivel M<sub>0</sub>, se consideran los distintos sistemas de información (cada uno considerado como un *legacy system*), que bien pueden ser productos de mercado para la planificación y control de proyectos (p. ej.: *Project Server* o *Redmine*), productos para la gestión de contenidos (*ECMs*) (como *Alfresco*<sup>TM</sup> o *Sharepoint*<sup>TM</sup>), *ERPs* (o *CRMs*, o *SCMs*) o desarrollos a medida. El foco de observación va a ser la capa de persistencia de cada sistema; hoy en día, la mayoría están soportados por *BD relacionales* [Codd 1970, 1992], p. ej.: *Oracle*<sup>TM</sup>, *SQL\*Server*<sup>TM</sup>, *mySQL*<sup>TM</sup>, *PostgreSQL*<sup>TM</sup> o *Hana*<sup>TM</sup> [Färber et al. 2012] en el caso del *ERP SAP*<sup>TM</sup>, situados en el nivel M<sub>0</sub>.

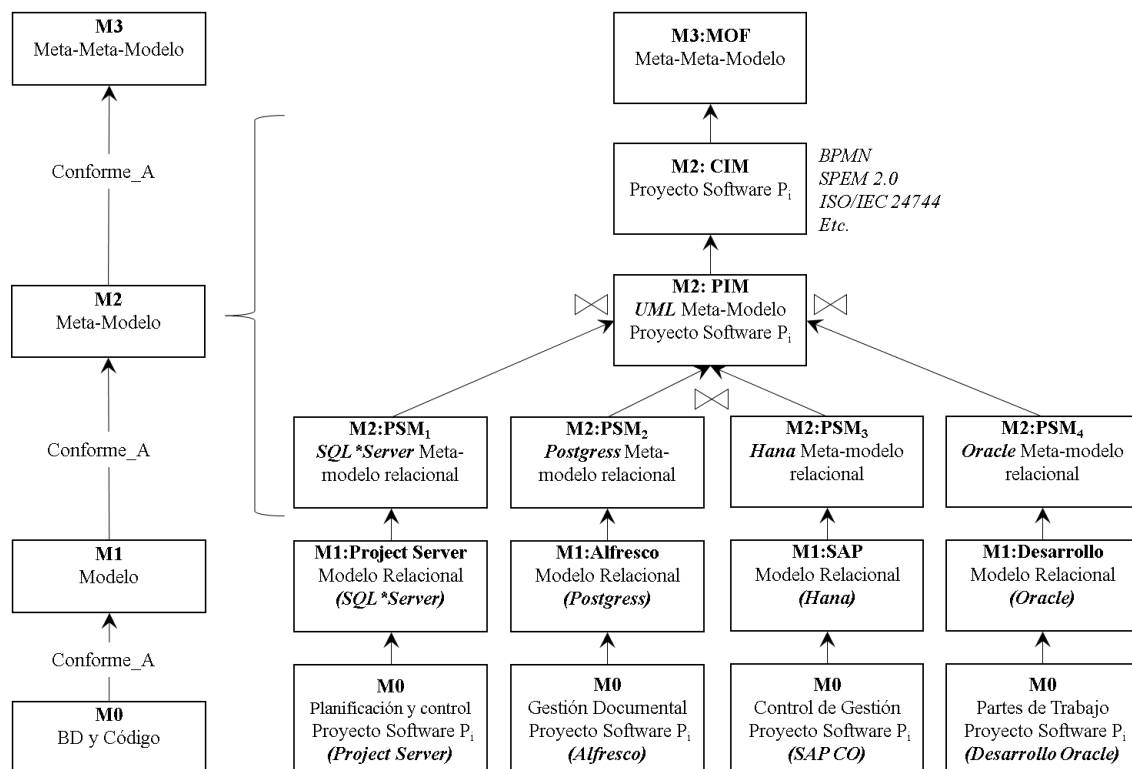


Figura. 7.2. Plataformas para ingeniería inversa

A nivel M<sub>2</sub>, es necesario establecer un primer nivel de *Meta-Modelo*, cercano a la plataforma tecnológica (*PSM*). Existen propuestas de *Meta-Modelos* del *modelo relacional* en «*Atlanmod Zoos*» [Atlanmod 2015] en formatos *Ecore* [Merks 2010; Eclipse:EMF 2015] y «*Kernel Meta-Meta-Model*» [Eclipse:KM3 2006] aunque son muy básicas. También existen en «*Information management metamodel (IMM)*» [OMG:IMM] que extiende a «*OMG Common Warehouse Metamodel (CWM)*» [OMG:CWM 2003; Bruneliere et al. 2010a; Pérez-Castillo et al. 2012b], presentando *perfiles UML2* [Alhir 2006; Giachetti et al 2008] para representar los artefactos (tablas, restricciones y rutinas) de los esquemas relacionales. [Arevalo et al. 2013] extienden el *Meta-Modelo* relacional [OMG:IMM] para soportar restricciones [Türker & Gertz 2001] sobre tablas y disparadores sobre la base de la gramática «*SQL:1999*» [Melton & Simon 2002] y «*SQL Persistent Storage Modules SQL/PSM*» [Eisenberg 1996], donde los triggers son un subtipo de procedimientos almacenados especificados en un lenguaje propietario derivado de *SQL/PSM*, como «*PL/SQL*» [Feuerstein & Pribyl 2005], «*Transact\*SQL*» [Colledge 2008; Ben-Gan 2012; Microsoft™:T-SQL 2014], «*PL/pgSQL*» [PL/pgSQL 2014] o «*mySQL Stored Procedures*» [Harrison & Feuerstein 2008]. En [Arevalo 2014; Arevalo et al. 2013, 2015a, 2015b] se proponen enfoques *MDE* para extraer reglas de negocio desde *BD relacionales*.

En el mismo nivel M<sub>2</sub>, pero cercano al técnico o analista en *IT*, como plataforma *PIM*, se va a utilizar un *Meta-Modelo de procesos* que soporte las reglas temporales de la taxonomía (Figura 6.1), y se desarrollará en la siguiente sección. Este *Meta-Modelo*



se especificará en *UML* y las reglas temporales en *OCL*. El foco es la modelación temporal y el soporte para la representación de modelos que vienen de un *legacy system*. Por último, como plataforma *CIM* destino, pueden considerarse distintos *Meta-Modelos* utilizados en la gestión de procesos de negocio, generales como *BPMN 2.0* [*OMG: BPMN 2013*; *Iso/Iec:19510 2013*] y específicos para el software como *SPEM 2.0* [*Bendraou et al. 2007*; *OMG: SPEM 2008*] e *Iso/Iec:24744* [*Iso/Iec:24744 2007, 2014*]. Desde la plataforma *PIM* utilizada queda abierta la posibilidad de llegar a otros *Meta-Modelos*.

## 7.2 Meta-Modelo de procesos (PIM-upl)

Los modelos obtenidos mediante otras propuestas basadas en *MDE* (ver trabajo relacionado en Capítulo 4) para hacer *ingeniería inversa* [*Chikofsky & Cross 1990*] de *bases de datos relacionales* capturan bien las tablas, claves ajenas y algunas reglas de negocio muy simples, pero, la mayoría, no ahondan en el concepto de proceso de negocio y sus perspectivas [*van der Aalst et al. 2012*]. En [*Boronat et al. 2005*] existe un precedente interesante que utiliza reescritura de fórmulas lógicas como soporte en el proceso de extracción de esquemas desde *legacy systems*. En este sentido, se pretende utilizar la semántica del *Meta-Modelo PIM-up* (Figura 6.11) que soporta las reglas temporales de la taxonomía (Figura 6.1). Este *Meta-Modelo*, cuyos artefactos están contenidos en el paquete «*Semantic*», se va a extender para soportar el proceso de ingeniería inversa basada en *MDE* que permitirá extraer la semántica temporal de *legacy systems*.

La Figura 7.3 muestra la extensión del *Meta-Modelo PIM-up*. Se añade el paquete «*Interchange*» (al igual que se diferencia en los *Meta-Modelos* estándar la semántica de las clases destinadas al intercambio de diagramas) con las clases «*LIS System*» «*Diagram*» «*Lane*» y «*Pool*», extraídas en la Tabla 5.5 que compara *Meta-Modelos* de procesos estándar. Se representa el diagrama de clases de este *Meta-Modelo UML con reglas temporales* que, en adelante, se denomina «*Meta-Modelo pivote PIM-upl*», porque se situará en ese nivel de abstracción *MDE* para soportar el proceso de ingeniería inversa desde los *Meta-Modelos* de procesos *PSM* que describen la capa de persistencia de cada *legacy system*.

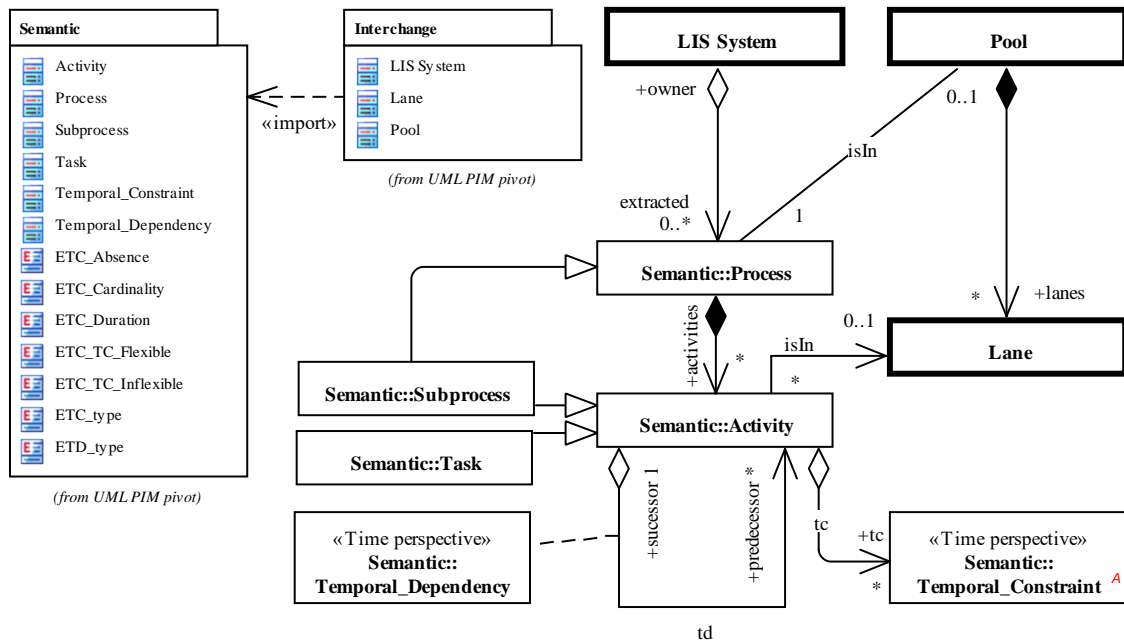


Figura. 7.3. Meta-Modelo pivote PIM-upl para ingeniería inversa desde legacy systems

La clase *LIS System* determina la procedencia de un modelo de procesos; es decir de qué *legacy system* o instancia viene. Un proceso, opcionalmente, puede situarse en una piscina (clase *Pool*) mediante la asociación *isIn*, por tratarse de procesos de negocio dentro de la misma organización. De manera similar, una actividad (clase *Activity*) puede situarse opcionalmente en una calle (clase *Lane*) de la piscina.

La utilización de *sintaxis abstracta* de *Meta-Modelos* y especificación de reglas *OCL* supone manejar un *enfoque declarativo* en detrimento de una *sintaxis concreta* y *lenguajes imperativos* [Fahland 2009a, 2009b; Pichler et al. 2011; Reijers et al. 2013]. Esto otorgará un mayor nivel de *interoperabilidad* [Geraci 1991] a los modelos obtenidos al aplicar métodos basados en *MDE* [Milanovic 2009; Elvesæter et al. 2006, 2010], pues se busca independizar, en la medida de lo posible, la heurística de extracción de la notación de representación de procesos. Este *Meta-Modelo*, que contiene un mínimo de clases para expresar la dimensión temporal, permite simplificar el proceso de extracción y la definición de nuevas restricciones que se identifican en las secciones siguientes, otorgando un mayor nivel de independencia de la solución y evitando tener que replicar la resolución de estos problemas en cada plataforma destino.

### 7.3 Plataforma origen piloto: MS Project Server

Para resolver la aplicación del proceso de extracción mediante ingeniería inversa *MDE* se ha elegido inicialmente *MS Project Server* porque es una herramienta de planificación y control de proyectos (*Sistemas que más artefactos tienen vinculados a la perspectiva temporal BPM* (ver Capítulo 5): Tablas 5.1, 5.2 y 5.3) común en el mercado, es utilizada frecuentemente por las organizaciones del software y, que ha servido para

aplicarla al caso real (AQUA-WS Project) [Cutilla et al. 2012] (ver Capítulo 8).

### 7.3.1 Arquitectura de MS Project Server

MS Project Server [Stover 2007] es un producto Microsoft™ con una robusta arquitectura (Figura 7.4):

- a) Permite trabajar con clientes basados en formularios Web (páginas .aspx) para crear y modificar proyectos, a la vez que permite intercambiar proyectos con MS Project Professional [Hansen & Hansen 2013] y con aplicaciones de terceras partes a través de SOAP.
- b) Los módulos de servidor facilitan la gestión de contenidos con Windows SharePoint Services y la lógica en servidor con páginas aspx. El módulo de planificación y control, que soporta los métodos PERT Tiempos y PERT Costes [Malcolm et al. 1959; PMI:PMBOK 2013], aplica el algoritmo CPM [Kelley & Walker 1959] para el cálculo del camino crítico de un proyecto (scheduling engine).

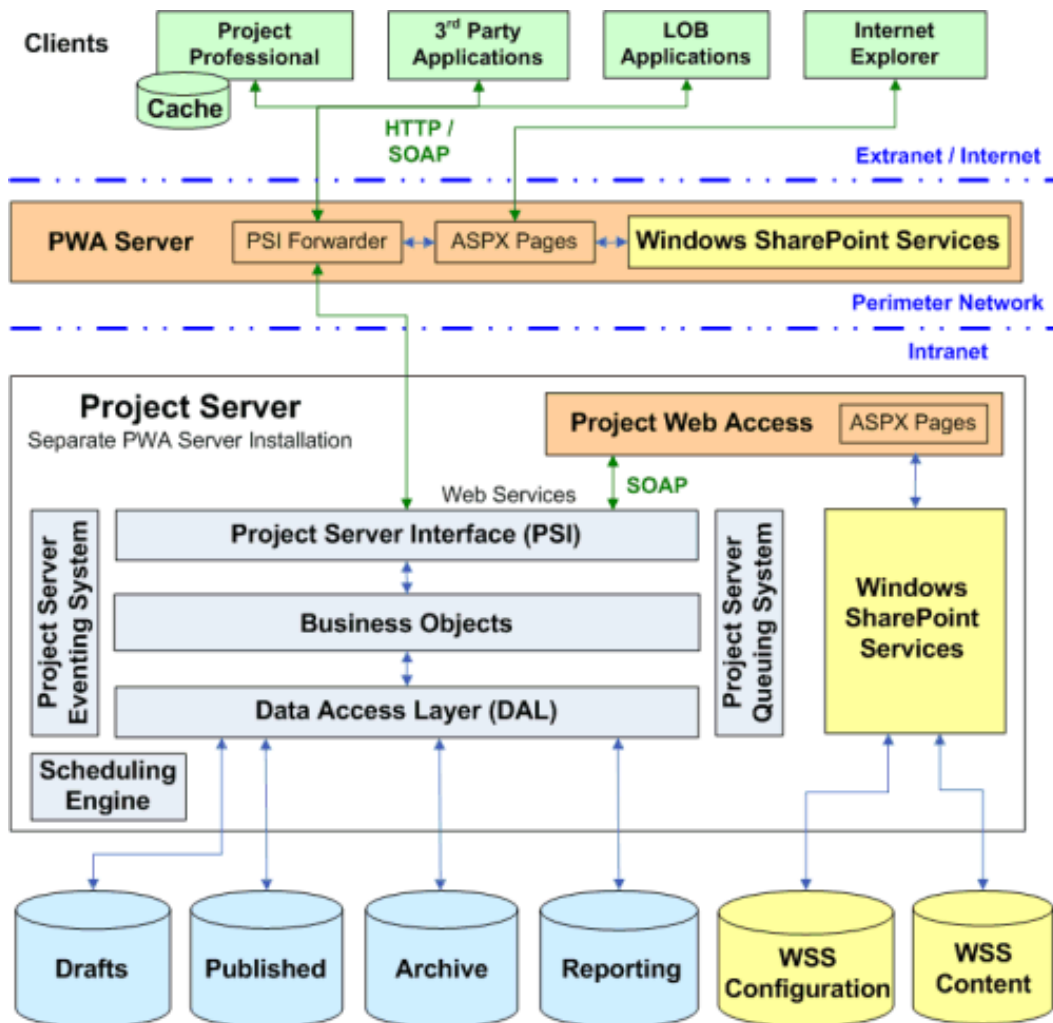


Figura. 7.4. Arquitectura de MS Project Server

- c) La capa de persistencia está soportada por el SGBD MS SQL\*Server™ [Colledge 2008; Ben-Gan 2012; Microsoft™: T-SQL 2014]. Una instalación configura cuatro tipos de instancias: *Drafts*, *Published*, *Archive* y *Reporting*. *Drafts* contiene los proyectos activos que gestiona cada responsable. *Published* contiene las versiones de proyectos que se publican en la web y soporta la lógica de control de autorizaciones. *Archive* contiene la base de datos de proyectos finalizados. Y, por último, *Reporting* es una instancia que toma datos de las otras BD para facilitar al usuario la extracción de información y generación de informes.

### 7.3.2 Meta–Modelo PSM de tareas

Tomando como referencia la BD relacional almacenada sobre la instancia *Published* (Ver anexo II.b), se han extraído las tablas que están vinculadas a la vista de proyectos, tareas y restricciones temporales (Figura 7.5).

La tabla *MSP\_PROJECTS* contiene los datos del proyecto: *PROJ\_UID* y *PROJ\_NAME*, los eventos planificados (*OPT\_DEF\_START\_TIME*, *OPT\_DEF\_FINISH\_TIME*) y actuales (*PROJ\_INFO\_START\_TIME*, *PROJ\_INFO\_FINISH\_TIME*) de inicio y finalización y la duración calculada por el método CPM (*OPT\_EST\_DUR*).

*MSP\_TASKS* contiene las tareas de un proyecto, donde se especifican las fechas planificadas: *TASK\_START\_DATE* y *TASK\_FINISH\_DATE* y las fechas actuales de la tarea si ha comenzado: *TASK\_ACT\_START* y *TASK\_ACT\_FINISH*. La clave ajena *Fk\_Msp\_Tasks\_Project\_Parent* liga al proyecto con las tareas que lo componen. Si un proyecto es además un subproyecto, podrá ser incluido también como una tarea de un proyecto de mayor nivel; esto se resuelve con la clave ajena *FK\_TaskIsSubproject*. Por otro lado, si una tarea es una tarea de grupo, puede contener una lista de tareas de menor nivel; esto se expresa mediante la clave ajena *FK\_Task\_Parent*. La tabla *MSP\_LINKS* almacena la dependencia temporal (relación de precedencia; ver Tabla 7.2) entre una tarea predecesora y una tarea sucesora; existen dos claves ajenas para resolver estas dependencias *FK\_Predecessor* y *FK\_Successor*.

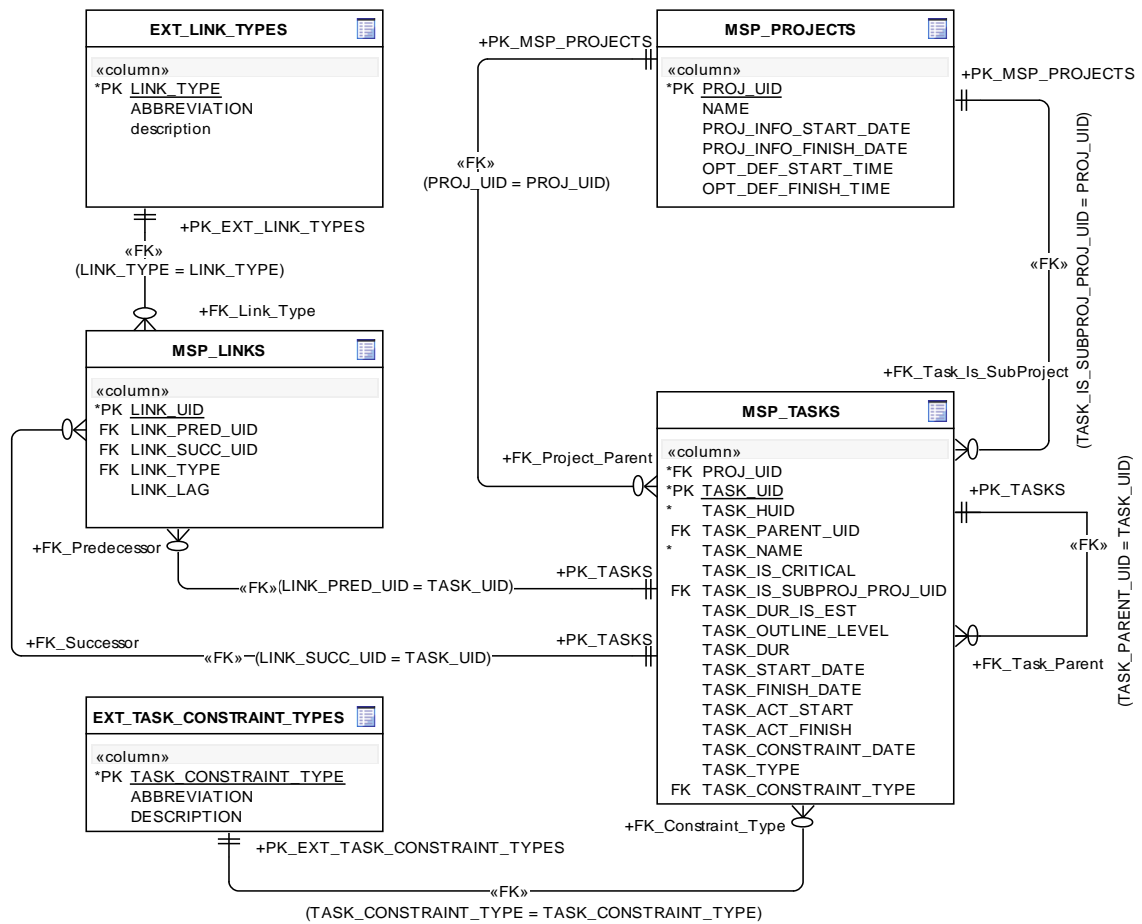


Figura. 7.5. Meta-Modelo de tareas MS Project Server

En cuanto a la especificación de restricciones, *MP\_TASKS* puede almacenar restricciones sobre la duración con el atributo *TASK\_DUR\_IS\_EST* (*true/false*) para modelar que la duración de la tarea es estimada y fija en caso contrario. También se puede especificar una restricción de tiempo sobre el inicio o finalización de la tarea (ver Tabla 7.1). Si no se especifica restricción de tiempo, por defecto se asigna *ASAP* para el inicio y terminación de la tarea.

Las Tablas 7.1 y 7.2 se han extraído de la documentación de *MS Project Server*. *Constraint\_Type* (Tabla 7.1) fija el tipo de restricción temporal aplicable en la tabla *MSP\_TASKS*. Los valores que puede tomar el atributo *MSP\_TASKS.CONSTRAINT\_TYPE* están fijados por *Short Value* en esta tabla. Hay que observar que es necesario especificar una fecha límite, *MSP\_TASKS.CONSTRAINT\_DATE*, en los casos *MSO*, *MFO*, *SNET*, *SNLT*, *FNET* y *FNLT* y no es necesaria para los casos *ASAP* y *ALAP*. Se ha extendido el *Meta-Modelo* de tareas de *MS Project Server*, añadiendo la tabla *EXT\_TASK\_CONSTRAINT\_TYPES* (en correspondencia con la Tabla 7.1) e interrelacionándola con *MSP\_TASKS*.

**Tabla. 7.1.** Tabla Constraint\_Type (MS Project Server)

Short Value	Abbreviation	Description	Constraint Date required
0 (Default)	ASAP	As soon as possible	No
1	ALAP	As late as possible	No
2	MSO	Must start on	Yes
3	MFO	Must finish on	Yes
4	SNET	Start no earlier than	Yes
5	SNLT	Start no later than	Yes
6	FNET	Finish no earlier than	Yes
7	FNLT	Finish no later than	Yes

La Tabla *LINK\_TYPE* (Tabla 7.2) muestra las dependencias temporales o relaciones de precedencia entre tareas. El atributo *MSP\_LINKS.LINK\_TYPE* toma los valores *int Value* de esta tabla. Obsérvese que, por defecto, la restricción es *FS* (*intValue=1*). Se ha extendido el *Meta-Modelo* de tareas de MS Project Server añadiendo la tabla *EXT\_LINK\_TYPES* (en correspondencia con la Tabla 7.2) e interrelacionándola con la tabla

**Tabla. 7.2.** Tabla Link\_Type (MS Project Server)*MSP\_LINKS.*

intValue	Name	Abbreviation	Description
0	FinishFinish	FF	The task identified by LINK_SUCC_UID cannot finish until the task identified by LINK_PRED_UID finishes.
1 (Default)	FinishStart	FS	The task identified by LINK_PRED_UID must finish before the task identified by LINK_SUCC_UID can start. This is the default dependency type.
2	StartFinish	SF	The task identified by LINK_PRED_UID must start before the task identified by LINK_SUCC_UID finishes. This is the most uncommon of the four dependency types. For example, in a scenario where you are setting a fence post, the successor is <b>Pour Cement</b> and the predecessor is <b>Set Post</b> . The successor ( <b>Pour Cement</b> ) must start first, however, the predecessor ( <b>Set Post</b> ) must finish before the successor ( <b>Pour Cement</b> ) finishes.
3	StartStart	SS	The task identified by LINK_SUCC_UID cannot start until the task identified by LINK_PRED_UID starts.

## 7.4 Extracción de procesos desde Legacy Databases

En el Capítulo 2 se ha planteado la arquitectura *MDE* del proceso de ingeniería inversa que debe transformar la *dimensión temporal* de una *legacy database* (nivel *PSM*) en un modelo de negocio cercano al experto en software (nivel *CIM*). En este esquema se propone partir de la capa de persistencia de cada *legacy system*, observando la *perspectiva temporal* en determinados artefactos del *Meta-Modelo* de cada base de datos relacional. En los niveles de abstracción utilizados para cada plataforma se descartarán estructuras y reglas propias de otras perspectivas *BPM* o aquellas estructuras accesorias en los *Meta-Modelos* considerados. En esta memoria se aborda como caso de estudio y sistema origen *MS Project Server* para obtener modelos *PSM* (Sección 7.3) basados en el *Meta-Modelo relacional* [Codd 1970, 1992].

La *heurística* para realizar dicha transformación se basará en la aplicación sucesiva de distintos procesos de abstracción *MDE*. La *heurística* se basará en el *Meta-Modelo de procesos pivote PIM-upl* (Figura 7.3) con *reglas temporales* (que se sitúa a nivel *PIM*). Se generará una primera capa del proceso de negocio, que representará un mapa de actividades, con características de flujo de control condicionado por las *reglas temporales*.

El enfoque que se considerará plantea dos tipos de transformación: a) *PSM*→*PIM* y b) *PIM*→*CIM*.

- i) **Transformación *PSM*→*PIM***. Esta transformación permite capturar la dimensión temporal de un *legacy system* en un modelo independiente de cualquier plataforma y conforme al *Meta-Modelo PIM-upl* y reglas temporales *OCL*. Inicialmente se está atacando la dimensión temporal, pero posteriormente, este *Meta-Modelo* podría ampliarse con otras perspectivas, como la organizacional y la de casos, utilizando trabajos como [Awad et al. 2009; Stroppi et al. 2012] que generan un *Meta-Modelo* basado en *BPMN* que captura estas dimensiones para establecer invariantes *OCL* que modelan hasta veintiocho patrones *Workflow Resource Patterns (WRPs)* [Gamma et al. 1995; Tran et al. 2007]. Sus *Meta-Modelos* podrían fusionarse con el *Meta-Modelo PIM-upl*.
- ii) **Transformación *PIM*→*CIM***. Esta transformación permite llegar a modelos independientes de la computación y cercanos al experto en el negocio del software, bien basados en un estándar de propósito general como *BPMN 2.0* [OMG:BPMN 2013; Iso/Iec:19510 2013], o bien en estándares específicos para la gestión de los procesos software como *SPEM 2.0* [Bendraou et al. 2007; OMG: SPEM 2008] e *Iso/Iec:24744* [Iso/Iec:24744 2007, 2014]. Pero también, desde el nivel *PIM* podría plantearse llegar a otros *Meta-Modelos* destino e incluso a sintaxis concretas que propusiera el experto.

#### 7.4.1 Extracción de modelos *PSM*→*PIM-upl*

La Figura 7.6 ilustra la transformación *PSM*→*PIM*. Se parte de un *legacy system* como caso de uso. En esta memoria se ha elegido inicialmente *MS Project Server* (por las razones aportadas en el Capítulo 5). Cada caso de uso requiere levantar un modelo de procesos con la dimensión temporal; para *MS Project* se ha realizado en la Sección 7.3, soportado por su correspondiente *Meta-Modelo relacional*.

Esta primera transformación generará un modelo de procesos *PIM pivote*, conforme a su *Meta-Modelo PIM-upl* (Figura 7.3), por lo que este modelo destino goza ya de las estructuras y formalización de las reglas temporales del *Meta-Modelo* en cuestión en el *lenguaje declarativo OCL*, situándose a un nivel mayor de abstracción y formalización de las reglas temporales. La *heurística* de transformación de cada *legacy database* a *PIM* debe tener en cuenta:

- i) Debe resolverse la detección de actividades (tareas elementales y subprocessos) y sus agrupaciones para el mapeo, como la pertenencia de una actividad a proceso o subprocesso).
- ii) Asignación de propiedades: específicas y/o por defecto (p. ej.: *ASAP* en inicio y terminación).
- iii) Detección de dependencias temporales específicas entre actividades, conforme a la taxonomía de reglas (Figura 6.1).
- iv) Asignación de procesos ad-hoc para agrupar actividades que queden aisladas o sin dependencias.

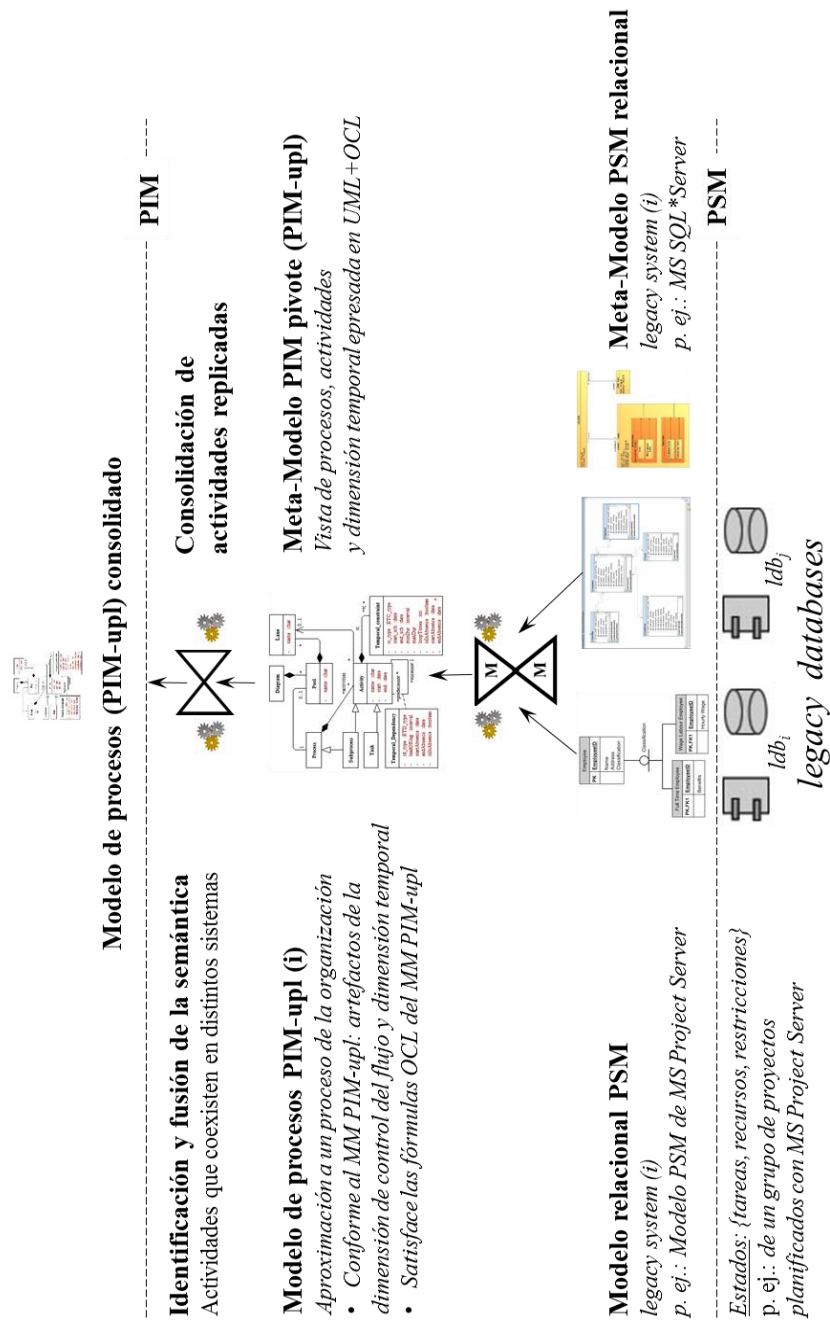


Figura. 7.6. Proceso de extracción de la dimensión temporal PSM→PIM



En los siguientes apartados se planteará un *Algoritmo* que describe la heurística y las reglas detalladas de mapeo entre *Meta-Modelos*.

#### 7.4.1.1 Algoritmo

El origen son las tablas del *Meta-Modelo* de procesos de *MS Project Server* (Figura 7.5). El destino es el *Meta-Modelo* de procesos *UML+OCL pivote PIM-upl* (Figura 7.3). La Figura 7.7 muestra el algoritmo general para extraer el modelo de negocio de cada proyecto existente en la *legacy database* origen. En la Sección siguiente figuran tablas de mapeo de las *legacy databases* origen a las clases destino.

```

Ψ≡{Proyectos seleccionados para realizar ingeniería inversa a modelo de procesos desde BD Published}
∀(Pi:Proyecto) ∈ Ψ {
    CREAR Proceso de negocio asociado al proyecto Pi
    /* Aplicando reglas de mapeo de las Tabla 7.3 (Proceso) */
    Si el Proyecto es un SubProyecto
        CREAR Aj:Actividad --El proyecto también es una Actividad (Tabla de mapeo 7.5).
        CREAR Subproceso -- (Tabla de mapeo 7.4 para SubProcess)
    Fin

    ∀(Ti:Tarea) ∃ (Pi:Proyecto) {
        CREAR Ai:Actividad que corresponde a Ti:Tarea y forma parte del proceso de negocio
        /* Aplicando reglas de mapeo de la Tabla de mapeo 6.5 para Activity*/
        Si la Tarea es una Tarea en Grupo
            CREAR Proceso -- (Tabla de mapeo 7.3 y datos de la fila del Proyecto vinculada)
            CREAR Subproceso -- (Tabla de mapeo 7.4)
        Si no existen precedencias entre sus tareas
            CREAR Subproceso AdHoc -- (Tabla de mapeo 7.4)
        Fin
    }
    Fin

    CREAR restricción de duración de la Actividad
    (Fija o Flexible, según atributo TASK_DUR_IS_EST)←(Fija o estimada )en la Tarea
    --(Tabla de mapeo 7.6)
    ∀(RTi:Restricción temporal) ∃ (Ti:Tarea) -- (Tabla de mapeo 7.7)
    {CREAR restricción de tiempo (Temporal_Constraint) para la Actividad (Activity)}
    ∀(TPj:Tarea predecesora de Ti:Tarea) -- (Tabla de mapeo 7.8)
    {CREAR dependencia temporal (Temporal_Dependency) para la Actividad (Activity)}
}
    
```

Figura. 7.7. Algoritmo para extracción de la dimensión temporal PSM→PIM

Partiendo de la instancia *Published* de la capa de persistencia de *MS Project Server* se determina un conjunto de proyectos planificados a los que se someterá al proceso; El algoritmo de la Figura 7.7 generará un modelo de negocio para cada proyecto conforme al *Meta-Modelo PIM-upl* (Figura 7.3).

- i) Cada *proyecto* (Tabla *MSP\_PROJECTS*) se transforma en un *proceso de negocio*. En este paso se crea la instancia *Process*, conforme a la Tabla 7.3 que complementa al algoritmo.
- ii) Un *proyecto* puede también ser un *subproyecto*. En este caso se ha de crear además la instancia *Activity* (Tabla 7.5) y la instancia *SubProcess* (Tabla 7.4).

- iii) En este paso hay que generar *Actividades* pertenecientes al *Proceso* de negocio para todas las *Tareas* (Tabla *MSP\_TASKS*) de un *proyecto*:
- a) Se crea la instancia de la clase *Activity*.
  - b) Un *proyecto* puede también ser un *subproyecto*. En este caso se ha de crear además la instancia *Process* (Tabla 7.3) y la instancia *SubProcess* (Tabla 7.4). Si no hay relaciones de precedencia entre las actividades agrupadas, el subproceso será de tipo *AdHoc* (*SubProcess.isAdHoc=True*), así, las *actividades subordinadas* a la *actividad de grupo* se ejecutarán en paralelo dentro del *subproceso*.
  - c) Toda *tarea* tiene una *duración*, bien *fija* o bien *estimada* (*TASK\_DUR\_IS\_EST=true/false*). De aquí se deriva una *restricción temporal de duración fija o flexible*. Se crea la instancia de *Temporal\_Constraint* con las propiedades correspondientes.
  - d) Una *tarea* se planifica por defecto según la restricción *As Soon As Possible* (*ASAP*). Opcionalmente pueden especificarse restricciones para el inicio y/o terminación de la tarea, bien fijas (*MSON* o *MFON*) o flexibles (*ASAP*, *ALAP*, *NET* o *NLT*). Se crean las instancias de *Temporal\_Constraint* con las propiedades correspondientes para el evento de inicio y terminación de la *actividad*.
  - e) Una *tarea* puede tener otras *Tareas Predecesoras* (Tabla *MSP\_LINKS*) que condicionan sus eventos de inicio o terminación. Por cada *tarea Predecesora* se crea una instancia de la clase *Temporal\_Dependency* ligada a la actividad (*Activity*) correspondiente a la tarea, asignando el tipo de dependencia temporal (*FF*, *FS*, *SF* o *SS*) y un adelanto o retraso temporal en función del atributo *LINK-LAG*.

#### 7.4.1.2 Reglas de Mapeo

En esta sección se presentan un conjunto de tablas que detallan las reglas de mapeo de estructuras y restricciones entre los dos niveles de abstracción *PSM* y *PIM*. El origen de la transformación son tablas relacionales de *MS Project Server*. Cada conjunto origen se expresa en álgebra relacional.

**Tabla. 7.3.** Proyecto como proceso de negocio.

$\sigma_{proj\_uid \in \Psi}^{Msp\_Projects}$	Nuevo proceso de negocio asociado al proyecto en legacy system:		Process
↓ Columna	Descripción	Transformación	↔ Atributo
Crear instancia de <b>Proceso con el Proyecto <math>P_i</math></b> ( $P_i:Process$ ) Si una <b>Actividad <math>A_j</math></b> es un Subproyecto o una tarea en grupo ( $A_j=SP_j=P_j$ ) Crear instancia de <b>Proceso (<math>P_j:Process</math>)</b> con el Proyecto $Msp\_tasks.TaskIsSubProj.Proj\_Uid= P_j$ o con la tarea padre $Msp\_tasks.TaskIsSubProj.Task\_Parent\_Uid= P_j$			
<i>proj_name</i>	Nombre proyecto	Nombre proceso de negocio	<i>name</i>
<i>proj_info_start_date</i>	Fecha inicio proyecto	Inicio proceso de negocio	<i>start</i>
<i>proj_info_finish_date</i>	Fecha finalización proyecto	Fin proceso de negocio	<i>end</i>
<i>opt_def_start_time</i>	Fecha planificada comienzo del proyecto	Inicio CPM	<i>startCPM</i>
<i>opt_def_start_time</i>	Fecha planificada de fin del proyecto	Fin CPM	<i>endCPM</i>
<i>opt_def_est_dur</i>	Duración camino crítico	Duración mínima camino crítico	<i>minDur</i>

**Tabla. 7.4.** Subproceso como Actividad Resumen y Proceso de Negocio.

$\sigma_{proj\_uid \in \Psi}^{Msp\_Projects}$	† Nuevo proceso de negocio asociado al proyecto en legacy system:		SubProcess
↓ Columna	Descripción	Transformación	↔ Atributo
Si una <b>Tarea <math>A_j</math></b> es un Subproyecto o una tarea en grupo ( $A_j=SP_j=P_j$ ) Crear instancias de <b>Subproceso (<math>SP_j:SubProcess</math>)</b> con el Subproyecto $Msp\_tasks.TaskIsSubProj.Proj\_Uid= P_j$ o con la tarea padre $Msp\_tasks.TaskIsSubProj.Task\_Parent\_Uid= P_j$  Si es una tarea en Grupo <span style="float:right">isAdhoc←true</span>			

† Un proyecto puede verse también como una actividad (Process→Activity) y puede formar parte de otros proyectos de mayor nivel como un subproceso (Subprocess→Process) . Se crea la primera actividad como actividad resumen del proyecto y, a continuación, se vinculan las instancias de proceso y subproceso de negocio.

**Tabla. 7.5.** Actividades del proceso de negocio.

vinculación al proceso, mapeo de subprocesos y de actividades agrupadas como procesos *AdHoc*

$\sigma_{proj\_uid \in \Psi} \left( \begin{matrix} Msp\_Projects \\ \times \\ Msp\_Tasks \end{matrix} \right)$	Conjunto de Actividades del proceso de negocio asociado al proyecto en LIS:		Activity
↓ Columna	Descripción	Transformación	↔ Atributo
<i>task_name</i>	Nombre de la tarea	Nombre de la actividad	<i>name</i>
<i>task_start_date</i>	Fecha de inicio de la tarea	Fecha de inicio de la actividad	<i>start</i>
<i>task_end_date</i>	Fecha de finalización de la tarea	Fecha de finalización de la actividad	<i>end</i>
<i>opt_def_start_time</i>	Fecha planificada comienzo del proyecto	Inicio CPM	<i>startCPM</i>
<i>opt_def_start_time</i>	Fecha planificada de fin del proyecto	Fin CPM	<i>endCPM</i>
⤵ <b>Clave ajena</b>	<b>Descripción</b>	<b>Transformación</b>	◊ Asociación
<i>FK_Msp_Project_Parent</i>	Tareas del proyecto	Composición proceso ◊ actividades	Process ◊ Activity
<i>FK_Msp_TaskIsSubproject</i>	El elemento es un <i>Subproyecto</i> o una <i>Tarea</i> (Ambos definidos como	( <b>task_is_subproj≠0</b> ) ⇒ i. Crear proceso. ii. Crear subproceso.	Process→Activity Subprocess→Activity Subprocess→Process

**Tabla. 7.5.** Actividades del proceso de negocio. vinculación al proceso, mapeo de subprocessos y de actividades agrupadas como procesos *AdHoc*

	actividad en modelo destino)	( <b>task_is_subproj=0</b> ) ⇒ iii. En caso contrario es una tarea. Secrea la subclase “Task”.	Task→Activity
		$p = \rho(Msp\_Tasks); c = \rho(Msp\_Tasks)$	
	La tarea es una tarea de grupo (tarea padre: p). Existe un conjunto de tareas que tienen el mismo valor de la columna <i>task_parent_uid</i>	$\emptyset \neq \left( \sigma_{proj\_uid=Pi} Msp\_projects \right) \bowtie_p \bowtie_{\substack{p.proj\_uid=c.proj\_uid \\ \wedge p.task\_id=c.task\_parent\_uid}} c \Rightarrow$	$c \Rightarrow$
FK_Msp_Tasks_Msp_Taks	<i>task_parent_uid</i>	Crear proceso y subprocesso, como subprocesso <i>AdHoc</i> . Se establecen los vínculos de la jerarquía de generalización y se asigna el valor del atributo del <i>subproceso</i> .	Process→Activity Subprocess→Activity Subprocess→Process Subprocess.isAdHoc←True
	La tarea depende de una tarea de grupo (tarea hija: c)	( <b>task_parent_uid≠null</b> ) ⇒ i. Crear instancia de la composición <i>subproceso</i> ♦ actividades para el <i>subproceso</i> ii. <i>AdHoc</i> padre	Process ♦ Activity

**Tabla. 7.6.** Restricción de tiempo de duración fija para cada tarea ‘FIXD’

$\sigma_{proj\_uid \in \Psi} \left( \begin{matrix} Msp\_Projects \\ \bowtie \\ Msp\_Tasks \end{matrix} \right)$	Restricción de duración fija $A_i$ para la tarea del proyecto	<b>Temporal_Constraint</b>	
↓ Columna	Descripción	Transformación	↔ Atributo
<i>task_dur_is_est</i>	La duración de la tarea es fija o estimada. Se crea una TC de duración	i. Crear instancia de <i>Temporal_Constraint</i> dependiente de la actividad $A_i$ ii. <i>Activity</i> ♦ <i>Temporal_Constraint</i> . iii. Asociar tipo de restricción temporal.	$tc\_type \leftarrow x$ $minDur \leftarrow y$ $maxDur \leftarrow task\_dur$
<i>task_dur</i>	Duración de la tarea	( <b>task_dur_is_est=true</b> ) ⇒ $x \leftarrow 'FLEXD'$ $y \leftarrow 0$ ( <b>task_dur_is_est=false</b> ) ⇒ $x \leftarrow 'FIXD'$ $y \leftarrow task\_dur$	

**Tabla. 7.7.** Restricciones de tiempo sobre inicio y terminación de la actividad

$\sigma_{proj\_uid \in \Psi} \left( \begin{matrix} Msp\_Projects \\ \bowtie \\ Msp\_Tasks \\ \bowtie \\ Ext\_Task\_Constraint\_Types \end{matrix} \right)$	Restricciones de tiempo sobre inicio y terminación de la actividad	<b>Temporal_Constraint</b>	
↓ Columna	Descripción	Transformación	↔ Atributo
<i>Abbreviation</i>	Restricciones sobre fecha inicio	$\left( \begin{matrix} abbreviation \in \{ 'MSON', 'SNET', 'SNLT' \} \\ abbreviation = 'ASAP' \Rightarrow abbreviation = 'SASAP' \\ abbreviation = 'ALAP' \Rightarrow abbreviation = 'SALAP' \end{matrix} \right) \Rightarrow$	$tc\_type \leftarrow abbreviation$ $start\_sch \leftarrow task\_constraint\_date$
<i>task_constraint_date</i>	Restricciones sobre fecha fin	$\left( \begin{matrix} abbreviation \in \{ 'MFON', 'FNET', 'FNLT' \} \\ abbreviation = 'ASAP' \Rightarrow abbreviation = 'FASAP' \\ abbreviation = 'ALAP' \Rightarrow abbreviation = 'FALAP' \end{matrix} \right) \Rightarrow$	$tc\_type \leftarrow abbreviation$ $end\_sch \leftarrow task\_constraint\_date$

**Tabla. 7.8.** Dependencias temporales sobre la actividad

$\sigma_{proj\_uid \in \Psi}$	$\left( \begin{array}{l} Msp\_Projects \\ \times \\ Msp\_Tasks \\ \times \\ Ext\_Task\_Link\_Types \end{array} \right)$	<b>Dependencias temporales sobre la actividad</b>	<b><i>Temporal_Dependenc</i></b> <b>y</b>
Crear instancia clase asociativa: <i>Temporal_Dependency</i> Definir roles <i>Predecessor</i> y <i>Successor</i> con claves ajenas <i>Link_Pred_Uid</i> y <i>Link_Succ_Uid</i>			
↓ Columna	Descripción	Transformación	◦-◦ Atributo
<i>link_type</i>	Tipo de enlace entre predecesora y sucesora		
<i>link_lag</i>	Adelanto (-) o retraso (+) de la relación entre predecesora y sucesora	$td\_type \leftarrow abbreviation$ $leadORlag \leftarrow link\_lag$	
<i>Link_Types.abbreviation</i>	Tipo de enlace entre predecesora y sucesora		

### 7.4.2 Actividades replicadas en legacy systems

#### 7.4.2.1 Identificación del problema

Una *actividad* puede ser planificada en más de un *proyecto*, o bien aparecer en distintos *gestores de contenido, en un ERP o en otros legacy systems* (p. ej.: podría estar planificada en *MS Project*, generar distintos documentos en *Alfresco* y tener un presupuesto asignado en el *ERP SAP*). En el proceso de ingeniería inversa desde estos sistemas se originarían distintos procesos de negocio para cada *legacy system*, con pertenencia de estas actividades a cada proceso de negocio, en cuyo caso puede aparecer cierta redundancia de actividades que es necesario controlar. Se trata de resolver el caso concreto: *Actividad de un proceso de negocio que está replicada en más de un legacy system*; p. ej.: aparece en dos proyectos, o en un proyecto y un sistema de gestión de contenidos, etc. No se plantea como un problema general de *Fusión de Modelos*, sino como un problema concreto, que responde a la siguiente cuestión: *¿Podría obtenerse un modelo global del proceso que contemplara una actividad independientemente del legacy system del que procede?*

#### 7.4.2.2 Resolución mediante extensión del Meta-Modelo PIM-upl

El experto de negocio debe captar estas redundancias e identificar el grupo de actividades replicadas que, conceptualmente, corresponden a una *actividad única o global* ( $A_{MA}$ ) desde el punto de vista del negocio. Una *actividad global* ( $A_{MA}=\{A_i\}$ ) es, por tanto, una *actividad* y también puede verse como un nuevo artefacto que agrupa a un conjunto de *actividades privadas* ( $A_i$ ) observadas en cada *legacy system*.

