Guía de Uso de SPEM 2 con EPF Composer

Versión 3.0

Francisco Ruiz, Javier Verdugo
1-abril-2008

Universidad de Castilla-La Mancha
Escuela Superior de Informática
Departamento de Tecnologías y Sistemas de Información
Grupo Alarcos

http://alarcos.inf-cr.uclm.es/

Índice

Índice 1
Índice de Figuras 3
Índice de Tablas 5
1 Introducción 7
1.1 Notas sobre Versiones 7
2 Contexto de Ingeniería de Procesos 9
3 Características Generales de SPEM 2 11
3.1 Mecanismos de Trabajo con SPEM 13
4 Metamodelo SPEM 17
4.1 Estructura de Paquetes 20
4.2 Organización de un Repositorio SPEM 2 22
4.3 Contenido de Método 22
4.3.1 Elementos de Contenido 23
4.3.1.1 Tareas 23
4.3.1.2 Roles 24
4.3.1.3 Productos de Trabajo 24
4.3.1.4 Guías 25
4.3.1.5 Categorías 26
4.3.2 Propiedades Principales 27
4.3.3 Asociaciones 27
4.4 Procesos 29
4.4.1 Aspectos Generales 29
4.4.2 Elementos de Desglose de Trabajo 30
4.4.2.1 Actividades 31
4.4.2.2 Contenido de Método en Uso 33
4.4.2.3 Fases, Iteraciones e Hitos 34
4.4.3 Tipos de Procesos 35
4.5 Reutilización y Variabilidad 36
4.5.1 Variabilidad de Elementos de Contenido 36
4.5.2 Reutilización y Variabilidad de Elementos de Proceso 41
4.6 Configuraciones de Método 42
5 El Editor EPF Composer 45
5.1 Poblando el Contenido de Método 45
5.1.1 Categorización de Elementos de Método 46
5.2 Creando Procesos 46
5.2.1 Estructuras de Desglose 47
5.2.1.1 Estructuras de Desglose de Trabajo 48
5.2.1.2 Propiedades de los Elementos de Desglose de Trabajo 48
5.2.1.3 Descriptores de Elementos Básicos 50
5.2.1.4 Otras Estructuras de Desglose 51
5.2.2 Reutilizando y Adaptando Actividades 52
5.2.3 Diagramas 53
5.2.3.1 Diagramas de Actividad 54
5.2.3.2 Diagramas de Detalle de Actividad 55
5.2.3.3 Diagramas de Dependencias de Producto de Trabajo 55
5.3 Creando Configuraciones de Método 56
5.4 Publicando Contenidos 57
5.4.1 Personalización de la Web. 59
6 Ejemplo de Aplicación – METRICA 3 61
6.1 Reglas de Aplicación 61

6.1.1 Aspectos Generales 61
6.1.2 Aspectos sobre Participantes 62
6.1.3 Aspectos sobre Técnicas y Prácticas 62
6.1.4 Aspectos sobre Procesos, Actividades y Tareas 63
6.1.5 Aspectos sobre Productos 64
6.2 Ejemplos de Resultados 64
6.2.1 Participante 64
6.2.2 Técnica 65
6.2.3 Práctica 66
6.2.4 Proceso 68
6.2.5 Actividad 69
6.2.6 Tarea 70
Anexo A: Términos e Iconos de SPEM 2 y EPF Composer 73
A.1: Español -> Inglés 73
A.2: Inglés -> Español 75
Anexo B: Pasos Recomendados en la Creación de Procesos con EPF Composer 79
Anexo C: Opciones de Exportación 83
C.1: Exportar Configuraciones y Plugins de Método 83
C.2: Exportar a Microsoft Project 84
C.3: Exportar a XML 85
Anexo D: Otras Preguntas Frecuentes 87
D.1: ¿Cómo generar distintas vistas de una metodología? 87
D.2: ¿Cómo adecuar una metodología a proyectos específicos? 87
D.3: ¿Cómo preparar versiones reducidas para entregar a terceros? 88
D.4: ¿Cómo incorporar documentos y enlaces externos en las descripciones de elementos y procesos? 88
D.5: ¿Cómo definir la estructura de la Web publicada? 90

Índice de Figuras

Figura 1. Aplicación de MOF a procesos software. 10
Figura 2. Ingeniería de Procesos. 10
Figura 3. Idea básica de proceso en SPEM 2. 11
Figura 4. Marco de trabajo general con SPEM 2. 12
Figura 5. Niveles de modelado y ejemplos de instanciaciones. 14
Figura 6. Ejemplo de representación gráfica de procesos usando el perfil UML de SPEM. 15
Figura 7. Aspectos principales para modelar con SPEM. 17
Figura 8. Ortogonalidad del Method Content y de los Processes (igual que ocurre en RUP). ... 17
Figura 9. Conceptos para representar la jerarquía de desglose del trabajo. 18
Figura 10. Ejemplo del mecanismo de plugins en SPEM para permitir variabilidad y extensibilidad. 19
Figura 11. Ejemplo de Componente de Proceso en SPEM 2. 19
Figura 12. Estructura de paquetes de SPEM 2. 20
Figura 13. Ejemplo de Jerarquía para organizar un repositorio (biblioteca) SPEM 2..... 22
Figura 14. Jerarquía de conceptos del method content. 23
Figura 15. Ejemplo de asociaciones de una tarea..... 24
Figura 16. Ejemplo de uso de categorías..... 26
Figura 17. Relaciones entre los Elementos de Contenido de Método. 28
Figura 18. Jerarquía de conceptos de procesos. 29
Figura 19. Ejemplos de Estructura de Desglose de Trabajo (izquierda) y de Flujo de Trabajo (derecha)..... 30
Figura 20. Flujo de trabajo (derecha) y anidamiento de actividades (izquierda) entre elementos de desglose de trabajo. 31
Figura 21. Relación entre actividades y roles y productos de trabajo. 32
Figura 22. Ejemplo de una actividad y sus asociaciones. 32
Figura 23. Ejemplo de contenido de método en uso. 34
Figura 24. Una tarea en uso dentro de una estructura de desglose de trabajo. 34
Figura 25. Ejemplos de variabilidad de tipo “contribuye”. 37
Figura 26. Resultado de la variabilidad de la Figura 25. 38
Figura 27. Ejemplo de variabilidad de tipo “reemplaza”. 38
Figura 28. Resultado de la variabilidad de la Figura 27. 39
Figura 29. Ejemplo de variabilidad de tipo “amplía”. 39
Figura 30. Resultado de la variabilidad de la Figura 29. 40
Figura 31. Ejemplo de variabilidad de tipo “amplía y reemplaza”. 40
Figura 32. Resultado de la variabilidad de la Figura 31. 41
Figura 33. Ejemplo de una configuración de método (elementos en rojo). 43
Figura 34. Pantalla de la ficha de una tarea en EPF Composer. 45
Figura 35. Estructura de Desglose de Trabajo (WBS) en forma de tabla en EPF Composer. 49
Figura 36. Ventana de la estructura de desglose de Asignación de Equipos. 51
Figura 37. Ventana de la estructura de desglose de Utilización de Productos de Trabajo. 52
Figura 38. Ventana de la Vista Consolidada. 52
Figura 39. Diagrama de Actividad y paleta de su editor gráfico en EPF Composer. 54
Figura 40. Diagrama de Detalle de Actividad y paleta de su editor gráfico en EPF Composer. 55
Figura 41. Diagrama de Dependencias de Producto de Trabajo y paleta de su editor gráfico en EPF Composer. 55
Figura 42. Ejemplo de sitio generado automáticamente con EPF Composer. 59
Figura 43. Opciones de exportación de EPF Composer. 83
Figura 44. Diálogo para añadir un enlace a una URL externa, un Archivo o un Elemento de método. 88
Figura 45. Ejemplo de material de soporte con enlaces a páginas externas. 89
Figura 46. Elemento publicado que incluye el material de soporte de la Figura 45 como guía.. 90

Figura 47. Opciones disponibles para adjuntar archivos a una plantilla. 90

Índice de Tablas

Tabla 1. Niveles de modelado de MOF..... 9
Tabla 2. Ejemplo de participante implementado como role..... 65
Tabla 3. Ejemplo de técnica implementada como guideline..... 66
Tabla 4. Ejemplo de práctica implementada como guideline..... 67
Tabla 5. Ejemplo de proceso implementado como capability pattern..... 69
Tabla 6. Ejemplo de actividad implementada como capability pattern..... 70
Tabla 7. Ejemplo de tarea implementada como task..... 71

1 Introducción

METRICA 3 (M3), el proceso unificado de Rational (RUP) y demás metodologías están basadas en un conjunto de ideas y conceptos subyacentes, que no están definidos de manera explícita, es decir, no es evidente el metamodelo subyacente utilizado. Además, el formato en que se maneja la información de dichas metodologías son documentos de texto en lenguaje natural. Esto tiene la gran desventaja de que toda la manipulación (creación, revisión, reutilización, adaptación, etc.) y generación de documentación (publicación) para dichas metodologías es un proceso puramente manual. Para evitar dicha situación, OMG (la organización industrial promotora de UML) ha desarrollado y aprobado recientemente el estándar SPEM (Software & Systems Process Engineering Metamodel) versión 2.0 [1], que pretende ser el estándar industrial para la representación de modelos de procesos de ingeniería del software e ingeniería de sistemas.
Por otro lado, dentro de la plataforma abierta ECLIPSE, se ha puesto en marcha el proyecto EPF (Eclipse Process Framework) [2], que ha desarrollado un editor de SPEM 2, llamado "EPF Composer", en adelante EPFC. EPFC se basa en el estándar SPEM 2 y permite definir, gestionar y reutilizar un repositorio de fragmentos de métodos y procesos. Con EPFC se pueden crear implementaciones en formato SPEM 2 de cualquier método, proceso o metodología de ingeniería del software.
En este documento se presentan las principales características y conceptos de SPEM 2, así cómo la manera en que sus conceptos y elementos se manejan con la herramienta EPFC. Como ejemplo de uso, se utiliza la metodología M3, estableciendo las reglas para implementar dicha metodología con SPEM 2, e incluyendo una implementación detallada en EPFC de una tarea completa de M3.

1.1 Notas sobre Versiones

Este documento es la versión 3. Respecto de la versión anterior, publicada en enero-2008, los principales cambios son los siguientes:

- Se ha incluido un apartado 5.4.1 explicando las opciones para personalizar las wbes publicadas con EPFC.
- Se han incluido tres nuevos anexos que intentan guiar a los usuarios de SPEM con EPFC a la hora de trabajar:
  - Pasos recomendados en la creación de procesos con EPF Composer (anexo B).
  - Opciones de exportación (anexo C) de configuraciones y plugins, y cómo exportar plantillas de proyectos a MS Project y a XML en general.
  - Otras preguntas frecuentes (anexo D): ¿cómo generar distintas vistas de una metodología?, ¿cómo adecuar una metodología a proyectos específicos?, ¿cómo preparar versiones reducidas para entregar a terceros?, ¿cómo incorporar documentos y enlaces externos en las descripciones de elementos y procesos? y ¿cómo generar la estructura de la Web publicada?

La versión 2 fue una actualización y ampliación en profundidad de la versión 1.0, elaborada por Francisco Ruiz en octubre de 2007.

Para cada concepto o término de SPEM 2 se presenta su equivalente en castellano y el original en inglés. La nomenclatura en castellano está basada en la versión española de EPFC 1.2, pero

[1] http://www.omg.org/cgi-bin/doc?ptc/07-08-07
[2] http://www.eclipse.org/epf/

no siempre se ha respetado, en cuyo caso se incluyen indicaciones explicando la razón de tal diferencia.
El documento está hecho con la versión de SPEM 2.0 publicada en agosto de 2007 en la web oficial. En cuanto a la herramienta EPFC se ha empleado la versión 1.2, tanto en inglés como en la traducción al español incluida en el NL Pack 1.2.

2 Contexto de Ingeniería de Procesos

SPEM 2 es un estándar de metamodelado que sirve para representa procesos de ingeniería de software, entiendo un Proceso Software (PS) como "Un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software".
SPEM 2 se encuadra dentro de la Ingeniería de Procesos de Software, en inglés *Software Processes Engineering* (SEP), que es un área nueva de la Ingeniería de Software dedicada a "la definición, implementación, medición y mejora de los procesos de Ingeniería de Software".

SPEM 2 está basado en MOF (Meta Object Facility) [3], otro estándar previo de la OMG que define una arquitectura de modelado de cuatro niveles conceptuales (ver Tabla 1). Así, SPEM es a los procesos software lo mismo que UML es a los sistemas software. UML es un metamodelo que sirve para representar modelos de sistemas software y SPEM es un metamodelo que sirve para representar modelos de procesos software.

Nivel	MOF	Ejemplo
M3	Modelo MOF (Meta-metamodelo)	
M2	Meta-modelo	UML, SPEM Entidad-Interrelación
M1	Modelo	Modelo en UML
M0	Datos	Datos de un proyecto concreto

Tabla 1. Niveles de modelado de MOF.

Entre los distintos niveles consecutivos existen relaciones de instanciación, de forma que un elemento de un nivel N_1 es una instancia de un elemento (más genérico, más abstracto) del nivel superior N. En la Figura 1 se muestra un ejemplo para el dominio de los procesos software.

Esto nos lleva a una nueva manera de trabajar. A partir de ahora, además de modelar, también hablaremos de METAMODELAR. Un Metamodelo describe un conjunto de conceptos genéricos y sus interrelaciones, que sirven de base para la definición de Modelos de un cierto Dominio. Por tanto, un metamodelo es un modelo de modelos. Mediante el uso de un metamodelo se pueden representar modelos del correspondiente dominio.
Aplicando estas ideas al ámbito de los procesos software, los modelos de PS se construyen mediante instanciación de los conceptos del metamodelo de PS genérico, es decir, de SPEM. Esta instanciación es determinada por las características propias del modelo que se quiere elaborar. En el diseño de un modelo de PS se deben respetar las relaciones entre los diferentes conceptos definidos en SPEM.

Gracias al uso de SPEM 2, se puede disponer de modelos de PS en formato procesable por computador, lo que proporciona capacidades para:
- Facilitar la comprensión y comunicación humana.
- Facilitar la reutilización.
- Dar soporte a la mejora de procesos.
- Dar soporte a la gestión de procesos.
- Guiar la automatización de procesos.
- Dar soporte para la ejecución automática.

[3] http://www.omg.org/technology/documents/formal/MOF_Core.htm

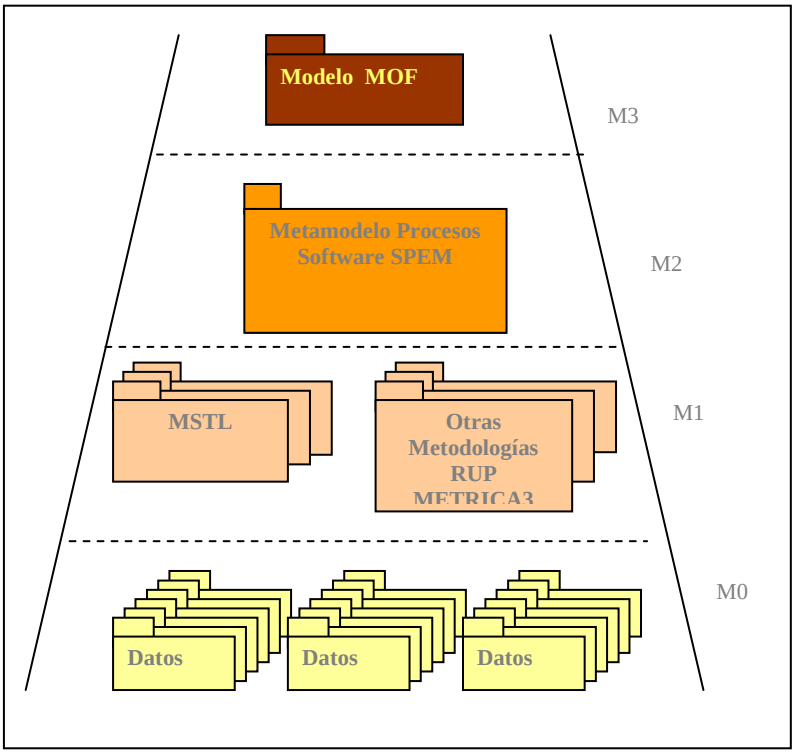


Figura 1. Aplicación de MOF a procesos software.