El experto podrá generar procesos de negocio específicos para cada *legacy system* o bien buscar una representación de procesos más genéricos y cercanos a la realidad del negocio, en cuyo caso, estas actividades globales aparecerán en los modelos heredando todo el comportamiento (*estado y restricciones temporales*) de cada uno de sus fragmentos. La aparición de este nuevo tipo de actividades puede introducir colisiones de integración de propiedades que hay que resolver con nuevas *restricciones*.

Se plantean extensiones al *Meta-Modelo UML pivote PIM-upl* (Figura 7.3) que i)

deriven las *restricciones y dependencias temporales* y ii) resuelvan las colisiones de integración, para generar procesos más generales que contemplen estas actividades replicadas. Contar con este *Meta-Modelo PIM* más compacto (un conjunto de clases mínimas que permiten expresar la dimensión temporal de los procesos) permite simplificar, en este caso, la resolución de colisiones, con lo que el proceso de abstracción *PIM*→*CIM* arrastrará este beneficio al nivel de abstracción que maneja el experto de negocio en la gestión de los procesos del software.

En la [Figura 7.8](#) se especifican las extensiones al *Meta-Modelo* y las restricciones *OCL* que debe cumplir una *actividad que integra o fusiona* (clase *Merged\_Activity* ≡ *A<sub>MA</sub>*) el comportamiento de todos sus fragmentos (clase *Activity* ≡ *A<sub>f</sub>*) procedentes de cada sistema *LIS<sub>f</sub>*.

- i) Una *actividad global* (clase *Merged\_Activity* ≡ *A<sub>MA</sub>*) es un subtipo de *actividad* (clase *Activity*) añadida por el experto de negocio al contexto. En la [Figura 7.8](#) aparecen dos recintos (□): a) el recinto *LIS Models* delimita a la clase *Process* y *Activity* que aparece en cada *legacy system* y b) el recinto *OverAll Model* que engloba al anterior y a la nueva actividad (*Merged\_Activity*) y restricciones adicionales. La actividad *Merged\_Activity* tiene como propiedades a) *unique\_Identifier:unique\_Id* que debe ser único en el sistema, ya que no siempre es posible identificar actividades sólo por el nombre; b) */tc:overAll:Temporal\_Constraint[0..\*]* que es la unión del conjunto de restricciones temporales de todos sus fragmentos y c) */td\_overAll:Temporal\_Dependency[0..\*]* como conjunto heredado de dependencias temporales de sus predecesoras. Estos dos últimos atributos son atributos calculados.
- ii) Cada sistema puede manejar distintos nombres o identificadores para cada uno de los fragmentos relacionados con esta actividad global. Por tanto es necesario que el experto identifique dichos fragmentos. Para ello se especificarán las asociaciones (*isMappedInLis* ≡ *A<sub>MA</sub>→A<sub>f</sub>*) entre la actividad global y sus fragmentos, donde el rol de la actividad global es *isOverAll (1)* y el rol de cada fragmento es *+fragmentInLis (1..\*)*.

Hay que tener en cuenta que *OCL* [[OMG:OCL 2014](#)] es un «lenguaje declarativo *Side-effect free*» [[Demuth et al. 1999, 2001](#); [Cabot et al. 2012](#)], lo que significa que no tiene capacidad de alterar el estado del sistema, por lo que en el proceso de identificación, el experto del dominio debe modificar los procesos para otorgar la consistencia necesaria.

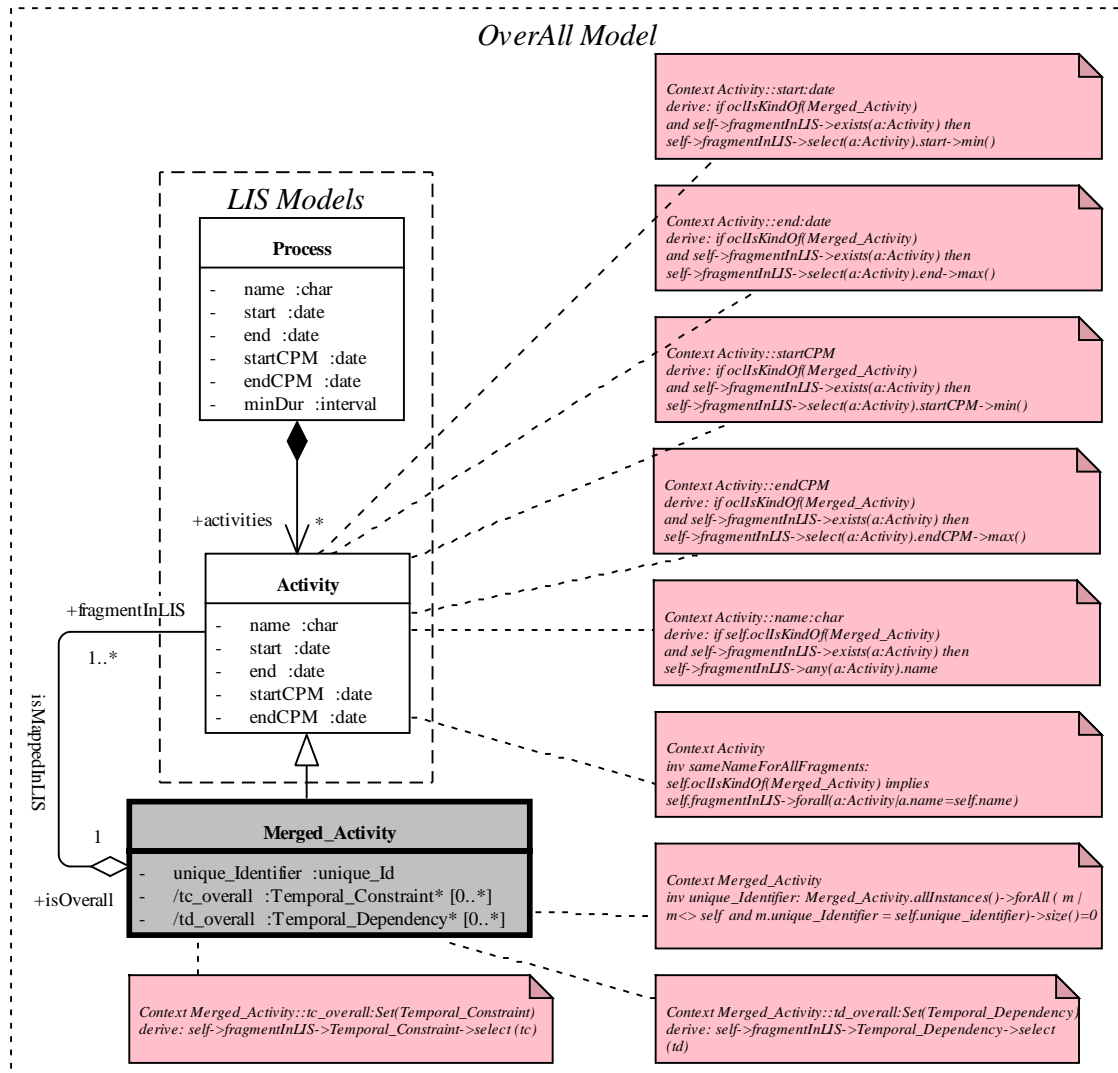


Figura. 7.8. Extensión al Meta-Modelo PIM-upl para resolver colisiones de actividades replicadas

A continuación se comenta cada una de las restricciones OCL para resolver los distintos tipos de colisiones que pueden aparecer:

i) Colisiones de identificación de actividades.

El contexto para la primera restricción de la Figura 7.9 es el atributo *Activity::name*. Si la actividad es del tipo *Merged\_Activity* y existe algún fragmento definido que provenga de un *LIS<sub>i</sub>*, entonces el atributo *name* se toma de cualquiera de estos fragmentos (regla *OCL derive*).

```
Context Activity::name:char
derive: if self.ocIsKindOf(Merged_Activity)
and self->fragmentInLIS->exists(a:Activity) then
self->fragmentInLIS->any(a:Activity).name
```

```
Context Activity
inv sameNameForAllFragments:
self.ocIsKindOf(Merged_Activity) implies
self.fragmentInLIS->forall(a:Activity|a.name=self.name)
```

Figura. 7.9. Resolución de colisiones del atributo *Activity.name*

El segundo invariante, cuyo contexto es la clase *Activity*, expresa que, obligatoriamente, el atributo *Activity.name* de una instancia *Merged\_Activity* debe coincidir con el de todos los fragmentos definidos. Estos dos invariantes aseguran que, para la jerarquía formada por una instancia de *Merged\_Activity* y el conjunto de instancias *Activity* (fragmentos) el nombre de la actividad debe ser único; ahora bien, ¿Cómo asegurar que otras actividades conceptualmente distintas, pero que coinciden en nombre, no se mezclan de manera indeseada?; para ello se define un tercer invariante (Figura 7.10):

```
Context Merged_Activity
inv unique_Identifier: Merged_Activity.allInstances()->forall ( m |
m<> self and m.unique_Identifier = self.unique_Identifier)->size()=0
```

Figura. 7.10. Resolución de colisiones del atributo *Activity.unique\_Identifier*

De este modo, se asegura que dentro del subconjunto de instancias *Merged\_Activity* no se repita el atributo *Merged\_Activity.unique\_Identifier*.

ii) Colisiones con los atributos *start* y *end* (Figura 7.11).

Los atributos *Activity.start* y *Activity.end* referencian los eventos reales de inicio y terminación de la actividad. Al existir diferentes fragmentos procedentes de diversos *legacy systems* es necesario establecer cuáles son realmente los eventos de la actividad única a nivel conceptual (*Merged\_Activity*).

```
Context Activity::start:date
derive: if ocIsKindOf(Merged_Activity)
and self->fragmentInLIS->exists(a:Activity) then
self->fragmentInLIS->select(a:Activity).start->min()
```

```
Context Activity::end:date
derive: if ocIsKindOf(Merged_Activity)
and self->fragmentInLIS->exists(a:Activity) then
self->fragmentInLIS->select(a:Activity).end->max()
```

Figura. 7.11. Resolución de colisiones con atributos *start* y *end* de la clase *Activity*



Se definen dos reglas de *derivación OCL* para compatibilizar todas las restricciones heredadas: a) La primera establece el inicio más temprano posible y b) La segunda que esta actividad termina cuando lo hace el último fragmento.

iii) Colisiones con los atributos *startCPM* y *endCPM* (Figura 7.12).

```
Context Activity::startCPM
derive: if oclIsKindOf(Merged_Activity)
and self->fragmentInLIS->exists(a:Activity) then
self->fragmentInLIS->select(a:Activity).startCPM->min()
```

```
Context Activity::endCPM
derive: if oclIsKindOf(Merged_Activity)
and self->fragmentInLIS->exists(a:Activity) then
self->fragmentInLIS->select(a:Activity).endCPM->max()
```

Figura. 7.12. Resolución de colisiones con atributos *startCPM* y *endCPM*

De manera similar al caso anterior referido a los atributos *Activity.start* y *Activity.end* con los atributos que definen los eventos planificados (por un planificador de tareas *CPM*) o previstos para una actividad, *Activity.startCPM* y *Activity.endCPM* deben tener el inicio del primer fragmento planificado para empezar y como evento de finalización el del último fragmento previsto en finalizar.

iv) Restricciones que tienen como contexto la clase *Merged\_Activity* para derivar automáticamente restricciones y dependencias temporales (Figura 7.13).

```
Context Merged_Activity::tc_overall:Set(Temporal_Constraint)
derive: self->fragmentInLIS->Temporal_Constraint->select(tc)
```

```
Context Merged_Activity::td_overall:Set(Temporal_Dependency)
derive: self->fragmentInLIS->Temporal_Dependency->select(td)
```

Figura. 7.13. Derivación de reglas para la clase *Merged\_Activity*

En este caso, los atributos */Merged\_Activity.tc\_overAll* y */Merged\_Activity.td\_overAll* que representan un conjunto de restricciones temporales y de dependencias temporales son atributos calculados mediante la regla *derive OCL*, generándose para la clase *Merged\_Activity* la unión de estos tipos de reglas.

#### 7.4.2.3 Escenario ejemplo

La Figura 7.14 ilustra con un escenario UML [OMG:UML 2011] las capacidades que ofrece ahora el *Meta-Modelo pivote PIM-upl* (Figura 7.3) con las extensiones del

apartado anterior. Una vez que el experto de negocio da conformidad a la agrupación de fragmentos mediante la definición de la instancia *Merged\_Activity* y la especificación de su identificador único (*unique\_Identifier*), puede optar por extraer modelos que sólo representan la semántica de un *legacy system* ( $LIS_i$ ) o bien por generar un modelo de procesos global compuesto por el conjunto de las actividades globales (clase *Merged\_Activity*  $\equiv A_{MA}$ ) seleccionadas, excluyendo sus fragmentos replicados ( $A_f | A_{MA} \equiv \{A_f\}$ ) y una selección de otro subconjunto de actividades que no están fragmentadas ( $A_n | A_n \notin A_{MA}$ ) pero sí tienen relación con el mismo proceso de negocio.

En la [Figura 7.14](#) aparecen tres recintos que engloban objetos: i) Procesos de negocios extraídos de cada  $LIS_i$  (con línea punteada negra), ii) Reglas temporales, restricciones y dependencias, derivadas para una actividad mediante la combinación controlada de sus fragmentos y iii) Recinto con la vista de procesos globales de negocio.

- i) En el primer recinto, el experto de negocio tiene acceso sólo a representar el proceso derivado de cada sistema, independientemente de los demás.
- ii) El segundo recinto contempla las clases *Temporal\_Constraint* y *Temporal\_Dependency* para una actividad global (*Merged\_Activity*) que engloba a todos sus fragmentos y su correspondiente comportamiento.
- iii) El último recinto contempla los artefactos de un proceso de negocio global que combina actividades globales, excluyendo sus fragmentos, y actividades que no tienen fragmentos. Este proceso estará más cercano al proceso real ejecutado en la organización, pues los distintos fragmentos pueden provenir de distintos sistemas y combinar una o más perspectivas BPM (además de la temporal, podría integrar la organizacional y la de casos).

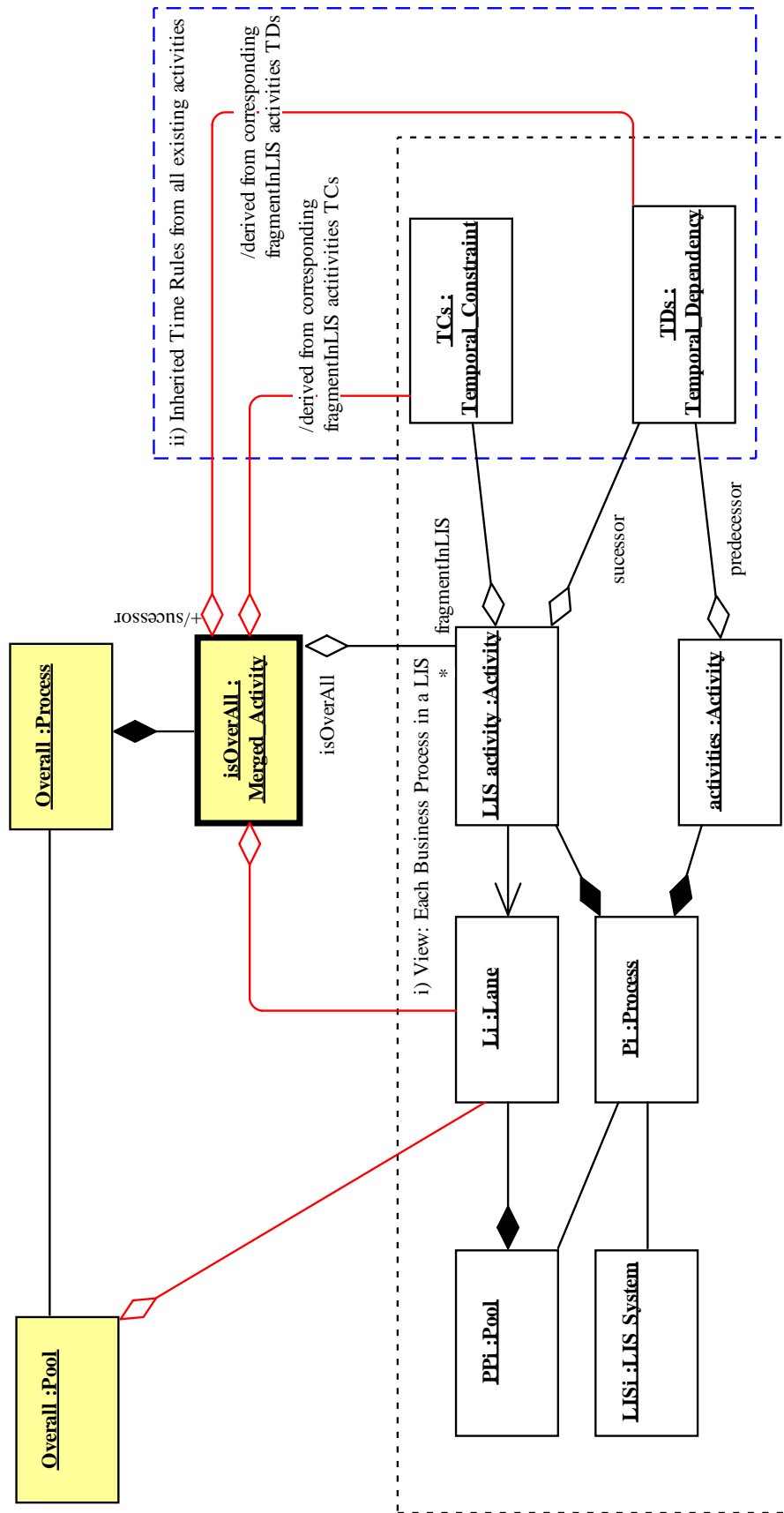


Figura. 7.14. Escenario de extracción de modelos de procesos con actividades replicadas

### 7.4.3 Transformación de modelos PIM-upl→CIM

El experto de negocio del software puede desear expresar sus modelos de procesos conforme a distintos lenguajes o estándares: a) *BPMN 2.0* [OMG:BPMN 2013; Iso/Iec:19510 2013], como estándar de propósito general; b) *SPEM 2.0* [Bendraou et al. 2007; OMG: SPEM 2008] o *Iso/Iec:24744* [Iso/Iec:24744 2007, 2014] como estándares específicos para la gestión de los procesos software; c) podría plantearse llegar a otros *Meta-Modelos* destino e incluso a sintaxis concretas que propusiera el experto como, en nuestro caso, *NDTQ-Framework* [Ponce et al. 2013; Aragón 2014], inspirado en *NDT* [Escalona 2004; Escalona & Aragón 2008] posee un *Meta-Modelo* compatible con [Iso/Iec:24744 2007, 2014].

Se puede pensar en realizar transformaciones específicas a cada *Meta-Modelo* de procesos, o bien realizarlos a través de los estándares más extendidos para el intercambio de metadatos de procesos, lo cual ampliará el rango de herramientas (*BPMS*) que podrían recoger el modelo del proceso obtenido. *XPDL 2.2* [Shapiro 2010; WfMC:XPDL 2012] tiene un alto nivel de conformidad con el estándar *BPMN 2.0*, aunque no será hasta la versión 3.0 donde se soporte completamente, y que *BPDM* [Harmon 2004; OMG:BPDM 2008] ha quedado en la versión 1.0 no ha sido ampliado desde entonces en favor del desarrollo que ha tenido *BPMN* para cubrir gran parte de los aspectos ligados a la definición y gestión de procesos de negocio. En el primer salto de abstracción *PSM*→*PIM* se ha optado por manejar *sintaxis abstracta* de los *Meta-Modelos*. También hay que tener en cuenta que el modelo destino debe soportar la semántica temporal del *Meta-Modelo pivote PIM-upl* (Figura 7.3), lo que obliga a extender el *Meta-Modelo* destino con estas reglas temporales. Por esta razón, se elige el *Meta-Modelo BPMN 2.0* como plataforma *CIM* y primer destino *PIM*→*CIM* que se desarrolla en esta memoria de tesis (posteriormente pueden desarrollarse otras transformaciones). La Figura 7.15 ilustra estas alternativas y el camino de este proceso de abstracción.

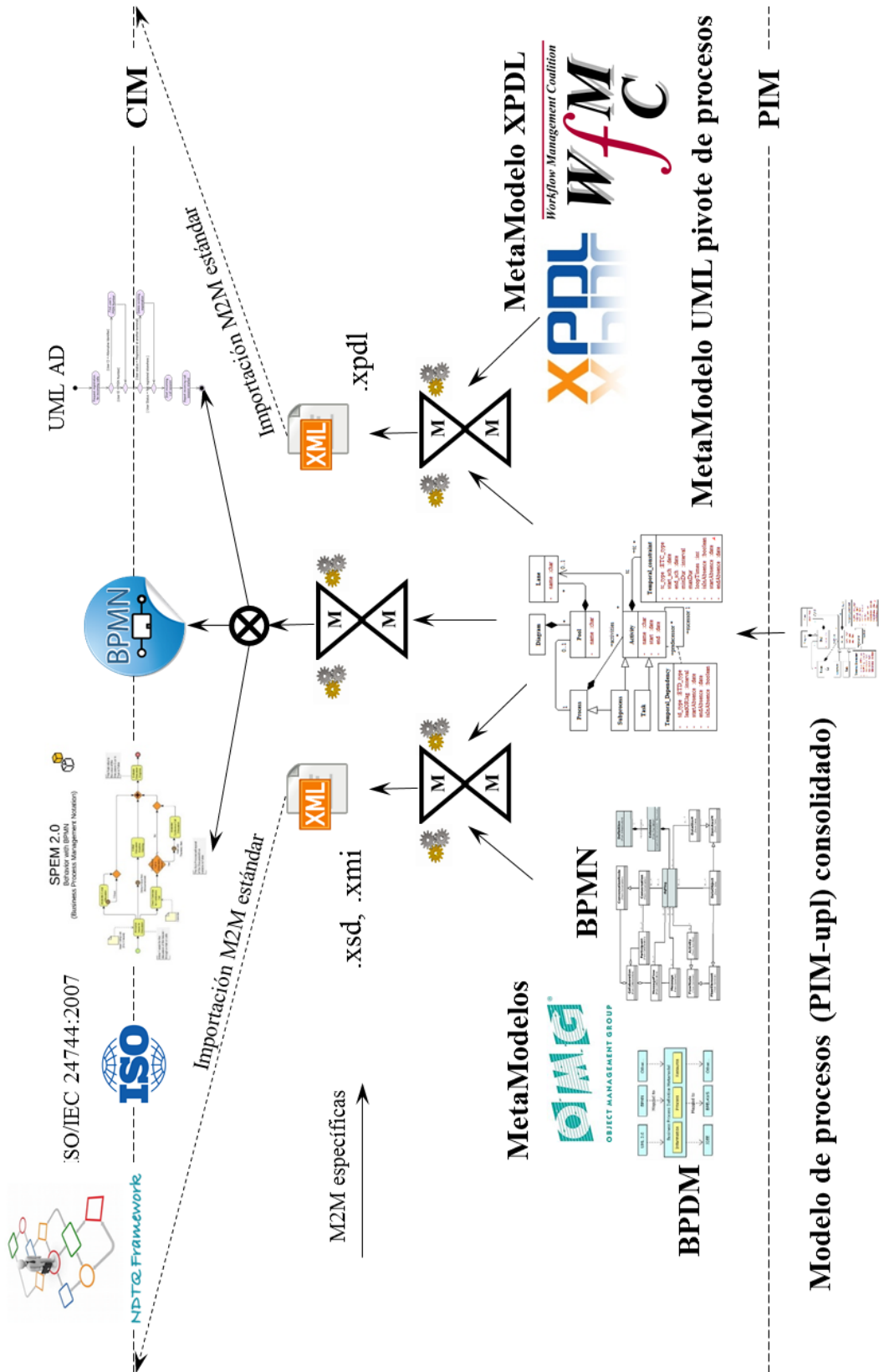


Figura. 7.15. Proceso de extracción de la dimensión temporal PIM→ CIM

La Figura 7.16 muestra un diagrama de clases con la vista del *Meta-Modelo BPMN 2.0* y los requisitos para extender éste con la dimensión temporal del *Meta-Modelo pivote PIM-upl* (Figura 7.3). En el diagrama de clases se acotan (Encerradas en el recinto “*BPMN Semantic+Temporal Extensions*”) las clases del estándar, existentes en el paquete *Semantic*; aparecen sin sombrear y con sus propiedades. En el exterior del recinto se sitúan las clases del *Meta-Modelo PIM* (con fondo en amarillo) con las que existe una relación y ofrecen los requisitos para extender *BPMN 2.0* con la dimensión temporal. Es necesario añadir dos clases nuevas al *Meta-Modelo BPMN*: *BPMN TE::Temporal\_Constraint* y *BPMN\_TE::Temporal\_Dependency* para soportar las restricciones y dependencias temporales sobre *actividades*. Se especifica la relación «*trace*» que establece la correspondencia entre clases de los dos *Meta-Modelos*.

De este modo, habría que extender las distintas clases del *Meta-Modelo BPMN 2.0*, como se especifica a continuación:

- i) **Clase Semantic::process.** Hay que añadir los atributos *name*, *start*, *end*, *startCPM*, *endCPM* y *minDur*.
- ii) **Clase Semantic::subProcess.** No es necesario añadir el atributo *isAdHoc* sino las instancias *Semantic::SubProcess* y *Semantic::tAdHocSubProcess* para expresar este hecho.
- iii) **Semantic::Activity.** Hay que añadir los atributos *start*, *end*, *startCPM*, *endCPM* y *minDur*. El atributo *name* lo tomará por herencia de la clase *tFlowElement*.
- iv) **Semantic::Temporal\_Constraint y Semantic::Temporal\_Dependency.** Soportan las *restricciones de tiempo* sobre una *actividad* y las *dependencias temporales* entre *actividades* mediante los invariantes *OCL* que heredan de las mismas clases en el *Meta-Modelo PIM-up* (Figura 6.11).

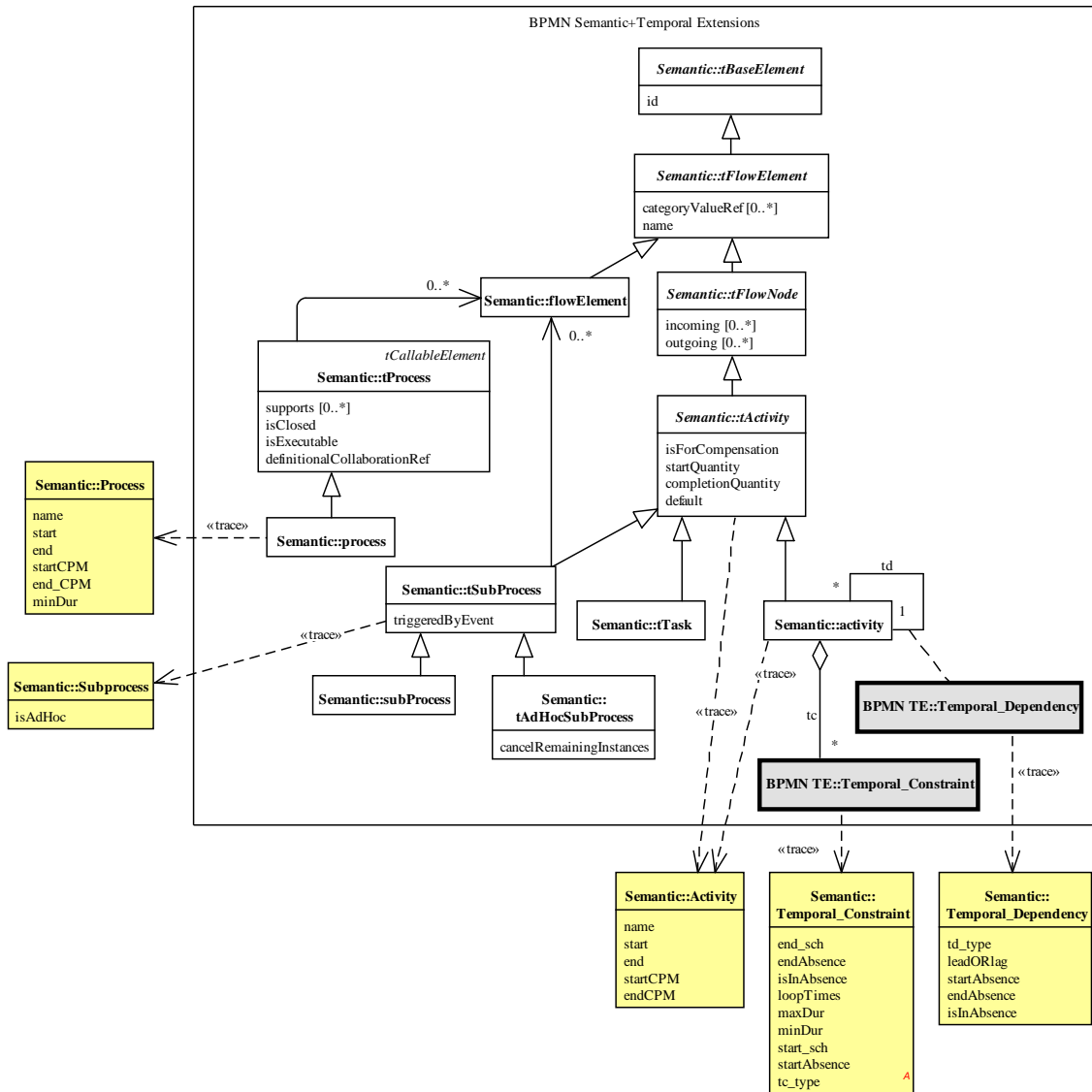


Figura. 7.16. Meta-Modelo BPMN 2.0 con extensiones temporales

Además hay que asociar los invariantes OCL a las clases extendidas en *BPMN*; así, se vincularán:

- i) Invariantes de la Figura 6.14 sobre *BPMN::process* y *BPMN::activity*. Estos invariantes definen la semántica de los estados de la actividad.
- ii) Invariantes de las Figuras 6.15, 6.16 y Tabla 6.6 a la clase *BPMN::Temporal\_Constraint* para expresar las restricciones temporales sobre la clase *BPMN::activity*.
- iii) Por último, invariantes de la Figura 6.17 y Tabla 6.7 para definir las dependencias temporales entre actividades y sus predecesoras, sobre la clase *BPMN::Temporal\_Dependency*.

Hay que tener en cuenta las clases *Pool* con la jerarquía correspondiente

(*laneSet*→*tLaneSet*) y la clase *Lane* con la correspondencia (*lane* → *tLane*) para ubicar procesos y actividades dentro de estos artefactos *BPMN*. Estas clases se pueden instanciar en los dos niveles de abstracción mediante el proceso automático de ingeniería inversa (sobre todo cuando se capturen otras perspectivas como la organizacional y la de casos) o bien a instancias del experto para enriquecer los modelos de procesos. En la memoria de tesis se desarrolla el salto al nivel de abstracción *BPMN 2.0*, transformación *M2M*, pero en otros casos podrían plantearse otro tipo de transformaciones *M2M* o *M2T*.

Ya se ha referenciado a *BPMN 2.0* no sólo por sus capacidades para la representación de procesos sino también por el soporte para el intercambio de metadatos de modelos de procesos, luego una vez se cuenta con un modelo conforme a este *Meta-Modelo extendido* se ofrecen facilidades para el intercambio de diagramas con otras herramientas en formato .xml (.xsd, .xmi).

#### 7.4.3.1 Algoritmo

El origen es el *Meta-Modelo pivote PIM-upl* (Figura 7.3). El destino es el *Meta-Modelo BPMN 2.0* (Figura 7.16). La Figura 7.17 representa el algoritmo general para generar modelos *CIM BPMN 2.0* con restricciones temporales expresadas mediante invariantes *OCL*. En la Sección siguiente figuran Tablas de mapeo de las tablas origen a las clases destino.

- i) Se seleccionan los procesos a transformar a modelos *BPMN*.  
Se crea la instancia de *proceso* (clase *process*), según la [Tabla de mapeo de propiedades 7.9](#), y se le asigna un *Pool* (clase *laneSet*), siguiendo las reglas específicas de la [Tabla 7.10](#).
- ii) Para todas las *actividades* (*Ai:Activity*) que forman parte del proceso (*Pi:Process*):
  - a) Se crea la instancia *BPMN activity*, utilizando la [Tabla de mapeo 7.11](#) para asignar sus propiedades.
  - b) Si la actividad tiene asignada una *calle* (clase *Lane*) dentro de una piscina (*Pool*), se crea la instancia correspondiente *BPMN* ([Tabla 7.12](#)) y se asigna a la clase *BPMN::activity*.
  - c) Si la actividad es un subproceso (clase *Subprocess*), se crea en el modelo *BPMN* ([Tabla 7.13](#)). Si se trata de un proceso *AdHoc* se resuelve la instanciación según la [Tabla 7.14](#).
  - d) Se mapean a *BPMN* las restricciones temporales sobre la clase *BPMN::Temporal\_Constraint*, utilizando la [Tabla de mapeo 7.15](#).



```

Ψ≡{Procesos seleccionados para la transformación PIM → CIM }
∀(Pi:Proceso) ∈ Ψ {
    CREAR Proceso de negocio BPMN Pi /* Aplicando reglas de mapeo de las Tabla 7.9 (Proceso) */
    CREAR Pool BPMN para proceso Pi --(Tabla de mapeo 7.10).

    ∀(Ai:Actividad) ∃ (Pi:Proceso) {
        CREAR 'Ai:Actividad' BPMN del proceso de negocio
        /* Aplicando reglas de mapeo de la Tabla de mapeo 7.11 para Activity*/
        Si ∄ Lanei para la Actividad Ai
            CREAR Lanei para la Actividad Ai dentro del Pool de su proceso -- Tabla 7.12
            ASOCIAR Lanei para Actividad Ai -- Tabla 7.12
        Fin

        Si la Actividad es un subproceso
            CREAR Subproceso -- (Tabla de mapeo 7.13 y datos de la fila del Proyecto vinculada)
        Si es un subproceso AdHoc CREAR Subproceso AdHoc BPMN -- (Tabla de mapeo 7.14
        Fin
        ∀(RTi:Restricción temporal) ∃ (Ai:Actividad) -- (Tabla de mapeo 7.15)
            { CREAR restricción de tiempo (Temporal_Constraint) para la Actividad (Activity) }
        ∀(DTi:Dependencia temporal) ∃ (Ai:Actividad) -- (Tabla de mapeo 7.16)
            { CREAR dependencia temporal (Temporal_Dependency) para la Actividad (Activity) }
    }
}
    
```

Figura. 7.17. Algoritmo de transformación PIM→CIM

- e) Se mapean a *BPMN* las dependencias temporales sobre la clase *BPMN::Temporal\_Constraint*, utilizando la [Tabla de mapeo 7.16](#).

### 7.4.3.2 Reglas de Mapeo

En esta sección se presentan un conjunto de tablas que detallan las reglas de mapeo de estructuras y restricciones entre los dos niveles de abstracción *PIM* y *CIM*. El origen de la transformación son las clases del *Meta-Modelo pivote PIM-upl* (Figura 7.3). El *Meta-Modelo* destino es *BPMN 2.0* (Figura 7.8) que soportará todas las nuevas estructuras y las reglas temporales expresadas en *OCL*.

**Tabla. 7.9.** Proceso de negocio BPMN

Crear instancias: <i>tProcess</i> ; <i>process</i> → <i>tProcess</i>			
<i>Process</i>	Proceso <i>BPMN</i>		<i>process</i>
o-oAtributo	Descripción	Transformación	o-oAtributo
<i>name</i>	Nombre proceso de negocio		<i>name</i>
<i>start</i>	Inicio proceso de negocio		<i>start</i>
<i>end</i>	Fin proceso de negocio		<i>end</i>
<i>startCPM</i>	Inicio CPM		<i>startCPM</i>
<i>endCPM</i>	Fin CPM		<i>endCPM</i>
<i>minDur</i>	Duración mínima camino crítico		<i>minDur</i>

**Tabla. 7.10.** Pool BPMN para el proceso

---

Crear instancias:

- tBaseElement*
- tLaneSet* → *tBaseElement*
- laneSet* → *tLaneSet*
- tProcess* → *laneSet*

---

**Tabla. 7.11.** Actividad BPMN

---

Crear instancias:

- tBaseElement*
- tFlowElement* → *tBaseElement*
- flowElement* → *tFlowElement* (La actividad es *tFlowNode* y *flowElement* como parte de *tProcess*)
- tProcess* → *flowElement* (Así queda definida la composición de proceso y sus actividades)
- tFlowNode* → *tFlowElement*
- tActivity* → *tFlowNode*

---

<i>Process</i>	Actividad BPMN	<i>activity</i>
o-o Atributo	Descripción	o-o Atributo
<i>name</i>	Nombre proceso de negocio	<i>tFlowElement.name</i> ← <i>name</i>
<i>start</i>	Inicio proceso de negocio	<i>start</i>
<i>end</i>	Fin proceso de negocio	<i>end</i>
<i>startCPM</i>	Inicio CPM	<i>startCPM</i>
<i>endCPM</i>	Fin CPM	<i>endCPM</i>

---

**Tabla. 7.12.** Calle para la actividad dentro del Pool del proceso

---

Crear instancias:

- tBaseElement* (Instancia base para la calle)
- tLane* → *tBaseElement*
- lane* → *tLane*
- tLaneSet* → *lane* (*tLaneSet* es el mismo que el Pool del proceso de la tabla 7.10)
- tLane* → *tBaseElement* (Instancia base de la actividad). Se crea la asociación *+partitionElement* (Figura 6.10 del Meta-Modelo BPMN 2.0).

---

**Tabla. 7.13.** Subproceso BPMN

---

Crear instancias:

- tSubProcess* → *tActivity*
- subProcess* → *tSubProcess*
- tSubProcess* → *flowElement* (Así queda definida la composición de subproceso y sus actividades)

---

**Tabla. 7.14.** Subproceso BPMN AdHoc

---

Crear instancias:

- tAdHocSubProcess* → *tSubProcess*

---

**Tabla. 7.15.** Restricciones temporales BPMN

Crear instancias:  
*BPMN::activity*→*BPMN::Temporal\_Constraint*

Temporal_Constraint	Restricciones temporales a BPMN	Temporal_Constraint
↔Atributo	Descripción Transformación	↔Atributo
<i>start_sch</i>	Inicio planificado	<i>start_sch</i>
<i>end_sch</i>	Fin planificado	<i>end_sch</i>
<i>startAbsence</i>	Comienzo ausencia	<i>startAbsence</i>
<i>endAbsence</i>	Fin ausencia	<i>endAbsence</i>
<i>isInAbsence</i>	Estado en ausencia	<i>isInAbsence</i>
<i>minDur</i>	Duración mínima	<i>minDur</i>
<i>maxDur</i>	Duración máxima	<i>maxDur</i>
<i>loopTimes</i>	Número máximo de iteraciones	<i>loopTimes</i>
<i>tc_Type</i>	Tipo restricción temporal	<i>tc_Type</i>

*BPMN::activity*→*BPMN::Temporal\_Dependency*

**Tabla. 7.16.** Dependencias temporales BPMN

Temporal_Dependency	Dependencias temporales a BPMN	Temporal_Dependency
↔Atributo	Descripción Transformación	↔Atributo
<i>td_Type</i>	Tipo dependencia temporal	<i>td_Type</i>
<i>startAbsence</i>	Comienzo ausencia	<i>startAbsence</i>
<i>endAbsence</i>	Fin ausencia	<i>endAbsence</i>
<i>isInAbsence</i>	Estado en ausencia	<i>isInAbsence</i>
<i>leadORlag</i>	Adelanto o retraso entre eventos	<i>leadORlag</i>

## 7.5 Resumen y conclusiones

En este Capítulo se ha desarrollado una propuesta basada en el paradigma *MDE* que permite extraer procesos desde la *legacy database* de *MS Project Server*; para ello, se ha realizado un estudio detallado de la documentación del producto, en particular de las reglas temporales que soporta. Los expertos en software utilizan las facilidades de estos productos para hacer planes de proyectos, compuestos por tareas, que en un contexto *BPM* pueden representarse como procesos y actividades del negocio. Dentro de la instancia *Published* de la *legacy database* (soportada por *MS SQL\*Server™* [*Colledge 2008; Ben-Gan 2012; Microsoft™:T-SQL 2014*], conforme al *Meta-Modelo Relacional* de *Codd* [*Codd 1970, 1992*]) pueden existir múltiples proyectos y, por tanto, múltiples procesos, que a su vez pueden agruparse en clases o modelos de procesos. Se ha extendido el *Meta-Modelo* de tareas de *MS Project Server* (nivel de abstracción *PSM*) con clases y referencias que faciliten el proceso de extracción mediante ingeniería inversa *MDE*.

Para sustentar la extracción se ha extendido el *Meta-Modelo PIM-up* (Figura 6.11) del Capítulo 6 (según la taxonomía de reglas de la Figura 6.1), añadiendo clases que permiten vincular los modelos obtenidos al *legacy system* origen, utilizándolo como modelo pivote intermedio (*PIM-upl*), constituyendo así una propuesta basada en sintaxis

abstracta y en un *lenguaje declarativo* como *OCL*.

Los modelos de procesos buscados por el experto en negocio de software pueden ser diversos, bien basados en estándares generales como *BPMN*, o bien basados en estándares específicos para gestión del software como *Iso/Iec:24744* o *SPEM 2.0*. Investigadores y expertos coinciden en que *BPMN* manifiesta cierta debilidad para representar la dimensión temporal, por ejemplo, no soporta bien la taxonomía de reglas (*Figura 6.1*) contemplada en el *Meta-Modelo PIM-up* (*Figura 6.11*); *Iso/Iec:24744* y *SPEM 2.0* están en la misma situación. No obstante, *BPMN* ha cobrado mucha popularidad en el ámbito científico y empresarial, además de poseer un robusto *Meta-Modelo* de procesos que, no solo, facilita la especificación de los mismos sino que también es un estándar para el intercambio de esquemas y metadatos sobre procesos

Utilizando el *Meta-Modelo pivote PIM-upl* (*Figura 7.3*), que trata la semántica temporal y es capaz de discriminar el legacy system origen y las características que permiten intercambiar procesos, se plantean dos saltos de nivel de abstracción, que se describen con un esquema, un algoritmo y tablas detalladas para el mapeo de artefactos: a) *PSM*→*PIM*, llegando a una plataforma independiente de la tecnología, con un conjunto mínimo de artefactos que extraen la semántica temporal de cada *legacy system* origen y b) *PIM*→*CIM BPMN 2.0*, arrastrando la semántica temporal detectada y llevándola al estándar. El *Meta-Modelo pivote PIM-upl* facilita el proceso de extracción, al contener un conjunto mínimo de artefactos que simplifica la especificación de la dimensión temporal sobre procesos y actividades y la resolución de colisiones de actividades replicadas en distintos *legacy systems*. Desde este nivel *PIM* podrían plantearse extensiones con otras perspectivas y atacar otros *legacy systems* (*ECMs*, *ERPs*, *SCMs*, *CRMs* y *desarrollos a medida*). Por otro lado, también podrían generarse especificaciones textuales (transformaciones *M2T*) o generar *logs de eventos de ejecución* útiles en enfoques como «*process mining*» que utiliza este artefacto como origen.

# Capítulo 8. CASO DE ESTUDIO: AQUA-WS PROJECT

*“La mayoría de las ideas fundamentales de la ciencia son esencialmente sencillas y, por regla general pueden ser expresadas en un lenguaje comprensible para todos”*  
**Albert Einstein** (1879-1955)

**E**l enfoque desarrollado extrae procesos del software desde *legacy systems*, como *MS Project*, que definen con rigor la dimensión temporal. En este Capítulo se va a aplicar la propuesta al caso de estudio *AQUA-WS Project*. Se partirá de la planificación del proyecto, almacenada en la instancia *Published* de *SQL\*Server* para una instalación de *MS Project Server*, con objeto de aplicar el proceso de ingeniería inversa *MDE* desarrollado en este trabajo de investigación, generando *procesos* que corresponden a *proyectos*, *subproyectos* o *grupos de tareas*. A continuación se discuten los resultados obtenidos, detectando limitaciones, y se proponen recomendaciones para mejorar la calidad de los modelos obtenidos. Por último se propone la reutilización de estos resultados con enfoques alternativos de *reingeniería de procesos*.

## 8.1 AQUA-WS Project

«*AQUA-WebServices (AQUA-WS Project)*<sup>40</sup>» [Cutilla et al. 2012] es un proyecto de modernización de software, que surge a raíz de la necesidad de unificar todos los antiguos sistemas de la Empresa Municipal de Abastecimiento de Aguas de Sevilla (*EMASESA*<sup>41</sup>), en un único sistema, el núcleo *AQUA*. El nuevo sistema está orientado a la web, y desarrollado en *Java J2EE*. Los principales sistemas integrados fueron *AQUA-SiC* (sistema de gestión de clientes, basado en una plataforma software local), *AQUA-ReD* (sistema de gestión de redes basado en tecnologías web) y *AQUA-Sigo* (sistema de gestión de obras y proyectos basado en tecnologías web). La participación del grupo *IWT2* consistió en la gestión y dirección de la oficina técnica de calidad del proyecto.

El Anexo IV contiene la información extraída de la planificación de este proyecto. Se realizaron planes para el control y seguimiento de las actividades (ciclo de vida de

---

<sup>40</sup> <http://iwt2.org/casos-de-exito/>

<sup>41</sup> <http://www.emasesa.com/>

cada uno de los subsistemas, revisión de los distintos tipos de entregables, y actividades específicas globales de aseguramiento de la calidad del proyecto).