El gran potencial de esta nueva manera de trabajar con procesos ha hecho que surja la Ingeniería de Procesos (también llamada Ingeniería de Métodos), que trata de aplicar a los procesos maneras y técnicas que antes han demostrado su utilidad en los productos (software). La Figura 2 muestra un resumen de las nuevas actividades y roles en esta nueva faceta.

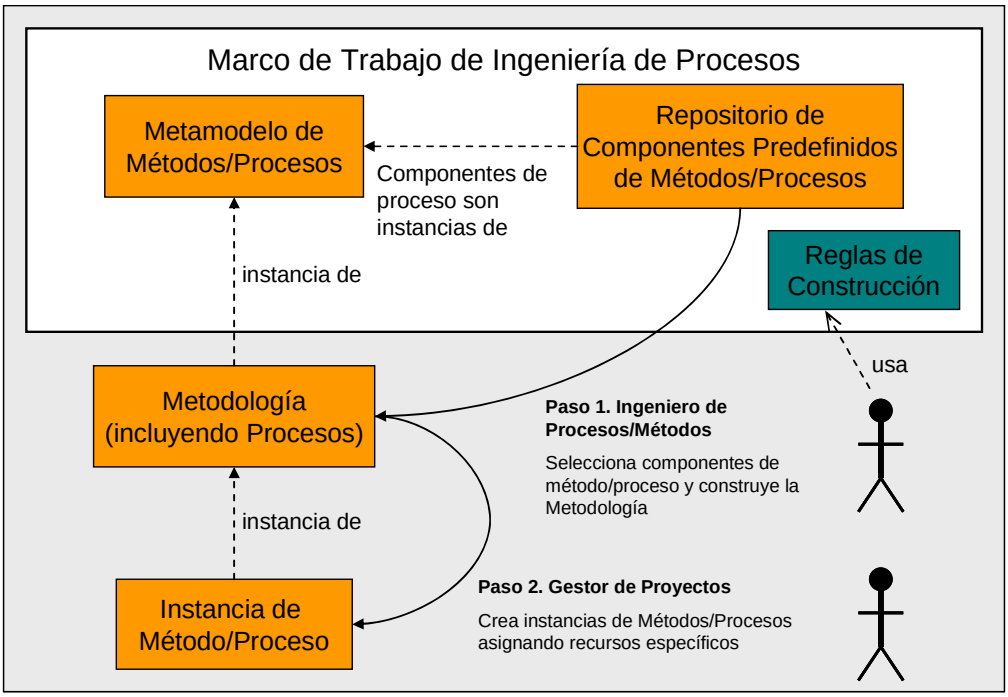


Figura 2. Ingeniería de Procesos.

3 Características Generales de SPEM 2

SPEM 2 (Software & Systems Process Engineering Metamodel Specification, v2.0) es un metamodelo para modelos de procesos de ingeniería del software y de ingeniería de sistemas. SPEM 2 sirve para definir procesos de desarrollo de software y sistemas y sus componentes. Su alcance se limita a los elementos mínimos necesarios para definir dichos procesos sin añadir características específicas de un dominio o disciplina particular; pero sirve para métodos y procesos de diferentes estilos, culturas, niveles de formalismo, o modelos de ciclos de vida. No es un lenguaje de modelado de procesos en general, ya que está orientado a los procesos software. No provee conceptos propios para modelado del comportamiento, pero incluye mecanismos para encajar el método externo elegido (diagramas de actividad de UML 2, BPMN/BPDM, ..).

La idea central de SPEM 2 para representar procesos está basada en tres elementos básicos: rol, producto de trabajo y tarea (ver Figura 3). Las tareas representan el esfuerzo a hacer, los roles representan quien lo hace y los productos de trabajo representan las entradas que se utilizan en las tareas y las salidas que se producen. La idea central subyacente es que un modelo de proceso consiste, básicamente, en decir quien (rol) realiza qué (tarea) para, a partir de unas entradas (productos de trabajo) obtener unas salidas (productos de trabajo).

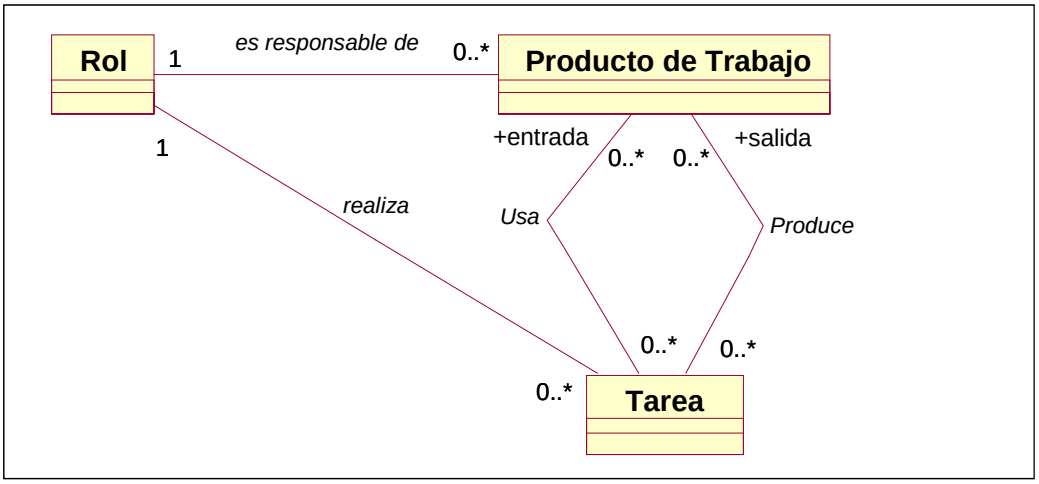


Figura 3. Idea básica de proceso en SPEM 2.

SPEM 2 puede ser una importante ayuda para que las empresas que llevan a cabo proyectos software pueden enfrentar mejor los problemas relacionados con los procesos. Entre dichos problemas, cabe destacar los siguientes:

- Miembros de los equipos no tienen acceso fácil y centralizado a la información de procesos que necesitan.
- Diferentes desarrolladores manejan diferentes fuentes o versiones de la misma información.
- Es difícil combinar e integrar informaciones y procesos que están en formatos propietarios diferentes.
- Cada libro, manual , herramienta utiliza un lenguaje y estilo diferente.
- Es duro definir una aproximación de desarrollo organizada y sistemática que se adapte a las necesidades.
- Cultura, prácticas establecidas, requisitos de certificación, legales, etc.

Además de un metamodelo para ingeniería de procesos, SPEM 2 también es un marco de trabajo conceptual que provee los conceptos necesarios para modelar, documentar, presentar, publicar, gestionar, intercambiar y realizar métodos y procesos software. Por ello, está destinado a ingenieros de procesos, jefes de proyectos, gestores de proyectos y programas; que son responsables de mantener e implementar procesos para sus organizaciones o para proyectos concretos.
La Figura 4 muestra un resumen del marco de trabajo general de SPEM, es decir, de los escenarios más habituales de su uso.