- i) En el Anexo IV.a figuran extractos de estos planes realizados con *MS Project*. El Anexo IV.a1 es la planificación global del proyecto, ejecutado por distintos contratistas externos (Figura IV.1: planificación por subsistemas y Figura IV.2: Planificación por fases técnicas *NDT* [Escalona 2004; Escalona & Aragón 2008]). El Anexo IV.a2 refleja un plan detallado para el control de calidad de una de las tareas del plan anterior: (64): *(ASI) Instalaciones y Equipos* (Figura IV.3). Por último, el Anexo IV.a3 contiene un patrón de proyecto (elaborado con *MS Project*) (Figura IV.4) que se aplica a todas las tareas de la categoría *(ASI)*, Análisis del Sistema, dentro de la Metodología *NDT*.
- ii) En el Anexo IV.b se ha extraído el esquema relacional (Figura IV.5) de la base de datos *SQL\*Server para MS Project Server (projectserver\_Published\_AQUA-WS)*.
- iii) El último Anexo IV.c contiene las instancias de las tablas almacenadas en la base de datos: a) Tipos de restricciones temporales sobre tareas (*Ext\_Task\_Constraint\_Types*) (Figura IV.6), b) Tipos de dependencias temporales (*Ext\_Link\_Types*) (Figura IV.7), c) Tabla de proyectos (*Msp\_Projects*) (Figura IV.8), d) Tabla de tareas (*Msp\_Tasks*) (Figura IV.9 y Tabla IV.1) y e) Tabla de dependencias entre tareas (*Msp\_Links*) (Figura IV.10 y Tabla IV.2).

*MS Project*, además de contemplar la dimensión temporal de un proyecto, permite definir recursos, grupos de recursos y realizar asignaciones a tareas. La Tabla IV.3 incluye una lista de tareas con la asignación de recursos a tareas del proyecto.

### 8.1.1 Proyectos definidos en *projectserver\_Published\_AQUA-WS*

En la base de datos *SQL\*Server 2008 para MS Project Server (projectserver\_Published\_AQUA-WS)* figuran tres proyectos definidos en la tabla *Ms\_Projects* (Figura IV.8):

- i) «(1) *AQUA-WS*». Es el proyecto que recoge el caso de estudio de este capítulo, compuesto por las tareas identificadas en el intervalo *task\_uid [1,105]* de la tabla *Msp\_Tasks* (Figura IV.9 y Tabla IV.1). Su planificación se recoge en las Figuras IV.1 y IV.2.
- ii) «(64) *ASI Instalaciones y Equipos*». Es la tarea (64) perteneciente al proyecto (1) anterior. Esta tarea aparece en el plan de este proyecto a ese nivel de desglose, pero a efectos del control de calidad detallado de entregables requiere de una planificación detallada. Esto se realiza a través de un subproyecto; es decir esta tarea es, simultáneamente, una tarea que forma parte del proyecto (1) y un proyecto (64) o subproyecto.

- iii) «(1001) **Patrón Fase NDT (ASI) Análisis del Sistema**». Es un proyecto o subproyecto patrón de la fase *NDT ASI*. La manera de llevar a cabo la planificación de estas tareas o fases *NDT (REQ, ASI, DSI, CSI, etc.)* es a través de la descomposición en tareas elementales según la metodología y definición de hitos para la revisión de entregables, tanto respecto a normas técnicas como para contrastar o balancear la trazabilidad de modelos (p. ej.: verificar modelos del *ASI* respecto a los requisitos). Cada patrón se aplica a todas las tareas de la misma tipología.

### 8.1.2 Extracción de tareas, restricciones y relaciones de precedencia

El proyecto tiene un conjunto de 173 tareas (Figura IV.9, Tabla IV.1), donde están embebidas las restricciones sobre tareas, y 141 relaciones de precedencia de donde se derivarán las dependencias temporales entre tareas. En esta sección se van a realizar extracciones de los artefactos relacionales (filas de tablas que contienen las reglas de tiempo) ligados a cada proyecto para alimentar el proceso de extracción descrito en el Capítulo 7. El destino serán modelos, de distinto nivel de abstracción, que contienen las reglas temporales en una representación más cercana al experto, que en este caso serán diagramas *BPMN* conforme al *Meta-Modelo* extendido (Figura 7.16).

Se plantean las siguientes extracciones, realizadas con consultas *SQL*:

- i) **Tareas de primer nivel de la planificación general**. Se van a definir dos subconjuntos de tareas: a) Las de primer nivel, ligadas a la organización del proyecto, control y seguimiento del proyecto y las fases en que se descomponen el proyecto.
- ii) **Grupo de tareas ligadas a un subsistema**. Se desciende hasta el máximo nivel de detalle de una de las fases del proyecto.
- iii) **Tarea elemental de la planificación general** que supone la realización de una fase *NDT* como *ASI (Análisis del sistema)*. Se trata de descomponer esta actividad en un subproyecto donde constan tareas para el desarrollo de los distintos modelos o entregables, así como los hitos de control de calidad asociados. El modo de abordar cada fase *NDT* es a través del desglose pormenorizado de sus tareas en un **patrón estándar de planificación** que es la última extracción.
- iv) **Patrón *NDT ASI***.

#### 8.1.2.1 Extracción E1: tareas generales y fases de la planificación general

La primera tarea de un proyecto MS Project es la actividad resumen, que equivale al proyecto, visto también como tarea de grupo.

En esta primera extracción (*proj\_uid=1* y *task\_uid=1*) son los datos identificativos del proyecto. En la Figura 8.1 se ha realizado una extracción *SQL* (*tareas dependientes de las tareas con task\_uid IN (1, 2, 3)*) de las tareas generales asociadas a la organización

del proyecto, que contiene las tareas asociadas al control de calidad, control y seguimiento del proyecto y las tareas que representan a los distintos subsistemas o subproyectos a controlar (Ver planning de Figuras IV.1 y IV.2).

task_uid	task_parent_uid	task_name	task_dur_js_est	task_dur	task_constraint_type	abbreviation	task_constraint_date
2	1	Seguimiento y Control del Proyecto	✓	567,00	6	FNET	19/01/2010 0:00:00
3	1	Organización y lanzamiento del Proyecto		109,00	0	ASAP	{null}
4	3	Reunión de Lanzamiento		1,00	0	ASAP	{null}
5	3	Elaboración y Revisión Manual de Calidad		23,00	6	FNET	10/12/2007 0:00:00
6	3	Elaboración y Revisión Plan de Gobierno del Proyecto		17,00	6	FNET	21/12/2007 0:00:00
7	3	Elaboración Plan Maestro del Proyecto		15,00	6	FNET	28/12/2007 0:00:00
8	3	Elaboración y Aceptación Plan de Calidad		3,00	6	FNET	11/01/2008 0:00:00
9	3	Elaboración del Plan de Comunicación		10,00	4	SNET	03/03/2008 0:00:00
10	3	Puesta en marcha del Plan de Comunicación		24,00	0	ASAP	{null}
11	1	Alfa 0.1: Capa Transversal. Subsistema Común	✓	220,00	0	ASAP	{null}
40	1	Alfa 0.2 : Gestión de Clientes.		254,75	0	ASAP	{null}
56	1	Alfa 0.3: Gestión de Obras y Proyectos		173,00	0	ASAP	{null}
61	1	Alfa 0.4: Clientes: Intervenciones en Red	✓	229,50	0	ASAP	{null}
80	1	Alfa 0.5: Clientes. Informes de Gestión		265,00	0	ASAP	{null}

Figura. 8.1. E1::Tareas Organizativas, Control de Calidad y Fases AQUA-WS

La Figura 8.2 es el resultado de la extracción SQL que genera como resultado el conjunto de relaciones de precedencia que ligan a las tareas de la Figura 8.1. Con las extracciones anteriores se puede generar un diagrama de procesos (Ver Figura 8.9).

Link_Succ_Uid	Sucesor	Link_Pred_Uid	Predecesor	Link_type	abbreviation	Link_lag
2	Seguimiento y Control del Proyecto	94	Beta 0.6: Núcleo Aqua	0	FF	{null}
2	Seguimiento y Control del Proyecto	3	Organización y lanzamiento del Proyecto	3	SS	{null}
5	Elaboración y Revisión Manual de Calidad	4	Reunión de Lanzamiento	1	FS	{null}
6	Elaboración y Revisión Plan de Gobierno del Proyecto	5	Elaboración y Revisión Manual de Calidad	1	FS	{null}
7	Elaboración Plan Maestro del Proyecto	6	Elaboración y Revisión Plan de Gobierno del Proyecto	1	FS	{null}
8	Elaboración y Aceptación Plan de Calidad	7	Elaboración Plan Maestro del Proyecto	1	FS	{null}
9	Elaboración del Plan de Comunicación	8	Elaboración y Aceptación Plan de Calidad	1	FS	{null}
10	Puesta en marcha del Plan de Comunicación	9	Elaboración del Plan de Comunicación	1	FS	{null}
11	Alfa 0.1: Capa Transversal. Subsistema Común	3	Organización y lanzamiento del Proyecto	3	SS	5,00
40	Alfa 0.2 : Gestión de Clientes.	11	Alfa 0.1: Capa Transversal. Subsistema Común	3	SS	30,00
56	Alfa 0.3: Gestión de Obras y Proyectos	40	Alfa 0.2 : Gestión de Clientes.	3	SS	30,00
61	Alfa 0.4: Clientes: Intervenciones en Red	56	Alfa 0.3: Gestión de Obras y Proyectos	3	SS	30,00
80	Alfa 0.5: Clientes. Informes de Gestión	61	Alfa 0.4: Clientes: Intervenciones en Red	3	SS	30,00
94	Beta 0.6: Núcleo Aqua	80	Alfa 0.5: Clientes. Informes de Gestión	3	SS	30,00

Figura. 8.2. E1::Precedencias en Tareas Organizativas, Control de Calidad y Fases AQUA-WS

### 8.1.2.2 Extracción E2: Grupo de tareas ligadas a un subsistema

Se elige la fase «Alfa 0.4: Clientes: Intervenciones en Red (task\_uid=61)». En la



Figura 8.3 se realiza la extracción SQL para el conjunto de tareas de detalle que corresponden al subproyecto o fase (Ver planning de Figuras IV.1 y IV.2). La Figura 8.4 representa las relaciones de precedencia para las tareas involucradas en la Figura 8.3.

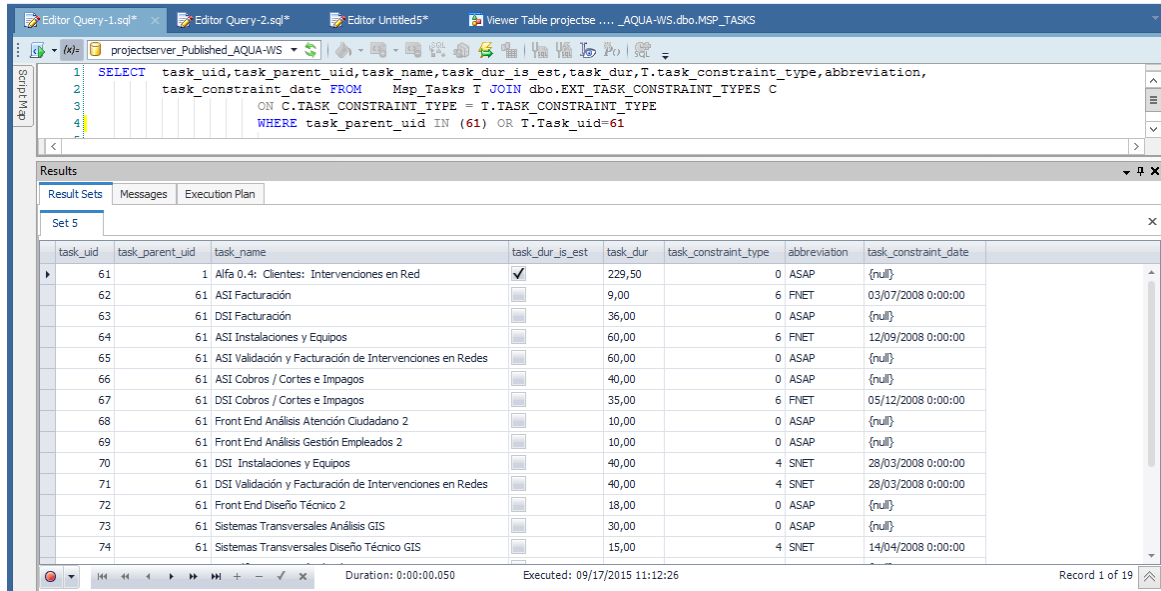


Figura. 8.3. Tareas de un grupo de actividades (AQUA-WS)

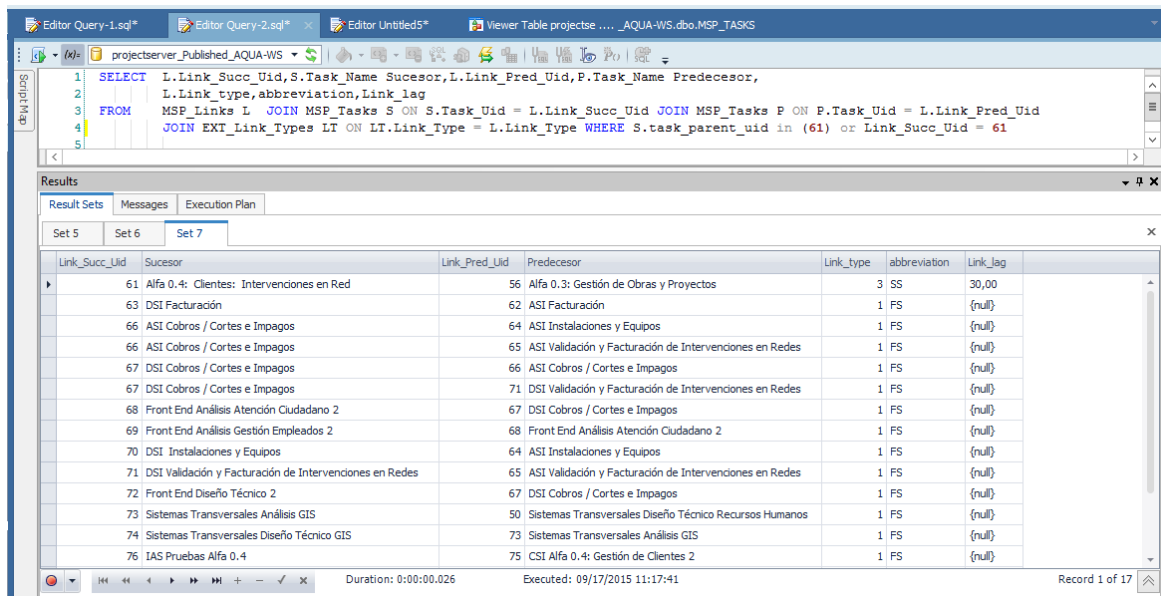


Figura. 8.4. Dependencias en tareas de grupo (AQUA-WS)

### 8.1.2.3 Extracción E3: Tarea elemental de la planificación general

La tarea «(64) ASI Instalaciones y Equipos» es una tarea elemental (no se desglosa más a ese nivel) en la planificación general. No obstante, para llevar a cabo el trabajo de control de calidad se necesita un nivel mayor de detalle que viene normalizado por la

aplicación de la *Metodología NDT a la fase de Análisis del Sistema (ASI)* (Ver Planning de la *Figura IV.3*). Esta tarea «(64)» es un caso concreto de aplicación del patrón *ASI* dentro de *NDT* (ver Sección siguiente). Se contempla como un subproyecto subordinado al proyecto general y que se modela como un subproyecto. Sus tareas están comprendidas en el intervalo  $task\_uid \in [64.001, 64.999]$  para el proyecto  $proj\_uid=64$ . La *Figura 8.5* representa el resultado de la extracción de tareas y la *Figura 8.6* la extracción de relaciones de precedencia.

task_uid	task_parent_uid	task_name	task_dur	abbreviation	task_start_date	task_finish_date
64001	{null}	(64): (ASI) Fase de Análisis - Instalaciones y Equipos	9,00	ASAP	23/06/2008 0:00:00	03/07/2008 0:00:00
64002	64001	(ASI-MN) Modelo de Navegación	2,00	ASAP	23/06/2008 0:00:00	24/06/2008 0:00:00
64003	64002	(ASI-MN-BA) Modelo de Navegación Básico	1,00	ASAP	23/06/2008 0:00:00	23/06/2008 0:00:00
64004	64003	Desarrollo	1,00	ASAP	23/06/2008 0:00:00	23/06/2008 0:00:00
64005	64003	Verificación QA	0,00	ASAP	23/06/2008 0:00:00	23/06/2008 0:00:00
64006	64002	(ASI-MN-FI) Modelo de Navegación Final	1,00	ASAP	24/06/2008 0:00:00	24/06/2008 0:00:00
64007	64006	Desarrollo	1,00	ASAP	24/06/2008 0:00:00	24/06/2008 0:00:00
64008	64006	Verificación QA	0,00	ASAP	24/06/2008 0:00:00	24/06/2008 0:00:00
64009	64002	Verificación QA-Requisitos	0,00	ASAP	24/06/2008 0:00:00	24/06/2008 0:00:00
64010	64001	(ASI-MC) Modelo de Contexto	2,00	ASAP	25/06/2008 0:00:00	26/06/2008 0:00:00
64011	64010	(ASI-MN-BA) Modelo de Contexto Básico	1,00	ASAP	25/06/2008 0:00:00	25/06/2008 0:00:00
64012	64011	Desarrollo	1,00	ASAP	25/06/2008 0:00:00	25/06/2008 0:00:00
64013	64011	Verificación QA	0,00	ASAP	25/06/2008 0:00:00	25/06/2008 0:00:00
64014	64010	(ASI-MC-FI) Modelo de Contexto Final	1,00	ASAP	26/06/2008 0:00:00	26/06/2008 0:00:00

Figura. 8.5. Tabla Msp\_Tasks para el subproyecto (64) (AQUA-WS)

Link_Succ_Uid	Sucesor	Link_Pred_Uid	Predecesor	Link_type	abbreviation	Link_lag
64005	Verificación QA	64004	Desarrollo	1	FS	{null}
64006	(ASI-MN-FI) Modelo de Navegación Final	64003	(ASI-MN-BA) Modelo de Navegación Básico	1	FS	{null}
64008	Verificación QA	64007	Desarrollo	1	FS	{null}
64009	Verificación QA-Requisitos	64003	(ASI-MN-BA) Modelo de Navegación Básico	1	FS	{null}
64009	Verificación QA-Requisitos	64006	(ASI-MN-FI) Modelo de Navegación Final	1	FS	{null}
64013	Verificación QA	64012	Desarrollo	1	FS	{null}
64014	(ASI-MC-FI) Modelo de Contexto Final	64011	(ASI-MN-BA) Modelo de Contexto Básico	1	FS	{null}
64016	Verificación QA	64015	Desarrollo	1	FS	{null}
64017	Verificación QA-Requisitos	64011	(ASI-MN-BA) Modelo de Contexto Básico	1	FS	{null}
64017	Verificación QA-Requisitos	64014	(ASI-MC-FI) Modelo de Contexto Final	1	FS	{null}
64021	Verificación QA	64020	Desarrollo	1	FS	{null}
64022	(ASI-MI-FI) Modelo de Interface Abstracta Final	64019	(ASI-MI-BA) Modelo de Interface Abstracta Básico	1	FS	{null}
64024	Verificación QA	64023	Desarrollo	1	FS	{null}

Figura. 8.6. Tabla Msp\_Links para el subproyecto (64) (AQUA-WS)

8.1.2.4 Extracción E4: Patrón NDT ASI (Análisis del sistema)

El proyecto y subproyecto «(10001) Patrón Fase NDT (ASI) Análisis del Sistema» representa a esta fase dentro de la *Metodología NDT*. Este proyecto patrón se replica para planificar cada instancia o caso de un ASI que sigue *NDT*. Se contempla como un subproyecto independiente del proyecto general, se modela como un subproyecto y su desglose sirve para facilitar la planificación de cada instancia. Sus tareas están comprendidas en el intervalo  $task\_uid \in [10001.001, 10001.999]$  para el proyecto  $proj\_uid = 10001$ . La *Figura 8.7* representa el resultado de la extracción de tareas y la *Figura 8.8* la extracción de relaciones de precedencia.

```

1 SELECT task_uid, task_parent_uid, task_name, task_dur, abbreviation, task_start_date, task_finish_date
2 FROM Msp_Tasks T JOIN dbo.EXT_TASK_CONSTRAINT_TYPES C
3 ON C.TASK_CONSTRAINT_TYPE = T.TASK_CONSTRAINT_TYPE
4 WHERE proj_uid = 10001
5

```

task_uid	task_parent_uid	task_name	task_dur	abbreviation	task_start_date	task_finish_date
1001001	{null}	(1001): (ASI) Fase de Análisis - Instalaciones y Equipos	{null}	ASAP	{null}	{null}
1001002	1001001	(ASI-MN) Modelo de Navegación	{null}	ASAP	{null}	{null}
1001003	1001002	(ASI-MN-BA) Modelo de Navegación Básico	{null}	ASAP	{null}	{null}
1001004	1001003	Desarrollo	{null}	ASAP	{null}	{null}
1001005	1001003	Verificación QA	{null}	ASAP	{null}	{null}
1001006	1001002	(ASI-MN-FI) Modelo de Navegación Final	{null}	ASAP	{null}	{null}
1001007	1001006	Desarrollo	{null}	ASAP	{null}	{null}
1001008	1001006	Verificación QA	{null}	ASAP	{null}	{null}
1001009	1001002	Verificación QA-Requisitos	{null}	ASAP	{null}	{null}
1001010	1001001	(ASI-MC) Modelo de Contexto	{null}	ASAP	{null}	{null}
1001011	1001010	(ASI-MN-BA) Modelo de Contexto Básico	{null}	ASAP	{null}	{null}
1001012	1001011	Desarrollo	{null}	ASAP	{null}	{null}
1001013	1001011	Verificación QA	{null}	ASAP	{null}	{null}
1001014	1001010	(ASI-MC-FI) Modelo de Contexto Final	{null}	ASAP	{null}	{null}

Figura. 8.7. Tabla Msp\_Tasks para el subproyecto patrón (1001): Fase NDT ASI

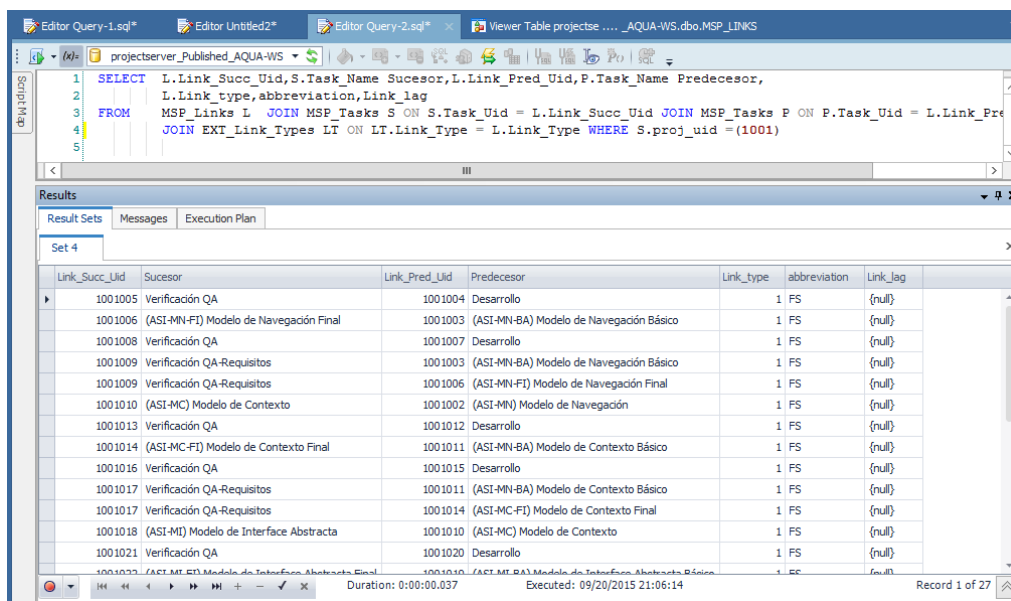


Figura. 8.8.Tabla Msp\_Links para el subproyecto patrón (1001): Fase NDT ASI

## 8.2 Generación de modelos de procesos

En esta sección se van a generar diagramas *BPMN* mediante el proceso de ingeniería inversa propuesto (Algoritmos y tablas de transformación del Capítulo 7), partiendo de las tablas relacionales y generando, en primer lugar, un modelo conforme al *Meta-Modelo pivote PIM-upl* (Figura 7.3), y posteriormente, generando un modelo *BPMN* conforme a su *Meta-Modelo extendido* (Figura 7.16) con reglas temporales. La Tabla 8.1 relaciona los artefactos involucrados en las transformaciones de los planes del proyecto en aproximaciones a los procesos de negocio expresados en *BPMN*. Se describe el subconjunto de tareas extraídas del plan, las tablas relacionales asociadas y, por último, el diagrama *BPMN* generado.









**Tabla. 8.1.** Mapeo de elementos de la planificación AQUA-WS a procesos *BPMN*

Proceso Extracción de datos	Planificación MS Project	Artefactos origen extraídos Tablas	Proceso <i>BPMN</i>
E1: Tareas generales y subsistemas de la planificación general	Figuras IV.1 y IV.2	Figuras 8.1, 8.2	Figura 8.9
E2: Grupo de tareas ligadas al subsistema «(61)Alfa 0.4: Clientes: Intervenciones en Red»	Figuras IV.1 y IV.2	Figuras 8.3, 8.4	Figura 8.10
E3: Tarea elemental de la planificación general, definido como subproyecto «(64) ASI Instalaciones y Equipos»	Figura IV.3	Figuras 8.5, 8.6	Figura 8.10
E4: Subproyecto «(10001) Patrón Fase NDT (ASI) Análisis del Sistema»	Figura IV.4	Figuras 8.7, 8.8	Figura 8.11

En los diagramas *BPMN* generados mediante la ingeniería inversa, un *proyecto* se mapea como un *proceso*, un *subproyecto* o *conjunto de actividades* como un *subproceso*

*BPMN* y cada tarea como una actividad *BPMN* (A efectos de aclarar la trazabilidad entre tareas y actividades, en el diagrama se incluye el identificador de la tarea origen delante del nombre). Las tareas contienen restricciones sobre duración y sobre los eventos de inicio y terminación de la tarea. Las relaciones de precedencia originan las dependencias temporales entre tareas. En resumen, cada extracción de datos de la [Tabla 8.1](#) va a dar lugar a un modelo que instancia al *Meta-Modelo BPMN extendido* ([Figura 7.16](#)).

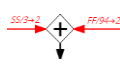
Se han considerado las siguientes convenciones para expresar la semántica temporal sobre los diagramas *BPMN* generados:

- i) **Grupos de tareas.** Las jerarquías de tareas (tarea de grupo y descomposición en tareas de un nivel jerárquico inferior) se mapean recursivamente como subprocesos *BPMN* hasta llegar a tareas elementales o en el nivel de profundidad más detallado. Si no existen relaciones de precedencia entre tareas de un subproceso, se supone ejecución en paralelo para las actividades en el diagrama generado.
- ii) **Restricciones temporales sobre actividades.** Se expresan con leyendas en azul, p. ej.: a) duraciones sobre actividades fijas  y  flexibles; y b) Inicio flexible  o terminación flexible  o . Esta semántica no existe en *BPMN*, pero sí en el *Meta-Modelo extendido* ([Figura 7.16](#)). Si la herramienta de modelación de procesos lo permitiese, podrían utilizarse decoradores (como en *Time-BPMN* [[Gagné & Trudel 2009a](#)] o en [[Cheikhrouhou et al. 2013b](#)]), aunque se ha decidido optar por mantener las características del estándar *BPMN 2.0*.
- iii) **Dependencias temporales.** Los flujos de control asociados a las relaciones de precedencia explícitas de la planificación aparecen con el arco en rojo  para la actividad sucesora. Se especifican leyendas aclarativas sobre estos arcos (normalmente usadas en *BPMN* para establecer un predicado o condición del flujo), también en rojo, para referenciar las reglas de dependencia respecto a la actividad predecesora; así:  representa una dependencia temporal (*Finish to Start*) entre dos actividades (es el flujo por defecto), y  especifica una dependencia temporal (*Start to Start*) entre la actividad predecesora (3) y la sucesora (2).

La [Figura 8.9](#) muestra un diagrama *BPMN* con el proceso general del proyecto, ligado a la extracción E1 de la base de datos:

- i) La actividad «(3) Organización y lanzamiento del Proyecto» es un Subproceso desplegado que contiene actividades generales del proyecto y las generales del control de calidad, como «(5) Elaboración y Revisión Manual de Calidad», «(6) Elaboración y Revisión Plan de Gobierno del Proyecto», «(7) Elaboración Plan Maestro del Proyecto», «(8) Elaboración y Aceptación Plan de Calidad», «9

*Elaboración del Plan de Comunicación», «(10) Puesta en marcha del Plan de Comunicación».*

- ii) El conjunto de subsistemas que se han mapeado como Subprocesos: «(11) Alfa 0.1: Capa Transversal. Subsistema Común», «(40) Alfa 0.2: Gestión de Clientes», «(56) Alfa 0.3: Gestión de Obras y Proyectos», «(61) Alfa 0.4: Clientes: Intervenciones en Red», «(80) Alfa 0.5: Clientes. Informes de Gestión» y «(94) Beta 0.6: Núcleo Aqua» representan las actividades globales de control de calidad que hay que realizar sobre los mismos. Estos subsistemas tienen dependencias temporales entre ellos, con reglas temporales que prevén su inicio con un período entre ellos de 30d, a excepción del primero que se inicia la semana siguiente al inicio del proyecto. Todos corren en paralelo, lo cual se expresa con las puertas BPMN correspondientes.
- iii) La actividad «(2) Seguimiento y Control del Proyecto» representa a las actividades elementales de control de calidad de esta naturaleza (control de entregables, hitos, análisis de desviaciones, etc.). Se inicia en paralelo (*dependencia SS*) con la actividad «(3) Organización y lanzamiento del Proyecto» y termina con el último subsistema (*dependencia FF*) «(94) Beta 0.6: Núcleo Aqua», lo cual se expresa con la puerta paralelo  correspondiente.

La Figura 8.10 corresponde al diagrama BPMN de detalle del subsistema «(61) Alfa 0.4: Clientes: Intervenciones en Red», asociado a la extracción de datos E2.

La Figura 8.11 corresponde al diagrama BPMN del subproyecto «(64) (ASI) Instalaciones y Equipos», asociado a la extracción de datos E3 y representa una instancia de esta fase NDT, cuyo patrón o subproceso genérico se representa en la Figura 8.12 (1001) Patrón NDT (ASI) Análisis del Sistema», asociado a la extracción de datos E4. Tanto en el patrón genérico del proceso NDT ASI como en la instancia de la actividad (64) aparecen actividades para el desarrollo de cuatro modelos y la verificación de los mismos (en la planificación son hitos o tareas con duración cero); podrían mapearse desarrollo+verificación con dos actividades ( $\square \rightarrow \square$ ), pero, en realidad, la verificación implica un punto de decisión, por lo que podría mapearse (*actividad con duración nula* ⇒ *puerta exclusiva*) ( $\square \rightarrow \diamond$ ). En la sección siguiente, esta última consideración ayudará a comparar la semántica capturada en los modelos obtenidos (Figuras 8.11 y 8.12) con el proceso real que modela un experto en software y NDT (Figura 8.13).

8.2.1 Extracción E1: tareas generales y fases de la planificación general

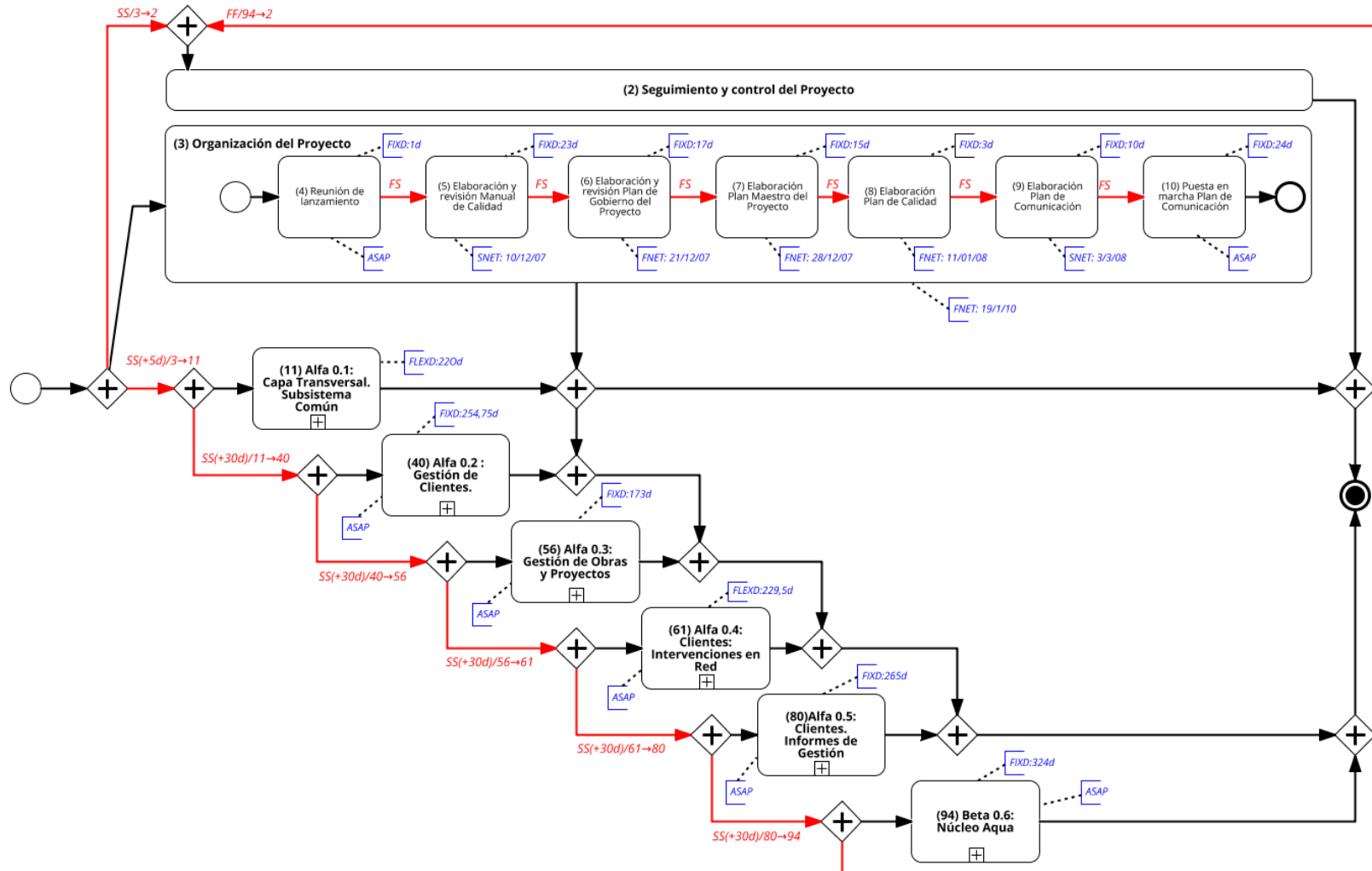


Figura. 8.9. Actividades generales y subsistemas AQUA-WS

8.2.2 Extracción E2: Grupo de tareas ligadas a un subsistema

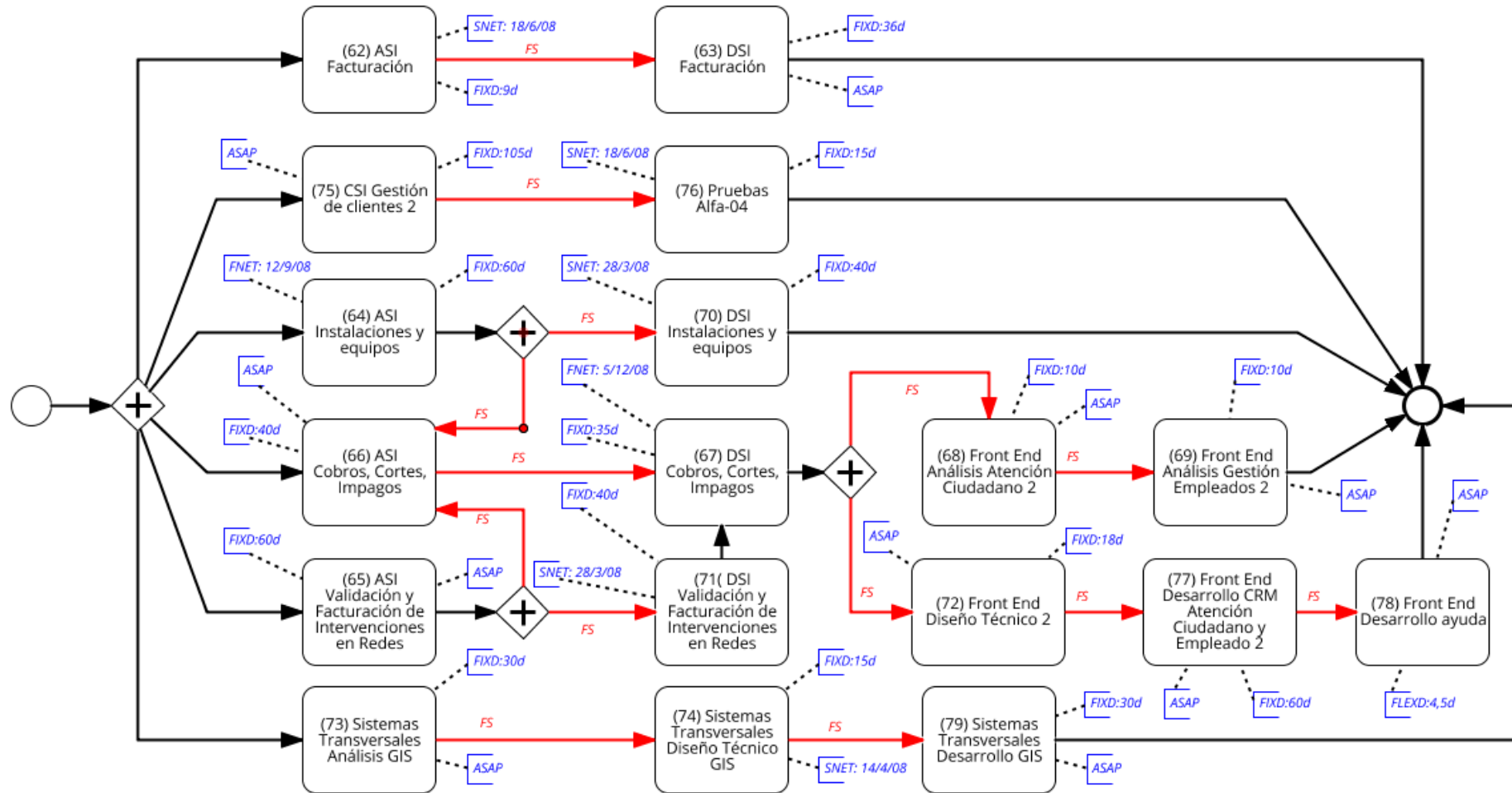


Figura. 8.10. Proceso BPMN (61) Fase Alfa 0.4: Clientes Intervenciones en Red (AQUA-WS)



8.2.3 Extracción E3: Tarea elemental de la planificación general

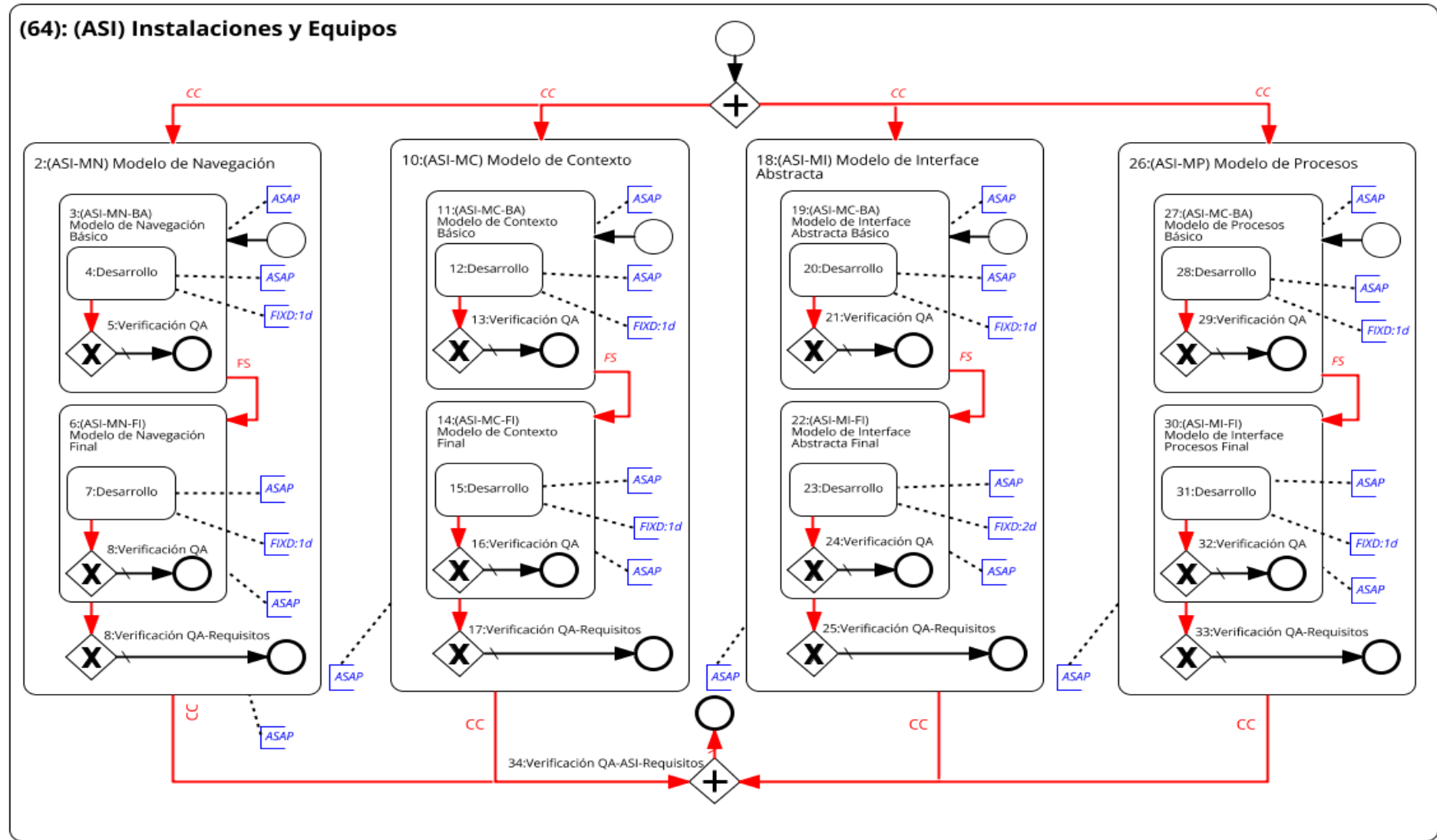


Figura. 8.11. Proceso de negocio BPMN (64) (ASI) Instalaciones y Equipos (AQUA-WS)

8.2.4 Extracción E4: Patrón NDT ASI

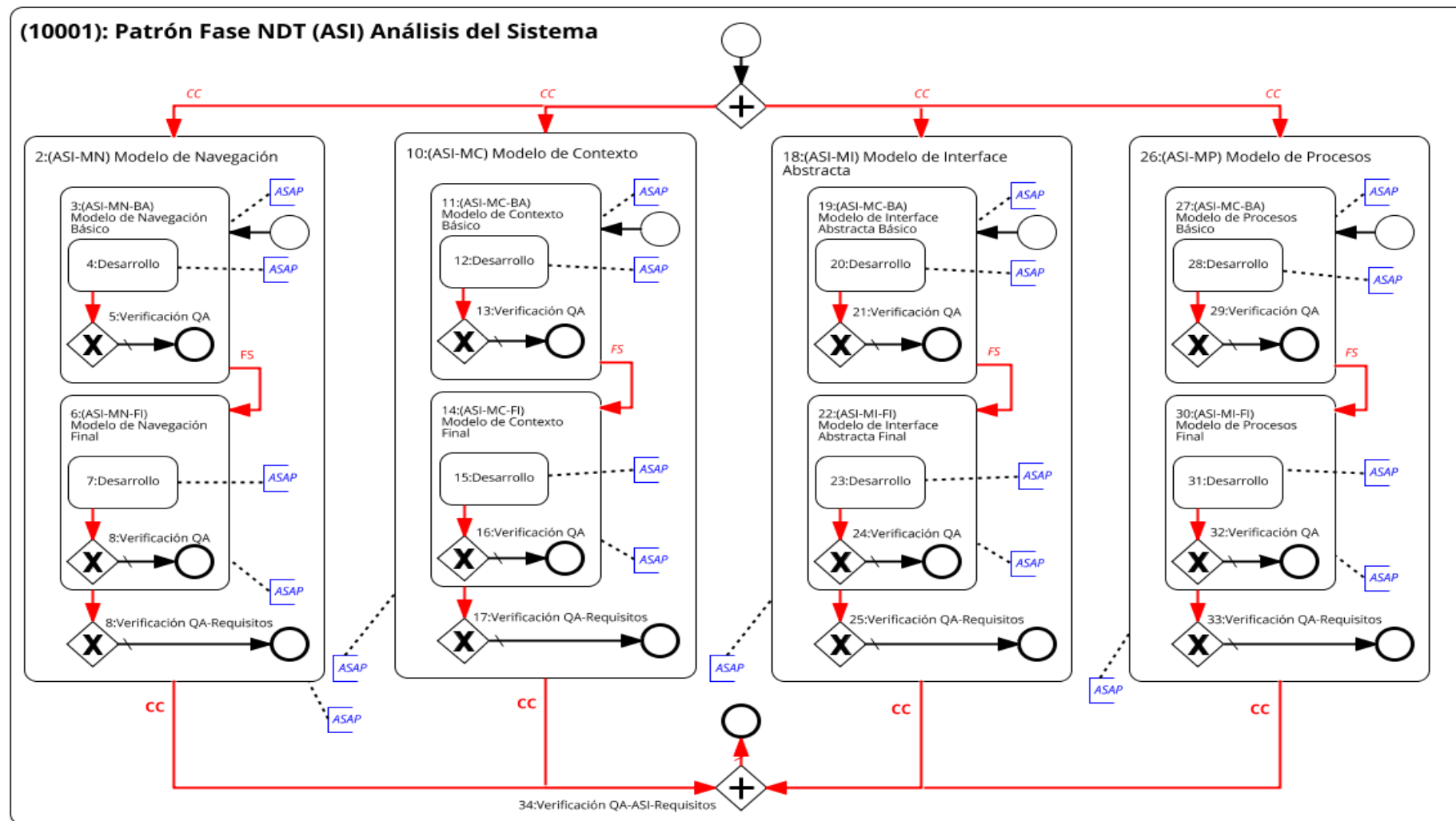


Figura. 8.12. Proceso de negocio BPMN (1001) Patrón NDT (ASI) Análisis del

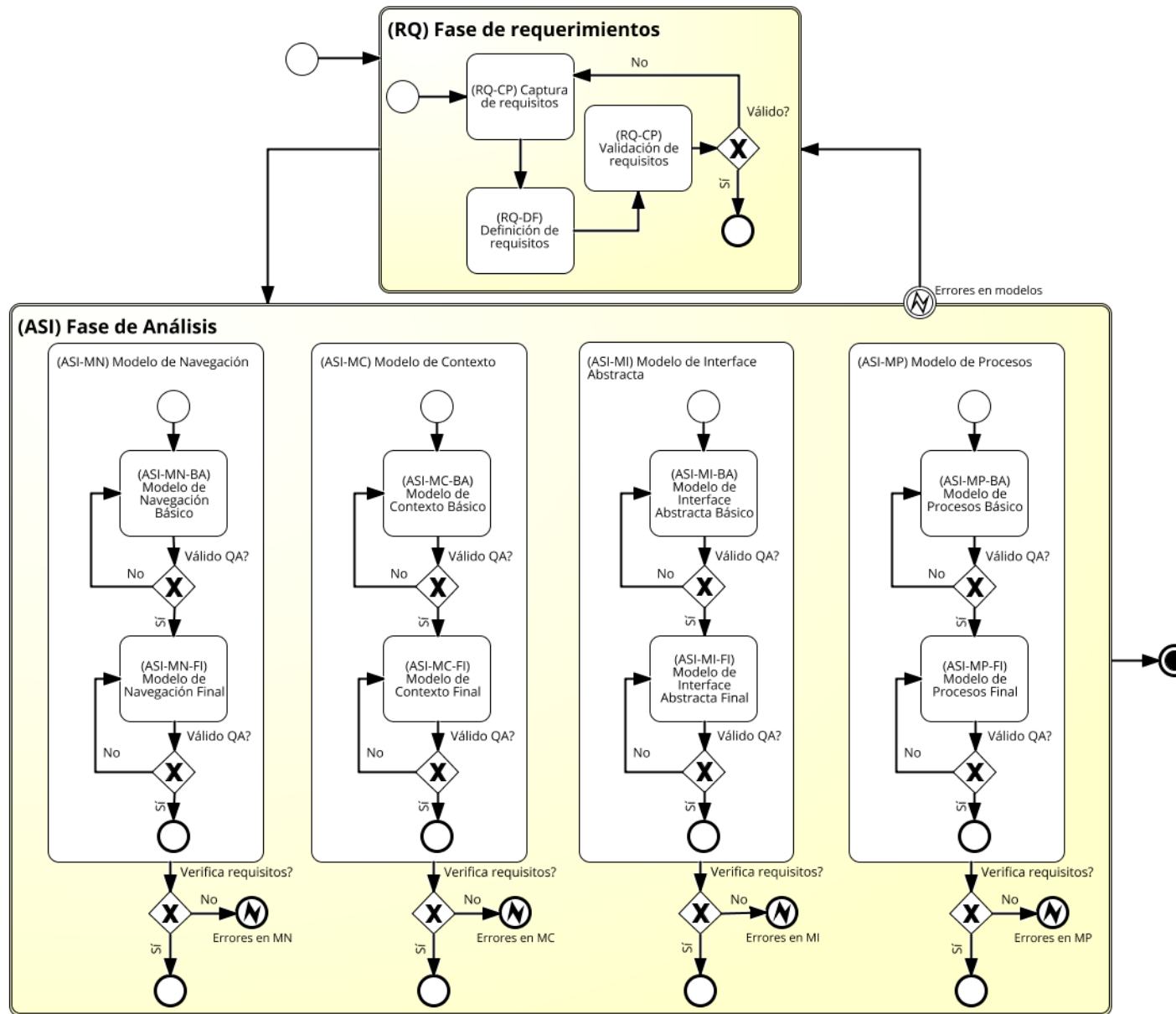


Figura. 8.13. Procesos de negocio NDT: Fases de Requerimientos y Análisis

### 8.3 Discusión de resultados

En los apartados anteriores se ha descrito el proceso de extracción de un conjunto de procesos de negocio *BPMN* desde *MS Project Server (legacy database)* en el caso *AQUA-WS Project*, basado en ingeniería inversa *MDE* y manejando las reglas asociadas a la *perspectiva temporal* que contiene un plan de proyecto.