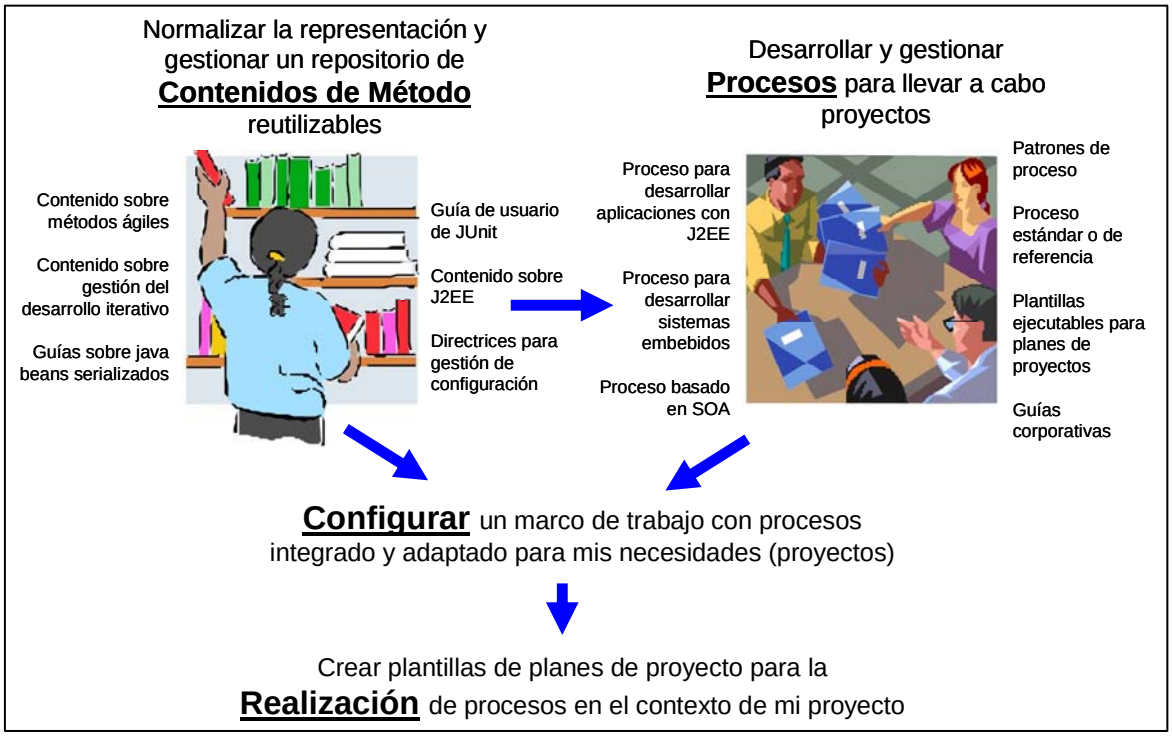


Figura 4. Marco de trabajo general con SPEM 2.

Al trabajar con SPEM existen 4 escenarios fundamentales:

a) Crear un repositorio de "contenidos de método" reutilizables, es decir, una colección organizada de roles, tareas, productos de trabajo, guías, fragmentos de método y procesos, etc.. Esto en sí solo ya es un valor para cualquier organización software, ya que supone disponer de un repositorio de conocimiento sobre procesos en un formato estandarizado. Además, es de gran ayuda a los desarrolladores de software porque en su trabajo necesitan conocer cómo hacer las tareas de desarrollo y mantenimiento de software, cómo gestionar el proyecto, cómo comprender los productos de trabajo que se deben crear en cada tarea, cuales son las habilidades requeridas en cada rol, y disponer de las guías, directrices, plantillas, etc. adecuadas en cada momento. Además, el repositorio basado en SPEM es una base de conocimiento ideal para la formación en procesos.

b) Dar soporte al desarrollo, gestión y crecimiento de procesos software. Esto implica combinar, reutilizar y extender los elementos de método anteriores para configurar los procesos que sirven para guiar los proyectos. Desde fragmentos de proceso elementales se puede llegar a generar todo un proceso completo o toda una metodología, incluyendo varios procesos. SPEM 2 ayuda a los equipos de desarrollo a definir o seleccionar un proceso, incluyendo opciones para que los mismos métodos puedan ser aplicados de forma diferente en momentos distintos o en proyectos distintos, para comprender claramente como se

relacionan unas tareas con otras, o para que los procesos puedan ser vistos como flujos de trabajo o como estructuras de desglose de trabajo (WBS), según interese en cada momento. Para lo anterior, SPEM soporta la creación sistemática de procesos basada en la reutilización de contenidos de método. También provee el fundamento conceptual para que los ingenieros de procesos y gestores de proyectos seleccionen, adapten y rápidamente ensamblen procesos para sus proyectos concretos. El ensamblado rápido de procesos es posible gracias a la implementación de catálogos de procesos predefinidos, "trozos" de proceso o patrones de proceso.

c) Establecer un marco de trabajo general de la organización a partir de los procesos y los elementos definidos anteriormente. Para ello, SPEM permite dar soporte al despliegue del contenido de método y proceso que justo se necesita en cada caso, teniendo en cuenta que ningún proyecto es exactamente como el anterior y nunca exactamente el mismo proceso software se ejecuta dos veces. En este punto es importante recordar que el nivel 3 de CMMI (proceso definido) necesita disponer de procesos estándares en la organización y de mecanismos de particularización (tailoring) y SPEM 2 provee de ambas cosas. Entre otras capacidades adicionales, SPEM 2 incorpora conceptos para:
   - Reutilización de procesos o patrones de procesos,
   - Variabilidad (procesos que incluyen partes alternativas configurables), y
   - Particularización (los usuarios definen sus propias extensiones, omisiones y puntos de variabilidad sobre procesos estándares reutilizados).

d) Generar plantillas para planes de proyecto concretos. Esto supone que a partir de ahora los jefes de proyecto pueden contar con mucha más información y disponible de manera automática a la hora de definir los planes de los proyectos. Para darles auténtico valor, las definiciones de los procesos deben ser desplegadas en formatos que permitan su realización automática (sistemas de gestión de proyectos y recursos, motores de flujos de trabajo, …). Para ello, SPEM incluye estructuras de definición de procesos que permiten expresar cómo un proceso será realizado de forma automática con estos sistemas. Ejemplos de ello son las iteraciones (una o varias definiciones de trabajo serán repetidas varias veces en un proyecto) y las ocurrencias múltiples (varias instancias de una definición de trabajo pueden llevarse a cabo a la vez de forma paralela).

3.1 Mecanismos de Trabajo con SPEM

Este estándar de OMG se describe de dos maneras (ver parte izquierda de la Figura 5):

- Como un Metamodelo MOF-compliant, es decir, un metamodelo MOF del nivel M2 de forma que todos sus elementos están definidos mediante instanciación de elementos del meta-metamodelo universal (del nivel M3). Este metamodelo define todas las estructuras y reglas de estructuración para representar contenidos de métodos y procesos. Es completo en sí mismo, es decir, no necesita de otro metamodelo, aunque en la práctica reutiliza algunas clases de UML 2 por razones de ahorro.

- Como Perfil de UML 2, es decir, como un conjunto de estereotipos UML 2 que permiten representar métodos y procesos usando UML 2. En este caso, la definición sólo abarca la presentación, mientras que las definiciones semánticas y restricciones están en el metamodelo anterior.