#### 8.3.1 Proceso general y fases del proyecto

En los diagramas *BPMN* (Figuras 8.9 y 8.10) obtenidos se observa:

- i) Se capturan aspectos ligados a dos perspectivas de un proceso: a) *Perspectiva de flujo de control* y b) *Perspectiva temporal* de un proceso. A la primera corresponde el ordenamiento y agrupamiento de actividades, al derivar procesos y subprocesos y organizar la ejecución secuencialmente o en paralelo según la organización jerárquica de tareas del plan y sus relaciones de precedencia. Por otro lado, se han capturado las *restricciones temporales* asociadas a la *duración*, *condiciones* sobre los *eventos de inicio* o *terminación* de la *actividad* y las *dependencias temporales* asociadas al *tipo de relación de precedencia*.

Los procesos obtenidos son un punto de partida para el experto en negocio del software. *BPMN* ofrece mayor claridad y semántica que un diagrama *PERT* [Malcolm et al. 1959; PMI:PMBOK 2013].

- ii) El nivel de desglose de la *planificación es muy general*, lo cual conduce a que los diagramas de proceso representados también lo sean. La Figura 8.9 representa la organización (*subproceso 3 expandido*) y control del proyecto (*Actividad 2*), algunas actividades generales de control de calidad (*Actividades 4 a 10*) y la organización en fases, que se representan como subprocesos *BPMN* colapsados. Descendiendo en profundidad en una de las fases «(61) Alfa 0.4: Clientes: *Intervenciones en Red*» se obtiene el diagrama correspondiente (Figura 8.10); aquí aparecen actividades ligadas al *ciclo del software* (Según *NDT: REQ, ASI, DSI, CSI, etc.*), como, por ejemplo, la «(64) (*ASI*) *Instalaciones y Equipos*», que en el plan global del proyecto se ve sólo a ese nivel. Ahora bien, la manera de abordar la organización y control de estas fases es mediante la metodología *NDT* que se aplicó al proyecto. De este modo, por ejemplo, la fase *NDT ASI* estableciendo los modelos a realizar por cada contratista y los tipos de verificación (aplicación de normas en cada modelo y el contraste de trazabilidad de estos modelos respecto al modelo de requisitos). A continuación se analizará una tarea de este tipo (*Actividad y subproyecto 64, Figura 8.11*), el proceso general que puede derivarse desde un proyecto patrón (*Actividad y subproyecto 10001, Figura 8.12*) y se contrastará con el proceso de negocio *ASI* definido manualmente en *NDT* (Figura 8.13).

### 8.3.2 Fase NDT ASI

Una buena práctica es establecer patrones o estándares de planificación de tareas que son repetitivas e inherentes al proceso en cuestión; en nuestro caso el *proceso del software* y suministrar este patrón al Jefe de Proyecto, para que las parametrize en cada caso, sustituyendo valores, como nombres de tareas, fechas, duraciones, etc., pero manteniendo el desglose e identificación de estas tareas estándar y sus relaciones de precedencia, así como el perfil de los recursos implicados. Si las tareas están identificadas, y asociadas a *tareas tipo o clases de tarea*, se facilitará el establecimiento de estos patrones y redundará en la sistematización y análisis de los planes de proyectos, pero también se ofrecerá otra manera de obtener un proceso de negocio, que estará mejor definido a partir de estos patrones de planificación configurados en *MS Project* (o en la herramienta que se utilice para planificar y controlar proyectos).

En nuestro caso de estudio, para la fase *NDT ASI*, se establece el patrón (*Actividad y subproyecto 10001*) atemporal (no se especifican fechas concretas de inicio y terminación de cada tarea) en *MS Project* (Figura IV.4). Se contemplan cuatro modelos: i) (*ASI-MN*) *Modelo de Navegación*, ii) (*ASI-MC*) *Modelo de Contexto*, iii) (*ASI-MI*) *Modelo de Interface Abstracta* y iv) (*ASI-MP*) *Modelo de Procesos*. Cada modelo tiene dos niveles de elaboración: i) *Modelo básico* y ii) *Modelo final*. Cada una de las versiones de los modelos tiene un hito de revisión, tanto para verificar la aplicación de normas internas del proyecto como para verificar la trazabilidad de estos modelos respecto al modelo de requisitos *NDT*. El análisis (*ASI*) se considera también un modelo de modelos, por lo que requiere el hito de verificación global que se cumple al validar todos los modelos que lo componen. La aplicación de nuestra propuesta de ingeniería inversa *MDE* lleva a obtener el proceso de negocio de la Figura 8.12.

Este patrón es de obligado cumplimiento en el proyecto para cada caso de desarrollo de un Análisis de Sistemas (*ASI*). El patrón se suministra a cada *Jefe de Proyecto* (rol de contratista externo) para que lo aplique a cada caso concreto. El *Jefe de Proyecto* otorga los valores concretos a cada caso de la planificación: nombre de la tarea, fechas de inicio, duraciones, etc. De este modo, la Figura IV.3 representa la planificación de un caso *NDT ASI* dentro del proyecto *AQUA-WS* para la actividad y subproyecto «(64) (*ASI*) *Instalaciones y Equipos*». La Figura 8.11 es el proceso obtenido mediante nuestra ingeniería inversa *MDE*.

Los procesos derivados en las Figuras 8.11 y 8.12 muestran la captura de los distintos modelos e hitos de verificación, pero no es posible deducir ciclos entre tareas, pues esta semántica (asociada al flujo de control) no está en una herramienta como *MS Project*. La Figura 8.13 que muestra un modelo realizado manualmente en *BPMN* por un experto de negocio en *NDT* ofrece mayor expresividad, pues considera *subprocesos-transacción* a la fase de requisitos (*REQ*) y a la de análisis (*ASI*), de modo que cancelaciones o errores en sus tareas no permiten finalizar con éxito la fase.

De cualquier modo, el resultado obtenido ofrece una herramienta útil para el experto de negocio del software, que le permitirá reflexionar sobre una aproximación a la definición de sus procesos y cómo, y a qué nivel de detalle y sistematización, están contemplados en la planificación y control de proyectos, o en el peor de los casos, a extraer una lista de actividades candidatas a formar parte de sus procesos de negocio.

### 8.3.3 Nivel de abstracción de los modelos

Los procesos obtenidos en las Figuras 8.9, 8.10 y 8.11 son modelos o *instancias* concretas (nivel  $M_0$  MOF) [Reichert & Weber 2012] conforme al *Meta-Modelo BPMN* extendido (Figura 7.16) (Nivel  $M_2$  MOF como plataforma *CIM*); dicho de otro modo, cada proyecto concreto, fase de este proyecto, o la planificación de una tarea concreta del ciclo de vida del software, como «(64) *ASI Instalaciones y Equipos*» permiten obtener un caso o una instancia de un proceso general (Nivel  $M_0$  MOF). Desde el punto de vista del experto en negocio es relevante conseguir modelos más generales, a nivel  $M_1$  MOF, que, por ejemplo, respondan a ¿Cómo se realiza, en general, el análisis en *NDT*? En nuestro caso, esto lo permite el tratamiento del patrón (*Actividad y subproyecto 10001*) atemporal y con *tareas bien codificadas e identificadas* para la fase *NDT ASI*, pero de no existir este nivel de abstracción en *MS Project*, es el experto quién tendría que manejar las distintas instancias de procesos, o bien utilizar técnicas alternativas, como «*process mining*» [van der Aalst et al. 2007, 2012; van der Aalst 2011, 2013, 2014, 2015; Reichert & Weber 2012], que utilizan estas instancias para generar *modelos de procesos* [Reichert & Weber 2012] de nivel  $M_1$  MOF.

### 8.3.4 Extensión y reutilización de modelos

Se acaba de apuntar la posible reutilización de los modelos de procesos obtenidos (si son a bajo nivel,  $M_0$ ) con técnicas como «*process mining*» que trabaja con el *log de eventos de ejecución* de un sistema que maneje modelos de procesos (*PAIS*) [Dumas et al. 2005]. *MS Project Server* no es un sistema de esta categoría, por lo que no cuenta con *log de eventos*. Los modelos  $M_0$  obtenidos podrían ser reutilizados para construir un *log* [van der Aalst 2015] estándar como «*Extensible Event Stream (XES)*<sup>42</sup>» [Verbeek et al. 2011] y utilizarlo con herramientas automatizadas para «*process mining*», como *ProM* [Verbeek et al. 2011; Kalenkova et al. 2014; van der Aalst 2015].

Tanto, utilizando patrones de planificación que nos lleven a derivar procesos genéricos (de nivel  $M_1$ ), como utilizando «*process mining*», es fundamental la clasificación de las tareas en categorías (identificación, tipificación, etc.).

Por otro lado, el foco de la tesis es la ingeniería inversa basada en *MDE*, utilizando la dimensión temporal de los procesos, pero *MS Project Server* (o el sistema de partida en cuestión, sea para planificación o control de proyectos, para la gestión documental,

---

<sup>42</sup> <http://www.xes-standard.org/>

etc.) tiene la capacidad de definir además: recursos, grupos de recursos y asignarlos a tareas, lo cual está relacionado con la *perspectiva organizacional o de recursos* (La [Tabla IV.3](#) es un ejemplo) de un proceso. Tratar esta información, de modo similar a como se ha tratado la *dimensión temporal*, y extender el *Meta-Modelo pivote PIM-upl* ([Figura 7.3](#)) usado en la ingeniería inversa propuesta, llevaría a enriquecer el contenido y calidad de los procesos, pues podrían aparecer los recursos como calles, los grupos de recursos como piscinas, etc. y a resolver la asignación de estos recursos a las actividades. Existen trabajos como [[Awad et al. 2009](#); [Stroppi et al. 2012](#)] que manejan patrones de asignación de recursos y contemplan *Meta-Modelos* que extienden *BPMN 2.0* con la perspectiva de los *recursos* y de *casos*. La fusión de estos *Meta-Modelos* con el *Meta-Modelo PIM-upl* ([Figura 6.11](#)) no requeriría mucho esfuerzo y sería compatible con nuestro enfoque; de este modo los procesos obtenidos gozarían de más semántica: *control del flujo*, *dimensión temporal* y *la de recursos y casos*.

## 8.4 Resumen y conclusiones

En este Capítulo se ha aplicado la solución propuesta al caso *AQUA-WS Project*. El proyecto es un caso de modernización de software, que sustituye plataformas tecnológicas antiguas por desarrollos Web. Nuestro grupo de investigación, *IWT2*, aportó la metodología «*Navigational Development Technique (NDT)*» y ejecutó los procesos de control de calidad. El análisis de los resultados refleja la viabilidad de la propuesta de investigación, que genera automáticamente procesos expresados con *BPMN* a partir de los *Meta-Modelos* de la base de datos relacional *MS SQL\*Server 2008 (legacy database)* donde se publican los proyectos realizados con *MS Project Server*. No obstante, hay que tener en cuenta que los procesos obtenidos son una aproximación a los procesos de la organización, pues no se han tratado otras perspectivas como: la *organizacional* o de *recursos*, la *de información*, la *funcional* o la de *casos*; además, *MS Project* es una herramienta para planificar y controlar proyectos, pero no es un sistema de la categoría *BPMS*, que están diseñados expresamente para modelar las características de un proceso.

En cuanto a la calidad de los procesos obtenidos, se ha detectado que está relacionada con la clasificación de las tareas del plan. La situación más desfavorable es partir de un plan poco detallado y con tareas no tipificadas, lo que conducirá a obtener una aproximación al proceso, representada por una instancia concreta; es decir: un *proyecto* corresponde a un *proceso*. Se ha analizado otro caso definido mediante *patrones de planificación para tareas repetitivas y pertenecientes al ciclo de vida del software*, por ejemplo, la fase *NDT (ASI)* que se replica para cada subsistema del proyecto global. En este caso, *el patrón de planificación para ASI* lleva, en vez de a generar un proceso concreto, a obtener un *proceso genérico, clase o modelo de proceso*. También se ha comparado el modelo obtenido con el modelo definido manualmente por un experto en *NDT*; se ha observado un grado de aproximación aceptable, pero con artefactos *BPMN*

simples, pues el enfoque no considera la semántica de transacciones y los eventos de cancelación que estas conllevan (que el experto en *NDT* ha utilizado sobre *BPMN*).

Por último, *MS Project Server* no es un sistema *PAIS*, razón por la cual no cuenta con uno de sus artefactos típicos como es el *log de eventos* o de actividad del sistema. Los procesos obtenidos, como instancias de nivel *MOF M<sub>0</sub>* pueden utilizarse como *inputs* para generar un *log de eventos estándar* como *XES*. El formato *XES* está soportado por sistemas automatizados de «*process mining*», como es el caso de *ProM*. «*Process mining*» genera modelos de procesos aplicando algoritmos basados en el análisis de los patrones detectados en el *log*, aunque, habitualmente, sólo extraen la perspectiva del *control del flujo de ejecución*.



## **PARTE IV. CONCLUSIONES**



# Capítulo 9. CONCLUSIONES Y TRABAJO FUTURO

*"Lo importante es no dejar de hacerse preguntas"*  
*Albert Einstein (1879-1955)*

**A**quí se resumen las contribuciones realizadas en esta tesis doctoral y se plantean trabajos futuros. La investigación se ha situado en el contexto de implantación del enfoque *BPM* en organizaciones especializadas en el negocio del software «*TI*», dirigiendo el esfuerzo a extraer aproximaciones a los procesos ejecutados en las «*TI*», partiendo de *legacy systems* como los de *planificación y control de proyectos informáticos*. Existen *legacy systems* (de la categoría *PAIS*) que soportan la definición y ejecución de procesos, pero la mayoría solo soportan funciones de negocio; sin embargo, todos estos sistemas albergan una capa de persistencia (*legacy database*) que representa los estados resultantes de procesos ejecutados (incluso manualmente o al margen de los sistemas informáticos) en las «*TI*». Este trabajo de tesis es una propuesta para la extracción automática de procesos desde *legacy systems* (como *MS Project Server*) basada en el paradigma *MDE*, poniendo el foco de estudio en la *dimensión temporal*. El resultado final son aproximaciones a los procesos del software expresados en lenguajes cercanos al experto en software, como *BPMN*, que se han contrastado con el caso de modernización de software *AQUA-WS Project*. El enfoque admite extensiones para contemplar otros *legacy systems*, manejar otras perspectivas de los procesos como: *la organizacional, la funcional, la de información y la de casos*, y especificar los procesos con lenguajes alternativos a *BPMN*.

## 9.1 Contribuciones

En esta sección se describen las contribuciones del trabajo de investigación, asociándolas a los objetivos definidos en el Capítulo 2.

### 9.1.1 Estudio del estado del arte (ob.0).

Se ha realizado un estudio del estado del arte relacionado con el ámbito del trabajo de investigación:

i) *Ingeniería del software dirigida por modelos (MDE)*. Exponiendo los problemas del

desarrollo tradicional de software y destacando las aportaciones del enfoque *MDE* [Kent 2002; Schmidt 2006] para paliarlos.

ii) *Gestión de procesos del software*. Se ha estudiado el enfoque *BPM* [van der Aalst 2004; Netjes et al. 2006] dirigido a la gestión del ciclo de vida del software en organizaciones *TI*, poniendo foco en los sistemas más aceptados por la comunidad científica y técnica para modelar y gestionar procesos del software: a) estándares generales como *UML AD* [OMG:UML 2011], *BPMN* [OMG:BPMN 2013; Iso/Iec:19510 2013] o *XPDL* [Shapiro 2010; WfMC:XPDL 2012]; b) específicos como *SPEM* [Bendraou et al. 2007; OMG: SPEM 2008] o *Iso/Iec:24744* [Iso/Iec:24744 2007, 2014]; así como c) nuestra metodología *NDT* [Escalona 2004; Escalona & Aragón 2008]. Se destaca el estudio de los Meta-Modelos de procesos de estos lenguajes como base para la propuesta del trabajo de investigación.

iii) *Dimensión temporal de los procesos*. Se han estudiado distintas propuestas para la especificación de reglas temporales, identificando un conjunto de reglas soportadas en los mismos. Este estudio ha permitido identificar carencias [Gagné & Trudel 2009a; Flores & Sepúlveda 2011; Watahiki et al. 2011; Cheikhrouhou et al. 2013b] para la representación de esta dimensión en sistemas habituales para la representación de procesos como *UML AD*, *BPMN*, *XPDL*, *SPEM*, *Iso/Iec 24744* y *NDTQ-Framework* [Ponce et al. 2013; Aragón 2014]. Se han identificado también propuestas que extienden *BPMN* para capturar la dimensión temporal, como las propuestas de a) [Flores & Sepúlveda 2011] que manejan patrones en *BPMN 1.2* y que hemos redefinido en *BPMN 2.0*, concluyendo que sobrecargan el flujo de control del proceso, dificultando su comprensión; y b) las de autores como [Gagné & Trudel 2009a] con *Time-BPMN* y [Cheikhrouhou et al. 2013b] que utilizan nuevos decoradores que extienden la semántica temporal del estándar, planteando una propuesta elegante y de fácil comprensión pero que posterga su uso hasta la integración en el estándar *BPMN*.

iv) *Otras dimensiones de los procesos: organizacional y casos*. Hemos estudiado propuestas que formalizan extensiones a *BPMN* para capturar en su *Meta-Modelo* la modelación de patrones de asignación de recursos, como son los trabajos de [Awad et al. 2009; Stroppi et al. 2012].

v) *Legacy systems*. Se ha identificado la problemática general de uso de estos sistemas en cualquier tipo de organización, especialmente la asociada a su mantenimiento para aportar nuevos artefactos que mejoren su estructura. Se han estudiado propuestas generales para modernizar estos sistemas como *OMG ADM* y otros trabajos que realizan una modernización específica de sus bases de datos (*legacy databases*) utilizando el paradigma *MDE*. En este apartado también hemos revisado la propuesta de arqueología de procesos [Pérez-Castillo et al. 2012a] en cuyo campo hemos contribuido en la publicación [Arévalo et al. 2014] para extraer *ECA Rules* desde *legacy databases*.

Concluimos tras este estudio que las propuestas existentes permiten derivar estructuras y

reglas más cercanas al experto del negocio, donde las clases generadas representan con cierta fidelidad las entidades del contexto, pues las reglas de negocio derivadas suelen estar cerca de estas estructuras, aunque es difícil capturar reglas de negocio que están embebidas en el código como algoritmos complejos. Esto último motiva nuestro trabajo con la dimensión temporal de los procesos, como base para extraer su definición desde *legacy databases*.

vi) *Minería de procesos*. Hemos revisado contribuciones científicas significativas en este nuevo campo, que es capaz de derivar modelos de procesos analizando múltiples instancias de estos desde un log de eventos. Estos modelos de procesos suelen representar el flujo de control pero carecen de otros artefactos ligados a otras perspectivas de los procesos. El *log de eventos* es un artefacto esencial donde han de figurar las trazas de los procesos y la asociación de cada tarea a una categoría o tipo de actividad. Hay que tener en cuenta que no siempre es posible contar con *legacy systems* que ofrezcan este artefacto.

### 9.1.2 Meta-Modelos de tareas para legacy system de uso habitual en «TI» (ob.1).

Se han estudiado «*legacy systems*» (Capítulo 5, Anexos II y III) de uso habitual en las «TI», como son: i) *Los sistemas de planificación y control de proyectos*, como *MS Project* [Hansen & Hansen 2013] y *RedMine*<sup>TM</sup> [Lang 2010]; ii) *Sistemas de gestión documental o de contenidos (ECMs)*, como «*Alfresco*<sup>TM</sup>» [Shariff 2013] o «*MS Sharepoint*<sup>TM</sup>» [Smith & Bates 2007]; y iii) Otros sistemas enfocados a la gestión de otras áreas de las organizaciones, como *ERPs*, *CRMs*, *SCMs* [Hendricks et al. 2007], de fabricantes como *SAP*, *ORACLE* o *Microsoft*, o bien *desarrollos a medida*.

En el Anexo II figuran los *Meta-Modelos* de tareas extraídos para: i) *MS Project*, sobre *SGBD MS-Access* [Hennig et al. 2010]; ii) *MS Project Server*, sobre *MS SQL\*Server* [Colledge 2008; Ben-Gan 2012; Microsoft<sup>TM</sup>:T-SQL 2014]; iii) El *ECM Alfresco*, sobre *PostgreSQL* [Momjian 2001]; y iv) El gestor de proyectos open source *RedMine*, sobre *MySQL* [Dubois 2005].

Estos sistemas albergan estructuras y reglas existentes en su capa de persistencia (*legacy database*) que están relacionadas con las distintas perspectivas de los procesos (Tabla 6.1), y en particular, las estructuras relacionadas con la dimensión temporal (Tabla 6.2); estas estructuras son: *tareas*, *agrupaciones de tareas*, *reglas* y *dependencias temporales* entre ellas. El análisis de las tablas 6.1 y 6.2 ha permitido concluir que los *sistemas de planificación y control de proyectos* son un buen punto de partida para detectar *Meta-Modelos* de tareas existentes en cada *legacy system*. Por esta razón, por su extendido uso en el mercado y en proyectos del grupo IWT2, se ha elegido *MS Project Server* [Stover 2007] como primer *legacy system* para validar el enfoque de la propuesta.

*MS Project Server* es un robusto sistema colaborativo para la gestión de proyectos, basado en la arquitectura *MS-SharePoint* [Smith & Bates 2007]; está sustentado sobre cuatro instancias de base de datos *MS SQL\*Server*: i) *Draft*, ii) *Published*, iii) *Archive* y iv) *Reporting*. Las estructuras de las instancias *Draft* y *Published* son muy similares. Se

ha elegido la instancia *Published*, donde figuran todos los proyectos con visibilidad compartida por equipos de proyectos.

De la instancia *Published*, se ha extraído un *Meta-Modelo* de tareas (Figura 7.5), situado en el nivel de abstracción *PSM*. La instancia relacional [Codd 1970, 1992], definida sobre *MS SQL\*Server*. Se contemplan *Meta-Modelos* relacionales basados en árboles de sintaxis abstracta (*ASTM*): i) generales (*GASTM*), como los que se proveen en [OMG:IMM] (Anexo IIIa); y ii) específicos (*SASTM*), desarrollados en [Arevalo et al. 2013] (Anexo IIIb), para soportar restricciones de integridad embebidas mediante código propietario como disparadores [Eisenberg 1996; Feuerstein & Pribyl 2005; Colledge 2008; Ben-Gan 2012; Microsoft™:T-SQL 2014]

### 9.1.3 Taxonomía de reglas temporales (ob.2.).

Los lenguajes de procesos basados en un lenguaje como *BPMN*, ofrecen notables ventajas [Lu & Sadiq 2007] a los expertos del dominio que prefieren estas notaciones gráficas ante otros lenguajes [Bonnet et al. 2014], aunque presentan ciertas debilidades para representar adecuadamente la dimensión temporal, razón por la cual existen multitud de propuestas que extienden el alcance, [Gagné & Trudel 2009a; Flores & Sepúlveda 2011; Watahiki et al. 2011; Cheikhrouhou et al. 2013b]. Para analizar los *legacy system* de la sección anterior y desarrollar el resto de contribuciones, se ha establecido una taxonomía de reglas (Figura 6.1) que contempla un amplio abanico de reglas y dependencias temporales. La debilidad de *BPMN* en relación a la dimensión temporal también es extensiva a *Meta-Modelos* de procesos contemplados en estándares del software como «*Software Engineering Meta-Model for Development Methodologies*» [Iso/Iec:24744 2007, 2014] o «*Software & Systems Process Engineering Meta-Model specification (SPEM 2.0)*» [Bendraou et al. 2007; OMG: SPEM 2008].

### 9.1.4 Meta-Modelo de procesos PIM-up con reglas temporales (ob.3.).

Utilizando la taxonomía de reglas temporales (Figura 6.1) como requisito se han estudiado los *Meta-Modelos* de procesos de estándares como *UML AD* [OMG:UML 2011], *BPMN* [OMG:BPMN 2013], y *XPDL 2.2* [WfMC:XPDL], extrayendo una tabla comparativa (Tabla 5.5) con las clases comunes que utilizan todos estos *Meta-Modelos*.

Se plantea un *Meta-Modelo de procesos*, diseñado con *UML*, que se sitúa en el nivel de abstracción *PIM*, denominándolo *Meta-Modelo PIM-up* (Figura 6.11). Este *Meta-Modelo* soporta la taxonomía de reglas (Figura 6.1), formalizándolas mediante fórmulas *OCL* [OMG:OCL 2014; Iso/Iec 19507:2012] como lenguaje de declarativo [Fahland 2009a, 2009b; Pichler et al. 2011; Reijers et al. 2013]. Mediante extensiones de este *Meta-Modelo* básico se resolverán los objetivos del trabajo de investigación.

### 9.1.5 Extensión al Meta-Modelo BPMN (ob.4.).

Inicialmente, se elige el *Meta-Modelo BPMN* para expresar procesos por las razones

expuestas [Lu & Sadiq 2007, Bonnet et al. 2014], aunque podrían elegirse otros *Meta-Modelos* [Iso/Iec:24744 2007, 2014; Bendraou et al. 2007; OMG: SPEM 2008; Ponce et al. 2013], según convenga al experto en software. La primera utilidad del *Meta-Modelo PIM-up* (Figura 6.11) es extender el *Meta-Modelo BPMN* para dotarlo con la semántica temporal de la taxonomía de reglas (Figura 6.1).

Llevar modelos a *BPMN* otorga la capacidad de representación gráfica y la interoperabilidad de procesos y modelos de procesos, basándose en su robusto *Meta-Modelo* y en las facilidades de intercambio de diagramas *XML* (*datos* y *meta-datos*).

### 9.1.6 Ingeniería inversa MDE para extraer procesos desde legacy systems (ob.5.)

Por las razones anteriormente expuestas, se han seleccionado las plataformas: i) *MS Project Server* (nivel *PSM*) y ii) *BPMN* (nivel *CIM*). Aunque, posteriormente, podría extenderse a otras plataformas *PSM* (otros *legacy systems*) y a otras plataformas *CIM*. Por esta razón, se han identificado nuevos requisitos que deben tenerse en cuenta para extraer la dimensión temporal de cada *legacy system*:

- i) Es necesario identificar el *legacy system* origen.
- ii) Una actividad podría aparecer replicada en más de un *legacy system* (p. ej.: podría estar planificada en *MS Project*, generar distintos documentos en *Alfresco* y tener un presupuesto asignado en el *ERP SAP*). O bien se contemplan como fragmentos distintos en cada *legacy system*, generando un proceso para cada *legacy system*, o bien se resuelve esta colisión, integrando todo el comportamiento en una actividad global, siendo posible entonces la representación de procesos más genéricos.

Para llevar a cabo la extracción de procesos desde *legacy systems* (ob.5.), se han resuelto los siguientes objetivos específicos:

#### 9.1.6.1 Meta-Modelo PIM-upl (ob.5.1)

Este *Meta-Modelo*, independiente de la plataforma (*PIM*) sirve para representar cualquier instancia de un proceso con las reglas temporales formalizadas en el *Meta-Modelo PIM-up* (ob.3, Figura 6.11), pero también debe referenciar el origen (*legacy system*) o procedencia del proceso, ya que es posible utilizar múltiples orígenes (distintos sistemas y extensiones con otras perspectivas, además de la temporal), pues *MS Project Server* sólo es un ejemplo de *legacy system*. Se denomina a esta extensión *Meta-Modelo pivote PIM-upl* (ob.6) (Figura 7.3).

Se han estudiado las posibles colisiones que pueden aparecer por motivo de replicación de una actividad en más de un *legacy system*, extendiendo el mismo *Meta-Modelo* para dar solución a cada tipo de colisión (Figura 7.8) contemplando la identificación de estas actividades y los *legacy systems* donde aparecen. Las colisiones se resuelven con restricciones de integridad *OCL* sobre esta extensión (Figuras 7.9, 7.10,

7.11, 7.12 y 7.13). Un experto del dominio puede manejar extracciones de actividades: i) Representando procesos privados para cada *legacy system*, o ii) Generando procesos globales que están compuestos por actividades privadas y estas actividades globales que aúnan el comportamiento de sus réplicas o fragmentos procedentes de distintos *legacy systems*.

#### 9.1.6.2 Algoritmo de transformación PSM→PIM-upl (ob.5.2)

Cada proyecto es una instancia del *Meta-Modelo* (ob.3) de tareas de *MS Project Server* (PSM). El *Meta-Modelo PIM-upl* tiene la semántica temporal necesaria (restricciones y dependencias temporales, además de soportar la resolución de colisiones por replicación de actividades en distintos *legacy systems*), formalizada con *OCL*. El algoritmo (Figura 7.7), complementado por las Tablas 7.3, 7.4, 7.5, 7.6, 7.7 y 7.8) detalla el mapeo de artefactos de un modelo de nivel PSM (basado en el *Meta-Modelo* de *MS Project Server*) en artefactos de nivel PIM (basado en el *Meta-Modelo PIM-upl*). De este modo, los modelos obtenidos gozan de la semántica temporal y de la resolución de colisiones por replicación de actividades.

#### 9.1.6.3 Algoritmo de transformación PIM-upl→CIM (ob.5.3)

El *Meta-Modelo* (ob.4) extiende *BPMN* con la semántica temporal de la taxonomía (ob.2) (Figura 6.1). (ob.6) es un *Meta-Modelo* con la misma semántica temporal. El algoritmo (Figura 7.17), complementado por las Tablas 7.9, 7.10, 7.11, 7.12, 7.13, 7.14, 7.15 y 7.16) detalla las transformaciones *PIM→CIM* en el caso de *BPMN*. Aunque el sistema destino podría ser otro (p. ej. *SPEM 2.0*, *NDTQ-Framework* [Ponce et al. 2013; Aragón 2014] o *Iso/Iec:24744*).

### 9.1.7 Verificación del enfoque en el caso AQUA-WS Project (ob.6)

El enfoque de ingeniería inversa *MDE* (*PSM→PIM→CIM*) sobre el caso de estudio *AQUA-WS Project* [Cutilla et al. 2012] ha permitido generar los siguientes resultados sobre subconjuntos de actividades del sistema origen (almacenadas en la instancia de la *BD Published de MS Project Server*):

- i) Grupo de tareas del Plan General (Proceso *BPMN* de la Figuras 8.9 y 8.10).
- ii) Subsistema que utiliza *NDT* como marco metodológico (Proceso *BPMN* de la Figura 8.11).
- iii) Patrón de la fase *ASI NDT* (Proceso *BPMN* de la Figura 8.12).

En los casos (i y ii) se obtienen procesos *BPMN* que son instancias concretas de un proceso (nivel de abstracción *MOF M<sub>0</sub>*), mientras que en el caso (iii), el proceso *BPMN* representa un modelo o clase de procesos (nivel de abstracción *MOF M<sub>1</sub>*) para la fase de Análisis de Sistemas de Información (ASI) contemplada en la metodología *NDT* [Escalona 2004; Escalona & Aragón 2008].

En la Figura 8.13 se provee un *modelo de procesos* realizado manualmente para las



fases de *requerimientos (RQ)* y *Análisis del sistema (ASI)*, realizado por un experto en negocio del software y en *NDT*. El contraste de este modelo (Figura 8.13) con el derivado mediante la aplicación de extracción automática ( $PSM \rightarrow PIM \rightarrow CIM$ , de la Figura 8.12) ofrece un grado de cumplimiento aceptable en la representación de la perspectiva de control del flujo de las actividades, añadiendo además la perspectiva temporal (restricciones y dependencias temporales). No obstante, el modelo generado carece de una semántica que existe en *BPMN*, como son las transacciones, eventos de error y cancelación, y que en este caso ha utilizado el experto.

Se ha observado una limitación respecto a planes de proyecto que no contemplan una clasificación de actividades: el resultado es una instancia de un proceso (*MOF  $M_0$* ). Por otro lado, si el experto clasifica las actividades (p. ej. la tarea  $x$  es una instancia de la clase de actividad *ASI NDT*), entonces es posible utilizar patrones de planificación de proyectos como modelos de proceso (*MOF  $M_1$* ). Para resolver este salto en el nivel de abstracción (*de  $M_0$ :instancia de proceso a  $M_1$ :modelo de proceso*) el experto tendría que clasificar las actividades del proyecto como instancias de un tipo o clase.

## 9.2 Trabajo futuro

En esta tesis doctoral se ha planteado un enfoque basado en el paradigma *MDE* que permite extraer procesos desde un *legacy system*; en particular desde la instancia de la base de datos *Published* definida sobre *MS SQL\*Server* de *MS Project* [Hansen & Hansen 2013], como sistema para la *planificación y gestión de proyectos*. Para el caso de aplicación *AQUA-WS Project* [Cutilla et al. 2012] se ha elegido la plataforma destino (nivel CIM) *BPMN*, por lo que se ha llevado a su *Meta-Modelo* (Figura 7.16) la semántica temporal representada en el *Meta-Modelo PIM-up* (Figura 6.11) y *PIM-upl* (Figura 7.3). Ahora bien, pueden plantearse trabajos futuros en distintos ámbitos:

- i) **Otros legacy systems.** Se ha trabajado con *MS Project* y *MS Project Server* [Stover 2007], pero podría extenderse a otros *legacy systems* con buena definición de la perspectiva temporal, en particular, sistemas de planificación y gestión de proyectos como *RedMine*<sup>TM</sup> [Lang 2010] (que se ha analizado en el Anexo II.d) sobre *MySQL*, o bien otros.

También podría pensarse en utilizar otra categoría de sistemas, p. ej.: *ECMs* como *Alfresco*<sup>TM</sup> [Shariff 2013] y *MS Sharepoint*<sup>TM</sup> [Smith & Bates 2007], u otros gestores de contenido, así como *ERPs*, *CRMs*, *SCMs* [Hendricks et al. 2007], o bien *desarrollos a medida*.

En cualquiera de los casos, el trabajo a realizar consistiría en particularizar el *Meta-Modelo* de tareas origen y la plataforma *PSM* de su *legacy database*. Por otro lado, habría que ajustar la transformación *PSM*→*PIM-upl*. Las transformaciones *PIM-upl*→*CIM* permanecerían inalteradas, a no ser que se modificara el destino *CIM*.

- ii) **Otros lenguajes de representación de procesos.** Podrían utilizarse otros sistemas de representación de procesos, p. ej. *SPEM 2.0*, *Iso/Iec:24744*, o nuestra plataforma *NDTQ-Framework* [Ponce et al. 2013; Aragón 2014]. *BPMN* posee un robusto *Meta-Modelo* de procesos pero, además es uno de los estándares más usados para intercambiar especificaciones de procesos (datos y metadatos), junto con el estándar *XPDL* que cada vez más cerca de soportar toda la expresividad y portabilidad de *BPMN*.

- iii) **Automatización del proceso de ingeniería inversa (PSM→PIM→CIM)**

El proceso diseñado se puede automatizar en distintas plataformas, p. ej.:

- a) En un entorno abierto como *Eclipse EMF* [Merks 2010; Eclipse:EMF 2015] o *Modisco* [Eclipse:Modisco 2015; Bruneliere et al. 2010b].
- b) Sobre el entorno *Enterprise Architect (EA)* [Sparx:EA 2015], donde tenemos experiencia de desarrollo de la plataforma *NDTQ-Framework*.
- c) Desarrollo a medida sobre otros entornos.

- iv) **Aplicación a otros casos de estudio.** Puede aplicarse el enfoque a otros proyectos donde se utilice *NDT* [Escalona 2004; Escalona & Aragón 2008] o bien otra metodología para detectar limitaciones de la propuesta y perfeccionarla. Por otro lado, podría estudiarse el encaje de esta propuesta en otros sectores productivos y de servicios.
- v) **Extensión con otras perspectivas de procesos.** La semántica utilizada en el proceso de ingeniería inversa *MDE* considera sólo la *perspectiva temporal* intrínseca a las *legacy databases*, representando procesos con características de *control del flujo* y esta *perspectiva temporal*. Trabajos como [Awad et al. 2009; Stroppi et al. 2012] manejan extensiones al *Meta-Modelo BPMN* que consideran la *perspectiva organizacional* y *de casos* mediante patrones y fórmulas *OCL*. Sería relativamente sencillo integrar estos *Meta-Modelos* con el *Meta-Modelo PIM-up* (Figura 6.11) propuesto en este trabajo. De este modo, los procesos obtenidos estarían enriquecidos con estas nuevas perspectivas. Según el análisis realizado en el Capítulo 5, también podemos encontrar estas perspectivas subyacentes en los *legacy systems*.
- vi) **Generación de log de eventos.** El *log de eventos* es una estructura típica de los sistemas orientados a procesos (*PAIS*), pero no es común encontrarlo en *legacy systems*. «*Process mining*» [van der Aalst et al. 2007, 2012; van der Aalst 2011, 2013, 2014, 2015; Reichert & Weber 2012] es un campo que está ahora en auge para descubrir y mejorar procesos existentes en las organizaciones, siendo una orientación muy valiosa para implantar un enfoque *BPM* en una organización, bien del sector del software o de cualquier otro sector. Existen formatos estándar de *log de eventos* como «*Extensible Event Stream (XES)*» [Verbeek et al. 2011]<sup>43</sup>, utilizado por herramientas automatizadas para «*process mining*» como *ProM* [Verbeek et al. 2011; Kalenkova et al. 2014; van der Aalst 2015].

Una vez resuelta la transformación *PSM*→*PIM-upl* desde un *legacy system*, podría considerarse una transformación *PIM-upl*→*XES* para generar un *log de eventos*, y alimentar a *ProM* o a otro sistema automatizado de «*process mining*» que acepte el formato *XES*, o bien generar el *log* en los formatos que soporte cada sistema. Esto permitiría derivar procesos de un modo alternativo y contrastarlos con los obtenidos mediante este trabajo de tesis doctoral, aunque estos métodos sólo derivan la *perspectiva de control del flujo*.

### 9.3 Conclusiones

El enfoque *BPM* ofrece ventajas estratégicas para cualquier organización en una sociedad globalizada que obliga a competir a los agentes de todos los sectores de la sociedad. Las «*organizaciones del software (TI)*» no son una excepción que,

<sup>43</sup> <http://www.xes-standard.org/>

progresivamente, están incorporando también esta tecnología para gestionar sus singulares procesos.

Las herramientas automatizadas que soportan el enfoque *BPM* [van der Aalst 2004], los *BPMS*, coexisten con otros sistemas automatizados no orientados a procesos (*legacy systems*), entre los que se encuentran las aplicaciones para *planificación y control de proyectos, gestión de documentación o de contenidos, sistemas de gestión económico-financiera, de relaciones con terceros (clientes, proveedores, colaboradores, bancos, etc.), como los ERPs, CRMs y SCMs*, además de los *desarrollos a medida*. El *ciclo de mejora continua BPM* [van der Aalst 2004] contempla tanto el descubrimiento de procesos como el contraste y mejora de los mismos.

El estudio realizado en el Capítulo 5 justifica la elección inicial de sistemas para la planificación y control de proyectos como *legacy systems* candidatos para extraer procesos. En este trabajo de investigación se ha propuesto un enfoque que *ayuda a extraer procesos desde la legacy database SQL\*Server de MS Project Server*, como ejemplo de *legacy system*, representándolos mediante lenguajes cercanos al lenguaje del experto en software, como *BPMN*. El enfoque se ha basado en la aplicación de ingeniería inversa *MDE* y el análisis de la *dimensión temporal de los procesos*, manejando *Meta-Modelos* que contemplan las estructuras y formulación de esta dimensión temporal con *UML* y *OCL*. Se han especificado algoritmos detallados que transforman artefactos de bajo nivel de abstracción (*tareas y reglas temporales sobre legacy databases relacionales*, a nivel *PSM*) en artefactos *BPMN* (nivel *CIM*).

El enfoque se ha aplicado satisfactoriamente al caso *AQUA-WS*, proyecto de modernización de software sobre plataforma Web, en la *Empresa Municipal de Abastecimiento de Aguas de Sevilla (EMASESA)*, donde el grupo de investigación *IWT2* ejerció las funciones de soporte metodológico con «*Navigational Development Technique (NDT)*» y, posteriormente, el control de calidad del proyecto global, constituido por múltiples subproyectos de desarrollo. Se ha verificado la validez del enfoque para obtener procesos (instancias) y modelos de procesos, así como alguna limitación cuando las actividades no están clasificadas (no pertenecen a una clase bien identificada).

El enfoque está limitado a la extracción de la dimensión temporal de procesos desde *MS Project Server* como *legacy system*, pero es susceptible de ser ampliado con otras dimensiones o perspectivas de procesos (*organizacional, funcional, información, casos*), así como de ser aplicado a otros tipos de *legacy systems*.

Los procesos generados son productos útiles para el experto en software, pero el enfoque también puede servir para generar un resultado intermedio como el *log de eventos* de *legacy systems* que carecen de ellos, como es el caso de *MS Project Server*. Este *log* podría utilizarse para aplicar «*process mining*» y generar modelos alternativos de los procesos, aunque estas técnicas suelen originar sólo la *dimensión del flujo de control* y no la *perspectiva temporal* que se ha extraído con el enfoque de la tesis.

## **REFERENCIAS**



- [*Abrahamsson 2002*] Abrahamsson, P. (2002): “*Agile Software Development Methods: Review and Analysis*”. VTT publications.
- [*Alalfi et al. 2008*] Alalfi, M.H., Cordy, J.R., Dean, T.R., (2008): “*SQL2XMI: Reverse engineering of UML-ER diagrams from relational database schemas*”. In: Working Conference on Reverse Engineering. 187–191.
- [*Alanen et al. 2003*] Alanen, M., Porres, I (2003): “*A Relation Between Context-Free Grammars and Meta-Object Facility Metamodels*”. Technical report, Turku Centre for Computer Science, 2003
- [*Alexander 1983*] Charles P. Alexander C.P. (1983): “*The New Economy*”. *Time* magazine, May 30, 1983.
- [*Alhir 2006*] Alhir, S. S. (2006): “*Guide to Applying the UML*”. Springer Science & Business Media.
- [*Allen 1983*] Allen, J.F. (1983): “*Maintaining Knowledge about Temporal Intervals*”. *Communications of the ACM* 26, 1983, 832-843
- [*Alvarez et al. 2014*] Alvarez, F., Antonelli, F., Facca, F. M., Fernandez, E., Giaffreda, R., Kim, E., ... & Potts, M. (2014). *DI. 1: FI-LINKS technology and business models map*.
- [*Antlr 2015*] Antlr Org (2015): “*ANTLR (ANother Tool for Language Recognition)*”, Recuperado de <http://www.antlr.org/>. Último acceso 31/10/2015
- [*Aragón 2014*] Aragón, G. (2014): “*Una Propuesta de Uso y Explotación del Paradigma Guiado Por Modelos para la Mejora de la Ingeniería de Requisitos Web*”. Tesis Doctoral. Departamento de Ciencias de la Computación, Universidad de Alcalá de Henares.  
[http://www.uah.es/escuela\\_doctorado/tesis/tesis.asp?CdPlan=D338&expe=2&proy=1](http://www.uah.es/escuela_doctorado/tesis/tesis.asp?CdPlan=D338&expe=2&proy=1)
- [*Arevalo et al. 2013*] Arévalo, C., Gómez-López, M. T., Reina Quintero, A.M., Ramos, I. (2013): “*An Architecture to Infer Business Rules from Event Condition Action Rules Implemented in the Persistence Layer*”. In R. Perez-Castillo, & M. Piattini (Eds.) *Uncovering Essential Software Artifacts through Business Process Archeology* (pp. 201-221). Hershey, PA: Business Science Reference. doi:10.4018/978-1-4666-4667-4.ch008.
- [*Arevalo 2014*] Arévalo, C. (2014). “*Extracting Business Models for Software Process Support*”. Ed. Riquelme, J., Ruiz, M., & García, M. T. *Actas JISBD, Doctoral Consortium, SISTEDES, Cádiz, 2014*, pp.40-47.
- [*Arevalo et al. 2015a*] Arevalo, C., Ramos, I., Escalona, M.J. (2015): “*Discovering Business Models for Software Process Management: An Approach for Integrating Time and Resource Perspectives from Legacy Information Systems*”. ICEIS 2015, Bcn.

- [Arevalo et al. 2015b] Arevalo, C., Ramos, I., Escalona, M.J. (2015): “*A metamodel to integrate business processes time perspective in BPMN 2.0*”. Elsevier, *Information and Software Technology*, (en revisión).
- [Arlow & Neustadt 2005] Arlow, J., & Neustadt, I. (2005): “*UML 2 and the unified process: practical object-oriented analysis and design*”. Pearson Education.
- [Armario et al. 2012] Armario, J., Gutiérrez, J., Alba, M., García-García, J. A., Vitorio, J., Escalona, M. J. (2012): “*Project Estimation with NDT*”. In S. Hammoudi, M. van Sinderen & J. Cordeiro (eds.), ICSOFT (p. /pp. 120-126). ISBN: 978-989-8565-19-8.
- [Atlanmod 2015] Atlanmod (2015): “*Atlanmod Zoos*”, Recuperado de <http://www.emn.fr/z-info/atlanmod/index.php/Zoos>, Último acceso 31/10/2015.
- [Attaran 2004] Attaran, M. (2004): “*Exploring the relationship between information technology and business process reengineering*”, *Information & Management*, Volume 41, Issue 5, May 2004, Pages 585-596, ISSN 0378-7206, [http://dx.doi.org/10.1016/S0378-7206\(03\)00098-3](http://dx.doi.org/10.1016/S0378-7206(03)00098-3).
- [Awad et al. 2009] Awad, A., Grosskopf, A., Meyer, A., & Weske, M. (2009): “*Enabling resource assignment constraints in BPMN*”. Hasso Plattner Institute, Potsdam.
- [Baisley 2005] Baisley, D. (2005): “*OMG and Business Rules*”. Presentation available at <http://www.omg.org/docs/omg/05-04-09.pdf>
- [Bandinelli et al. 1993] Bandinelli, S., Fuggetta, A., Ghezzi, C. (1993): “*Process Model Evolution in the SPADE Environment*”. *IEEE Trans. Software Eng. (TSE)* 19(12):1128-1144
- [Barba et al. 2012] Barba, I., Lanz, A., Weber, B., Reichert, M., & Del Valle, C. (2012): “*Optimized time management for declarative workflows*”. In *Enterprise, Business-Process and Information Systems Modeling* (pp. 195-210). Springer Berlin Heidelberg.
- [Barba et al. 2013] Barba, I., Weber, B., Del Valle, C., & Jiménez-Ramírez, A. (2013). User recommendations for the optimized execution of business processes. *Data & Knowledge Engineering*, 86, 61-84.
- [Bauer 1972] Bauer, F.L. (1972): “*Software Engineering. Advanced Course*”. *Software Engineering*: 522-545
- [Bendraou et al. 2007] Bendraou, R., Combemale, B., Cregut, X., Gervais, M. “*Definition of an executable SPEM 2.0*.” *Software Engineering Conference, APSEC 2007 14th Asia-Pacific, IEEE*, pp. 390-397, 2007a.
- [Ben-Gan 2012] Ben-Gan, I. (2012): “*Microsoft SQL Server 2012 T-SQL Fundamentals*”. Pearson Education.



- [Bermúdez et al. 2012] Bermúdez Ruiz, F.J., Molina, J.G. (2012): “*Un framework basado en modelos para la modernización de datos*”. In: Actas de las XII JISBD. pp. 477–490 (2012)
- [Bermúdez et al. 2014] Bermúdez Ruiz, F.J., Molina, J.G., Díaz, O. (2014): “*DB-Main/Models: Un caso de estudio sobre la interoperabilidad de herramientas basada en MDE*”, SISTEDES 2014, JISBD, pp. 227-240
- [Bézivin & Kurtev 2005] Bézivin, J., Kurtev, I. (2005): “*Model-based technology integration with the technical space concept*”. In: Procs. Of the Metainformatics Symposium. Springer-Verlag, 2005
- [Bisbal et al. 1997] Bisbal, J., Lawless, D., Wu, B., Grimson, J., Wade, V., Richardson, R., & Sullivan, D. O. (1997): “*An overview of legacy information system migration*”. In Software Engineering Conference, 1997. Asia Pacific... and International Computer Science Conference 1997. APSEC'97 and ICSC'97. Proceedings (pp. 529-530). IEEE.
- [Bisbal et al. 1999] Bisbal, J., Lawless, D, Wu, B., Grimson, J. (1999): “*Legacy Information Systems: Issues and Directions*”. 0740-7459/99 © 1999 IEEE
- [Boehm 2002] Boehm, B. (2002): “*Get ready for agile methods, with care*”. Computer, 35(1), 64-69.
- [Bonnet et al. 2014] Bonnet, F., Decker, G., Dugan, L., Kurz, M, Misiak, Z., Ringuette, S. (2014): “*Making BPMN a True lingua franca*”. BPM Trends, 2014.
- [Boronat et al. 2005] Boronat, A., Carsí, J.A., Ramos, I. (2005): “*Automatic Reengineering in MDA Using Rewriting Logic as Transformation Engine*”. CSMR 2005: 228-231
- [Brambilla et al. 2012] Brambilla, M., Cabot, J., Wimmer, M. (2012): “*Model-Driven Software Engineering in Practice*”. Synthesis Lectures on Soft. Eng., Morgan & Claypool Publishers, 2012
- [Bruneliere et al. 2010a] Bruneliere, H., Cabot, J., Clasen, C., Jouault, F., Bézivin, J. (2010): “*Towards Model Driven Tool Interoperability: Bridging Eclipse and Microsoft Modeling Tools*”. ECMFA 2010:32-47
- [Bruneliere et al. 2010b] Bruneliere, H., Cabot, J., Jouault, F., & Madiot, F. (2010): “*MoDisco: a generic and extensible framework for model driven reverse engineering*”. In Proceedings of the IEEE/ACM international conference on Automated software engineering (pp. 173-174). ACM.
- [Burtica et al. 2012] Burtica, R., Mocanu, E. M., Andreica, M. I., & Țăpuș, N. (2012, March): “*Practical application and evaluation of no-SQL databases in Cloud Computing*”. In Systems Conference (SysCon), 2012 IEEE International (pp. 1-6). IEEE.
- [Cabanillas et al. 2011a] Cabanillas, C., Resinas, M., & Ruiz-Cortés, A. (2011). “*Defining and analysing resource assignments in business processes with RAL*”. In Service-Oriented Computing (pp. 477-486). Springer Berlin Heidelberg.

- [Cabanillas et al. 2011b] Cabanillas, C., Resinas, M., & Ruiz-Cortés, A. (2011, April). “Towards the definition and analysis of resource assignments in BPMN 2.0.” In *9th International Conference on Business Process Management BPM11*.
- [Cabanillas et al. 2014] Cabanillas, C., Norta, A., Resinas, M., Mendling, J., & Ruiz-Cortés, A. (2014): “Towards process-aware cross-organizational human resource management”. In *Enterprise, Business-Process and Information Systems Modeling* (pp. 79-93). Springer Berlin Heidelberg.
- [Cabanillas et al. 2015a] Cabanillas, C., Resinas, M., del-Río-Ortega, A., & Ruiz-Cortés, A. (2015). “Specification and automated design-time analysis of the business process human resource perspective”. *Information Systems*, 52, 55-82.
- [Cabanillas et al. 2015b] Cabanillas, C., Knuplesch, D., Resinas, M., Reichert, M., Mendling, J., & Ruiz-Cortés, A. (2015): “RALph: A Graphical Notation for Resource Assignments in Business Processes”. In *Advanced Information Systems Engineering* (pp. 53-68). Springer International Publishing.
- [Cabot 2010] Cabot, J. (2010): “Has the success of BPMN 2.0 killed BPDM (Business Process Definition Metamodel)?”. In *Modeling Languages*, <http://modeling-languages.com/el-xito-de-bpmn-20-ha-acabado-con-bpdm-business-process-definition-metamodel/>
- [Cabot et al. 2012] Cabot, J., Gogolla, M. (2012): “Object Constraint Language (OCL): A Definitive Guide”. SFM pages:58-90
- [Castellanos et al. 2009] Castellanos, M., Medeiros, K.A., Mendling, J., Weber, B., Weitjers, A.J.M.M.(2009): “Business Process Intelligence”, in *Handbook of Research on Business Process Modeling*, 2009, Idea Group Inc. p. 456-480.
- [Castells 1999] Castells, M. (1999). “La sociedad red. La era de la información: economía, sociedad y cultural”. Madrid, Alianza Editorial, enero.
- [Castells 2002] Castells, M. (2002): “The Internet galaxy: Reflections on the Internet, business, and society”. Oxford University Press.
- [Castells 2011] Castells, M. (2011). *The rise of the network society: The information age: Economy, society, and culture* (Vol. 1). John Wiley & Sons.
- [Castells 2013] Castells, M. (2013): “Networks of outrage and hope: Social movements in the Internet age”. John Wiley & Sons.
- [Cheikhrouhou et al. 2013a] Cheikhrouhou, S., Kallel, S., Guermouche, N., Jmaiel, M. (2013): “A Survey on Time-aware Business Process Modeling”. ICEIS 2013:236-242
- [Cheikhrouhou et al. 2013b] Cheikhrouhou, S., Kallel, S., Guermouche, N., Jmaiel, M. (2013): “Toward a Time-centric modeling of Business Processes in BPMN 2.0”. iiWAS 2013:154.
- [Cheikhrouhou et al. 2014a] Cheikhrouhou, S., Kallel, S., Guermouche, N., Jmaiel, M. (2014): “Enhancing Formal Specification and Verification of Temporal Constraints in Business Processes”. IEEE SCC, 2014:701-708

- [*Cheikhrouhou et al. 2014b*] Cheikhrouhou, S., Kallel, S., Guermouche, N., Jmaiel, M. (2014): “*Toward a Verification of Time-Centric Business Process Models*”. WETICE 2014:326-331
- [*Chen 1976*] Chen, P. (1976): “*The Entity-Relationship Approach: Towards a unified behavior of data*”. ACM Transactions on Database Systems. 1:1. pp. 9-36. enero, 1976.
- [*Chen 2015*] Chen, J. X. (2015): “*The Advancement of Computing*”. *Computing in Science & Engineering*, 17(5), 4-4.
- [*Chikofsky & Cross 1990*] Chikofsky, E., Cross, J.H. (1990): “*Reverse engineering and design recovery: A taxonomy*”, IEEE Software, 7(1):13-17
- [*Chinosi & Trombetta 2012*] Chinosi, M., & Trombetta, A. (2012): “*BPMN: An introduction to the standard*”. *Computer Standards & Interfaces*, 34(1), 124-134.
- [*Chomsky 1956*] Chomsky, Noam (1956): “*Three models for the description of language*”. IRE Transactions on Information Theory 2 (2): 113–24. doi:10.1109/TIT.1956.1056813
- [*Chrissis et al. 2011*] Chrissis, M. B., Konrad, M., & Shrum, S. (2011): “*CMMI for development: guidelines for process integration and product improvement*”. Pearson Education.
- [*Clifford & van Bon 2008*] Clifford, D., van Bon, J. (2008): “*Implementing Iso/Iec 20000 Certification: The Roadmap. ITSM Library*”. van Haren Publishing. ISBN 90-8753-082-X.
- [*Codd 1970*] Codd, E. F. (1970): “*A relational model of data for large shared data banks*”. *Communications of the ACM*, 13(6), 377-387.
- [*Codd 1992*] Codd, E.F. (1992): “*The Relational Model for Database Management*”, Addison-Wesley, 1992.
- [*Colledge 2008*] Colledge, R. (2008): “*SQL\*Server, Administration in action*”, Ed. Manning, 2008.
- [*Conradi et al. 1992*] Conradi, R., Jaccheri, M.L., Mazzi, C., Nguyen, M.N., Aarsten, A. (1992): “*Design, Use and Implementation of SPELL, a language for Software Process Modelling and Evolution*”. EWSPT 1992:167-177
- [*Cosentino 2013*] Cosentino, V. (2013): “*A Model-Based Approach for Extracting Business Rules out of Legacy Information Systems*”, PhD, Laboratoire d’informatique de Nantes-Atlantique (LINA), (2013).
- [*Cosentino & Martínez 2013*] Cosentino, V., Martínez, S. (2013): “*Extracting UML/OCL Integrity Constraints and Derived Types from Relational Databases*”. OCL@MoDELS 2013:43-52.
- [*Criado et al. 2010*] Criado, J. I., Gascó, M., & Jiménez, C. E. (2010): “*Bases para una estrategia iberoamericana de interoperabilidad*”. Caracas, CLAD (Documentos Clave: Estado, Administración Pública y Sociedad, (18).

- [Cruz et al. 2014] Cruz M, Bernárdez B, del-Río-Ortega A, Durán A. (2014): “Una revisión de la notación PPINOT para indicadores de rendimiento mediante su aplicación a un caso real”. In: Actas de las X Jornadas de Ciencia e Ingeniería de Servicios (JCIS). Cádiz, España: Sistedes;. p. 167-76.
- [Cuadrado et al. 2012] Cuadrado, J. S., García, O. Á., Canovas, J., & Herrera, A. S. B. (2012): “Parametrización de las transformaciones horizontales en el modelo de herradura”. In *Jornadas de Ingeniería del Software y Bases de Datos*.
- [Cutilla et al. 2012] Cutilla, C.R., García-García, J.A., Gutiérrez, J.J., Domínguez-Mayo, P., Cuaresma, M.J., Rodríguez-Catalán, L., Domínguez Mayo, F.J., (2012): “Model-Driven Engineering applied in functional testing: the practical experience of the AQUA-WS Project”, Proceedings of the 7th International Conference on Software Paradigm Trends, ICSOFT 2012, 2012-07-27, ISBN 978-9898565198, (2012).
- [De Haas & van Horen 2012] De Haas, R., & van Horen, N. (2012): “International shock transmission after the Lehman Brothers collapse: Evidence from syndicated lending”. In *American Economic Review Papers & Proceedings* (Vol. 102, No. 3, pp. 231-237).
- [Dean & Bowen 1994] Dean, J. W., & Bowen, D. E. (1994): “Management theory and total quality: improving research and practice through theory development”. *Academy of management review*, 19(3), 392-418.
- [Del Rio Ortega et al. 2010] Del-Río-Ortega, A., Resinas, M., & Ruiz-Cortés, A. (2010): “Defining process performance indicators: An ontological approach”. In *On the Move to Meaningful Internet Systems: OTM 2010* (pp. 555-572). Springer Berlin Heidelberg.
- [Del Rio Ortega et al. 2012] del-Río-Ortega, A., de Reyna, M. R. A., Toro, A. D., & Ruiz-Cortés, A. (2012): “Defining process performance indicators by using templates and patterns”. In *Business Process Management* (pp. 223-228). Springer Berlin Heidelberg.
- [Del Rio Ortega et al. 2013a] del-Río-Ortega, A., Resinas, M., Cabanillas, C., & Cortés, A. R. (2013): “Defining and Analysing Resource-Aware Process Performance Indicators”. In *CAiSE Forum* (pp. 57-64).
- [Del Rio Ortega et al. 2013b] del-Río-Ortega, A., Resinas, M., Cabanillas, C., & Ruiz-Cortés, A. (2013): “On the definition and design-time analysis of process performance indicators”. *Information Systems*, 38(4), 470-490.
- [Deltombe et al. 2012] Deltombe, G., Le Goaer, O., Barbie, F. (2012): “Bridging KDM and ASTM for Model-Driven Software Modernization”, SEKE’12.
- [Demuth et al. 1999] Demuth, B., Hußmann, H. (1999): “Using UML/OCL constraints for relational database design”. UML. doi: 10.1007/3-540-46852-8\_42.
- [Demuth et al. 2001] Demuth, B., Hußmann, H., Loecher, S. (2001): “OCL as a specification language for business rules in database applications”. UML. doi:10.1007/3-540-45441-1\_9.

- [Díaz et al. 2013] Díaz, O., Puente, G., Cánovas Izquierdo, JL, García Molina, J. (2013): “*Harvesting models from web 2.0 databases*”. *Software and System Modeling (SOSYM)* 12(1):15-34.
- [Dijkstra 1972] Dijkstra, E. W. (1972): “*The humble programmer*”. *Communications of the ACM*, 15(10), 859-866.
- [Doelitzscher et al. 2011] Doelitzscher, F., Sulistio, A., Reich, C., Kuijs, H., & Wolf, D. (2011): “*Private cloud for collaboration and e-Learning services: from IaaS to SaaS*”. *Computing*, 91(1), 23-42.
- [Dominguez et al. 2015] Dominguez, M., Escalona M.J., Ramos, I., Arevalo, C. “*Systematic Literature Review for Web Application Estimation with Use Case Points. Looking for a Model-Driven Perspective*”. *Software Quality Journal*, 2015 (En revisión).
- [Dooley & Johnson 2001] Dooley, K., & Johnson, D. (2001): “*Changing the new product development process: reengineering or continuous quality improvement?*”. *Measuring Business Excellence*, 5(4), 32-38.
- [Du et al. 2011] Du, Y., Xiong, P., Fan, Y., & Li, X. (2011): “*Dynamic checking and solution to temporal violations in concurrent workflow processes*”. *Systems, Man and Cybernetics, Part A: Systems and Humans*, IEEE Transactions on, 2011.41 (6), 1166-1181.
- [Dubois 2005] DuBois, P. (2005). *MySQL (Developer's Library)*. Sams.
- [Dumas et al. 2005] Dumas, M., van der Aalst, W.M.P., Hofstede, A.H.M. (2005): “*Process-Aware Information Systems. Bridging People and Software Through Process Technology*”. John Wiley & Sons, Inc., 2005, ISBN-13 978-0-471-66306-5.
- [Dumas et al. 2013] Dumas, M., La Rosa, M., Mendling, J., Reijers, H. A. (2013): “*Fundamentals of business process management*”. Heidelberg: Springer.
- [Dunning & Lundan 2008] Dunning, J. H., & Lundan, S. M. (2008): “*Multinational enterprises and the global economy*”. Edward Elgar Publishing.
- [Eckerson 1995] Eckerson, Wayne W. (1995): “*Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications.*” *Open Information Systems* 10, 1 (January 1995): 3(20)
- [Eclipse:ATL 2015] Eclipse:ATL (2015): “*MMT/ATL Transformation Language (ATL)*”. Recuperado de <http://www.eclipse.org/atl/>. Último acceso 31/10/2015.
- [Eclipse:EMF 2015] Eclipse:EMF (2015): “*Eclipse Modeling Framework (EMF)*”. Recuperado de <http://www.eclipse.org/modeling/emf/> . Último acceso 31/10/2015.
- [Eclipse:KM3 2006] Eclipse:KM3 (2006): “*Kernel Meta Model (KM3)*”. Recuperado de <http://wiki.eclipse.org/KM3> . Último acceso 31/10/2015.

- [*Eclipse:Modisco 2015*] Eclipse Org.(2015): “*Modisco: a Model Driven Reverse Engineering Framework*”, Recuperado de <http://hal.inria.fr/hal-00972632> . Último acceso 31/10/2015.
- [*Eclipse:Xtext 2015*] Eclipse Org. “Xtext”. <http://www.eclipse.org/Xtext/>. Último acceso 31/10/2015.
- [*Eder & Tahamtan 2008a*] Eder, J., & Tahamtan, A. (2008): “*Temporal conformance of federated choreographies*”. In Database and Expert Systems Applications (pp. 668-675). Springer Berlin Heidelberg.
- [*Eder & Tahamtan 2008b*] Eder, J., & Tahamtan, A. (2008): “*Temporal consistency of view based interorganizational workflows*”. (pp. 96-107). Springer Berlin Heidelberg.
- [*Eisenberg, 1996*] Eisenberg, A. (1996): “*New standard for stored procedures in SQL*”. ACM SIGMOD Record 25 (4): 81–88. doi:10.1145/245882.245907, (1996).
- [*Eloranta et al. 2006*] Eloranta, L., Kallio, E., & Terho, I. (2006): “*A notation evaluation of BPMN and UML activity diagrams*”. Special course in information systems.
- [*Elrad et al. 2001*] Elrad, T., Filman, R. E., & Bader, A. (2001): “*Aspect-oriented programming: Introduction*”. Communications of the ACM, 44(10), 29-32.
- [*Elvesæter et al. 2006*] Elvesæter, B., Hahn, A., Berre, A. J., Neple, T. (2006): “*Towards an interoperability framework for model-driven development of software systems*”. In Interoperability of enterprise software and applications (pp. 409-420). Springer London.
- [*Elvesæter et al. 2010*] Elvesæter, B., Berre, AJ. (2010): “*OMG Specifications for Interoperability*”. In Interoperability for Enterprise Software and Applications: Proceedings of the Workshops and the Doctorial Symposium of the I-ESA International Conference 2010 (p. 31). John Wiley & Sons.
- [*Escalona & Aragón 2008*] Escalona, M.J., Aragón, G., 2008. *NDT. A Model-Driven Approach for Web Requirements*. IEEE TSE, vol. 34, no. 3, pp. 377-390
- [*Escalona 2004*] Escalona, M.J. (2004): “*Modelos y técnicas para la especificación y el análisis de la navegación en sistemas software*”. European thesis. Department of Computer Language and Systems, University of Seville
- [*Escalona et al. 2013*] Escalona, M.J., García-García, J. A., Más, F., Oliva, M., Del Valle, C. (2013): “*Applying model-driven paradigm: CALIPSOneo experience*”. Proceedings of the Industrial Track of the Conference on Advanced Information Systems Engineering 2013 (CAiSE'13), vol. 1017, pp. 25-32..

- [Estevez et al. 2013] Estévez, A., Avila-García, O., Sánchez-Barbudo Herrera, A., de Vega Rodrigo, M. (2013): “*Modernización Dirigida por la Arquitectura en la Transformación de un Core Bancario (Architecture-Driven Modernization in a Core Banking)*”. In García, J., García, F.O., Pelechano, V., Vallecillo, A., Vara, J.M., Vicente-Chicote, C. (2013): “*Desarrollo de software dirigido por modelos: Conceptos, Métodos y Herramientas*”. Ra-Ma Editorial, ISBN 978-84-9964-215-4. Chp. 24, pp 515-527.
- [Eveleens & Verhoef 2009] Eveleens, J. L., & Verhoef, C. (2009): *The rise and fall of the chaos report figures*. IEEE software, (1), 30-36.
- [Fabro et al. 2006] Fabro, M.D.D., Bézivin, J., Valduriez, P. (2006): “*Model-driven tool interoperability: An application in bug tracking*”. In: Meersman, R., Tari, Z. (eds.) OTM Conferences (1). LNCS, vol. 4275, pp. 863–881. Springer
- [Fahland et al. 2009a] Fahland, D., Mendling, J., Reijers, H.A., Weber, B., Weidlich, M., Zugal, S. (2009): “*Declarative versus Imperative Process Modeling Languages: The Issue of Maintainability*”. Business Process Management Workshops 2009:477-488
- [Fahland et al. 2009b] Fahland, D., Lübke, D., Mendling, J., Reijers, H.A., Weber, B., Weidlich, M., Zugal, S. (2009): “*Declarative versus Imperative Process Modeling Languages: The Issue of Understandability*”. BMMDS/EMMSAD 2009:353-366
- [Färber et al. 2012] Färber, F., Cha, S. K., Primsch, J., Bornhövd, C., Sigg, S., & Lehner, W. (2012): “*SAP HANA database: data management for modern business applications*”. ACM Sigmod Record, 40(4), 45-51.
- [Favre 2004] Favre, J. M. (2004): “*Foundations of model (driven)(reverse) engineering: Models*”. In Proceedings of the International Seminar on Language Engineering for Model-Driven Software Development, Dagstuhl Seminar 04101.
- [Favre 2010] Favre, L. (2010): “*Model Driven Architecture for Reverse Engineering Technologies: Strategic Directions and System Evolution*”. Premier Reference Source. IGI Global,
- [Feuerstein & Pribyl 2005] Feuerstein, S., Pribyl, B (2005): “*Oracle PL/SQL Programming (4th ed.)*”. O'Reilly & Associates. ISBN 0-596-00977-1.
- [Filman et al. 2004] Filman, R., Elrad, T., & Clarke, S. (2004): “*Aspect-oriented software development*”. Addison-Wesley Professional.
- [Fitz-Gibbon 1990] Fitz-Gibbon, C. T. (Ed.). (1990): “*Performance indicators (Vol. 2)*”. Multilingual Matters.
- [Flores & Sepúlveda 2011] Flores, C., & Sepúlveda, M. (2011): “*Temporal Specification of Business Processes through Project Planning Tools*”. In Business Process Management Workshops (pp. 85-96). Springer Berlin Heidelberg.
- [Fondemenet & Silaghi 2004] Fondemenet F. Silaghi R. (2004): “*Defining Model Driven Engineering Process*”. 3th Workshop in Software Model Engineering (WISME2004). October 11-15, Lisbon, Portugal.

- [Foster et al. 2008] Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008): “Cloud computing and grid computing 360-degree compared”. In *Grid Computing Environments Workshop, 2008. GCE'08* (pp. 1-10). Ieee.
- [Fuggetta 2000] Fuggetta, A. (2000): “Software process: a roadmap. The future of software engineering”, ACM, pp. 25–34.
- [Gagné & Trudel 2008] Gagné, D. & Trudel, A. (2008): “The Temporal Perspective: Expressing Temporal Constraints and Dependencies in Process Models”. In *Process Models in BPM and Workflow Handbook*.
- [Gagné & Trudel 2009a] Gagné, D. & Trudel, A. (2009). “Time-bpmn”. In *Commerce and Enterprise Computing, 2009. CEC'09. IEEE Conference on* (pp. 361-367). IEEE.
- [Gagné & Trudel 2009b] Gagné, D. & Trudel, A. (2009): “A Formal Temporal Semantics for Microsoft Project based on Allen's Interval Algebra”. In *BPSC* (pp. 32-45).
- [Gamma et al. 1995] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995): “Design Patterns”. 1995. Reading, Massachusetts: Addison-Wesley. ISBN 0-201-63361-2
- [Gantt 1919] Gantt, H. L. (1919): “Organizing for work”. Harcourt, Brace and Howe.
- [García et al. 2013] García, J., García, F.O., Pelechano, V., Vallecillo, A., Vara, J.M., Vicente-Chicote, C. (2013): “Desarrollo de software dirigido por modelos: Conceptos, Métodos y Herramientas”. Ra-Ma Editorial, ISBN 978-84-9964-215-4.
- [García-Borgoñón et al. 2013] García-Borgoñón, L., García-García, J. A., Alba, M., Escalona, M. J. (2013): “Building Sustainable Information Systems”, Chapter “Software Process Management: A Model-Based Approach”, In *Proceedings of the 2012 International Conference on Information Systems Development*, Springer Link, pp. 167-178, DOI 10.1007/978-1-4614-7540-8\_13.
- [García-Borgoñón et al. 2014a] García-Borgoñón, L., García-García, J.A., Escalona, M.J., Barcelona, M.A. (2014): “A model based solution for Business Process Modeling in Software Development Organizations”. *Information and Software Technology (IST)*.
- [García-Borgoñón et al. 2014b] García-Borgoñón, L., Barcelona, M.A., García-García, J.A., Alba, M., Escalona, M.J. (2014): “Software Process Modeling Languages: a Systematic Literature Review”. *ISS Vol. 56*. 103-116.
- [García-García 2015] García-García, J. (2015): “Una propuesta para el uso del paradigma guiado por modelos (MDE) para la definición y ejecución de procesos de negocio”, European thesis. Department of Computer Language and Systems, University of Seville



- [García-García et al. 2012a] García-García, J. A., Ortega, M. A., García-Borgoñón, L., & Escalona, M. J. (2012). “*NDT-Suite: a model-based suite for the application of NDT*”. Web Engineering (pp. 469-472). Springer Berlin Heidelberg.
- [García-García et al. 2012b] García-García, J.A., Cutilla, C.R., Escalona, M.J., Alba, M., Torres, J. (2012): “*NDT-Driver, a Java Tool to Support QVT Transformations for NDT*”. In the 20th International Conference on Information Systems Development (ISD), DOI 10.1007/978-1-4614-4951-5\_8, pp. 170-176., 2012.
- [García-García et al. 2014] García-García, J. A., Escalona, M., Domínguez-Mayo, F., Salido, A. (2014): “*NDT-Suite: A Methodological Tool Solution in the Model-Driven Engineering Paradigm*”. Journal of Software Engineering and Applications, 7, 206-217. doi: 10.4236/jsea.2014.74022. 2014.
- [Geraci 1991] Geraci, A. (1991): “*IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries*”. IEEE Press, Piscataway, NJ, USA, 1991
- [Giachetti et al. 2008] Giachetti G. Marín B. Pastor O. (2008): “*Perfiles UML y Desarrollo Dirigido por Modelos: Desafíos y Soluciones para Utilizar UML como Lenguaje de Modelado Específico de Dominio*”. Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos. Vol. 2. No. 3,
- [Gómez-López et al. 2010] Gómez-López, M. T., Gasca, R. M., & Arévalo, C. (2010): “*A survey using constraints to decision-making for fault tolerance in Business processes*”. International Journal of Software Engineering and Its Applications, 4(4), 79-92.
- [Gómez-López et al. 2014a] Gómez-López, M. T., Borrego, D., & Gasca, R. M. (2014): “*Data state description for the migration to activity-centric business process model maintaining legacy databases*”. In Business Information Systems (pp. 86-97). Springer International Publishing.
- [Gómez-López et al. 2014b] Gómez-López, M. T., Parody, L., Gasca, R. M., & Rinderle-Ma, S. (2014): “*Prognosing the Compliance of Declarative Business Processes Using Event Trace Robustness*”. In *On the Move to Meaningful Internet Systems: OTM 2014 Conferences* (pp. 327-344). Springer Berlin Heidelberg.
- [Greenfield et al. 2004] Greenfield, J., Short, K., Cook, S., Kent, S., & Crupi, J. (2004): “*Software factories: assembling applications with patterns, models, frameworks, and tools*” (pp. 16-27). Wiley Pub.
- [Grune 1999] Grune, Dick (1999): “*Parsing Techniques: A Practical Guide*”. US: Springer
- [Günther & van der Aalst 2007] Günther, C.W., van der Aalst, W.M.P. (2007): “*A generic import framework for process event logs*”. In: Business Process Intelligence Workshop (BPI'06), LNCS 4103, pp.81–92.

- [*Hammer & Champy 1993*] Hammer, M., Champy, J. (1993): “*Reengineering the Corporation: A Manifesto for Business Revolution*”, Harper Business Books, New York, 1993. ISBN 0-06-662112-7.
- [*Hansen & Hansen 2013*] Hansen, C., Hansen, P. K. (2014): “*Microsoft Project 2013*”. Target Gruppens Forlag.
- [*Harmon 2004*] Harmon, P. (2004): “The OMG’s model driven architecture and BPM”. Business Process Trends
- [*Harrison & Feuerstein2008*] Harrison, G., Feuerstein, S. (2008): “*MySQL Stored Procedure Programming*”. O’Reilly Media. p. 49. ISBN 978-0-596-10089-6.
- [*Havey 2005*] Havey, M. (2005): “*Essential Business Process Modeling*”. O’Reilly Media, Inc.
- [*Heckel et al. 2008*] Heckel, R., Correia, R., Matos, C.M.P., El-Raml, M., Koutsoukos, G., Andrade, L.F. (2008): “*Architectural Transformations: From Legacy to Three-Tier and Services*”. Software Evolution 2008:139-170
- [*Hendricks et al. 2007*] Hendricks, K. B., Singhal, V. R., & Stratman, J. K. (2007): “*The impact of enterprise systems on corporate performance: A study of ERP, SCM, and CRM system implementations*”. Journal of Operations Management, 25(1), 65-82.
- [*Hennig et al. 2010*] Hennig, T., Cooper, R., Griffith, G. L., & Dennison, J. (2010): “*Access 2010 Programmer's Reference*”. John Wiley & Sons.
- [*Heuvel 2006*] Heuvel, W.-J.v.d. (2006): “*Aligning Modern Business Processes and Legacy Systems: A Component-Based Perspective*”. (Cooperative Information Systems). 2006: The MIT Press.
- [*Hill et al. 2006*] Hill, J.B., Sinur, J., Flint, D., Melenovsky, M.J. (2006): “*Gartner’s position on business process management*”, Business Issues Gartner, Stamford, CT, 2006.
- [*Hornung et al. 2006*] Hornung, T., Koschmider, A., Mendling, J. (2006): “*Integration of heterogeneous BPM Schemas: The Case of XPDL and BPEL*”. CAiSE Forum 2006
- [*Huai et al. 2010*] Huai, W., Liu, X., Sun, H. (2010): “*Towards trustworthy composite service through business process model verification*”. Proceedings of the 7th International Conference on Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC) 2010. p422-427.
- [*Hunter 1971*] Hunter, G. (1971): “*Metalogic: An Introduction to the Metatheory of Standard First-Order Logic*”. Berkeley:University of California Press ISBN 978-0-520-01822-8
- [*Ingerman 1967*] Ingerman, Peter Zilahy (1967): “*Pāṇini-Backus Form Suggested*”. Communications of the ACM (Association for Computing Machinery) 10 (3): 137. doi:10.1145/363162.363165

- [Ingvaldsen & Gulla 2008] Ingvaldsen, J.E., Gulla, J.A., (2008): “*Preprocessing Support for Large Scale Process Mining of SAP Transactions*”. In: Business Process Intelligence Workshop (BPI’07), LNCS 4928, pp. 30–41.
- [Iso/Iec 12207:2008] “*Iso/Iec 12207:2008 Systems and software engineering -- Software life cycle processes*”, International Organization for Standardization. [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=43447](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=43447) . Último acceso 31/10/2015.
- [Iso/Iec 19506:2009] “*Iso/Iec 19506:2009: Knowledge Discovery Meta-Model (KDM), v1.1 (Architecture-Driven Modernization)*”, [http://www.iso.org/iso/catalogue\\_detail.1128htm?csnumber=32625](http://www.iso.org/iso/catalogue_detail.1128htm?csnumber=32625) , 2009, Iso/Iec. p. 302. Último acceso 31/10/2015.
- [Iso/Iec 19507:2012] “*Iso/Iec 19507:2012 Information technology - Object Management Group Object Constraint Language (OCL)*”. International Organization for Standardization, formal/2012-05-09, 2012, [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=57306](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=57306), Último acceso 31/10/2015.
- [Iso/Iec 19510:2013] “*Iso/Iec 19510:2013, OMG Business Process Model and Notation*”. International Organization for Standardization. [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=62652](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=62652), Último acceso 31/10/2015.
- [Iso/Iec 20000:2012] “*Iso/Iec 20000:2012 Information technology -- Service management*”, International Organization for Standardization. [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=51987](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=51987). Último acceso 31/10/2015.
- [Iso/Iec 24744:2007] “*Iso/Iec 24744:2007: “Software Engineering - Metamodel for Development Methodologies*”. International Organization for Standardization. [http://www.iso.org/iso/home/store/catalogue\\_ics/catalogue\\_detail\\_ics.htm?csnumber=38854](http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=38854). Último acceso 31/10/2015.
- [Iso/Iec 24744:2014] “*Iso/Iec 24744:2014: “Software Engineering - Metamodel for Development Methodologies*”. International Organization for Standardization. [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=62644](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=62644). Último acceso 31/10/2015.
- [Iso/Iec 27000:2014] “*Iso/Iec 27000:2014: Information technology, Security techniques, Information security management systems & Overview and vocabulary*”. International Organization for Standardization. [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=63411](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=63411). Último acceso 31/10/2015.
- [Iso/Iec 29119:2014] “*Iso/Iec 29119:2014 Software Testing. The new international software testing standard*”. International Organization for Standardization, <http://www.softwaretestingstandard.org>. Último acceso 31/10/2015.

- [*Iso/Iec 33001:2015*] “*Iso/Iec 33001:2015 Information technology -- Process assessment -- Concepts and terminology*”. International Organization for Standardization,  
[http://www.iso.org/iso/home/store/catalogue\\_ics/catalogue\\_detail\\_ics.htm?csnumber=54175](http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=54175), Último acceso 31/10/2015.
- [*Iso/Iec 9001:2015*] Iso/Iec. ISO 9001:2015 “*Quality management systems, Requirements*”. International Organization for Standardization,  
[http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=62085](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=62085) , Último acceso 31/10/2015.
- [*ITIL 2015*] *Information Technology Infrastructure Library (ITIL)*. <Http://www.itil-officialsite.com> . Último acceso 31/10/2015.
- [*Iwt2 2008*] IWT2, (2008): “*Consejería de Cultura de la Junta de Andalucía. Comparativa de herramientas de modelado. Proyecto Calidad*”.
- [*Izquierdo & Molina 2009a*] Izquierdo, J. L. C., & Molina, J. G. (2009): “*A domain specific language for extracting models in software modernization*”. In *Model Driven Architecture-Foundations and Applications* (pp. 82-97). Springer Berlin Heidelberg.
- [*Izquierdo & Molina 2009b*] Izquierdo, J. L. C., & Molina, J. G. (2009): “*Gra2MoL: Una Herramienta para la Extracción de Modelos en Modernización de Software*”. JISBD 2009:162-165
- [*Izquierdo & Molina 2014*] Izquierdo, J. L. C., & Molina, J. G. (2014): “*Extracting models from source code in software modernization*”. *Software & Systems Modeling*, 13(2), 713-734.
- [*Izquierdo et al. 2008*] Izquierdo, J. L. C., Cuadrado, J. S., & Molina, J. G. (2008): “*Gra2MoL: A domain specific transformation language for bridging grammarware to modelware in software modernization*”. In *Workshop on Model-Driven Software Evolution*.
- [*Izquierdo et al. 2012*] Izquierdo, J. L. C., Jouault, F., Cabot, J., & Molina, J. G. (2012): “*API2MoL: Automating the building of bridges between APIs and Model-Driven Engineering*”. *Information and Software Technology*, 54(3), 257-273.
- [*Jablonski & Bussler 1996*] Jablonski, S., Bussler, C. (1996): “*Work flow Management. Modeling Concepts, Architecture and Implementation*”. International Thomson Computer Press, London.
- [*Jakobson 1980*] Jakobson, R. (1980): “*Metalanguage as a linguistic problem*”. Roman Jakobson. *The framework of language*. Ann Arbor: Michigan Studies in the Humanities, 81-92.
- [*Jakobson et al. 1999*] Jacobson, I., Booch, G., Rumbaugh, J. (1999): “*The unified software development process (Vol. 1)*”. Reading: Addison-wesley.

- [Jouault et al. 2008] Jouault, F., Allilaire, F., Bézivin, J., & Kurtev, I. (2008): “*ATL: A model transformation tool*”. *Science of computer programming*, Volume 72, Issues 1–2, Pages 31–39. DOI: 10.1016/j.scico.2007.08.002.
- [Kagermann et al. 2013] Kagermann, H., Helbig, J., Hellinger, A., & Wahlster, W. (2013): “*Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry; Final Report of the Industrie 4.0 Working Group*”. Forschungsunion.
- [Kaiser et al. 1990] Kaiser, G., Barghuti, N. and Sokolsky M. (1990): “*Preliminary Experience with Process Modeling in the Marvel SDE Kernel*” in *Proceedings of IEEE 23th Hawaii ICSS Software Track*.
- [Kalenkova et al. 2014] Kalenkova, A. A., de Leoni, M., & van der Aalst, W. M. (2014): “*Discovering, Analyzing and Enhancing BPMN Models Using ProM★*”
- [Kallel et al. 2009] Kallel, S., Charfi, A., Dinkelaker, T., Mezini, M., & Jmaiel, M. (2009): “*Specifying and monitoring temporal properties in web services compositions*”. In *Web Services, 2009. ECOWS'09. Seventh IEEE European Conference on* (pp. 148-157). IEEE.
- [Kaloxyllos et al. 2012] Kaloxyllos, A., Eigenmann, R., Teye, F., Politopoulou, Z., Wolfert, S., Shrank, C., ... & Kormentzas, G. (2012): “*Farm management systems and the Future Internet era*”. *Computers and electronics in agriculture*, 89, 130-144.
- [Karner 1993] Karner, Gustav. (1993): “*Resource Estimation for Objectory Projects*”. Objective Systems SF AB
- [Kazhamiakin et al. 2006] Kazhamiakin, R., Pandya, P., & Pistore, M. (2006): “*Representation, verification, and computation of timed properties in web. Web Services*”. ICWS'06. International Conference on. IEEE, p 497-504.
- [Kelley & Walker 1959] Kelley, J., Walker, M. (1959): “*Critical-path planning and scheduling*”. Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference. ACM, 1959. p. 160-173.
- [Kent 2002] Kent, S. (2002): “*Model driven engineering. In Integrated formal methods*” (pp. 286-298). Springer Berlin Heidelberg.
- [Kitchenham et al. 2009] Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). “*Systematic literature reviews in software engineering—a systematic literature review*”. *Information and software technology*, 51(1), 7-15.
- [Kleppe et al. 2003] Kleppe, A. G., Warmer, J. B., & Bast, W. (2003): “*MDA explained: the model driven architecture: practice and promise*”. Addison-Wesley Professional.
- [Ko 2009] Ko, Ryan K. L. (2009): “*A computer scientist's introductory guide to business process management (BPM)*”. *Crossroads* v.15 n.4, p.11-18.

- [Ko et al. 2009] Ko, R. K., Lee, S. S., & Wah Lee, E. (2009): “*Business process management (BPM) standards: a survey*”. *Business Process Management Journal*, 15(5), 744-791.
- [Korten 1998] Korten, D. C. (1998): “*When corporations rule the world*”. *European Business Review*, 98(1).
- [Kruchten 2004] Kruchten, P. (2004): “*The rational unified process: an introduction*”. Addison-Wesley Professional.
- [Kumaran et al. 2008] Kumaran, S., Liu, R., & Wu, F. Y. (2008): “*On the duality of information-centric and activity-centric models of business processes*”. In *Advanced Information Systems Engineering* (pp. 32-47). Springer Berlin Heidelberg.
- [Kurtev et al. 2002] Kurtev, I., Bézivin, J., Aksit, M. (2002): “*Technological spaces: An initial appraisal*”. In: *CoopIS, DOA’2002 Federated Conferences, Industrial track*
- [Lang 2010] Lang, J. P. (2010): “*Redmine-Overview*”.
- [Lanz et al. 2014] Lanz, A., Weber, B., & Reichert, M. (2014): “*Time patterns for process-aware information systems*”. *Requirements Engineering*, 19(2), 113-141.
- [Lauesen & Vinter 2001] Lauesen, S., & Vinter, O. (2001): “*Preventing requirement defects: An experiment in process improvement*”. *Requirements Engineering*, 6(1), 37-50.
- [Liddle 2011] Liddle, S. W. (2011): “*Model-driven software development*”. In *Handbook of Conceptual Modeling* (pp. 17-54). Springer Berlin Heidelberg.
- [Lu & Sadiq 2007] Lu, R., & Sadiq, S. (2007): “*A survey of comparative business process modeling approaches*”. In *Business information systems* (pp. 82-94). Springer Berlin Heidelberg.
- [Magoulas et al. 2007] Magoulas, G., Lepouras, G., Vassilakis, C., “*Virtual reality in the e-Society*”, Springer London, 2007, Issn: 1359-4338
- [Makni et al. 2010] Makni, M., Tata, S., Yeddes, M. M., and Hadj-Alouane, N. B. (2010): “*Satisfaction and Coherence of Deadline Constraints in Inter-Organizational Workflows*”. In *Proceedings of the Confederated International Conferences: CoopIS on the Move to Meaningful Internet Systems: OTM 2010*, volume 6426 of LNCS, pages 523–539. Springer.
- [Makni et al. 2011] Makni, M., Hadj-Alouane, N. B., Tata, S., and Yeddes, M. M. (2011): “*Negotiating Deadline Constraints in Interorganizational Logistic Systems: A Healthcare Case Study*”. In *Proceedings of the International Workshops and Education Track on Business Process Management Workshops (BPM)*, 2011, volume 100 of LNBIP, pages 108–118. Springer.
- [Malcolm et al. 1959] Malcolm, D. G., Roseboom, J. H., Clark, C. E., & Fazar, W. (1959): “*Application of a technique for research and development program evaluation*”. *Operations research*, 7(5), 646-669.

- [*Malinova & Mendling, 2013*] Malinova, M., & Mendling, J. (2013): “*A Qualitative Research Perspective on BPM Adoption and the Pitfalls of Business Process Modeling*”. In *Business Process Management Workshops* (pp. 77-88). Springer Berlin Heidelberg.
- [*Manzón & Trujillo 2007*] Manzón, J.-N. & Trujillo, J. (2007): “*Model-driven reverse engineering for data warehouse design*”. JISBD 2007.
- [*MAP 2005*] M. A. P. (2005): “*Métrica Versión 3. Metodología de planificación, desarrollo y mantenimiento de sistemas de información*”. Ministerio de Administraciones Públicas.
- [*McNeil & Hanman 2013*] McNeil, K., & Hanman, S. (2013): “*Why do software projects fail? A case study*”. MHD Supply Chain Solutions, 43(2): 62-65.
- [*Mellor et al. 2003*] Mellor, S. J., Clark, A. N., Futagami, T. (2003): “*Model-driven development - Guest editor's introduction*”. IEEE Softw. 20, 5, 14–18.
- [*Melton & Simon 2002*] Melton, J., Simon, A.R. (2002): “*SQL:1999*”. Morgan Kaufmann. pp. 541–542. ISBN 978-1-55860-456-8.
- [*Mendling et al. 2010*] Mendling, J., Reijers, H.A., van der Aalst, W.M.P. (2010): “*Seven process modeling guidelines (7PMG)*”, Information and Software Technology, Volume 52, Issue 2, ISSN 0950-5849, <http://dx.doi.org/10.1016/j.infsof.2009.08.004>127-136.
- [*Merks 2010*] Merks, E. (2010): “*Eclipse modeling framework (EMF)*”. Recuperado de <http://jaxenter.com/eclipse-modeling-framework-interview-with-ed-merks-10027.html>. Último acceso 31/10/2015
- [*Microsoft™: T-SQL 2014*] Microsoft: T-SQL (2014): “*Transact-SQL Reference (Database Engine)*”. Recuperado de <http://msdn.microsoft.com/en-us/library/bb510741.aspx>. Último acceso 31/10/2015.
- [*Milanovic et al. 2009*] Milanovic, N., et al. (2009): “*Model-based interoperability of heterogeneous information systems: An industrial case study*”. In: Paige, R.F., et al. (Eds.) ECMDA-FA. LNCS, vol. 5562, pp. 325–336. Springer, 2009
- [*Millas 2013*] Millas, J. L. L. (2013): “*Microsoft. Net Framework 4. 5 Quickstart Cookbook*”. Packt Publishing Ltd.
- [*Mohapatra 2012*] Mohapatra, S. (2012): “*Business process reengineering: automation decision points in process reengineering*”. Springer Science & Business Media.
- [*Moløkken & Jørgensen 2003*] Moløkken, K., & Jørgensen, M. (2003): “*A review of software surveys on software effort estimation*”. In *Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on* (pp. 223-230). IEEE.
- [*Momjian 2001*] Momjian, B. (2001): “*PostgreSQL: introduction and concepts (Vol. 192)*”. New York: Addison-Wesley.

- [*Monteiro et al. 2013*] Monteiro, J., Swatman, P. M., & Tavares, L. V. (Eds.). (2013): “*Towards the Knowledge Society: ECommerce, EBusiness and Egovernment*”. The Second IFIP Conference on E-Commerce, E-Business, E-Government (I3E 2002) October 7–9, 2002, Lisbon, Portugal (Vol. 105). Springer.
- [*Moore 1965*] Moore, G. E. (1965): “*Cramming more components onto integrated circuits*”, *Electronics*, Volume 38, Number 8, April 19, 1965.
- [*Moreno-Montes de Oca et al. 2015*] Moreno-Montes de Oca, I., Snoeck, M., Reijers, H.A., Rodríguez-Morffi, A. (2015): “*A systematic literature review of studies on business process modeling quality*”, *Information and Software Technology*, Volume 58, ISSN 0950-5849, <http://dx.doi.org/10.1016/j.infsof.2014.07.011>, Pages 187-205
- [*Müller et al. 2000*] Müller, H. A., Jahnke, J. H., Smith, D. B., Storey, M. A., Tilley, S. R., & Wong, K. (2000): “*Reverse engineering: A roadmap*”. In *Proceedings of the Conference on the Future of Software Engineering* (pp. 47-60). ACM.
- [*Netjes et al. 2006*] Netjes, M., Reijers, H. A., & van der Aalst, W. M. (2006): “*Supporting the BPM life-cycle with FileNet*”. In *Proceedings of the CAiSE* (Vol. 6, pp. 497-508).
- [*Newcomb 2005*] Newcomb, P. (2005): “*Architecture-Driven Modernization (ADM)*”, in *WCRE'05*. p. 237.
- [*OASIS:BPPEL 2007*] OASIS (2007): “*Web Services Business Process Execution Language Version 2.0.*”. Organization for the Advancement of Structured Information Standards. URL: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>.
- [*OASIS:BPPEL4People 2010*] OASIS (2010): “*WS-BPEL Extension for People (BPPEL4People) Specification Version 1.1.*” Organization for the Advancement of Structured Information Standards. URL: <http://docs.oasis-open.org/bpel4people/bpel4people-1.1.html>
- [*OMG:ADM 2005*] OMG:ADM (2005): “*OMG Architecture-driven modernization (ADM)*”. Recuperado de <http://adm.omg.org>. Último acceso 31/10/2015.
- [*OMG:ASTM 2011*] OMG:ASTM (2011): “*Syntax tree metamodel. v. 1.0.*”. Recuperado de <http://www.omg.org/spec/ASTM/1.0>. Último acceso 31/10/2015.
- [*OMG:BPDM 2008*] OMG:BPDM. (2008): “*Business Process definition Metamodel (BPDM), Version 1.0.*”. Recuperado de <http://www.omg.org/spec/BPDM/>. Último acceso 31/10/2015
- [*OMG:BPMN 2013*] OMG:BPMN (2013): “*BPMN, Business Process Modeling Notation, Version 2.0.2*”. Recuperado de <http://www.omg.org/spec/BPMN/>. Último acceso 31/10/2015.