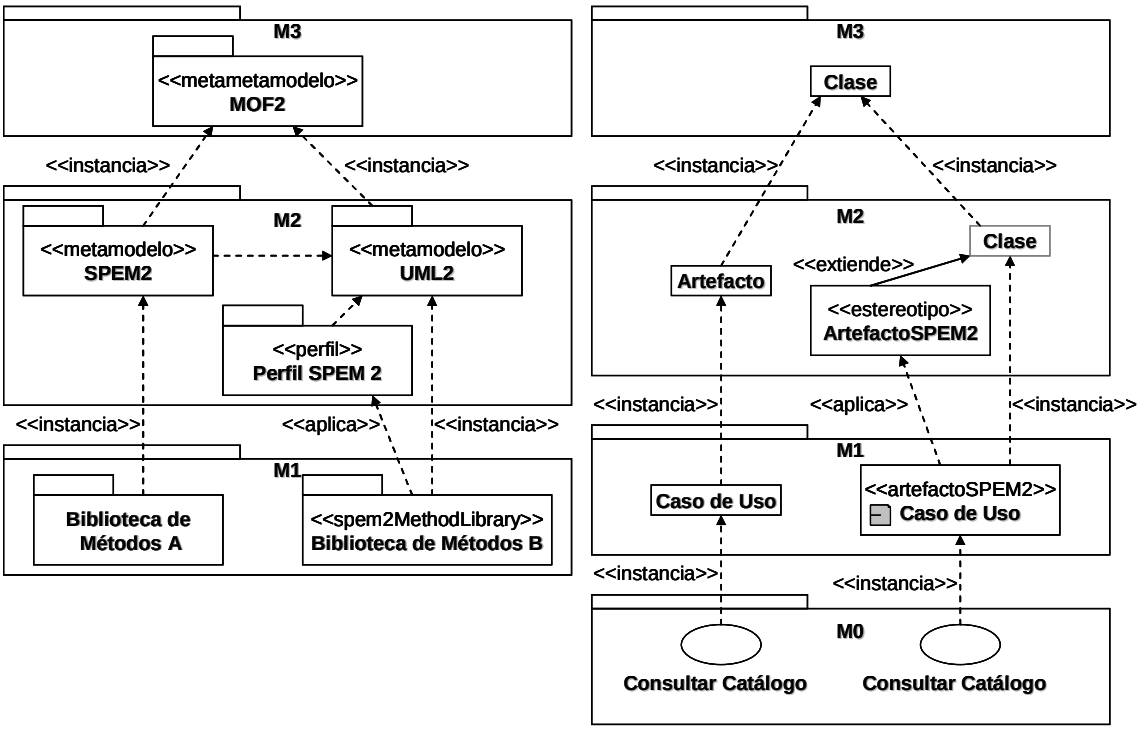


Figura 5. Niveles de modelado y ejemplos de instanciaciones.

Es posible crear bibliotecas de métodos, es decir, colecciones de elementos y fragmentos de procesos, con ambos mecanismos. Así, la "Biblioteca de Métodos A" en el nivel M1 de la Figura 5 es un ejemplo de instancia concreta (modelo) del metamodelo SPEM 2 usando SPEM 2 como un esquema para representar su contenido (primer mecanismo). Esto implica que, mientras que SPEM 2 define los conceptos de Rol, Tarea y Artefacto y las relaciones entre ellos, en la "Biblioteca de Métodos A" se incluyen instancias concretas de dichos conceptos, tales como "Analista de Sistema" (instancia de Rol) y "Caso de Uso" (instancia de Artefacto). En el lado derecho de la Figura 5 puede comprobarse que "Caso de Uso" (nivel M1) es una instancia directa de la metaclase "Artefacto" (nivel M2) de SPEM2, que a su vez es instancia de la meta-metaclase "Clase" de la capa M3. Por último, una instancia de "Caso de Uso" que se podría crear durante un proyecto concreto de desarrollo de un sistema software sería "Consultar Catálogo" (nivel M0).

El empleo del perfil UML tiene la ventaja de que no es necesario desarrollar herramientas CASE especiales para crear y mantener los métodos y procesos, bastando con usar una herramienta genérica de modelado UML. Pero esta opción tiene una importante desventaja: las reglas específicas de estructuración de SPEM, que sí están en el metamodelo pero no en el perfil, no pueden ser manejadas con herramientas genéricas basadas en UML, salvo que se empleen restricciones OCL, lo que supone aumentar significativamente el nivel de dificultad de UML. Por ejemplo, para la asociación entre roles y tareas, no es posible controlar que debe haber siempre un realizador principal y, opcionalmente, varios adicionales. Además, ya existen varias herramientas disponibles para editar métodos y procesos en SPEM 2, entre las que se encuentra Eclipse Process Framework Composer. Por todo ello, de manera general, es mejor opción utilizar SPEM como metamodelo en vez de cómo perfil de UML 2.

Lo estereotipos incluidos en el perfil UML 2 de SPEM incluyen iconos para la representación visual. Esto permite utilizar diagramas UML con dichos estereotipos para representar los métodos y procesos. La Figura 6 muestra el contexto de la tarea "Use Case Analysis", indicando los roles que intervienen (System Analista y Designer) y los productos de entrada (Analysis

Model, Use Case) y de salida (Analysis Model y Use Case Realization). En apartados posteriores se mostrará como la herramienta EPFC hace uso de estos mismos iconos para identificar también a los elementos del metamodelo de manera visual al generar documentación web.

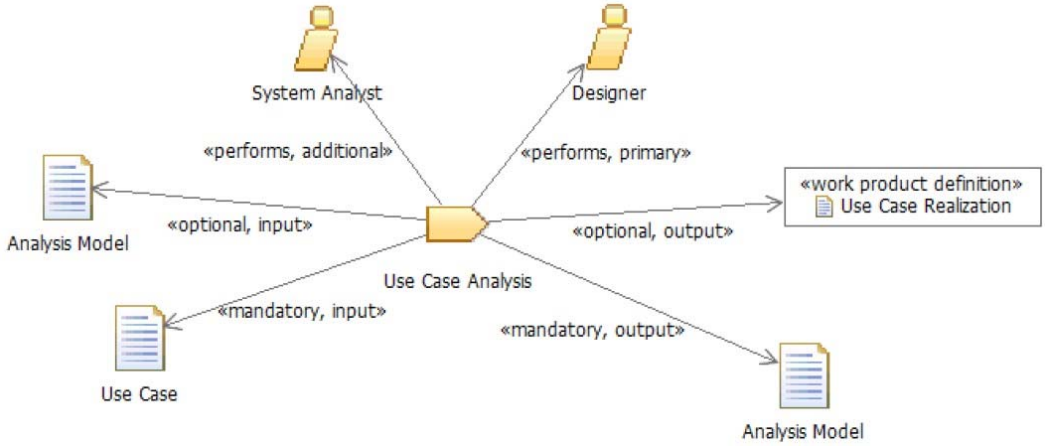


Figura 6. Ejemplo de representación gráfica de procesos usando el perfil UML de SPEM.

De ahora en adelante, esta guía se dedica exclusivamente al metamodelo basado en MOF, es decir, al primer y más potente mecanismo de uso de SPEM.

4 Metamodelo SPEM

Tal como se deduce del marco de trabajo (ver Figura 4), en SPEM 2 se distinguen dos grupos de conceptos a la hora de implementar una metodología (ver Figura 7 y Figura 8):

a) Primero, se puebla el Method Content (contenido del método) con Content Elements (elementos de contenido), es decir, los elementos primarios o constructores básicos.
b) Después, se combinan y reutilizan dichos elementos para obtener Processes (procesos).

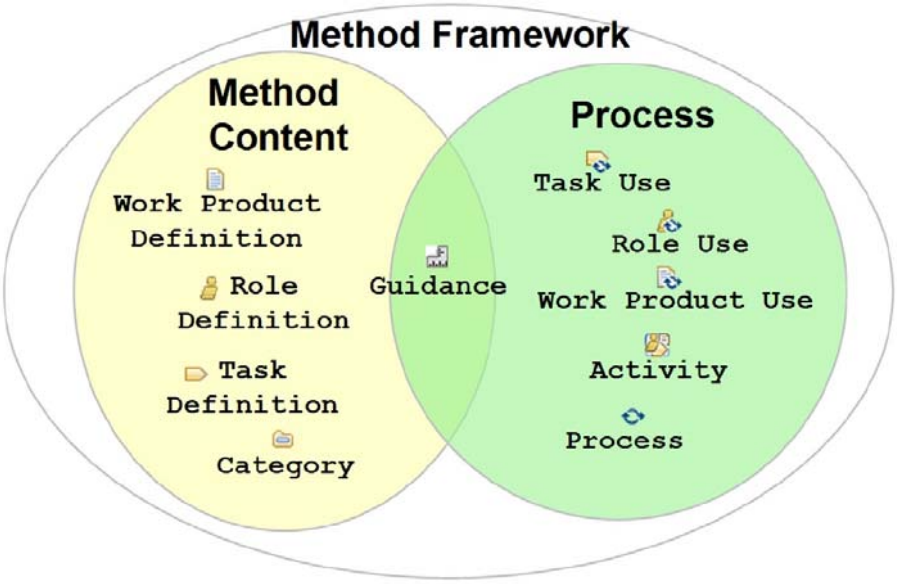


Figura 7. Aspectos principales para modelar con SPEM.

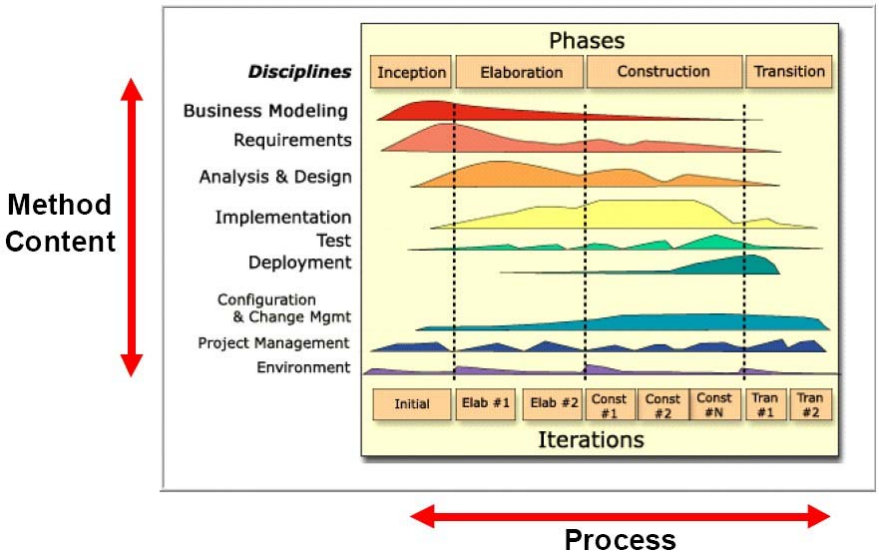


Figura 8. Ortogonalidad del Method Content y de los Processes (igual que ocurre en RUP).

En SPEM 2 la jerarquía de desglose del trabajo, es decir, los conceptos existentes para representar el esfuerzo a realizar a distintos niveles de detalle, son los siguientes del más general al más particular (ver Figura 9):

- Delivery Process (proceso de despliegue): Representa un proceso tal complejo como se necesite, que será el que sirva de base para realizar cierto tipo de proyectos.
- Capability Pattern (patrón de capacidad): representa un patrón de proceso, es decir, un fragmento de proceso que puede ser reutilizado más de una vez en un delivery process.
- Activity (actividad): en el elemento central para definir procesos ya que permite organizar los elementos básicos de proceso (roles, productos de trabajo y tareas).
- Task (tarea): es la porción más pequeña de trabajo en un modelo de proceso en SPEM 2.

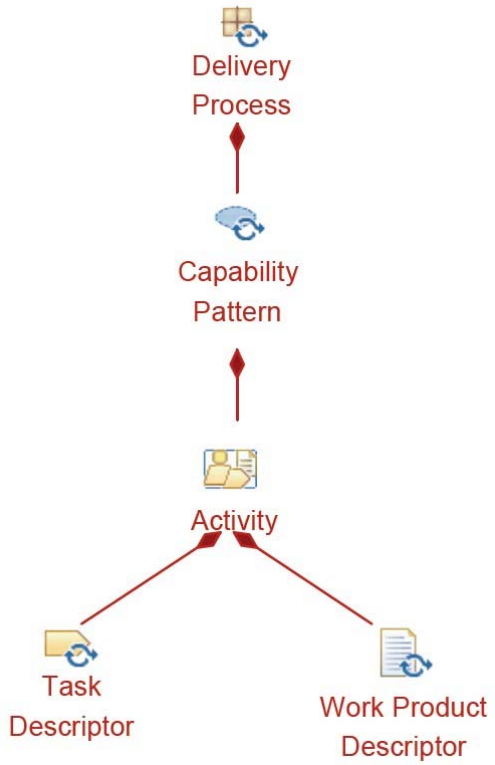


Figura 9. Conceptos para representar la jerarquía de desglose del trabajo.

Además de la separación clara entre la definición de contenidos de método y su aplicación en procesos, ya comentada, otras características avanzadas del metamodelo SPEM 2 son:

- Mantenimiento consistente de muchos procesos alternativos. Para ello, SPEM incluye: i) un conjunto extendido de interrelaciones de reutilización y variabilidad con semántica de herencia y orientación a aspectos; ii) conceptos de patrones de proceso, y iii) plugins de métodos. Estas opciones permiten tener diferentes variantes de procesos específicos, basados en los mismos contenidos de método y estructuras de procesos, pero aplicados con diferente detalle y escala.
- Muchos ciclos de vida diferentes. SPEM permite trabajar con distintos tipos de ciclos de vida del software: Cascada, Iterativo, Incremental, Evolutivo, etc. Para ello incluye un conjunto de atributos que permiten especificar aspectos temporales para los elementos de proceso que luego pueden ser asociados a los planes de proyectos. Un ejemplo de atributo para clases de ciclos de vida es la propiedad Iteración, que permite representar que la ejecución de una o varias descripciones se trabajo se puede repetir más de una vez.
- Variabilidad y extensibilidad. Para lo cual SPEM incluye un mecanismo de plugins de dos tipos (ver Figura 10): a) Method plug-ins para particularizar y adaptar contenidos de método sin modificar el original; y b) Process plug-ins para procesos, pudiendo añadir o sustituir elementos de trabajo en el WBS (estructura de desglose de trabajo del proceso) sin afectar al original.

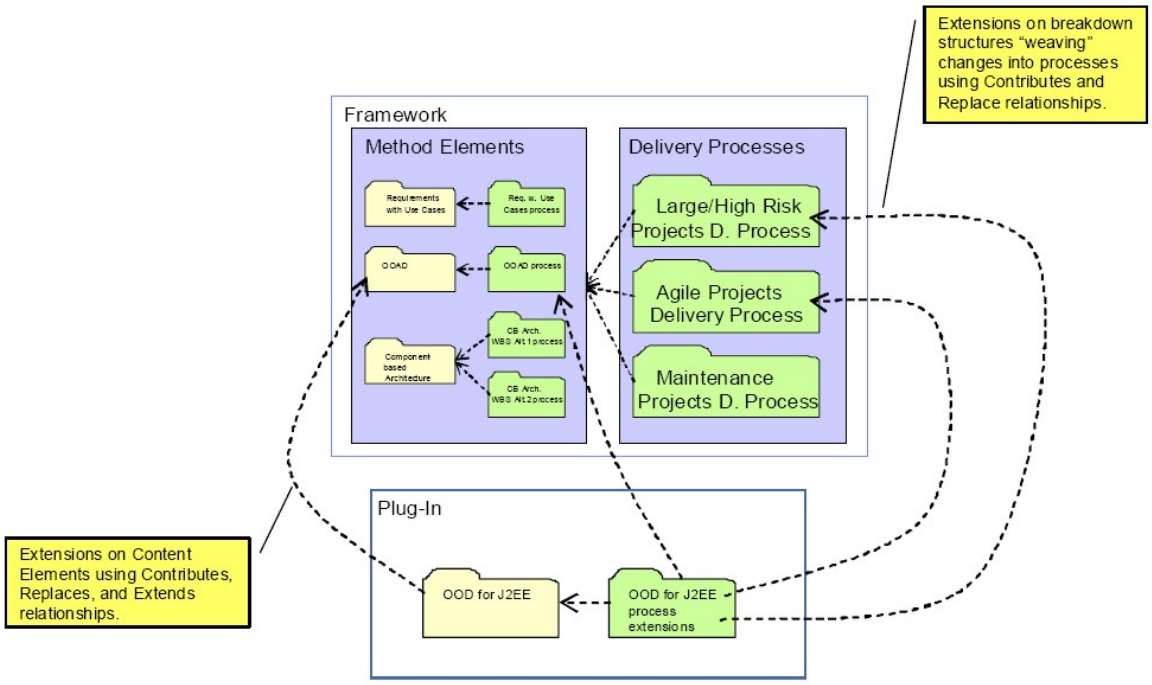


Figura 10. Ejemplo del mecanismo de plugins en SPEM para permitir variabilidad y extensibilidad.