- [*OMG:CWM 2003*] OMG:CWM (2003): “*OMG Common Warehouse Metamodel (CWM)*”, Recuperado de <http://www.omg.org/spec/CWM/1.1> . Último acceso 31/10/2015.
- [*OMG:IMM 2006*] OMG:IMM (2006): “*Information management metamodel (IMM)*”, Recuperado de <http://www.omgwiki.org/imm/doku.php> Último acceso 31/10/2015.
- [*OMG:MDA 2011*] OMG:MDA (2011): “*Modern Driven Architecture (MDA)*”, Recuperado de [http://www.omg.org/mda/.](http://www.omg.org/mda/) Último acceso 31/10/2015.
- [*OMG:MOF 2011*] OMG:MOF (2011): “*Meta Object Facility (MOF)*”, Recuperado de <http://www.omg.org/spec/MOF/2.4.1> . Último acceso 31/10/2015.
- [*OMG:MOFM2T 2008*] OMG:MOFM2T(2008): “*MOF Model to Text Transformation Language (MOFM2T), 1.0.*” Object Management Group. URL: <http://www.omg.org/spec/MOFM2T/1.0/>.
- [*OMG:OCL 2014*] OMG:OCL (2014). “*Object Constraint Language (OCL), Version 2.4*”. Recuperado from <http://www.omg.org/spec/OCL/2.4/> (2014). Last accessed 07/2014
- [*OMG:PRR 2009*] OMG (2009): “*UML, Production Rule Representation (PRR) 1.0*”. Recuperado de <http://www.omg.org/spec/PRR/1.0/> . Último acceso 31/10/2015.
- [*OMG:QVT 2011*] OMG:QVT (2011): “*QVT, Meta Object Facility (MOF) 2.0 Query/View/Transformation*”. Recuperado de <http://www.omg.org/spec/QVT/1.1/> . Último acceso 31/10/2015.
- [*OMG:SBVR 2015*] OMG:SBVR (2015): “*Semantics of Business Vocabulary and Rules (SBVR) V 1.3*”, <http://www.omg.org/spec/SBVR/1.3/>, Último acceso 31/10/2015.
- [*OMG:SPEM 2008*] OMG:SPEM (2008): “*SPEM, Software & Systems Process Engineering Metamodel specification 2.0* ”. Recuperado de <http://www.omg.org/spec/SPEM/>. Último acceso 31/10/2015.
- [*OMG:UML 2011*] OMG (2011): “*UML, Unified Modeling Language, Versión 2.4.1*”. Recuperado de <http://www.omg.org/spec/UML/>. Último acceso 31/10/2015.
- [*OMG:XMI 2013*] OMG:XMI (2013): “*MOF/XMI Mapping, Version 2.4.1*.” Recuperado de <http://www.omg.org/spec/XMI/2.4.1/>. Último acceso 31/10/2015.
- [*O'reilly 2007*] O'reilly, T. (2007): “*What is Web 2.0: Design patterns and business models for the next generation of software*”. *Communications & strategies*, (1), 17.
- [*Oshri et al. 2015*] Oshri, I., Kotlarsky, J., & Willcocks, L. P. (2015): “*The Handbook of Global Outsourcing and Offshoring 3rd Edition*”. Palgrave Macmillan.

- [*Osterweil 1987*] Osterweil, L. (1987): “*Software processes are software too*”. In proceedings of the 9th International Conference on Software Engineering, volume 3 of ICSE '87, pages 2-13. IEEE Computer Society Press, IEEE Computer Society Press.
- [*Paradauskas & Laurikaitis 2006*] Paradauskas, B., Laurikaitis, A. (2006): “*Business Knowledge extraction from Legacy Information Systems*”. Journal of Information Technology and Control 35(3) 214–221
- [*Parr & Fisher 2011*] Parr, T., & Fisher, K. (2011). “*LL (\*): the foundation of the ANTLR parser generator*”. In ACM SIGPLAN Notices (Vol. 46, No. 6, pp. 425-436). ACM.
- [*Pelechano 2012*] Pelechano, V. (2012): “*Automating the development of information systems with the MOSKitt open source tool*”. RCIS 2012:1-3
- [*Pérez et al. 2003*] Pérez, J, Anaya, V, Cubel, J.M., Ramos, I, Carsí, J.A. (2003): “*Data Reverse Engineering of Legacy Databases to Object Oriented Conceptual Schemas*”. Electr. Notes Theor. Comput. Sci. 72(4): 7-19, 2003
- [*Pérez-Castillo et al. 2009*] Pérez-Castillo, R., de Guzmán, I. G. R., Ávila-García, O., & Piattini, M. (2009): “*MARBLE: a modernization approach for recovering business processes from legacy systems*”. In International Workshop on Reverse Engineering Models from Software Artifacts (REM'09) (pp. 17-20).
- [*Pérez-Castillo et al. 2011a*] Pérez-Castillo, R., Piattini, M. et.al, (2011): “*MARBLE. A business process archeology tool*”. ICSM 2011:578-581.
- [*Pérez-Castillo et al. 2011b*] Pérez-Castillo, R., Piattini, M. et.al (2011): “*Knowledge discovery metamodel-Iso/Iec 19506: A standard to modernize legacy systems*”. Computer Standards & Interfaces, 33, 519–532. doi:10.1016/j.csi.2011.02.007
- [*Pérez-Castillo et al. 2012a*] Pérez-Castillo, R., Piattini, M. et.al (2012): “*A family of case studies on business process mining using MARBLE*”. Journal of Systems and Software (JSS) 85(6):1370-1385.
- [*Pérez-Castillo et al. 2012b*] Pérez-Castillo. R., Piattini, M. et.al (2012): “*Database schema elicitation to modernize relational databases*”. ICEIS
- [*Pérez-Castillo et al. 2012c*] Pérez-Castillo, R., Piattini, M. et.al (2012): “*Software modernization by recovering web services from legacy databases*”. J. Softw. Maint. Evol.: Res. Pract.
- [*Pichler et al. 2011*] Pichler, P., Weber, B., Zugal, S., Pinggera, J., Mendling, J., Reijers, H.A. (2011): “*Imperative versus Declarative Process Modeling Languages: An Empirical Investigation*”. Business Process Management Workshops 2011:383-394, Springer 2012.
- [*PL/pgSQL 2014*] PostgreSQL (2014): “*PL/pgSQL - SQL Procedural Language*”. Recuperado de <http://www.postgresql.org/docs/current/static/plpgsql.html>. Último acceso 31/10/2015.

- [*PMI:PMBOK 2013*] Project Management Institute (2013): “*A Guide to the Project Management Body of Knowledge (Pmbok Guide)*”. 5th Edition.
- [*Ponce et al. 2013*] Ponce, J., García-Borgoñon, L., García-García, J.A., Escalona, M.J., Domínguez-Mayo, F.J., Alba, M., and Aragon, G. (2013): “*A Model-Driven Approach for Business Process Management*”. *Covenant Journal of Engineering & Technology (CJICT)* Vol. 1, No. 2, pp. 32-52. 2013.
- [*Pressman 2010*] Pressman, R.S. (2010). “*Ingeniería del software. Un enfoque practico*”. McGraw-Hill, 7ªEd, ISBN: 978-607-15-0314-5.
- [*Rademakers 2012*] Rademakers, T. (2012): “*Activiti in Action: Executable business processes in BPMN 2.0.*”. Manning Publications Co.
- [*Recker 2010*] Recker, J. (2010): “*Opportunities and constraints: the current struggle with BPMN*”. *Business Process Management Journal*, 16(1), 181-201.
- [*Reichert & Weber 2012*] Reichert, M., & Weber, B. (2012): “*Enabling flexibility in process-aware information systems: challenges, methods, technologies*”. Springer Science & Business Media.
- [*Reijers 2013*] Reijers, H.A., Slaats, T., Stahl, C. (2013): “*Declarative Modeling-An Academic Dream or the Future for BPM?*”. *BPM 2013*:307-322
- [*Reina et al. 2004*] Reina, A. M., Torres, J., & Toro, M. (2004): “*Towards developing generic solutions with aspects*”. In *Proceedings of 5th Aspect-Oriented Modeling Workshop (AOM) in Conjunction with the UML 2004 Conference*.
- [*Reus et al. 2006*] Reus T., Geers H., van Deursen A. (2006): “*Harvesting Software Systems for MDA-Based Reengineering*”. *ECMDA-FA* pages:213-225
- [*Rifkin 2011*] Rifkin, J. (2011): “*The third industrial revolution: how lateral power is transforming energy, the economy, and the world*”. Macmillan.
- [*Ritzer 1991*] Ritzer, G. (1991). “*Metatheorizing in sociology*”. Lexington Books.
- [*Ross 2013*] Ross, R.G. (2013): “*BUSINESS RULE CONCEPTS: Getting to the Point of Knowledge*”. Business Rule Solutions, LLC (April 2013), ISBN: 0-941049-14-0, (4th Edition).
- [*Rossi et al. 2006*] Francesca Rossi; Peter van Beek; Toby Walsh (2006): “*Handbook of constraint programming*”. Elsevier. p. 157. ISBN 978-0-444-52726-4.
- [*Rolstadås 1995*] Rolstadås, A. (1995): “*Business process modeling and reengineering*”. in: *Performance Management: A Business Process Benchmarking Approach*. p. 148-150.
- [*Rubinger et al. 2014*] Rubinger, A. L., Knutsen, A. (2014): “*Continuous Enterprise Development in Java*”. O'Reilly Media, Inc.

- [Ruiz et al. 2003] Ruiz, F., Vizcaino, A., Garcia, F.: Piattini, M. (2003): “Using XMI and MOF for representation and interchange of software processes”. In Proceedings of the 14th International Workshop on database and expert systems applications, pages 739-744.
- [Ruiz-González & Cánfora 2004] Ruiz-González, F., Canfora, G. (2004): “Software Process: Characteristics”. Technology and Environments. SPT - Software Process Technology, vol. 5, pp. 5-10.
- [Salido et al. 2014] Salido, A., García-García, J.A., Gutiérrez, J., Ponce, J., (2014): “Tests Management in CALIPSONeo: A MDE Solution”. Journal of Software Engineering and Applications.
- [Sánchez Ramón et al. 2013] Ramón, O. S., Ruiz, F. J. B., & Molina, J. G. (2013): “Una valoración de la Modernización de Software Dirigida por Modelos”. JISBD.
- [Sánchez Ramón et al. 2014] Sánchez Ramón, O., Sánchez Cuadrado, J., García Molina, J. (2014): “Model-driven reverse engineering of legacy graphical user interfaces”. Autom. Softw. Eng. 21(2): 147-186, 2014
- [Schmidt, 2006] Schmidt, D.C. (2006): “Model-Driven Engineering”, Computer, vol.39, no.2, pp.25-31, 2006.
- [Schwaber 2004] Schwaber, K. (2004): “Agile project management with Scrum”. Microsoft Press.
- [Seacord et al. 2005] Seacord, R.C., Plakosh, D., Lewis, G.A. (2005): “Modernizing Legacy Systems”. Software Technologies, Engineering Processes, and Business Practices. SEI series in software engineering, Addison-Wesley 2003, ISBN 978-0-321-11884-4, pp. I-XV, 1-332
- [Seidewitz 2003] Seidewitz, E. (2003): “What models mean”. IEEE Software, 20(5):26-32.
- [Selic 2003] Selic, B. (2003). “The pragmatics of model-driven development”. IEEE Software, 20(5):19-25.
- [Selic 2008] Selic, B. (2008): “Mda manifestations”. The European Journal for the Informatics Professional (UPGRADE), 9(2):12-16.
- [Shapiro 2010] Shapiro, R.M. (2010): “XPDL 2.2: Incorporating BPMN 2.0 Process Modeling Extensions”. In 2010 BPM and Workflow Handbook, Spotlight on Business Intelligence, Ed.: Layna Fischer, Future Strategies Inc, ISBN-13: 9780981987057, pp. 201-213.
- [Shariff 2013] Shariff, M. (2013): “Alfresco 4 Enterprise Content Management Implementation”. Packt Publishing Ltd.
- [Smith & Bates 2007] Smith, T., & Bates, S. (2007): “SharePoint 2007 user's guide: learning Microsoft's collaboration and productivity platform”. Apress.

- [*SOB 2009*] SOB (2009): “*Managing Successful Projects with PRINCE2*”. Stationery Office Books. ISBN13: 978-0113310593. 2009.
- [*Sommerville & Kotonya 1998*] Sommerville, I., & Kotonya, G. (1998): “*Requirements engineering: processes and techniques*”. John Wiley & Sons, Inc.
- [*Soule 2010*] Soule, P. (2010): “*Autonomics development: a domain-specific aspect language approach*”. Springer Science & Business Media.
- [*Sparx Systems:EA 2015*] Sparx Systems (2015): “Enterprise Architect (EA). Visual modeling platform”. Retrieved at <http://www.sparxsystems.com/products/ea/index.html>. Último acceso 31/10/2015.
- [*Sprinkle 2004*] Sprinkle, J. (2004): “*Model-integrated computing*”. Potentials, IEEE, 23(1), 28-30.
- [*Standish 2012*] The Standish Group (2012): “The Standish Group Report Chaos”, <https://www.standishgroup.com/>, Último acceso 31/10/2015.
- [*Stavru et al. 2013*] Stavru S., Krasteva I., Ilieva S. (2013): “*Challenges of Model-Driven Modernization: An Agile Perspective*”, MODELSWARD\_2013.
- [*Stover 2007*] Stover, T. (2007): “*Microsoft® office project 2007 inside out*”. Microsoft Press.
- [*Stroppi et al. 2012*] Stroppi, L., Chiotti, O., Villarreal, P.(2012): “*Extended Resource Perspective Support for BPMN and BPEL*”. CIBSE 2012:56-69
- [*Sutton et al. 1995*] Sutton, S.M., Heimbigner, D., Osterweil, L.J., (1995): “*APPL/A: A Language for Software Process Programming*”. ACM Trans. Softw. Eng. Methodol. (TOSEM) 4(3):221-286
- [*Svatos 2012*] Svatos, O (2012). “*Modeling Suspension and Continuation of a Process*”. Journal of Systems Integration 3 (2) , 2012., pp. 62–73, ISSN: 1804-2724
- [*Thiry & Thirion 2009*] Thiry L., Thirion B. (2009): “*Functional Metamodels for Systems and Software*”. Journal of Systems and Software, Vol. 82, Iss. 7, pp 1125–1136. DOI: <http://dx.doi.org/10.1016/j.jss.2009.01.042>.
- [*Tilley & Smith 1995*] Tilley, S. R., & Smith, D. (1995): “*Perspectives on legacy system reengineering*”. Technical report, Software Engineering Institute, Carnegie Mellon University (1995)
- [*Tran et al. 2007*] Tran, H.N., Coulette, B., Dong, B.T. (2007): “*Modeling process patterns and their application*”. Cap Esterel, France, 2007.
- [*Trkman 2010*] Trkman, P. (2010): “*The critical success factors of business process management*”. *International Journal of Information Management*, 30(2), 125-134.

- [Türker & Gertz 2001] Türker, C., Gertz, M. (2001): “*Semantic integrity support in SQL: 1999 and commercial (object-) relational database management systems*”, VLDB J. (VLDB) 10(4):241-269
- [Ulrich & Newcomb 2010] Ulrich, W. M., & Newcomb, P. (2010). “*Information Systems Transformation: Architecture-Driven Modernization Case Studies*”. Morgan Kaufmann.
- [Ulrich 2002] Ulrich, W.M. (2002): “*Legacy systems: transformation strategies (p. 448)*”. Englewood Cliffs: Prentice Hall.
- [Usländer et al. 2013] Usländer, T., Berre, A. J., Granell, C., Havlik, D., Lorenzo, J., Sabeur, Z., & Modafferi, S. (2013): “*The future internet enablement of the environment information space*”. In *Environmental Software Systems. Fostering Information Sharing* (pp. 109-120). Springer Berlin Heidelberg.
- [US GPO 1918] Division of Vocational Education (1918): “*Trade and Industrial Series Vol 1-7*”. U.S. Government Printing Office.
- [US DoD 1989] United States Department of Defense (1989): “*Total Quality Management: A Guide for Implementation*”, Springfield, Virginia: National Technical Information Service, OCLC 21238720, DoD 5000.51-G
- [Valderas & Pelechano 2011] Valderas, P., Pelechano, V. (2011): “*A Survey of Requirements Specification in Model-Driven Development of Web Applications*”. TWEB 5(2):10 (2011)
- [van der Aalst 2004] van der Aalst, W.M.P. (2004): “*Business process management: a personal view*”. Business Process Management Journal, vol. 10, no. 2, p. 5.
- [van der Aalst 2011] van der Aalst, W.M.P. (2011): “*Process Mining: Discovery, Conformance and Enhancement of Business Processes*”. Springer, Heidelberg.
- [van der Aalst 2013] van der Aalst, W.M.P. (2013): “*A General Divide and Conquer Approach for Process Mining*”. In M. Ganzha, L. Maciaszek, and M. Paprzycki, editors, Federated Conference on Computer Science and Information Systems (FedCSIS 2013), pages 1-10. IEEE Computer Society.
- [van der Aalst 2014] van der Aalst, W.M.P. (2014): “*Process Mining in the Large: A Tutorial*”. In E. Zimanyi, editor, Business Intelligence (eBISS 2013), volume 172 of Lecture Notes in Business Information Processing, pages 33-76. Springer-Verlag, Berlin.
- [van der Aalst 2015] van der Aalst, W.M.P. (2015): “*Extracting Event Data from Databases to Unleash Process Mining*”. In J. Vom Brocke and T. Schmiedel, editors, Business Process Management Roundtable 2014, Springer-Verlag, Berlin, 2015.
- [van der Aalst et al. 2005] van der Aalst, W.M.P., Weske, M., Grunbauer, D. (2005): “*Case handling: A new paradigm for business process support*”. Data and Knowledge Eng. 53

- [van der Aalst et al. 2007] van der Aalst, W., Reijers, H., Weijters, A. (2007). “*Business process mining: an industrial application*”. *Information Systems* 32 (5), 713–732
- [van der Aalst et al. 2012] van der Aalst, W., Adriansyah, A., de Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., & Pontieri, L. (2012): “*Process Mining Manifesto*”. In *Business process management workshops* (pp. 169-194). Springer Berlin Heidelberg.
- [van der Aalst & Nikolov 2007] van der Aalst, W. M., & Nikolov, A. (2007): “*EMailAnalyzer: an e-mail mining plug-in for the ProM framework*”. BPM Center Report BPM-07-16, BPMCenter. Org.
- [van der Straeten et al. 2009] van der Straeten R., Mens, T., van-Baelen S. (2009): “*Challenges in Model-Driven Software Engineering. Models in Software Engineering*”, *Lecture Notes in Computer Science*, no. 5421, pp. 35-47.
- [van Deursen 1997] van Deursen, A. (1997): “*Domain specific languages versus object oriented frameworks: A financial engineering case study*”. *Proceedings of Smalltalk and Java in Industry and Academia*, p. 35---39.
- [van Deursen et al. 2000] van Deursen, A.; Klint, P. & Visser, J. (2000). “*Domain specific languages: an annotated bibliography*”. *SIGPLAN Not.*, 35(6):26-36.
- [Verbeek et al. 2011] Verbeek, H. M. W., Buijs, J. C., van Dongen, B. F., & van der Aalst, W. M. (2011): “*Xes, xesame, and prom 6*”. In *Information Systems Evolution* (pp. 60-75). Springer Berlin Heidelberg.
- [Viana et al. 2007] Viana, S., Rady de Almeida Jr., J., Pavón, J. (2007): “*A Rule Repository for Active Database Systems*”. *CLEI Electron. J. (CLEIEJ)* 10(2)
- [von Rosing et al. 2014] von Rosing, M., von Scheel, H., Scheer, A. (2014): “*The Complete Business Process Handbook, Body of Knowledge from Process Modeling to BPM, Volume I, 1<sup>st</sup> Edition*”. Morgan Kaufmann, <http://store.elsevier.com/product.jsp?isbn=9780127999593>.
- [Völter et al. 2006] Völter, M., Stahl, T., Bettin, J., Haase, A., Helsen, S., & Czarnecki, K. (2006): “*Model-Driven Software Development: Technology. Engineering, Management*”. Wiley, 5, 6.
- [Wagner 2003] Wagner, G. (2003): “*The agent-object-relationship metamodel: towards a unified view of state and behavior*”. *Inf. Syst.*, 28(5):475–504.
- [Wagner 2005] Wagner, G. (2005): “*Rule modeling and markup*”. *Proceedings of the First international conference on Reasoning Web 2005*: 251–274. Springer-Verlag.
- [Wahl & Sindre 2005] Wahl, T. & Sindre, G. (2005): “*An Analytical Evaluation of BPMN Using a Semiotic Quality Framework*”. *CAiSE’05 Workshops. v.1*, p.533–544.
- [Wallis 2010] Wallis, S. E. (2010): “*Toward a science of metatheory. Integral Review: A Transdisciplinary and Transcultural Journal for New Thought*”, *Research, and Praxis*, 6(3).

- [Watahiki et al. 2011] Watahiki, K., Ishikawa, F., Hiraishi, K. (2011): “*Formal verification of business processes with temporal and resource constraints*”. In Systems, Man, and Cybernetics (SMC), IEEE International Conference, 2011 (pp. 1173-1180).
- [Weber et al. 2007] Weber, B., Rinderle, S., Reichert, M. (2007): “*Change patterns and change support features in process-aware information systems*”. In: Krogstie et al (2007), pp 574–588
- [Weber et al. 2009] Weber, B., Sadiq, S.W., Reichert, M. (2009): “*Beyond rigidity - dynamic process lifecycle support*”. Computer Science - R&D (IFE) 23(2):47-65
- [Weske 2012] Weske, M. (2012): “*Business Process Management - Concepts, Languages, Architectures*”, 2nd Edition. Springer 2012
- [Westergaard 2014] Westergaard, M., van Dongen, B.F. (2014): “*KeyValueSets: Event Logs Revisited*”
- [WfMC:XPDL 2012] WfMC:XPDL 2.2 (2012): “*XML Process Definition Language (XPDL) 2.2*”, Workflow Management Coalition, [http://www.xpdl.org/standards/xpdl-2.2/XPDL%202.2%20\(2012-08-30\).pdf](http://www.xpdl.org/standards/xpdl-2.2/XPDL%202.2%20(2012-08-30).pdf), vÚltimo acceso 31/10/2015.
- [White & Bock 2011] White, S. A., & Bock, C. (2011): “*BPMN 2.0 Handbook Second Edition: Methods, Concepts, Case Studies and Standards*”. In Business Process Management Notation. Future Strategies Inc.
- [White 2008] White, S. A. (2008): “*BPMN modeling and reference guide: understanding and using BPMN*”. Future Strategies Inc.
- [Wicks et al. 2007] Wicks, M.N., Dewar, R.G. (2007): “*A new research agenda for tool integration*”. Journal of Systems and Software 80(9), 1569–1585, 2007
- [Wimmer & Kramler 2005] Wimmer, M., Kramler, G. (2005): “*Bridging grammarware and modelware*”. In: J.-M. Bruel (Ed.): MoDELS 2005 Workshops, LNCS 3844, pp. 159–168, 2006. Springer-Verlag.
- [Wong & Gibbons 2009] Wong, P., Gibbons, J. (2009): “*A relative timed semantics for BPMN*”. Electronic Notes in Theoretical Computer Science, 2009, vol. 229, no 2, p. 59-75.
- [Zikopoulos & Eaton 2011] Zikopoulos, P., & Eaton, C. (2011): “*Understanding big data: Analytics for enterprise class hadoop and streaming data*”. McGraw-Hill Osborne Media.
- [Zissis & Lekkas 2012] Zissis, D., & Lekkas, D. (2012): “*Addressing cloud computing security issues*”. Future Generation computer systems, 28(3), 583-592.
- [Zur Muehlen & Recker, 2008] Zur Muehlen, M., & Recker, J. (2008): “*How much language is enough? Theoretical and practical use of the business process modeling notation*”. In Advanced information systems engineering (pp. 465-479). Springer Berlin Heidelberg.



[Zur Muehlen & Recker, 2013] Zur Muehlen, M., Recker, J. (2013): “*We Still Don’t Know How Much BPMN Is Enough, But We Are Getting Closer*”. Seminal Contributions to Information Systems Engineering. Springer Berlin Heidelberg, p. 445-451.



## **PARTE V. ANEXOS**



## **ANEXO I: ACRÓNIMOS**



- ADM.** «*Architecture Driven Modernization*». Iniciativa OMG para modernización de *legacy systems*.
- Analista TIC.** Persona del equipo de proyecto que, partiendo del dominio del problema, está capacitado para elaborar modelos con diferentes niveles de abstracción.
- BPEL.** «*Business Process Execution Language*». Lenguaje estándar de *OASIS* para la ejecución de procesos.
- BPEL4People.** «*WS-BPEL Extension for People BPEL4People*». Lenguaje estándar de *OASIS* para la ejecución de procesos.
- ASTM.** «*Abstract Syntax Tree Meta-Model*». Estándar *OMG* para la definición de árboles de sintaxis abstracta.
- BPDM.** «*Business Process Definition Meta-Model*». Estándar *OMG* para el intercambio de artefactos de distintos *Meta-Modelos* de procesos.
- BPM.** «*Business Process Management*». Identifica una estrategia de gestión que incluye aquellos métodos, técnicas y herramientas de soporte al ciclo de vida de los procesos de negocio, el cual a su vez incluye el diseño, aprobación, gestión y análisis de los procesos de negocio operacionales que involucran a personas, organizaciones, aplicaciones, documentos y cualquier otra fuente de información.
- BPMN.** «*Business Process Model and Notation*». Identifica un lenguaje estándar de propósito general para el modelado de procesos de negocio.
- BPMS.** «*Business Process Management System*» o «*Business Process Management Suite*» que puede ser definido como una nueva categoría de software empresarial que permite a las empresas modelar, implementar y ejecutar conjuntos de actividades interrelacionadas – es decir, Procesos – de cualquier naturaleza, sea dentro de un departamento o permeando la entidad en su conjunto, con extensiones para incluir los clientes, proveedores y otros agentes como participantes en las tareas de los procesos.
- BPR.** «*Business Process Reengineering*» o práctica en las organizaciones de cualquier sector productivo, que supone orientar la organización a la gestión de sus procesos buscando mejores productos o servicios, mejor calidad y atención al cliente. Para ello se busca la redefinición de los procesos de negocio, eliminando tareas innecesarias, minimizar el tiempo de ejecución y espera de tareas y del proceso global, haciéndolo más eficiente y beneficioso para la organización.
- CIM.** «*Computer Independent Meta-Model*». Término *MDA* para *Meta-Modelo* independiente de la computación. Es el nivel de abstracción más cercano al dominio del negocio.
- CMMI.** «*Capability Maturity Model Integration*». Modelo de madurez que establece niveles para medir la calidad de los procesos de una organización dedicada al software.

- CPM.** «*Critical Path Method*». Método para calcular el camino crítico o de duración mínima de un grafo **PERT**.
- CWM.** «*Common Warehouse Meta-Model*». Estándar *OMG* para la definición e intercambio de almacenes de datos.
- Experto de negocio.** Persona de la organización que conoce el dominio del problema. Tiene el conocimiento para poder especificar cualquier proceso y sus variantes, detallando al máximo nivel cualquier regla del negocio.
- Experto TIC.** Experto en tecnologías de información y comunicaciones (consultor, analista, programador, técnico de sistemas, etc.).
- GASTM.** «*Generic Abstract Syntax Tree Meta-Model*». Estándar *OMG* para la definición de árboles de sintaxis abstracta con instrucciones compartidas por distintos lenguajes.
- ISO.** «*International Organization for Standardization*». Organismo internacional para la estandarización.
- ITIL.** «*Information Technology Infrastructure Library*».
- IWT2.** Grupo de investigación «*Ingeniería Web y Testing Temprano*» del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla.
- IMM.** «*Information Management Meta-Model*». Estándar *OMG* para soportar la interoperabilidad entre distintos *Meta-Modelos*. Es una evolución de *OMG CWM*.
- KDM.** «*Knowledge Discovery Meta-Model*». Estándar *OMG* para la extracción de conocimiento desde *legacy systems*. Es un *Meta-Modelo* integrado en *ADM*.
- Legacy database.** Base de datos heredada o base de datos de un *legacy system*. Una *legacy database* da persistencia a estados que son consecuencia de la ejecución de distintos procesos de una organización.
- Legacy system (o legacy information system).** *Sistema heredado* o aplicación informática antigua que se sigue utilizando y no es fácil de sustituir.
- Mapeo.** Este término es la traducción al español utilizado en esta tesis del término *mapping operation*, definido en *QVT* como una operación que implementa una parte de una transformación.
- MDA.** «*Model-Driven Architecture*». Iniciativa de *OMG* que soporta el paradigma *MDE* con un conjunto de *Meta-Modelos* y lenguajes.
- MDE.** «*Model-Driven Engineering*». Estas siglas denotan una filosofía de desarrollo guiada por modelos, en la cual, el principal objetivo de una fase (análisis, diseño, etc.) es desarrollar los modelos adecuados a partir del refinamiento de los modelos obtenidos en la fase anterior.



- MDD.** «*Model-driven Development*» o «*Model-driven Software Development*».
- MDS.** «*Model-driven Software Development*» o «*Model-driven Development*».
- MDSM.** «*Model Driven Software Modernization*».
- Meta–Modelo.** En el contexto de este trabajo un *Meta–Modelo* es la definición de una gramática (elementos y reglas de construcción) con la que poder elaborar modelos que sean conformes a dicho metamodelo.
- MOF.** «*Meta-Object Facility*» que representa conjunto de interfaces estándares que pueden ser utilizadas para definir y manipular metamodelos interoperables y sus correspondientes modelos.
- MOFM2T.** «*MOF Model to Text Transformation Language*». Ver también *M2T*. que identifica un estándar propuesto por el OMG para definir reglas de derivación para generar una versión textual de modelos.
- Modelo.** En el contexto de este trabajo, un modelo es una representación mediante un lenguaje concreto, con un mayor o menor grado de abstracción, de un aspecto de un sistema de información. Por ejemplo, un modelo de pruebas será la representación de un conjunto de pruebas a realizar sobre el sistema.
- M2M.** «*Model to Model*». Transformaciones desde artefactos de un modelo a otro modelo, basadas en *Meta–Modelos*.
- M2T.** «*Model to Text*»; ver también *MOFM2T*. Transformaciones realizadas desde un modelo a una especificación en un lenguaje textual.
- NDT.** «*Navigational Development Techniques*» que identifica la propuesta metodológica que ha servido como influencia para la realización de este trabajo de tesis y que será fundamental en trabajos futuros posteriores.
- NDTQ-Framework.** Entorno tecnológico del grupo de investigación *IWT2* basado en la metodología *NDT*, basado en *MDE*, estándares, modelos de madurez y buenas prácticas.
- OASIS.** «*Advanced Open Standards for the Information Society*». Organismo de estandarización que, entre otros estándares, gestiona los asociados a *BPEL* y arquitecturas *BPM*.
- OCL.** «*Object Constraint Language*» que identifica un lenguaje para la descripción formal de expresiones en los modelos UML. Su papel principal es el de completar los diferentes artefactos de la notación *UML* con requerimientos formalmente expresados.
- OMG.** «*Object Management Group*», que identifica a un consorcio sin ánimo de lucro formado por diversas compañías y organizaciones. Se dedica al cuidado y el establecimiento de diversos estándares de tecnologías orientadas a objetos, así como a fomentar el uso de tecnología orientada a objetos mediante guías y especificaciones

para las mismas.

**PAIS.** «*Process Aware Information System*». Sistema informático que contempla la gestión de procesos.

**Perfil UML.** Herramienta de extensión del lenguaje *UML* con el que es posible describir un problema de modelado en particular y facilitar la construcción de modelos en ese dominio.

**PERT.** «*Project Evaluation and Review Techniques*». Técnicas de evaluación y control de proyectos ideadas por la Oficina de Proyectos Especiales de la Marina de Guerra del Departamento de Defensa de los EE.UU.

**PIM.** «*Platform Independent Meta-Model*». Término *MDA* para *Meta-Modelo* independiente de la plataforma tecnológica.

**PIP.** «*Process Improvement Process*». Metodología que busca la mejora continua de los procesos existentes en una organización.

**PMI.** Project Management Institute.

**PMBOK.** «*Project Management Body of Knowledge*». Iniciativa del **PMI**: Procedimientos o buenas prácticas para la gestión de proyectos.

**PRINCE2.** «*PRojects IN Controlled Environments 2*». Metodología de gestión de proyectos que manejan una carga importante de variabilidad y de incertidumbre, en entornos controlados.

**Proceso Software.** Se define como un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos necesarios para concebir, desarrollar, desplegar y mantener un producto software.

**PSM.** «*Platform Specific Meta-Model*». Término *MDA* para *Meta-Modelo* específico de una plataforma.

**PRR.** «*Production Rule Representation*». Estándar *OMG* para la definición de reglas de producción.

**QVT.** «*Query/View/Transform*» que identifica el lenguaje de transformaciones de modelo a modelo definido por la *OMG* y utilizado en este trabajo de tesis.

**SASTM.** «*Specific Abstract Syntax Tree Meta-Model*». Estándar *OMG* para la definición de árboles de sintaxis abstracta específicos de un lenguaje.

**SBVR.** «*Semantics of Business Vocabulary and Rules*». Estándar *OMG* para la definición de vocabularios y reglas de negocio.

**Sintaxis abstracta.** En este trabajo de tesis, este término es sinónimo del término metamodelo y se utiliza con el mismo significado.

**Sintaxis concreta.** En este trabajo de tesis, este término es sinónimo del término modelo

y se utiliza con el mismo significado.

**Sistema heredado.** *Legacy system o legacy information system.* Es un sistema o aplicación informática antigua que se sigue utilizando y no es fácil de sustituir.

**SPEM.** «*Software & Systems Process Engineering Meta-Model specification (SPEM 2.0)*».

**SQL.** «*Structured Query Language*». Estándar *ISO* para la definición u manipulación de bases de datos relacionales.

**TI.** Tecnologías de la información (Sector **TI**: organizaciones dedicadas al negocio del software).

**TIC.** Tecnologías de la información y comunicaciones (Sector **TIC**: abarca a todo tipo de organización dedicada a software, hardware o de las comunicaciones).

**Transformación (O transformación MDE).** En el contexto de este trabajo una transformación es una especificación de cómo construir un modelo conforme con un metamodelo, tomando como entrada otro modelo conforme a otro metamodelo, que puede ser el mismo o distinto.

**UML.** Acrónimo de «*Unified Modelling Language*» y que identifica a un lenguaje estándar de modelado gráfico, utilizado en este trabajo principalmente para definir los metamodelos y modelos presentados.

**WfMC.** «*Workflow Management Coalition*». Organismo de estandarización en el mundo *BPM*.

**WS-BPEL.** «*Web Services Business Process Execution Language*». Lenguaje estándar de *OASIS* para la ejecución de procesos.

**XPDL.** «*XML Process Definition Language*». Lenguaje del *WfMC* para la definición e intercambio de modelos de procesos.



## **ANEXO II: META-MODELOS DE TAREAS EN LEGACY DATABASES (LDB)**



### Anexo II.a: LDB MS Access/MS Project

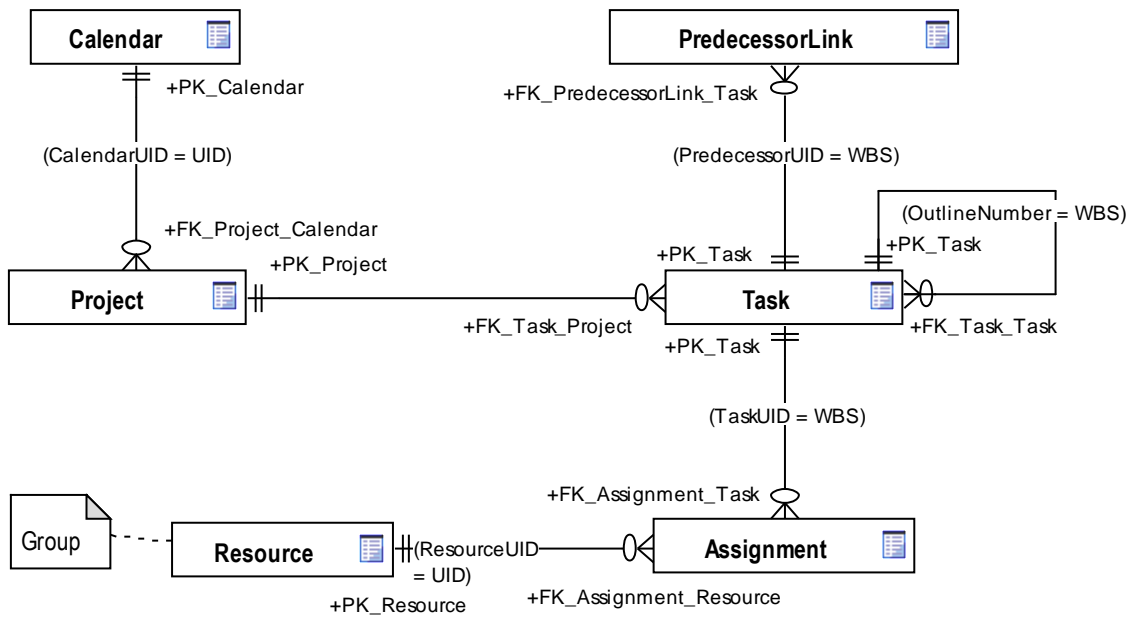


Figura. AII.1. Meta-Modelo de tareas de MS Project (MS Access)

### Anexo II.b: LDB SQL\*Server/MS Project Server Published instance

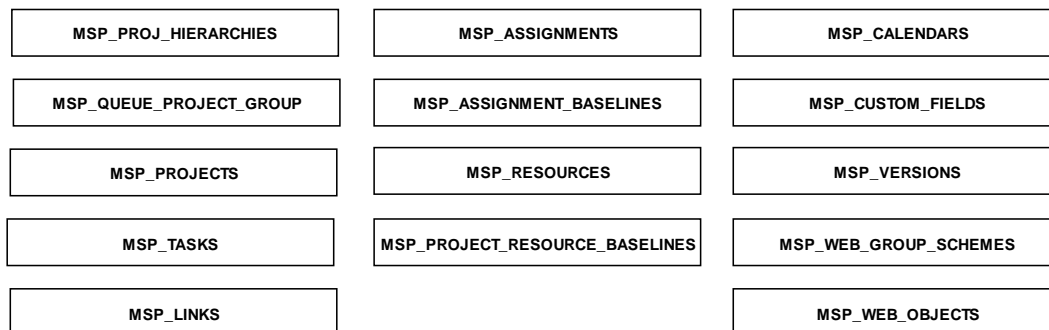


Figura. AII.2. Meta-Modelo de tareas de MS Project Server (SQL\*Server™). Tablas principales

### Anexo II.c: LDB PostgreSQL/ECM Alfresco

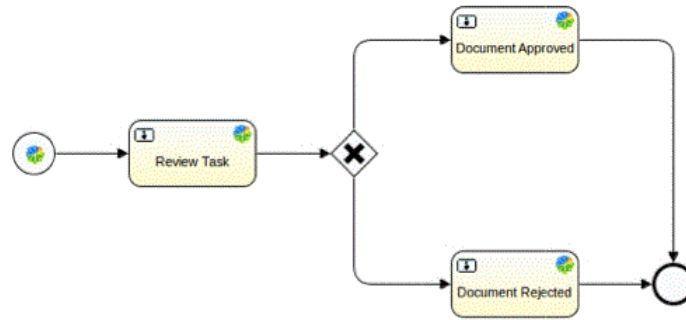


Figura. AII.3. Workflow básico de Alfresco

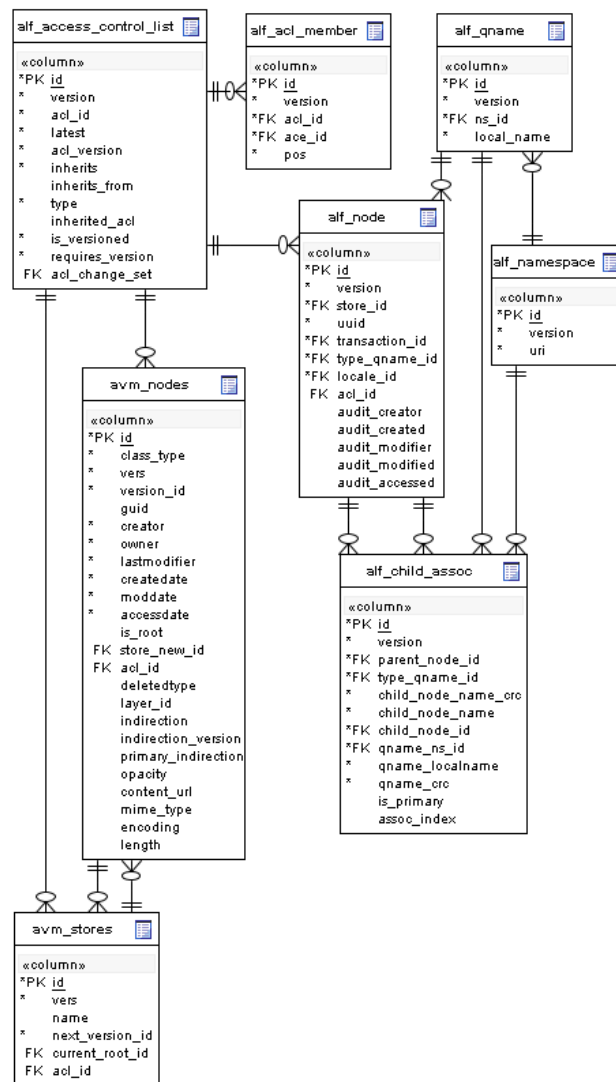


Figura. AII.4. Meta-Modelo del ECM:Alfresco (PostgreSQL)