- Patrones de proceso. Son bloques (trozos de proceso) reutilizables para crear nuevos procesos. La selección y aplicación de un patrón de proceso puede ser hecha de dos formas: a) puede ser copiado y modificado, permitiendo individualizar el contenido del patrón según las necesidades de cada momento; o b) puede ser aplicado por medio del mecanismo de Actividad en Uso (Use Activity), que es una forma avanzada de reutilizar estructuras de proceso. Una Actividad en Uso define tipos de interrelaciones para que cuando el patrón esté siendo revisado o modificado, todos los cambios se reflejen automáticamente en todos los procesos en que se aplica el patrón.
- Componentes de proceso. Son piezas de proceso sustituibles y reutilizables basadas en los principios de encapsulación y caja negra. No se especifica la descripción de trabajo interna del componente, sino que sólo se especifican los productos de trabajo de entrada y salida que habrá mediante los llamados puertos de productos de trabajo (ver Figura 11). Esta opción de SPEM permite manejar las situaciones en que un proyecto requiere que partes del proceso no sean decididas hasta la ejecución o nunca (caso típico: outsourcing).

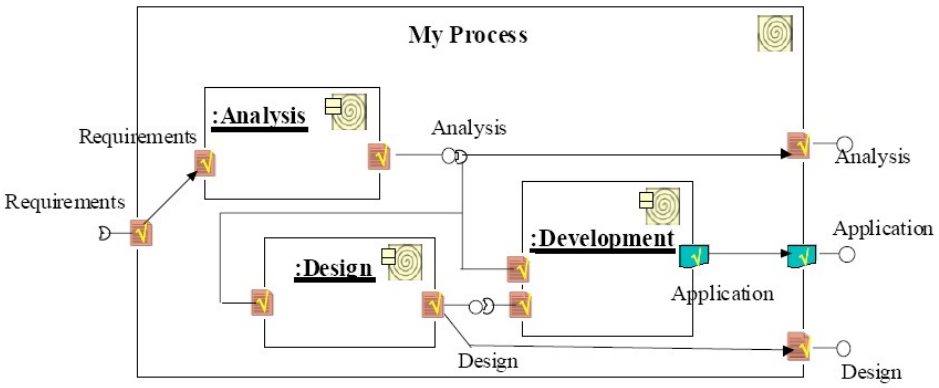


Figura 11. Ejemplo de Componente de Proceso en SPEM 2.

4.1 Estructura de Paquetes

Al igual que UML, con fines de organización y reutilización, SPEM2 está organizado en 7 paquetes (packages), que se muestran en la Figura 12. Cada paquete es una unidad lógica que extiende los paquetes de los que depende, proveyendo estructuras y capacidades adicionales. Los paquetes definidos en un nivel inferior pueden ser realizados en una implementación parcial de SPEM 2 sin necesidad de los paquetes de los niveles superiores. Como regla general, cada clase del metamodelo (constructor) se incluye en el paquete del nivel más inferior posible. En algunos casos, las clases se extienden (especializan) en paquetes de nivel superior vía el mecanismo "merge" (idéntico al de UML 2) para incluir más propiedades e interrelaciones. Un ingeniero de procesos puede elegir utilizar diferentes niveles de capacidades, conjuntos de conceptos, y niveles de formalismo para expresar sus procesos utilizando unos u otros paquetes.

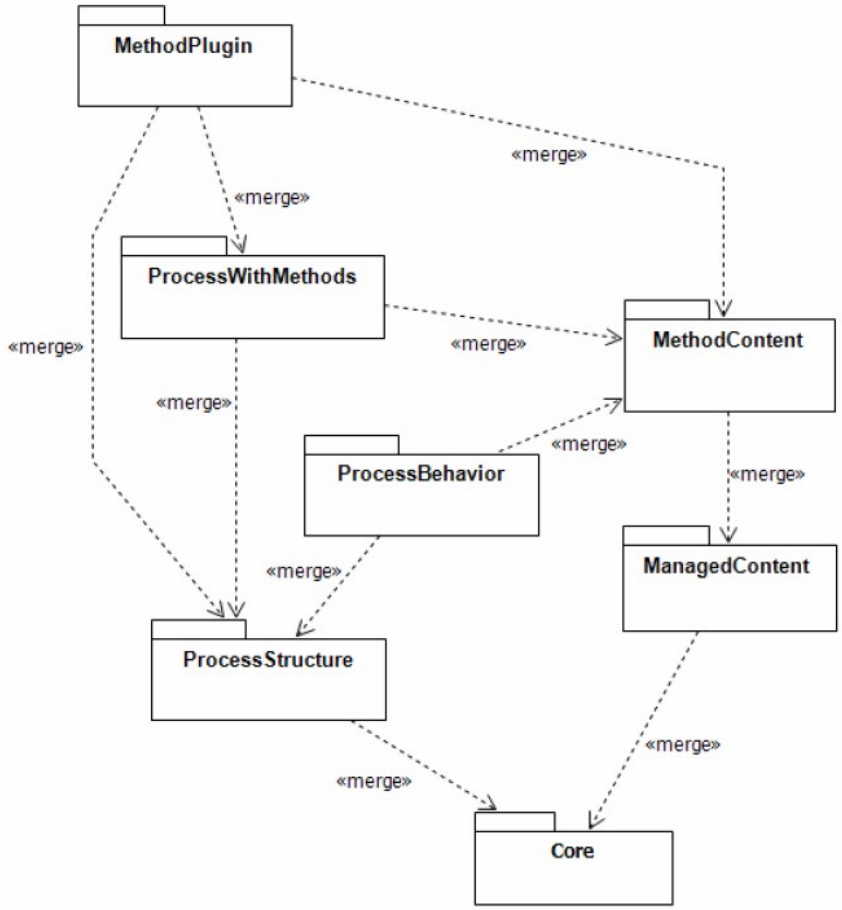


Figura 12. Estructura de paquetes de SPEM 2.

De forma resumida, los contenidos de cada paquete son los siguientes:

- Core: Contiene las clases y abstracciones que sirven de base para las clases de los demás paquetes. Provee dos capacidades básicas: a) crear cualificaciones definidas por el usuario (kinds) que permiten establecer tipos diferentes entre las instancias de una clase; y b) una colección de clases abstractas para definir el trabajo.
- Process Structure: Define la base para la creación de modelos de proceso (MP) flexibles y sencillos, es decir, define la estructura de desglose de trabajo estática mediante anidamiento de actividades y dependencias de precedencia entre ellas. Dicha estructura también incluye referencias a la lista de Roles que realizan cada actividad y a los

Productos de Trabajo que son entradas y/o salidas. También provee capacidades para reutilización mediante ensamblado de procesos usando conjuntos de actividades enlazadas de forma dinámica.

- Process Behavior: Permite extender las estructuras del paquete anterior con modelos de comportamiento externos: diagramas de actividad de UML 2 (comportamiento de proceso), máquina de estados (ciclo de vida de un producto de trabajo), etc. En vez de incluir un mecanismo propio para representar el comportamiento, se opta por reutilizar los ya existentes (de UML o de otro tipo).
- Managed Content: Permite incorporar y gestionar descripciones en lenguaje natural, documentos y otras informaciones útiles para la comprensión por humanos. Esta posibilidad es muy interesante y útil porque ciertos valores y culturas no pueden ser formalizados con modelos, sólo pueden ser capturados con documentación en lenguaje natural. Hay libertad total para combinar modelos estructurales de proceso con contenidos en lenguaje natural. Así, un proceso puede estar formado sólo por una colección de guías definiendo buenas prácticas (esto es especialmente apto en el caso de métodos ágiles poco estructurados); sólo una estructura de actividades sin ningún tipo de documento textual; o una combinación interrelacionada de ambas cosas.
- Method Content: Incluye los conceptos para crear elementos de método, que son los items elementales que sirven de base para el ensamblado de procesos, metodologías, ciclos de vida, etc. Los principales elementos de método se derivan del patrón básico de SPEM: alguien (rol) hace algo (tarea) para obtener algo (producto de trabajo) basándose o ayudándose en algo (guía). Por tanto, los elementos de método permiten describir, con el detalle necesario, cómo se alcanzan los objetivos del proceso haciendo qué tareas por qué roles usando qué recursos y obteniendo qué resultados. Los procesos reutilizan y relacionan entre sí los elementos de método de distintas maneras para diferentes tipos de proyectos. Los procesos ad-hoc tienen estructura de descomposición pero no reutilizan elementos de método, al no estar previamente definidos. Por su especial importancia, los elementos de este paquete se explican en detalle en el apartado 4.3.
- Process with Methods: Incorpora nuevas estructuras para poder integrar los procesos definidos con el paquete “Process Structure” con los elementos de método (instancias del paquete “Method Content). Al asociar elementos de método a partes específicas de procesos, se crean nuevas clases (tarea en uso, rol en uso, etc.) que heredan de los elementos de método pero pueden tener cambios individualizados.
- Method Plugin: Incorpora conceptos (Method Plu-gin, Process Component, Variability, …) para diseñar y gestionar bibliotecas o repositorios de contenidos de método y de procesos, que sean mantenibles a gran escala, reutilizables y configurables. Con estos conceptos los ingenieros de procesos pueden definir una o varias Configuraciones de Método (ver apartado 5.4), de cada proceso. Esto permite tener vistas diferentes de un mismo proceso adaptadas a distintas audiencias o tipos de usuarios.