### Anexo II.d: LDB MySQL/RedMine

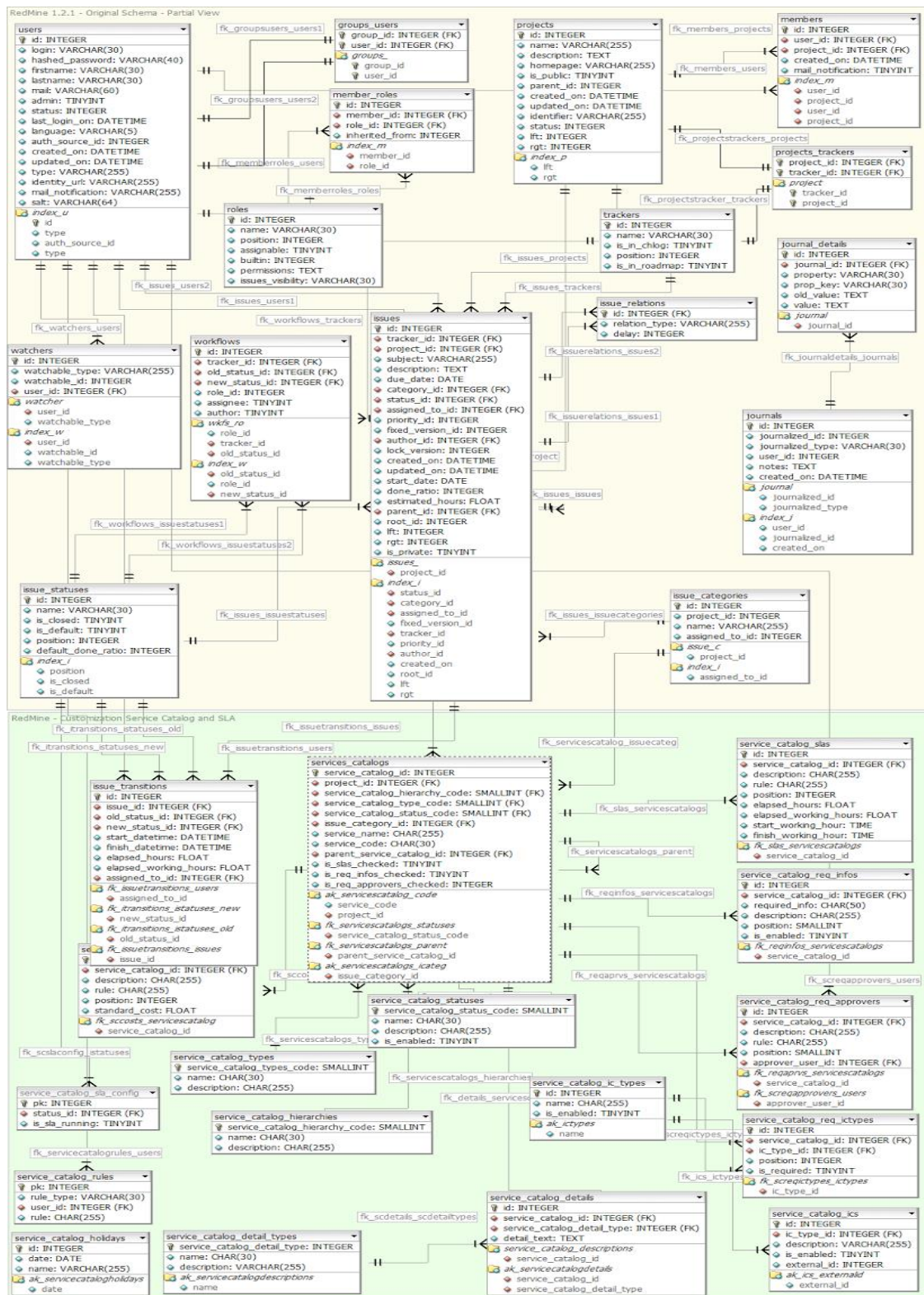


Figura. AII.6. Meta-Modelo de tareas de RedMine

# **ANEXO III: META-MODELOS RELACIONALES SQL**



### Anexo III.a: Meta-Modelos relacionales OMG:IMM

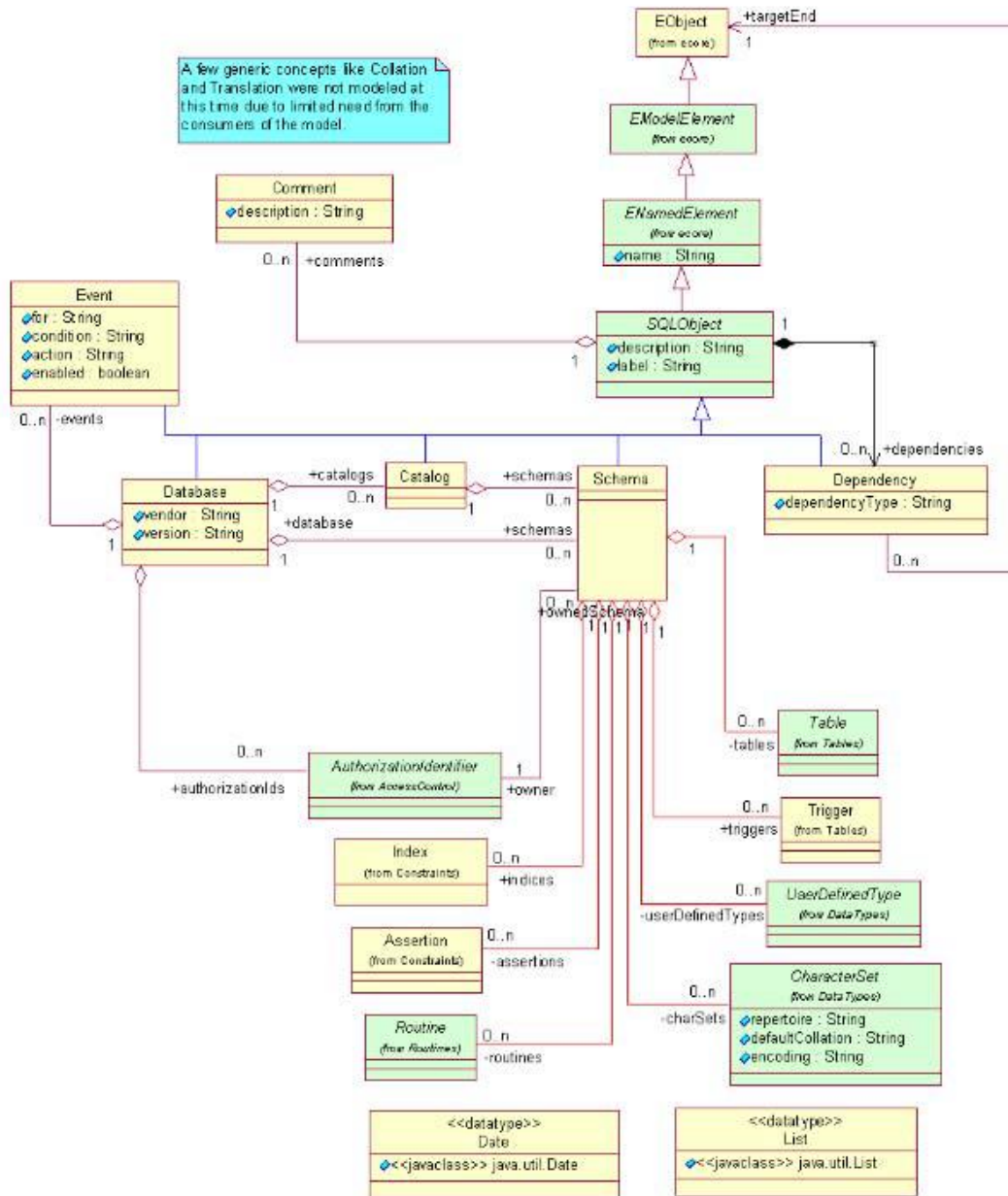


Figura. III.1. Meta-Modelo OMG:IMM PSM Relacional. Main Schemata

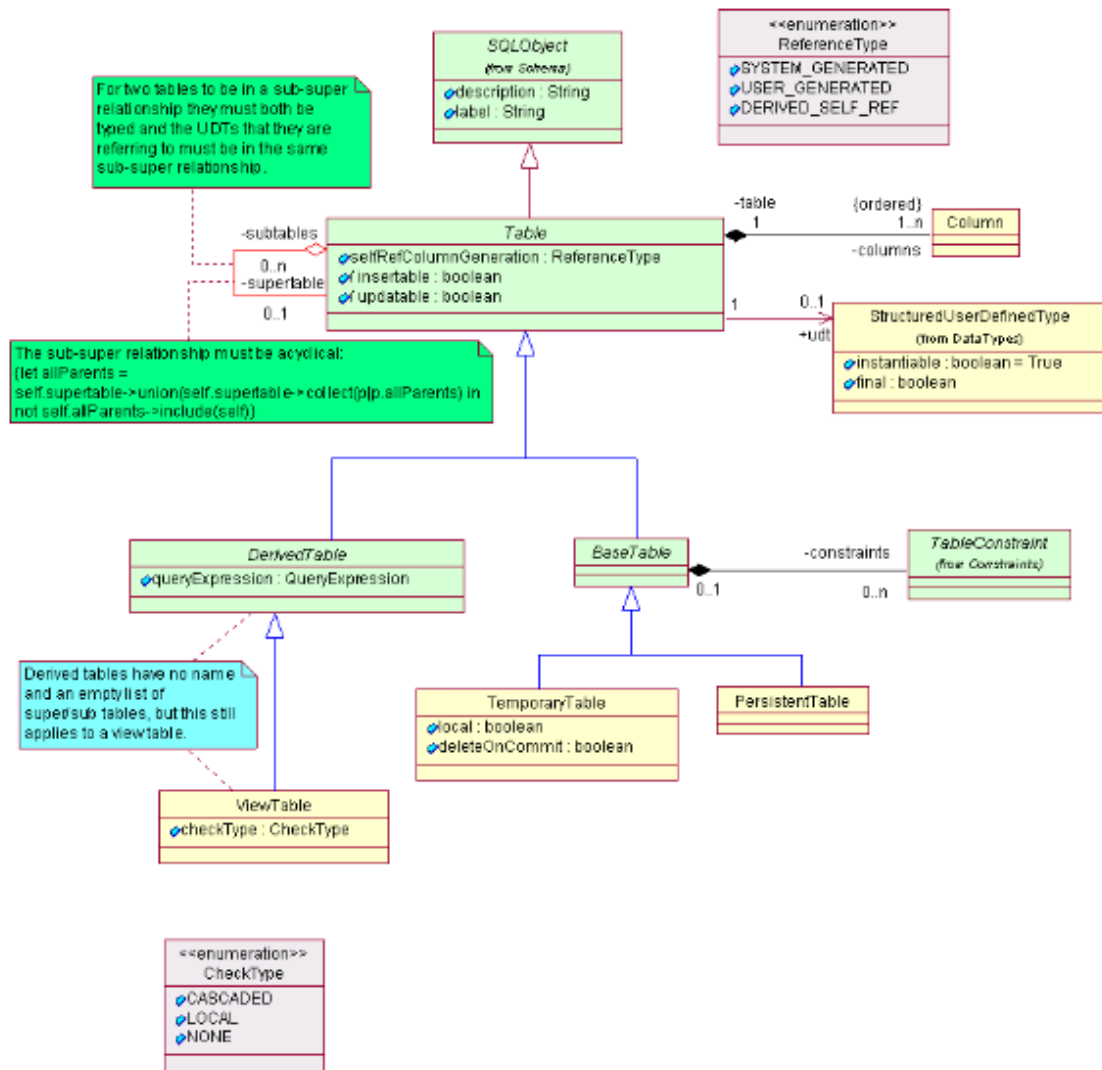


Figura. III.2. Meta-Modelo OMG:IMM PSM Relacional. Tables

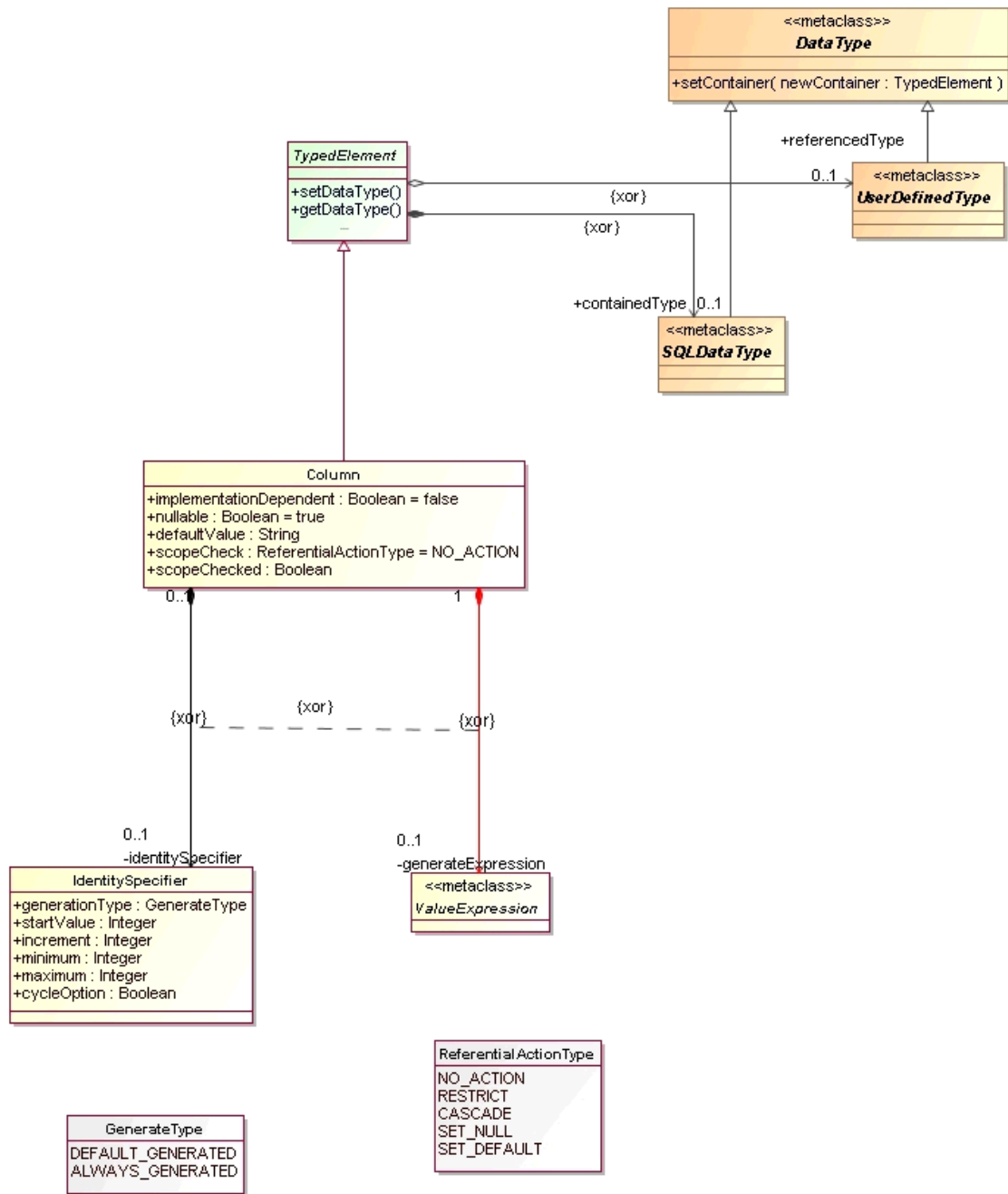


Figura. III.3. Meta-Modelo OMG:IMM PSM Relacional. Columns

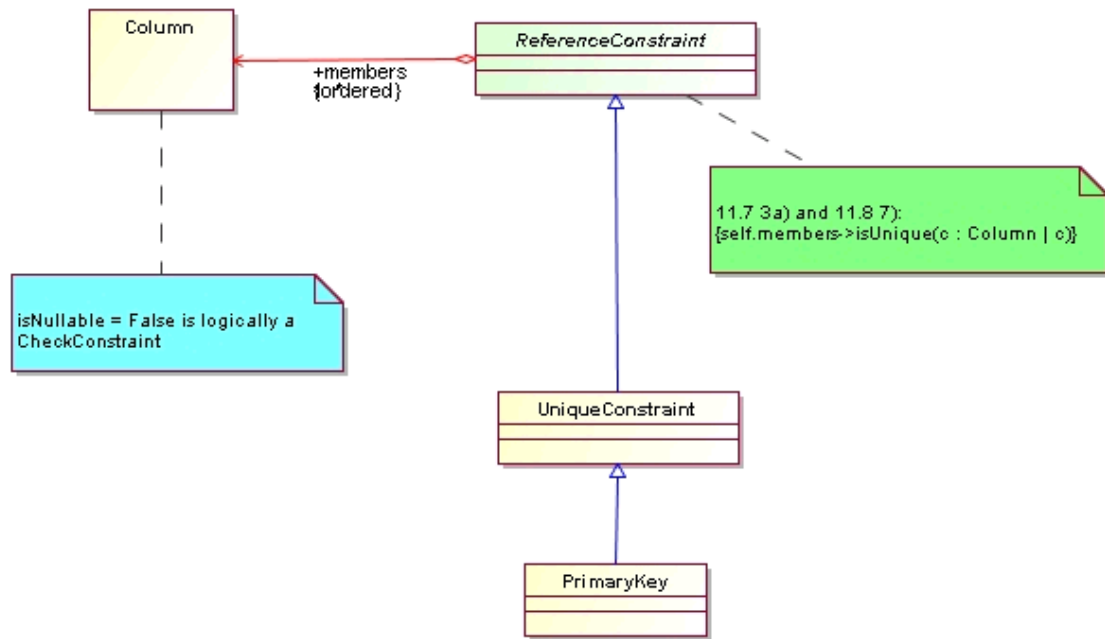


Figura. III.4. Meta-Modelo OMG:IMM PSM Relacional. Constraint Unique Keys



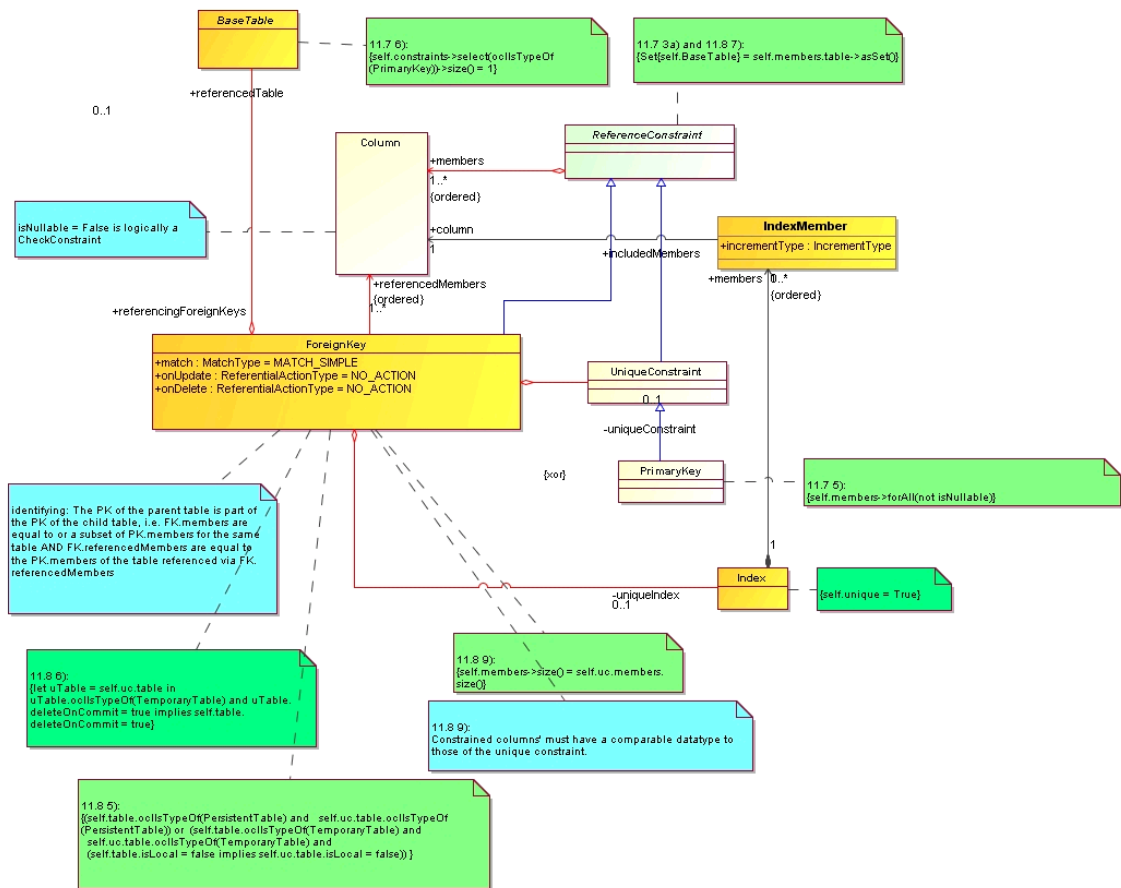


Figura. III.5. Meta-Modelo OMG:IMM PSM Relacional. Constraint Foreign Keys

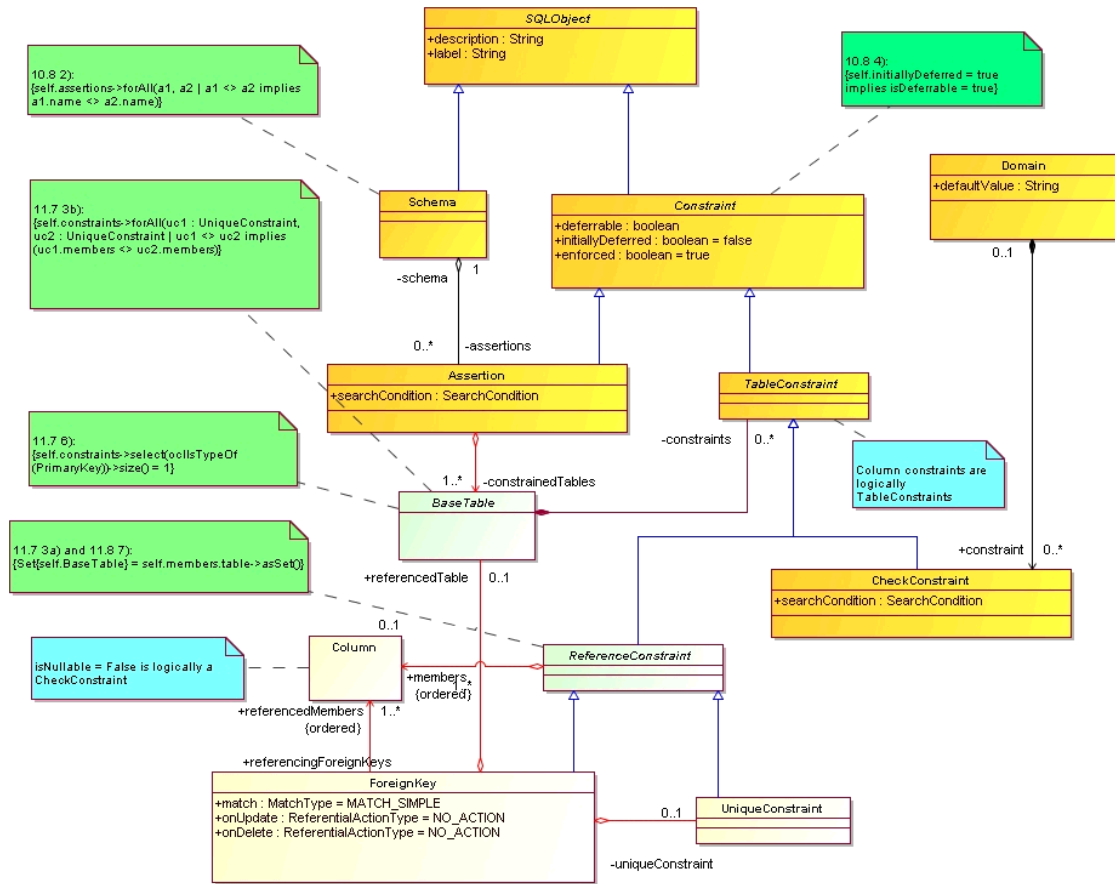


Figura. III.6. Meta-Modelo OMG:IMM PSM Relacional. Constraint Other Constraints

### Anexo III.b: Extensiones a Meta-Modelos relacionales. Restricciones

Este Meta-Modelo [Arevalo et al. 2013] se ha utilizado para la extracción de ECA Rules desde bases de datos relacionales.

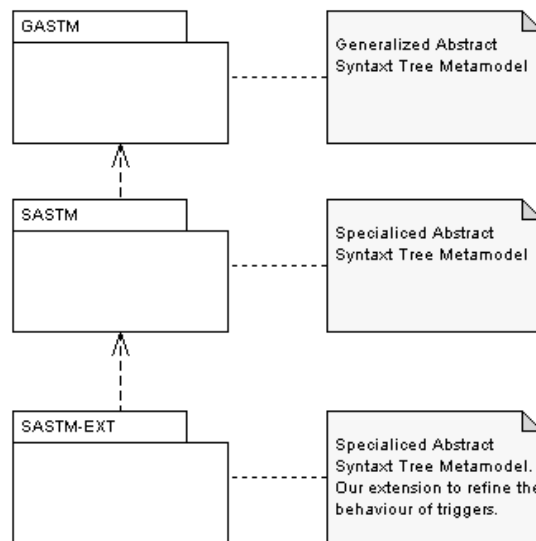


Figura. III.7. Meta-Modelo PSM Relacional. Paquetes

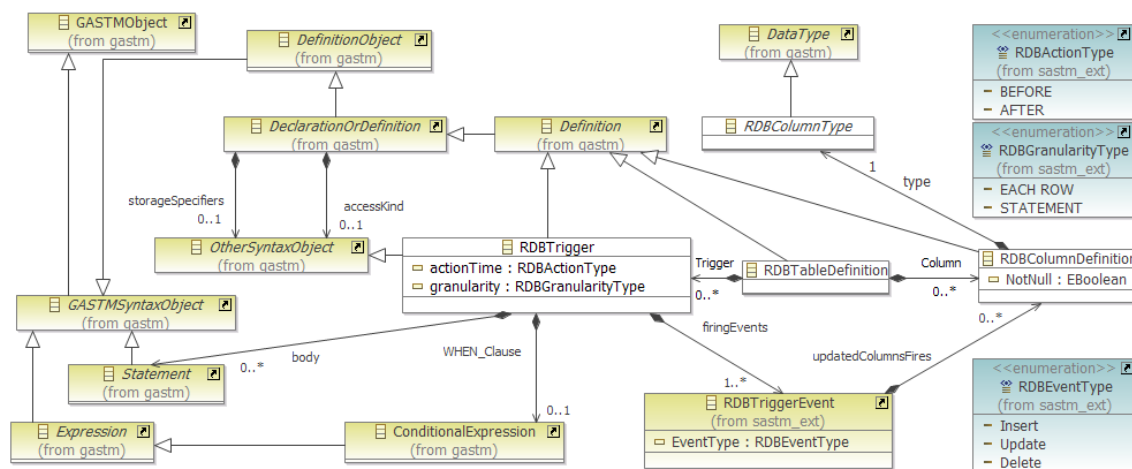


Figura. III.8. Meta-Modelo PSM Relacional. Extensiones para triggers



## **ANEXO IV: PROYECTO AQUA-WS**





### Anexo IV.a: Planificación del proyecto (MS Project)

#### Anexo IV.a1. Planificación general del Proyecto (Tareas generales y subsistemas)

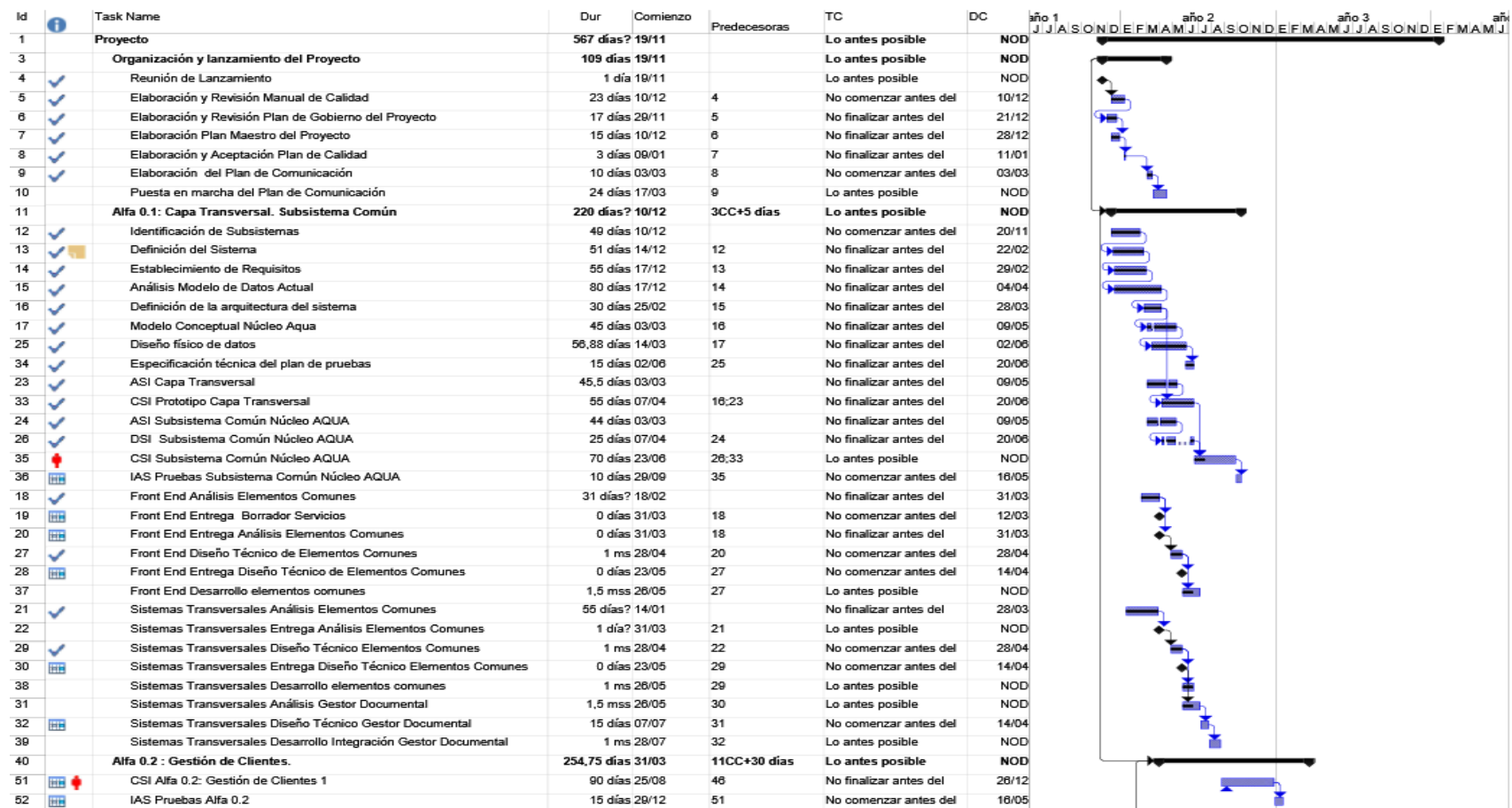
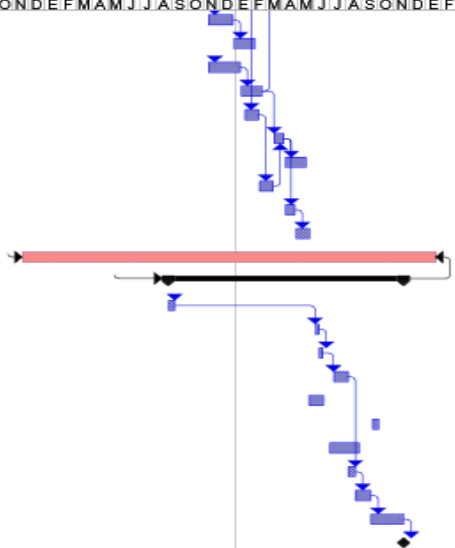
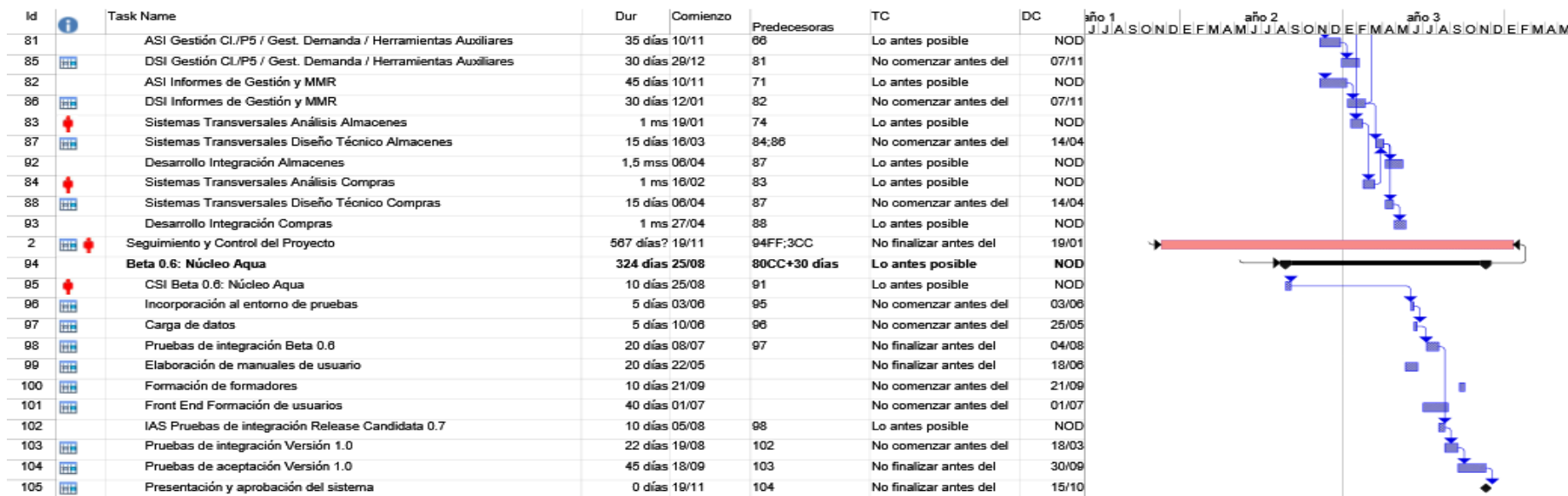


Figura. IV.1. Planificación General Proyecto AQUA-WS por subsistemas







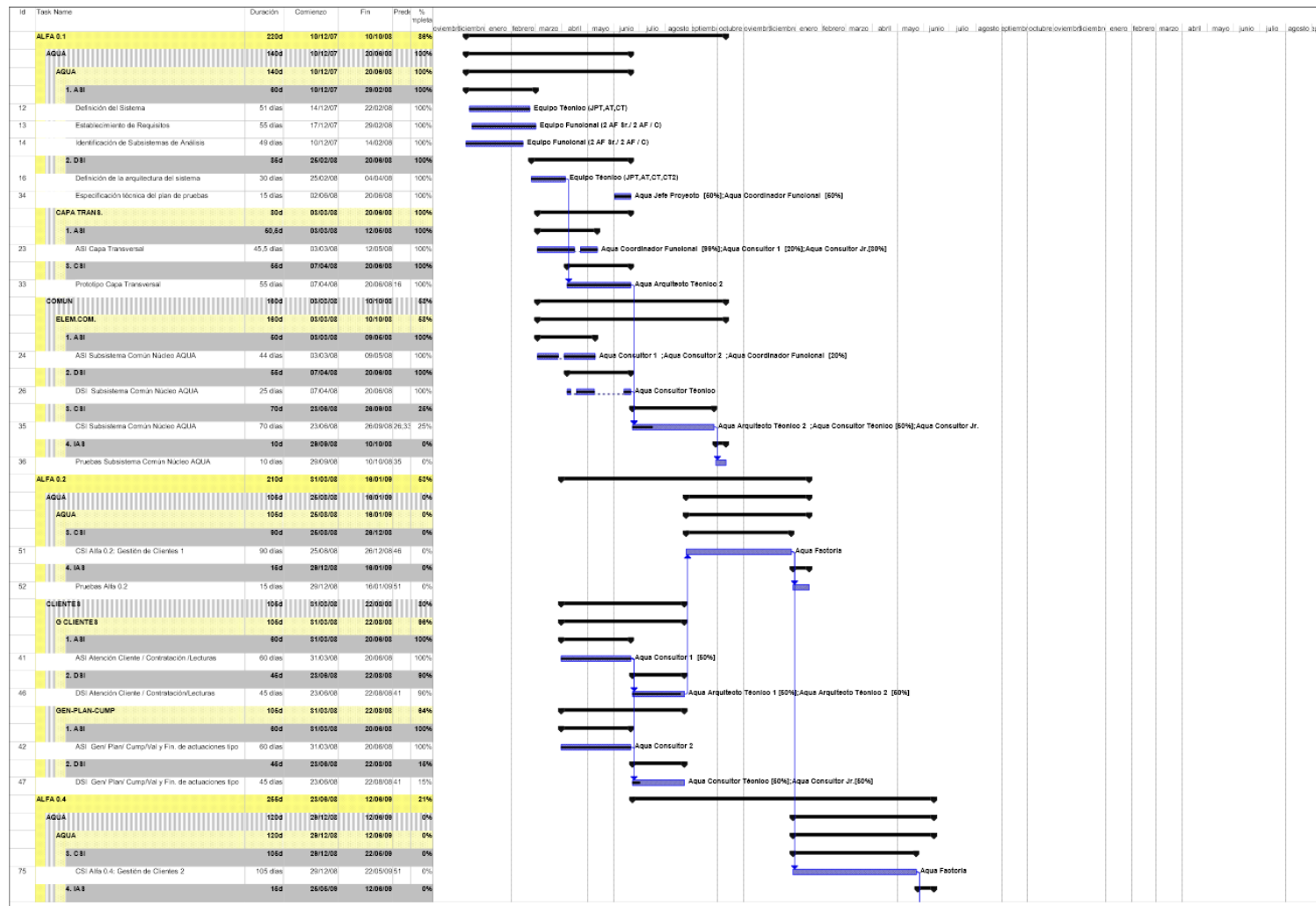
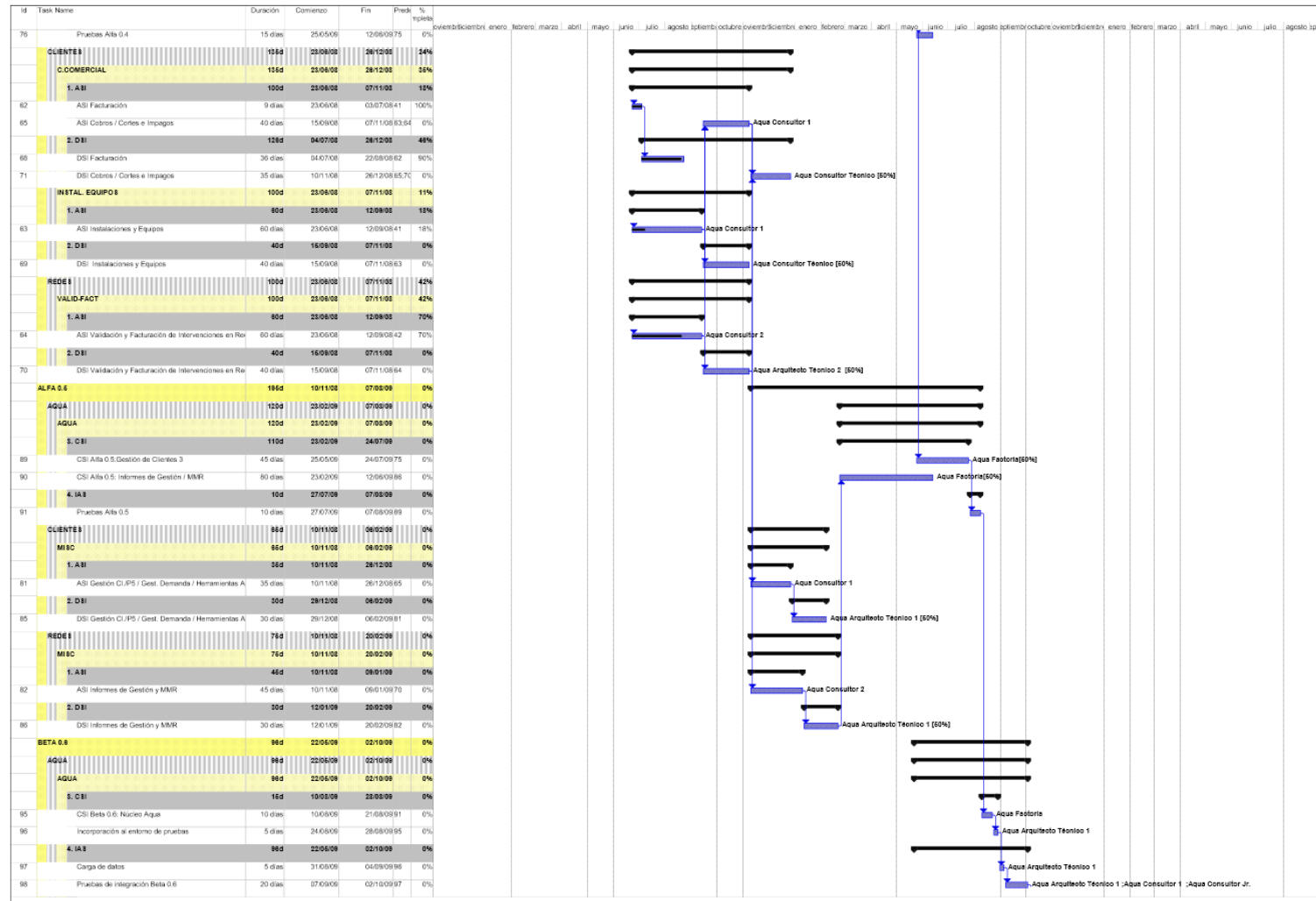


Figura. IV.2. Planificación General Proyecto AQUA-WS por fases





**Anexo IV.a2. Planificación Control Calidad Tarea (64): (ASI) Instalaciones y equipos**

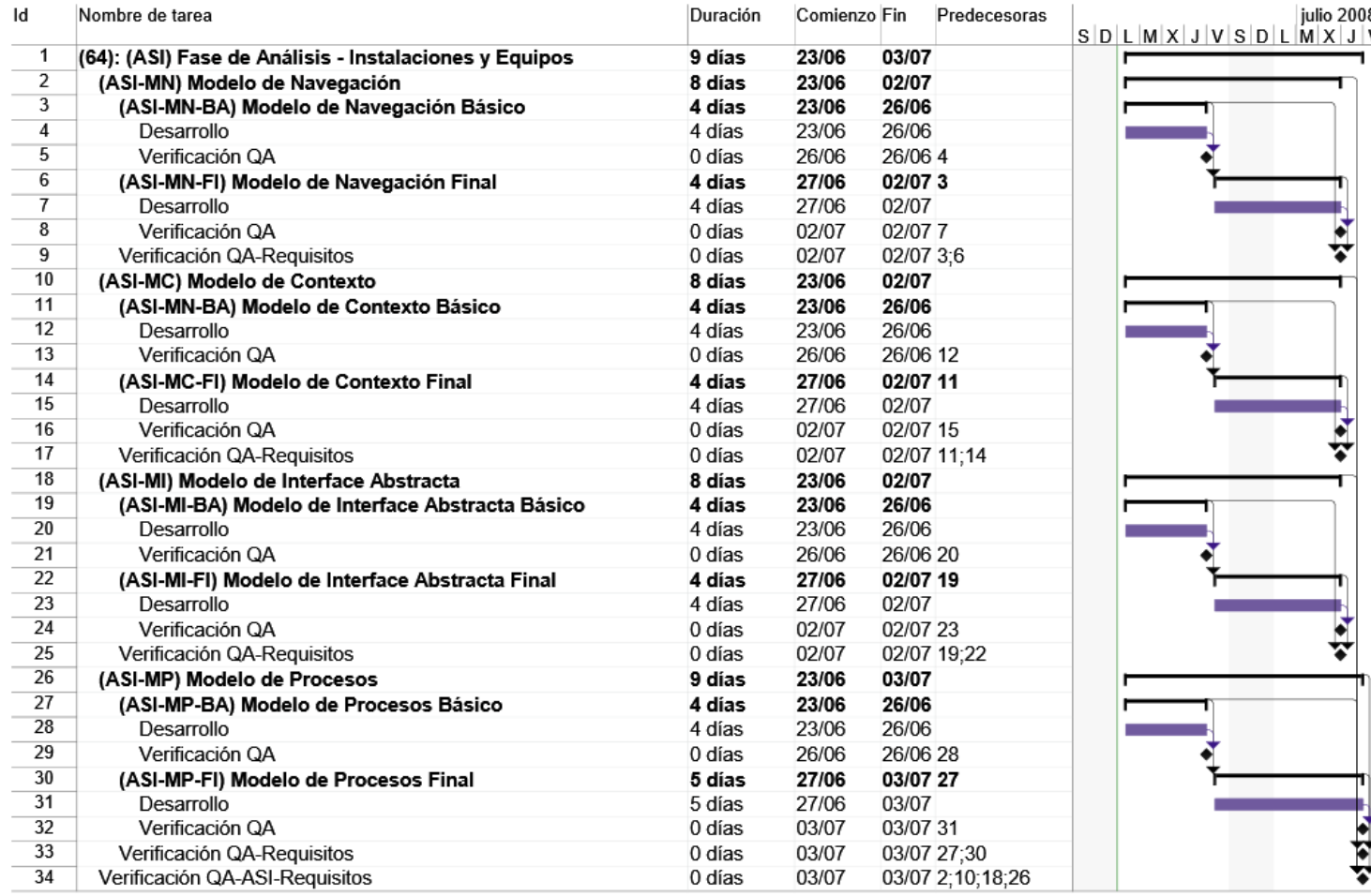


Figura. IV.3. Planificación Control Calidad (64) (ASI) Instalaciones y Equipos AQUA-WS

**Anexo IV.a3. Patrón de Planificación para Control Calidad de la fase NDT ASI**

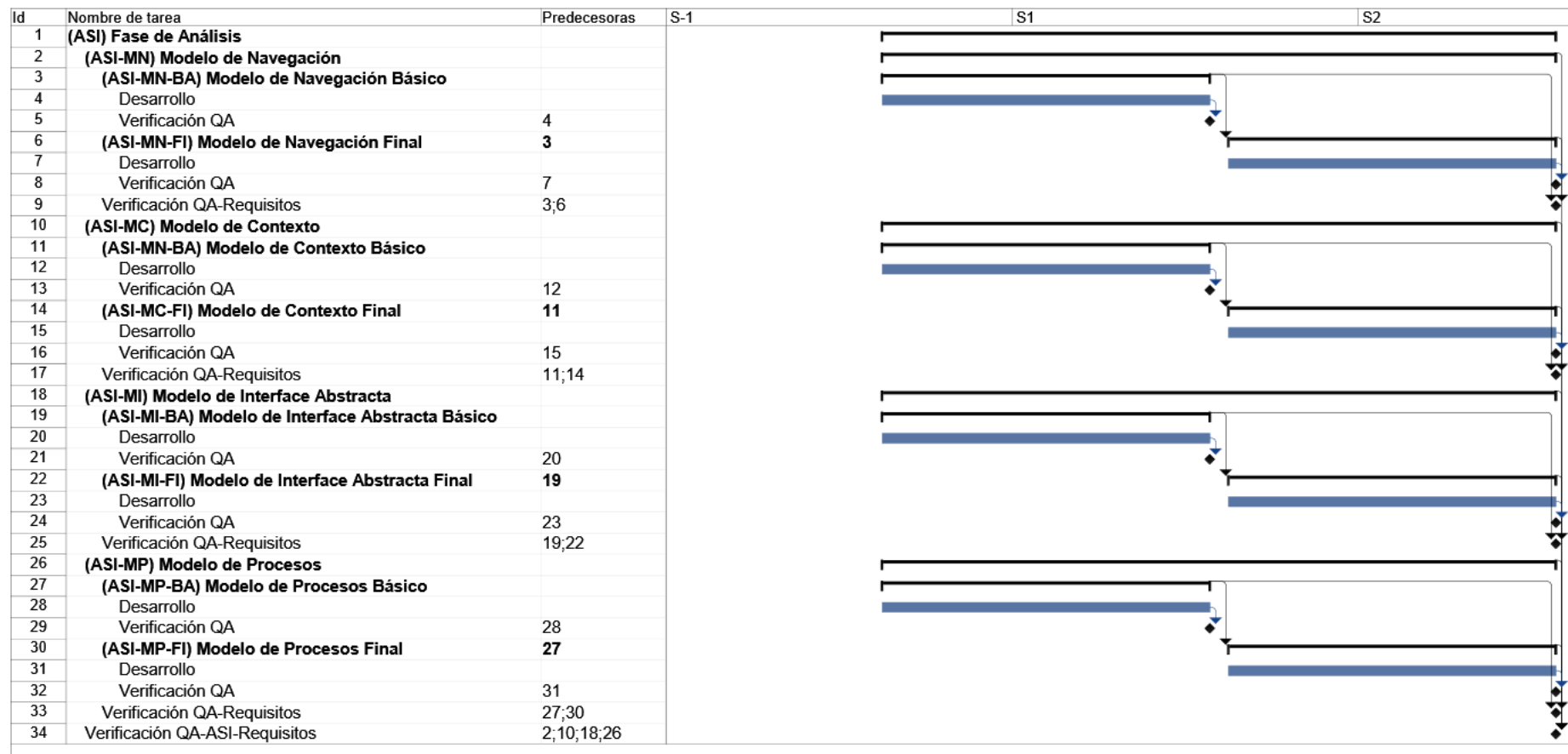


Figura. IV.4. Patrón de Planificación para Control Calidad de la fase NDT (ASI)

### Anexo IV.b: Diagrama Relacional BD SQL\*Server Published-AQUA

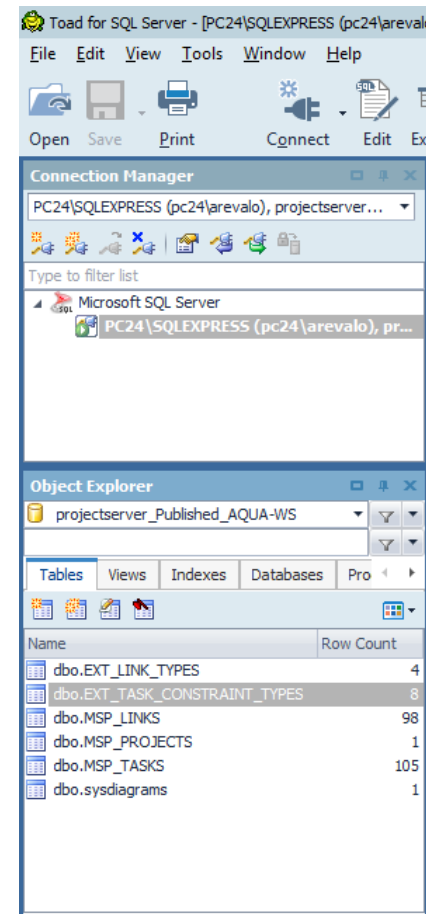
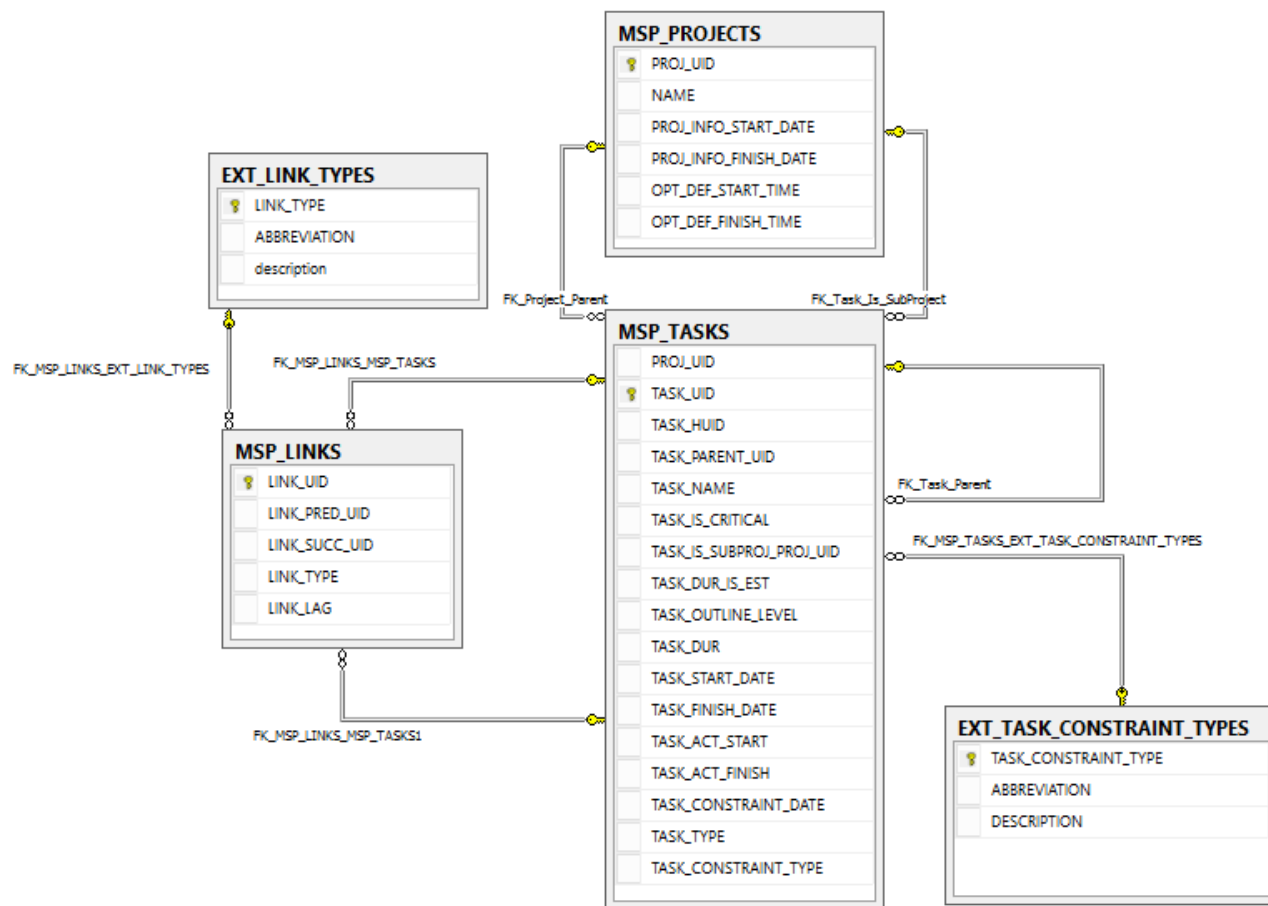
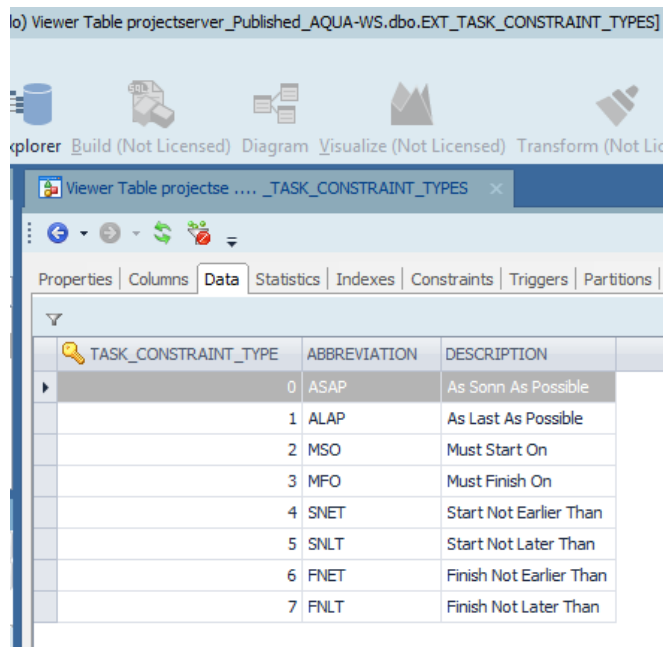


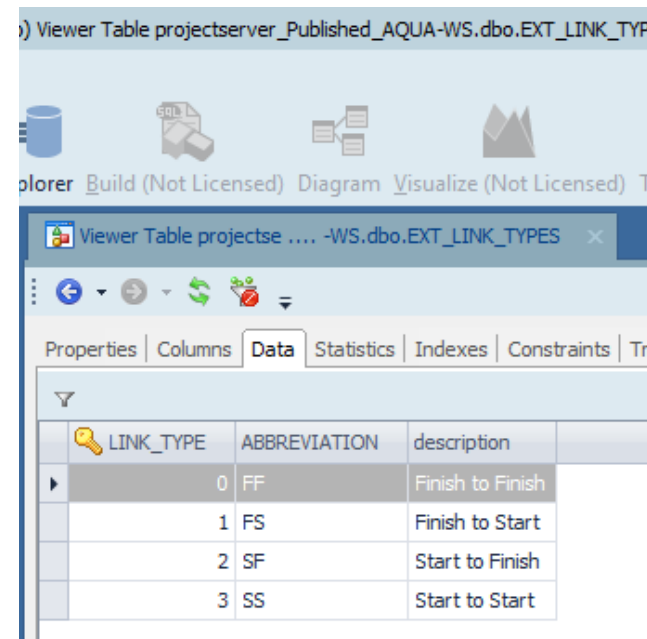
Figura. IV.5. Diagrama Relacional BD SQL\*Server Published-AQUA

### Anexo IV.c: Tablas BD SQL\*Server Published-AQUA



TASK_CONSTRAINT_TYPE	ABBREVIATION	DESCRIPTION
0	ASAP	As Soon As Possible
1	ALAP	As Last As Possible
2	MSO	Must Start On
3	MFO	Must Finish On
4	SNET	Start Not Earlier Than
5	SNLT	Start Not Later Than
6	FNET	Finish Not Earlier Than
7	FNLT	Finish Not Later Than

Figura. IV.6. Tabla Ext\_Task\_Constraint\_Types (AQUA-WS)



LINK_TYPE	ABBREVIATION	description
0	FF	Finish to Finish
1	FS	Finish to Start
2	SF	Start to Finish
3	SS	Start to Start

Figura. IV.7. Tabla Ext\_Link\_Types (AQUA-WS)



PROJ_ID	NAME	PROJ_INFO_START_DATE	PROJ_INFO_FINISH_DATE	OPT_DEF_START_TIME	OPT_DEF_FINISH_TIME
1	AQUA-WS	19/11/2007 0:00:00	19/01/2010 0:00:00	19/11/2007 0:00:00	{null}
64	ASI Instalaciones y Equipos	23/06/2008 0:00:00	03/07/2008 0:00:00	23/06/2008 0:00:00	{null}
10001	Patrón Fase NDT (ASI) Análisis del Sistema	{null}	{null}	{null}	

Figura. IV.8. Tabla Msp\_Projects (AQUA-WS)

PROJ_UID	TASK_UID	TASK_HUID	TASK_PARENT_UID	TASK_NAME	TASK_IS_CRITICAL	TASK_IS_SUBPROJ_PROJ_UID	TASK_DUR_IS_EST
1	1	1	{null}	Proyecto	<input type="checkbox"/>	{null}	<input checked="" type="checkbox"/>
1	2	1.1	1	Seguimiento y Control del Proyecto	<input type="checkbox"/>	{null}	<input checked="" type="checkbox"/>
1	3	1.2	1	Organización y lanzamiento del Proyecto	<input type="checkbox"/>	{null}	<input type="checkbox"/>
1	4	1.2.1	3	Reunión de Lanzamiento	<input type="checkbox"/>	{null}	<input type="checkbox"/>
1	5	1.2.2	3	Elaboración y Revisión Manual de Calidad	<input type="checkbox"/>	{null}	<input type="checkbox"/>
1	6	1.2.3	3	Elaboración y Revisión Plan de Gobierno del Pro...	<input type="checkbox"/>	{null}	<input type="checkbox"/>
1	7	1.2.4	3	Elaboración Plan Maestro del Proyecto	<input type="checkbox"/>	{null}	<input type="checkbox"/>
1	8	1.2.5	3	Elaboración y Aceptación Plan de Calidad	<input type="checkbox"/>	{null}	<input type="checkbox"/>
1	9	1.2.6	3	Elaboración del Plan de Comunicación	<input type="checkbox"/>	{null}	<input type="checkbox"/>
1	10	1.2.7	3	Puesta en marcha del Plan de Comunicación	<input type="checkbox"/>	{null}	<input type="checkbox"/>
1	11	1.3	1	Alfa 0. 1: Capa Transversal. Subsistema Común	<input type="checkbox"/>	{null}	<input checked="" type="checkbox"/>
1	12	1.3.1	11	Identificación de Subsistemas	<input type="checkbox"/>	{null}	<input type="checkbox"/>
1	13	1.3.2	11	Definición del Sistema	<input type="checkbox"/>	{null}	<input type="checkbox"/>
1	14	1.3.3	11	Establecimiento de Requisitos	<input type="checkbox"/>	{null}	<input type="checkbox"/>
1	15	1.3.4	11	Análisis Modelo de Datos Actual	<input type="checkbox"/>	{null}	<input type="checkbox"/>
1	16	1.3.5	11	Definición de la arquitectura del sistema	<input type="checkbox"/>	{null}	<input type="checkbox"/>
1	17	1.3.6	11	Modelo Conceptual Núcleo Aqua	<input type="checkbox"/>	{null}	<input type="checkbox"/>
1	18	1.3.7	11	Front End Análisis Elementos Comunes	<input type="checkbox"/>	{null}	<input checked="" type="checkbox"/>
1	19	1.3.8	11	Front End Entrega Borrador Servicios	<input type="checkbox"/>	{null}	<input type="checkbox"/>

Figura. IV.9. Tabla Msp\_Tasks (AQUA-WS)

**Tabla. IV.1.** Tabla SQL\*Server Msp\_Tasks (AQUA-WS)

PROJ_UID	TASK_UID	TASK_HUID	TASK_PAREN_T_UID	TASK_NAME	TASK_DUR	TASK_OUTLINE_LEVEL	TASK_RT_EST	TASK_ST_ART_DAT	TASK_FINISH_DAT	TASK_ACT_T_START	TASK_ACT_FINISH	TASK_CONSTRRAINT_DATE	TASK_CONSTRAINT_TYPE
1	1	1		Proyecto	567,00	1	True	19-11-07	19-1-10	19-11-07			0
1	2	1.1	1	Seguimiento y Control del Proyecto	567,00	2	True	19-11-07	19-1-10	19-11-07		19-1-10	6
1	3	1.2	1	Organización y lanzamiento del Proyecto	109,00	2	False	19-11-07	17-4-08	19-11-07			0
1	4	1.2.1	3	Reunión de Lanzamiento	1,00	3	False	19-11-07	19-11-07	19-11-07	19-11-07		0
1	5	1.2.2	3	Elaboración y Revisión Manual de Calidad	23,00	3	False	10-12-07	9-1-08	10-12-07	9-1-08	10-12-07	6
1	6	1.2.3	3	Elaboración y Revisión Plan de Gobierno del Proyecto	17,00	3	False	29-11-07	21-12-07	29-11-07	21-12-07	21-12-07	6
1	7	1.2.4	3	Elaboración Plan Maestro del Proyecto	15,00	3	False	10-12-07	28-12-07	10-12-07	28-12-07	28-12-07	6
1	8	1.2.5	3	Elaboración y Aceptación Plan de Calidad	3,00	3	False	9-1-08	11-1-08	9-1-08	11-1-08	11-1-08	6
1	9	1.2.6	3	Elaboración del Plan de Comunicación	10,00	3	False	3-3-08	14-3-08	3-3-08	14-3-08	3-3-08	4
1	10	1.2.7	3	Puesta en marcha del Plan de Comunicación	24,00	3	False	17-3-08	17-4-08				0
1	11	1.3	1	Alfa 0.1: Capa Transversal. Subsistema Común	220,00	2	True	10-12-07	10-10-08	10-12-07			0
1	12	1.3.1	11	Identificación de Subsistemas	49,00	3	False	10-12-07	14-2-08	10-12-07	14-2-08	20-11-07	4
1	13	1.3.2	11	Definición del Sistema	51,00	3	False	14-12-07	22-2-08	14-12-07	22-2-08	22-2-08	6
1	14	1.3.3	11	Establecimiento de Requisitos	55,00	3	False	17-12-07	29-2-08	17-12-07	29-2-08	29-2-08	6
1	15	1.3.4	11	Análisis Modelo de Datos Actual	80,00	3	False	17-12-07	4-4-08	17-12-07	4-4-08	4-4-08	6
1	16	1.3.5	11	Definición de la arquitectura del sistema	30,00	3	False	25-2-08	4-4-08	25-2-08	4-4-08	28-3-08	6
1	17	1.3.6	11	Modelo Conceptual Núcleo Aqua	45,00	3	False	3-3-08	9-5-08	3-3-08	9-5-08	9-5-08	6
1	18	1.3.7	11	Front End Análisis Elementos Comunes	31,00	3	True	18-2-08	31-3-08	18-2-08	31-3-08	31-3-08	6
1	19	1.3.8	11	Front End Entrega Borrador Servicios	0	3	False	31-3-08	31-3-08			12-3-08	4
1	20	1.3.9	11	Front End Entrega Análisis Elementos Comunes	0	3	False	31-3-08	31-3-08			31-3-08	6
1	21	1.3.10	11	Sistemas Transversales Análisis Elementos Comunes	55,00	3	True	14-1-08	28-3-08	14-1-08	28-3-08	28-3-08	6
1	22	1.3.11	11	Sistemas Transversales Entrega Análisis Elementos Comunes	1,00	3	True	31-3-08	31-3-08				0
1	23	1.3.12	11	ASI Capa Transversal	45,50	3	False	3-3-08	12-5-08	3-3-08	12-5-08	9-5-08	6
1	24	1.3.13	11	ASI Subsistema Común Núcleo AQUA	44,00	3	False	3-3-08	9-5-08	3-3-08	9-5-08	9-5-08	6
1	25	1.3.14	11	Diseño físico de datos	56,88	3	False	14-3-08	2-6-08	14-3-08	2-6-08	2-6-08	6
1	26	1.3.15	11	DSI Subsistema Común Núcleo AQUA	25,00	3	False	7-4-08	20-6-08	7-4-08	20-6-08	20-6-08	6
1	27	1.3.16	11	Front End Diseño Técnico de Elementos Comunes	30,00	3	False	28-4-08	23-5-08	28-4-08	23-5-08	28-4-08	4
1	28	1.3.17	11	Front End Entrega Diseño Técnico de Elementos Comunes	0	3	False	23-5-08	23-5-08			14-4-08	4
1	29	1.3.18	11	Sistemas Transversales Diseño Técnico Elementos Comunes	30,00	3	False	28-4-08	23-5-08	28-4-08	23-5-08	28-4-08	4
1	30	1.3.19	11	Sistemas Transversales Entrega Diseño Técnico Elementos Comunes	0	3	False	23-5-08	23-5-08			14-4-08	4

**Tabla. IV.1.** Tabla SQL\*Server Msp\_Tasks (AQUA-WS)

PROJ_UID	TASK_UID	TASK_HUID	TASK_PAREN_T_UID	TASK_NAME	TASK_DURATION_EST	TASK_OUTLINE_LEVEL	TASK_DURATION	TASK_START_DATE	TASK_FINISH_DATE	TASK_ACT_T_START	TASK_ACT_FINISH	TASK_CONSTRAINT_DATE	TASK_CONSTRAINT_TYPE
1	31	1.3.20	11	Sistemas Transversales Análisis Gestor Documental	False	3	45,00	26-5-08	4-7-08	26-5-08			0
1	32	1.3.21	11	Sistemas Transversales Diseño Técnico Gestor Documental	False	3	15,00	7-7-08	25-7-08			14-4-08	4
1	33	1.3.22	11	CSI Prototipo Capa Transversal	False	3	55,00	7-4-08	20-6-08	7-4-08	20-6-08	20-6-08	6
1	34	1.3.23	11	Especificación técnica del plan de pruebas	False	3	15,00	2-6-08	20-6-08	2-6-08	20-6-08	20-6-08	6
1	35	1.3.24	11	CSI Subsistema Común Núcleo AQUA	False	3	70,00	23-6-08	26-9-08	23-6-08			0
1	36	1.3.25	11	IAS Pruebas Subsistema Común Núcleo AQUA	False	3	10,00	29-9-08	10-10-08			16-5-08	4
1	37	1.3.26	11	Front End Desarrollo elementos comunes	False	3	45,00	26-5-08	4-7-08	26-5-08			0
1	38	1.3.27	11	Sistemas Transversales Desarrollo elementos comunes	False	3	30,00	26-5-08	20-6-08	26-5-08			0
1	39	1.3.28	11	Sistemas Transversales Desarrollo Integración Gestor Documental	False	3	30,00	28-7-08	22-8-08				0
1	40	1.4	1	Alfa 0.2 : Gestión de Clientes.	False	2	254,75	31-3-08	20-3-09	31-3-08			0
1	41	1.4.1	40	ASI Atención Cliente / Contratación /Lecturas	False	3	60,00	31-3-08	20-6-08	31-3-08	20-6-08	20-6-08	6
1	42	1.4.2	40	ASI Gen/ Plan/ Cump/Val y Fin. de actuaciones tipo	False	3	60,00	31-3-08	20-6-08	31-3-08	20-6-08	20-6-08	6
1	43	1.4.3	40	Front End Análisis Atención Ciudadano y Empleado	False	3	20,00	20-6-08	17-7-08			20-6-08	4
1	44	1.4.4	40	Sistemas Transversales Análisis FICO	False	3	60,00	29-12-08	20-2-09				0
1	45	1.4.5	40	Sistemas Transversales Análisis Recursos Humanos	False	3	30,00	23-2-09	20-3-09			17-10-08	6
1	46	1.4.6	40	DSI Atención Cliente / Contratación/Lecturas	False	3	45,00	23-6-08	22-8-08	23-6-08		18-7-08	6
1	47	1.4.7	40	DSI Gen/ Plan/ Cump/Val y Fin. de actuaciones tipo	False	3	45,00	23-6-08	22-8-08	23-6-08		22-8-08	6
1	48	1.4.8	40	Front End Diseño Técnico 1	False	3	18,00	25-8-08	9-9-08				0
1	49	1.4.9	40	Sistemas Transversales Diseño Técnico FICO	False	3	15,00	20-10-08	7-11-08			20-10-08	4
1	50	1.4.10	40	Sistemas Transversales Diseño Técnico Recursos Humanos	False	3	15,00	10-11-08	28-11-08			10-11-08	4
1	51	1.4.11	40	CSI Alfa 0.2: Gestión de Clientes 1	False	3	90,00	25-8-08	26-12-08			26-12-08	6
1	52	1.4.12	40	IAS Pruebas Alfa 0.2	False	3	15,00	29-12-08	16-1-09			16-5-08	4
1	53	1.4.13	40	Front End Desarrollo CRM Atención Ciudadano y Empleado 1	False	3	40,00	10-9-08	15-10-08				0
1	54	1.4.14	40	Sistemas Transversales Desarrollo Integración FICO	False	3	45,00	10-11-08	19-12-08			10-11-08	4
1	55	1.4.15	40	Sistemas Transversales Desarrollo Integración Recursos Humanos	False	3	30,00	1-12-08	26-12-08				0
1	56	1.5	1	Alfa 0.3: Gestión de Obras y Proyectos	False	2	173,00	17-3-08	12-11-08	17-3-08			0
1	57	1.5.1	56	ASI Gestión de Obras y Proyectos	False	3	45,00	17-3-08	16-5-08	17-3-08	16-5-08		0
1	58	1.5.2	56	DSI Gestión de Obras y Proyectos	False	3	48,00	19-5-08	23-7-08	19-5-08	23-7-08		0
1	59	1.5.3	56	CSI Gestión de Obras y Proyectos	False	3	90,00	26-6-08	29-10-08				0
1	60	1.5.4	56	Pruebas Gestión de Obras y Proyectos	False	3	10,00	30-10-08	12-11-08			18-6-08	4

**Tabla. IV.1.** Tabla SQL\*Server Msp\_Tasks (AQUA-WS)

PROJ_UID	TASK_UID	TASK_HUID	TASK_PAREN_T_UID	TASK_NAME	TASK_DUR	TASK_OUTLINE_LEVEL	TASK_DURATION	TASK_START_DATE	TASK_FINISH_DATE	TASK_ACT_T_START	TASK_ACT_FINISH	TASK_CONSTRAINT_DATE	TASK_CONSTRAINT_TYPE
1	61	1.6	1	Alfa 0.4: Clientes: Intervenciones en Red	229,50	2	229,50	28-4-08	13-3-09	28-4-08			0
1	62	1.6.1	61	ASI Facturación	9,00	3	9,00	23-6-08	3-7-08	23-6-08	3-7-08	3-7-08	6
1	63	1.6.2	61	DSI Facturación	36,00	3	36,00	4-7-08	22-8-08	4-7-08			0
1	64	1.6.3	61	ASI Instalaciones y Equipos	60,00	3	60,00	23-6-08	12-9-08	23-6-08		12-9-08	6
1	65	1.6.4	61	ASI Validación y Facturación de Intervenciones en Redes	60,00	3	60,00	23-6-08	12-9-08	23-6-08			0
1	66	1.6.5	61	ASI Cobros / Cortes e Impagos	40,00	3	40,00	15-9-08	7-11-08				0
1	67	1.6.6	61	DSI Cobros / Cortes e Impagos	35,00	3	35,00	10-11-08	26-12-08			5-12-08	6
1	68	1.6.7	61	Front End Análisis Atención Ciudadano 2	10,00	3	10,00	29-12-08	9-1-09				0
1	69	1.6.8	61	Front End Análisis Gestión Empleados 2	10,00	3	10,00	12-1-09	23-1-09				0
1	70	1.6.9	61	DSI Instalaciones y Equipos	40,00	3	40,00	15-9-08	7-11-08			28-3-08	4
1	71	1.6.10	61	DSI Validación y Facturación de Intervenciones en Redes	40,00	3	40,00	15-9-08	7-11-08			28-3-08	4
1	72	1.6.11	61	Front End Diseño Técnico 2	18,00	3	18,00	29-12-08	13-1-09				0
1	73	1.6.12	61	Sistemas Transversales Análisis GIS	30,00	3	30,00	1-12-08	26-12-08				0
1	74	1.6.13	61	Sistemas Transversales Diseño Técnico GIS	15,00	3	15,00	29-12-08	16-1-09			14-4-08	4
1	75	1.6.14	61	CSI Alfa 0.4: Gestión de Clientes 2	105,00	3	105,00	28-4-08	19-9-08				0
1	76	1.6.15	61	IAS Pruebas Alfa 0.4	15,00	3	15,00	22-9-08	10-10-08			18-6-08	4
1	77	1.6.16	61	Front End Desarrollo CRM Atención Ciudadano y Empleado 2	60,00	3	60,00	14-1-09	6-3-09				0
1	78	1.6.17	61	Front End Desarrollo ayuda	4,50	3	4,50	9-3-09	13-3-09				0
1	79	1.6.18	61	Sistemas Transversales Desarrollo GIS	30,00	3	30,00	19-1-09	13-2-09				0
1	80	1.7	1	Alfa 0.5: Clientes. Informes de Gestión	265,00	2	265,00	9-6-08	12-6-09				0
1	81	1.7.1	80	ASI Gestión Cl./P5 / Gest. Demanda / Herramientas Auxiliares	35,00	3	35,00	10-11-08	26-12-08				0
1	82	1.7.2	80	ASI Informes de Gestión y MMR	45,00	3	45,00	10-11-08	9-1-09				0
1	83	1.7.3	80	Sistemas Transversales Análisis Almacenes	30,00	3	30,00	19-1-09	13-2-09				0
1	84	1.7.4	80	Sistemas Transversales Análisis Compras	30,00	3	30,00	16-2-09	13-3-09				0
1	85	1.7.5	80	DSI Gestión Cl./P5 / Gest. Demanda / Herramientas Auxiliares	30,00	3	30,00	29-12-08	6-2-09			7-11-08	4
1	86	1.7.6	80	DSI Informes de Gestión y MMR	30,00	3	30,00	12-1-09	20-2-09			7-11-08	4
1	87	1.7.7	80	Sistemas Transversales Diseño Técnico Almacenes	15,00	3	15,00	16-3-09	3-4-09			14-4-08	4
1	88	1.7.8	80	Sistemas Transversales Diseño Técnico Compras	15,00	3	15,00	6-4-09	24-4-09			14-4-08	4
1	89	1.7.9	80	CSI Alfa 0.5:Gestión de Clientes 3	45,00	3	45,00	9-6-08	8-8-08				0
1	90	1.7.10	80	CSI Alfa 0.5: Informes de Gestión / MMR	80,00	3	80,00	23-2-09	12-6-09			29-12-08	4

**Tabla. IV.1.** Tabla SQL\*Server Msp\_Tasks (AQUA-WS)

PROJ_UID	TASK_UID	TASK_HUID	TASK_PAREN_T_UID	TASK_NAME	TASK_DURATION_EST	TASK_OUTLINE_LEVEL	TASK_DURATION	TASK_START_DATE	TASK_FINISH_DATE	TASK_ACT_T_START	TASK_ACT_FINISH	TASK_CONSTRAINT_DATE	TASK_CONSTRAINT_TYPE
1	91	1.7.11	80	IAS Pruebas Alfa 0.5	False	3	10,00	11-8-08	22-8-08			18-6-08	4
1	92	1.7.12	80	Desarrollo Integración Almacenes	False	3	45,00	6-4-09	15-5-09				0
1	93	1.7.13	80	Desarrollo Integración Compras	False	3	30,00	27-4-09	22-5-09				0
1	94	1.8	1	Beta 0.6: Núcleo Aqua	False	2	324,00	25-8-08	19-11-09				0
1	95	1.8.1	94	CSI Beta 0.6: Núcleo Aqua	False	3	10,00	25-8-08	5-9-08				0
1	96	1.8.2	94	Incorporación al entorno de pruebas	False	3	5,00	3-6-09	9-6-09			3-6-09	4
1	97	1.8.3	94	Carga de datos	False	3	5,00	10-6-09	16-6-09			25-5-09	4
1	98	1.8.4	94	Pruebas de integración Beta 0.6	False	3	20,00	8-7-09	4-8-09			4-8-09	6
1	99	1.8.5	94	Elaboración de manuales de usuario	False	3	20,00	22-5-09	18-6-09			18-6-09	6
1	100	1.8.6	94	Formación de formadores	False	3	10,00	21-9-09	2-10-09			21-9-09	4
1	101	1.8.7	94	Front End Formación de usuarios	False	3	40,00	1-7-09	25-8-09			1-7-09	4
1	102	1.8.8	94	IAS Pruebas de integración Release Candidata 0.7	False	3	10,00	5-8-09	18-8-09				0
1	103	1.8.9	94	Pruebas de integración Versión 1.0	False	3	22,00	19-8-09	17-9-09			18-3-08	4
1	104	1.8.10	94	Pruebas de aceptación Versión 1.0	False	3	45,00	18-9-09	19-11-09			30-9-09	6
1	105	1.8.11	94	Presentación y aprobación del sistema	False	3	0	19-11-09	19-11-09			15-10-09	6

LINK...	LINK_PRED_UID	LINK_SUCC_UID	LINK_TYPE	LINK_LAG
1	94	2	0	{null}
2	3	2	3	{null}
3	4	5	1	{null}
4	5	6	1	{null}
5	6	7	1	{null}
6	7	8	1	{null}
7	8	9	1	{null}
8	9	10	1	{null}
9	3	11	3	5,00
10	12	13	1	{null}
11	13	14	1	{null}
12	14	15	1	{null}
13	15	16	1	{null}
14	16	17	1	{null}
15	18	19	1	{null}
16	18	20	1	{null}
17	21	22	1	{null}
18	17	25	1	{null}
19	24	26	1	{null}
20	20	27	1	

Figura. IV.10. Tabla Msp\_Links (AQUA-WS)

**Tabla. IV.2.** Tabla SQL\*Server Msp\_Tasks (AQUA-WS)

LINK_UID	LINK_PRE_D_UID	LINK_SUCC_UID	LINK_TYPE	LINK_LAG
1	94	2	0	
2	3	2	3	
3	4	5	1	
4	5	6	1	
5	6	7	1	
6	7	8	1	
7	8	9	1	
8	9	10	1	
9	3	11	3	5,00
10	12	13	1	
11	13	14	1	
12	14	15	1	
13	15	16	1	
14	16	17	1	
15	18	19	1	
16	18	20	1	
17	21	22	1	
18	17	25	1	
19	24	26	1	
20	20	27	1	
21	27	28	1	
22	22	29	1	
23	29	30	1	
24	30	31	1	
25	31	32	1	
26	16	33	1	

**Tabla. IV.2.** Tabla SQL\*Server Msp\_Tasks (AQUA-WS)

LINK_UID	LINK_PRE_D_UID	LINK_SUCC_UID	LINK_TYPE	LINK_LAG
27	23	33	1	
28	25	34	1	
29	26	35	1	
30	33	35	1	
31	35	36	1	
32	27	37	1	
33	29	38	1	
34	32	39	1	
35	11	40	3	30,00
36	51	44	1	
37	44	45	1	
38	41	46	1	
39	41	47	1	
40	42	47	1	
41	43	48	1	
42	47	48	1	
43	48	49	1	
44	49	50	1	
45	46	51	1	
46	51	52	1	
47	48	53	1	
48	49	54	1	
49	50	55	1	
50	40	56	3	30,00
51	57	58	1	-15,00
52	58	59	1	-20,00

**Tabla. IV.2.** Tabla SQL\*Server Msp\_Tasks (AQUA-WS)

LINK_UID	LINK_PRE_D_UID	LINK_SUCC_UID	LINK_TYPE	LINK_LAG
53	59	60	1	
54	56	61	3	30,00
55	62	63	1	
56	64	66	1	
57	65	66	1	
58	66	67	1	
59	71	67	1	
60	67	68	1	
61	68	69	1	
62	64	70	1	
63	65	71	1	
64	67	72	1	
65	50	73	1	
66	73	74	1	
67	75	76	1	
68	72	77	1	
69	77	78	1	
70	74	79	1	
71	61	80	3	30,00
72	66	81	1	
73	71	82	1	
74	74	83	1	
75	83	84	1	
76	81	85	1	
77	82	86	1	
78	84	87	1	



**Tabla. IV.2.** Tabla SQL\*Server Msp\_Tasks  
(AQUA-WS)

LINK_UID	LINK_PRE_D_UID	LINK_SUCC_UID	LINK_TYPE	LINK_LAG
79	86	87	1	
80	87	88	1	
81	86	90	1	
82	89	91	1	
83	87	92	1	
84	88	93	1	
85	80	94	3	30,00
86	91	95	1	
87	95	96	1	
88	96	97	1	
89	97	98	1	
90	98	102	1	
91	102	103	1	
92	103	104	1	
93	104	105	1	

**Tabla. IV.3.** Tabla de asignación de recursos (AQUA-WS)

Nombre_Tarea	Duración
Proyecto	567 días?
Seguimiento y Control del Proyecto	567 días?
Organización y lanzamiento del Proyecto	109 días
Reunión de Lanzamiento	1 día
Elaboración y Revisión Manual de Calidad	23 días
Elaboración y Revisión Plan de Gobierno del Proyecto	17 días
Elaboración Plan Maestro del Proyecto	15 días
Elaboración y Aceptación Plan de Calidad	3 días
Elaboración del Plan de Comunicación	10 días
Puesta en marcha del Plan de Comunicación	24 días
Alfa 0.1: Capa Transversal. Subsistema Común	220 días?
Identificación de Subsistemas	49 días
Definición del Sistema	51 días
Establecimiento de Requisitos	55 días
Análisis Modelo de Datos Actual	80 días
Definición de la arquitectura del sistema	30 días
Modelo Conceptual Núcleo Aqua	45 días
Front End Análisis Elementos Comunes	31 días?
Front End Entrega Borrador Servicios	0 días
Front End Entrega Análisis Elementos Comunes	0 días
Sistemas Transversales Análisis Elementos Comunes	55 días?
Sistemas Transversales Entrega Análisis Elementos Comunes	1 día?
ASI Capa Transversal	45,5 días
ASI Subsistema Común Núcleo AQUA	44 días
Diseño físico de datos	56,88 días

**Tabla. IV.3.** Tabla de asignación de recursos (AQUA-WS)

Nombre_Tarea	Duración
DSI Subsistema Común Núcleo AQUA	25 días
Front End Diseño Técnico de Elementos Comunes	1 ms
Front End Entrega Diseño Técnico de Elementos Comunes	0 días
Sistemas Transversales Diseño Técnico Elementos Comunes	1 ms
Sistemas Transversales Entrega Diseño Técnico Elementos Comunes	0 días
Sistemas Transversales Análisis Gestor Documental	1,5 mss
Sistemas Transversales Diseño Técnico Gestor Documental	15 días
CSI Prototipo Capa Transversal	55 días
Especificación técnica del plan de pruebas	15 días
CSI Subsistema Común Núcleo AQUA	70 días
IAS Pruebas Subsistema Común Núcleo AQUA	10 días
Front End Desarrollo elementos comunes	1,5 mss
Sistemas Transversales Desarrollo elementos comunes	1 ms
Sistemas Transversales Desarrollo Integración Gestor Documental	1 ms
Alfa 0.2 : Gestión de Clientes.	254,75 días
ASI Atención Cliente / Contratación /Lecturas	60 días
ASI Gen/ Plan/ Cump/Val y Fin. de actuaciones tipo	60 días
Front End Análisis Atención Ciudadano y Empleado	20 días
Sistemas Transversales Análisis FICO	2 mss
Sistemas Transversales Análisis Recursos Humanos	0,99 mss
DSI Atención Cliente / Contratación/Lecturas	45 días
DSI Gen/ Plan/ Cump/Val y Fin. de actuaciones tipo	45 días
Front End Diseño Técnico 1	0,6 mss
Sistemas Transversales Diseño Técnico FICO	15 días

**Tabla. IV.3.** Tabla de asignación de recursos (AQUA-WS)

Nombre_Tarea	Duración
Sistemas Transversales Diseño Técnico Recursos Humanos	15 días
CSI Alfa 0.2: Gestión de Clientes 1	90 días
IAS Pruebas Alfa 0.2	15 días
Front End Desarrollo CRM Atención Ciudadano y Empleado 1	1,3 mss
Sistemas Transversales Desarrollo Integración FICO	1,5 mss
Sistemas Transversales Desarrollo Integración Recursos Humanos	1 ms
Alfa 0.3: Gestión de Obras y Proyectos	173 días
ASI Gestión de Obras y Proyectos	45 días
DSI Gestión de Obras y Proyectos	48 días
CSI Gestión de Obras y Proyectos	90 días
Pruebas Gestión de Obras y Proyectos	10 días
Alfa 0.4: Clientes: Intervenciones en Red	229,5 días?
ASI Facturación	9 días
DSI Facturación	36 días
ASI Instalaciones y Equipos	60 días
ASI Validación y Facturación de Intervenciones en Redes	60 días
ASI Cobros / Cortes e Impagos	40 días
DSI Cobros / Cortes e Impagos	35 días
Front End Análisis Atención Ciudadano 2	10 días
Front End Análisis Gestión Empleados 2	10 días
DSI Instalaciones y Equipos	40 días
DSI Validación y Facturación de Intervenciones en Redes	40 días
Front End Diseño Técnico 2	0,6 mss
Sistemas Transversales Análisis GIS	1 ms
Sistemas Transversales Diseño Técnico GIS	15 días
CSI Alfa 0.4: Gestión de Clientes 2	105 días

**Tabla. IV.3.** Tabla de asignación de recursos (AQUA-WS)

Nombre_Tarea	Duración
IAS Pruebas Alfa 0.4	15 días
Front End Desarrollo CRM Atención Ciudadano y Empleado 2	1,9 mss
Front End Desarrollo ayuda	4,5 días?
Sistemas Transversales Desarrollo GIS	1 ms
Alfa 0.5: Clientes. Informes de Gestión	265 días
ASI Gestión CI./P5 / Gest. Demanda / Herramientas Auxiliares	35 días
ASI Informes de Gestión y MMR	45 días
Sistemas Transversales Análisis Almacenes	1 ms
Sistemas Transversales Análisis Compras	1 ms
DSI Gestión CI./P5 / Gest. Demanda / Herramientas Auxiliares	30 días
DSI Informes de Gestión y MMR	30 días
Sistemas Transversales Diseño Técnico Almacenes	15 días
Sistemas Transversales Diseño Técnico Compras	15 días
CSI Alfa 0.5:Gestión de Clientes 3	45 días
CSI Alfa 0.5: Informes de Gestión / MMR	80 días
IAS Pruebas Alfa 0.5	10 días
Desarrollo Integración Almacenes	1,5 mss
Desarrollo Integración Compras	1 ms
Beta 0.6: Núcleo Aqua	324 días
CSI Beta 0.6: Núcleo Aqua	10 días
Incorporación al entorno de pruebas	5 días
Carga de datos	5 días
Pruebas de integración Beta 0.6	20 días
Elaboración de manuales de usuario	20 días
Formación de formadores	10 días
Front End Formación de usuarios	40 días

**Tabla. IV.3.** Tabla de asignación de recursos (AQUA-WS)

Nombre_Tarea	Duración
IAS Pruebas de integración Release Candidata 0.7	10 días
Pruebas de integración Versión 1.0	22 días

**Tabla. IV.3.** Tabla de asignación de recursos (AQUA-WS)

Nombre_Tarea	Duración
Pruebas de aceptación Versión 1.0	45 días
Presentación y aprobación del sistema	0 días

**ANEXO V. PUBLICACIONES Y  
PROYECTOS**



## Capítulos de libros

Arévalo Maldonado, C., Gómez-López, M. T., Quintero, A. M. R., & Ramos, I. (2013). An Architecture to Infer Business Rules from Event Condition Action Rules Implemented in the Persistence Layer. *Uncovering Essential Software Artifacts through Business Process Archeology*, 201.

**Abstract**

The business rules that govern the behaviour of a business process can be hardcoded in different ways in a software application. The modernization or improvement of these applications to a process-oriented perspective implies typically the modification of the business rules. Frequently, legacy systems are not well-documented, and almost always, the documentation they have is not updated. As a consequence many times is necessary the analysis of source code and databases structures to be transformed into a business language more understandable by the business experts involved in the modernization process.

Database triggers are one of the artefacts in which business rules are hardcoded. We focus on this kind of artefacts, having in mind to avoid the manual analysis of the triggers by a database expert, and bringing it closer to business experts. To get this aim we need to discover business rules that are hardcoded in triggers, and translate it into vocabularies that are commonly used by business experts. In this paper we propose an ADM-based architecture to disco

**Keywords**

Model Driven Modernization, Legacy Systems, Architecture-Driven Modernization (ADM), SQL, Metamodel, ECA Rules, Triggers.

## Publicaciones en revistas internacionales

Gómez-López, M. T., Gasca, R. M., & Arévalo Maldonado, C. (2010). A survey using constraints to decision-making for fault tolerance in Business processes. *International Journal of Software Engineering and Its Applications*, 4(4), 79-92.

### Abstract

Sometimes the business processes do not work how it is expected. In these cases, a diagnosis process has to be executed to determine the responsible activity or activities of the fault in order to substitute it or them for a correct activity. The aim of this paper is describe the necessary steps to find out another service that can replace it in an efficient way. In order to automate the search and substitution of activities, we propose to describe the functionality of the tasks using constraints, making easier the determination of the possible activities that could substitute everyone faulty activities in the business process. In this paper, it is also analyzed how to adapt the communication protocol with XML messages to a behavior described using constraints.

### Keywords

Business Process, Fault tolerance, Constraint Satisfaction Problem.



C. Arevalo, M.J. Escalona, I. Ramos and M. Domínguez-Muñoz (2015): “A metamodel to integrate business processes time perspective in *BPMN 2.0*”. In Elsevier Information and software technology (IST). Manuscript number INFOSOF-D-15-00235 (En revisión).

**Abstract**

*Context:* Business Process Management (BPM) is becoming a strategic advantage for organizations to streamline their operations. Most business experts are betting for OMG Business Process Model and Notation (BPMN) as de-facto standard (ISO/IEC 19510:2013) and selected technology to model processes. The temporal dimension underlies in any kind of process however, technicians need to shape this perspective that must also coexist with task control flow aspects, as well as resource and case perspectives. BPMN poorly gathers temporary rules. This is why there are contributions that extend the standard to cover such dimension. BPMN is mainly an imperative language. There are research contributions showing time constraints in BPMN, such as (i) BPMN patterns to express each rule with a combination of artifacts, thus these approaches increase the use of imperative BPMN style, and (ii) new decorators to capture time rules semantics giving clearer and simpler comprehensible specifications. Nevertheless, these extensions cannot yet be found in the present standard.

*Objective:* To define a time rule taxonomy easily found in most business processes and look for an approach that applies each rule with current BPMN 2.0 standard in a declarative way.

*Method:* A model-driven approach is used to propose a BPMN metamodel extension to address time-perspective.

*Results:* We look at a declarative approach where new time specifications may overlie the main control flow of a BPMN process. This proposal is totally supported with current BPMN standard, giving a BPMN metamodel extension with OCL constraints. We also use AQUA-WS as a software project case study which is planned and managed with MS Project. We illustrate business process extraction from project plans.

*Conclusion:* This paper suggests to handle business temporal rules with current BPMN standard, along with other business perspectives like resources and cases. This approach can be applied to reverse engineering processes from legacy databases.

**Keywords**

BPM, BPMN, Time perspective, Metamodel, OCL, MDE, Legacy system

The screenshot shows the Elsevier manuscript submission system interface. At the top, there is a navigation bar with the Elsevier logo and the text 'INFORMATION AND SOFTWARE TECHNOLOGY'. Below this, there are links for 'home', 'main menu', 'submit paper', 'guide for authors', 'register', 'change details', and 'log out'. A user profile section shows the username 'carlosarevalo@us.es' and a 'Switch To: Author' button. A table titled 'Revisions Being Processed for Author C. Arevalo, MEng Industrial Engineer' is displayed. The table has columns for 'Action', 'Manuscript Number', 'Title', 'Date Submission Began', and 'Current Status'. One revision is listed with manuscript number 'INFOSOF-D-15-00235R1', title 'A metamodel to integrate business processes time perspective in BPMN 2.0', and status 'Under Review'. The page number 'Page: 1 of 1 (1 total revisions being processed)' is shown at the bottom of the table.

Action	Manuscript Number	Title	Date Submission Began	Current Status
<a href="#">Action Links</a>	INFOSOF-D-15-00235R1	A metamodel to integrate business processes time perspective in BPMN 2.0	Dec 04, 2015	Under Review

M. Domínguez-Muñoz, M.J. Escalona, I. Ramos, and C. Arevalo. (2015): “Systematic Literature Review for Web Application Estimation with Use Case Points. Looking for a Model-Driven Perspective”. Springer, Software Quality Journal, Submission. Manuscript number SQJO-S-15-00125, (En revisión).

**Abstract**

Context: Software effort estimation has always been a matter of interest for companies since they need to carry out their projects on time and budget. In contrast, Web Applications development could be different to regular Software development as Web Application effort estimations might need different techniques and measures.

Objective: This paper aims to analyse how Use Case Points (UCP) technique is used in real Web Application effort estimation and evaluate whether the research community is applying new tendencies, like the Model-Driven paradigm.

Method: A Systematic Literature Review (SLR) process has been executed to revise the current situation for UCP usage in Web Applications effort estimations as well as the use of Model-Driven Engineering (MDE) techniques.

Results: The SLR compiles a large number of studies about effort estimation using UCP, some studies regarding Web Application and few works related to MDE-based techniques or enterprise uses.

Conclusion: UCP does not usually fit Web Application effort estimation approaches properly, although it is one of the most popular effort estimation techniques, therefore, it is not applied to enterprises. The lack of related studies reveals that effort estimation in Web Applications using UCP constitutes a real gap the research community is not currently filling in.

**Keywords**

Effort Estimation, Use Case Points, Web Application, Model-Driven Engineering, Systematic Literature Review.

## Publicaciones en conferencias internacionales

Arevalo, C., Ramos, I. and M.J. Escalona (2015). “Discovering Business Models for Software Process Management An Approach for Integrating Time and Resource Perspectives from Legacy Information Systems”. In 17th International Conference on Enterprise Information Systems (ICEIS 2015), Barcelona. Paper Nr.271.

Abstract

Business Process Management (BPM) is becoming the modern core to support business in all type of organizations and software business is not a n exception. Software companies are often involved in important and complex collaborative projects carried out by many stakeholders. Each actor (customers, suppliers or government instances, among others) works with individual and shared processes. Everyone needs dynamic and evolving approaches for managing their software projects lifecycle. Nevertheless, many companies still use systems that are out of the scope of BPM for planning and control projects and managing enterprise content (Enterprise Content Management, ECM) as well as all kinds of resources (ERP). Somehow systems include scattered artifacts that are related to BPM perspectives: control and data flow, time, resource and case, for example. It is aimed to get interoperable BPM models from these classical Legacy Information Systems (LIS). Model-Driven Engineering (MDE) allows going from application code to higher-level of abstraction models. Particularly, there are standards and proposals for reverse engineering LIS. This paper illustrates LIS cases for software project planning and ECM, looking at time and resource perspectives. To conclude, we will propose a MDE-based approach for taking out business models in the context of software process management.

Keywords

Software Process Management, *BPMN*, Interoperability, Legacy Systems, Reverse Engineering, Model-Driven Engineering.

## Publicaciones en conferencias nacionales

Arévalo, C. (2014). "Extracting Business Models for Software Process Support". Ed. Riquelme, J., Ruiz, M., & García, M. T. Actas JISBD, SISTEDES, Doctoral Consortium., Cádiz, 2014, pp.40-47.

### Abstract

Organizations oriented to software business have similar problems to other productive organizations; they are subject to globalization, competitiveness and they also need dynamic and evolutionary computer-aided software tools for managing life cycle of their projects that may quickly adapt to complex and changing environments. They also use the BPM (Business Process Management) approach for definition and execution of their own business process based on *BPMN* (Business Process Model Notation) language. A business expert, within these complex organizations, manages knowledge that lies in these multiple and heterogeneous automated tools, distributed all over the world, each one with different languages and data structures that usually are relational databases. Model Driven Engineering (MDE) paradigm and several standards of the Object Management Group (OMG) based on the Model Driven Architecture (MDA) together with other research works offer concepts, metamodels and methods for software reverse engineering going from application code and data structures to models of higher level of abstraction nearest to the business expert level. We will look for proposals to discover this knowledge, gathering artifacts that lie in relational databases such as table structures, constraints and hardcoded triggers.

The desired result must be a global data business view of the whole organization containing common and essential data structures (business models, business processes structures, resources involved and metrics for processes, etc.). This global view must have a high level of independency from every used computer-aided tool. It will also allow extracting global reports and efficient migrations between pairs of these tools and it will be more stable along time.

### Keywords

Software Process Support, Legacy System, Model Driven Architecture (MDA), Architecture-Driven Modernization (ADM), SQL, Trigger, Interoperability.

## Proyectos

### Proyectos de investigación

[P08-Tic-04095] "Opbus: Mejora De La Calidad De Procesos De Negocio Mediante Tecnologías De La Optimización y tolerancia a fallos".	
Información del proyecto	Responsable: Rafael Martínez Gasca Tipo de proyecto: Proyectos de Excelencia de la Junta de Andalucía Referencia: TIN2010-20057-C03-02 Fecha de comienzo: 13-1-2009 Fecha de finalización: 31-7-2011
Financiado por	Junta de Andalucía (Consejería de Innovación, Ciencia y Empresas)

[TIN2010-20057-C03-02] "Testing temprano y modelos de simulación híbrida en la producción de software. Tempros".	
Información del proyecto	Responsable: María José Escalona Cuaresma Tipo de proyecto: Plan Nacional del 2010 Referencia: TIN2010-20057-C03-02 Fecha de comienzo: 01-01-2011 Fecha de finalización: 31-12-2013
Financiado por	Ministerio de Ciencia e Innovación

[TIN2013-46928-C3-3-R] "Mecanismos Guiados En Etapas Tempranas Para La Mejora Del Software. Megus".	
Información del proyecto	Responsable: María José Escalona Cuaresma Tipo de proyecto: Programa Estatal de Fomento de la Investigación Científica y Técnica de Retos.  Referencia: TIN2013-46928-C3-3-R Fecha de comienzo: 01-01-2014 Fecha de finalización: 31-12-2016
Financiado por	Ministerio de Economía y Competitividad

### Proyectos de transferencia tecnológica

[P047-14/E09] "Proyecto EMPOWER".	
Información del proyecto	Responsable: María José Escalona Cuaresma; García-García, Julián Alberto  Tipo de proyecto: Contrato art. 11/45 LRU - 68/83 LOU.  Referencia: P047-14/E09  Fecha de comienzo: 01-03-2015 Fecha de finalización: 31/12/2016
Financiado por	CDTI / CTA

### Redes nacionales

[TIN2015-71938-REDT] "Red Temática para el desarrollo de soluciones software de calidad en entornos PLM".	
Información del proyecto	Responsable: María José Escalona Cuaresma Tipo de proyecto: Plan Estatal 2013-2016 Excelencia - Redes.  Referencia: TIN2015-71938-REDT Fecha de comienzo: 01-12-2014 Fecha de finalización: 31-12-2016
Financiado por	Ministerio de Economía y Competitividad

### Redes internacionales

000000000244467] "Red Temática Mexicana En Ingeniería Del Software".	
Información del proyecto	Responsable: María José Escalona Cuaresma Tipo de proyecto: Registro y Estructuración de Redes Temáticas CONACYT 2014.  Referencia: TIN2013-46928-C3-3-R Fecha de comienzo: 01-12-2014 Fecha de finalización:
Financiado por	Consejo Nacional De Ciencia Y Tecnologia (Conacyt). Mexico