Utilizando unos u otros paquetes, un ingeniero de procesos puede disponer de diferentes capacidades, conjuntos de conceptos y niveles de formalismo para expresar sus procesos. Los escenarios más habituales son los siguientes:

a) Core + Managed Content + Method Content: En este caso no hay modelos de proceso formales definidos, sino tan solo un repositorio para gestionar la documentación de descripciones de métodos de desarrollo y mantenimiento, técnicas y mejores prácticas.
b) Anterior + Process Structure + Process Behavior: Ahora ya existe un proceso definido pero no son necesarios mecanismos avanzados para organizar o gestionar un repositorio de métodos, o para permitir diferentes vistas de un mismo proceso.
c) Los 7 paquetes completos se emplean cuando se quiere disponer de toda la potencia y funcionalidad de SPEM 2.

4.2 Organización de un Repositorio SPEM 2

Un repositorio o biblioteca de métodos y procesos (Method Library) en SPEM 2 es una colección de uno o más plugins y una o varias configuraciones (Configuration). Cada plugin se almacena en un directorio de disco diferente e incluye contenido de método (Method Content) y procesos (Processes), cada uno de los cuales se refiere a las dos partes, izquierda y derecha, mostradas en la Figura 7. A su vez, el contenido de método está formado por paquetes de contenido (Content Package), categorías estándar (Standard Category) y categorías personalizadas (Custom Category); y el apartado de Procesos contiene patrones de proceso (Capability Pattern) y procesos para despliegue (Delivery Process). Un ejemplo de toda esta estructura se muestra en la Figura 13.

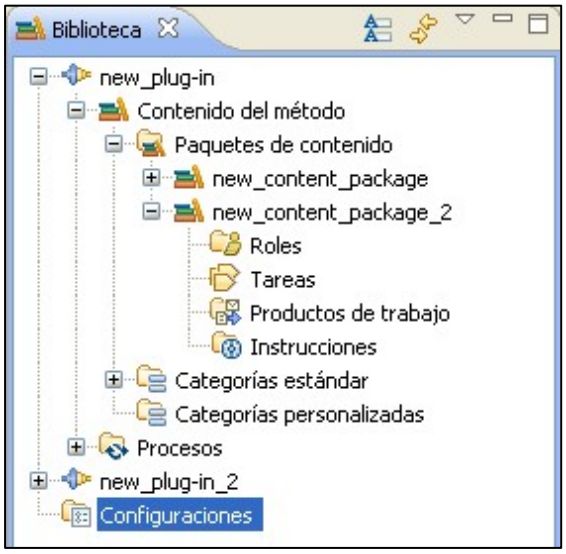


Figura 13. Ejemplo de Jerarquía para organizar un repositorio (biblioteca) SPEM 2.

4.3 Contenido de Método

A continuación se presentan los conceptos del Contenido de Método (Method Content) de SPEM 2. No es una revisión exhaustiva, ya que no se incluyen aquellos conceptos que quedan "ocultos" al usuario cuando modela procesos con algún editor de SPEM, por ejemplo, con EPF Composer. Los conceptos que se presentan corresponden principalmente con los paquetes "Managed Content", "Method Content", "Process with Methods" (en parte) y con el "Base Plug-in", que es un plugin preconstruido incluyendo diversas especializaciones predefinidas en SPEM 2 para actividad, categoría, guía y producto de trabajo.

El contenido de método puede ser organizado a voluntad del usuario mediante una jerarquía de paquetes de contenido (Content Package), cada uno de los cuales puede incluir roles, tareas, productos de trabajo y guías [4] (ver Figura 13).

En la Figura 14 se muestra la jerarquía completa de conceptos empleados en el Method Content, incluidos los elementos de contenido (Content Element).

[4] En EPF Composer 1.2 se llaman "instrucciones".

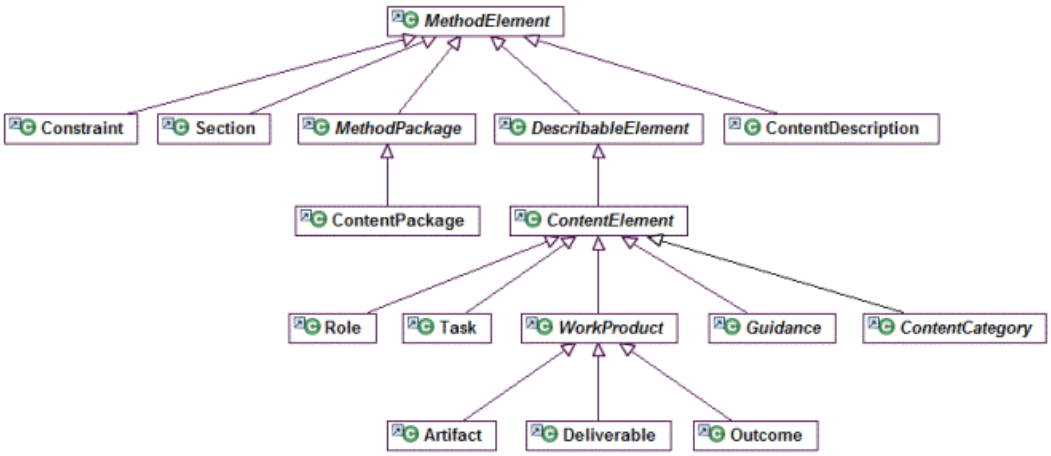


Figura 14. Jerarquía de conceptos del method content.

Los elementos describibles (Describable Element) son elementos de método (o de proceso) que pueden tener descripciones textuales mediante una Descripción de Contenido (Content Derscription) opcional, que incluye nombre de presentación, descripción breve, descripción principal (que permite texto enriquecido), y propósito. Un elemento describible puede tener 0..* Secciones (Section), que a su vez pueden estar anidadas. Además, puede estar asociado con 0..* Guías (Guidance) y con 0..* Categorías (Category). Éstas últimas sirven para poder clasificarlos y agruparlos de múltiples formas.

4.3.1 Elementos de Contenido

Los elementos de contenido (Content Element) son los constructores básicos y se derivan del patrón primitivo de trabajo: alguien (rol) hace algo (tarea) para obtener algo (producto de trabajo) basándose o ayudándose en algo (guía). En consecuencia, los cuatro tipos de elementos de contenido son: Tarea, Rol, Producto de Trabajo y Guía. Adicionalmente, existe un quinto tipo de elemento de contenido incluido con fines de clasificación y agrupación, llamado Categorías. En los siguientes subapartados se presenta cada uno de ellos.

4.3.1.1 Tareas

Una Tarea (Task Definition) describe una unidad de trabajo asignable y gestionable, es decir, es la unidad atómica de trabajo para definir procesos. Su granularidad es de unas pocas horas a unos pocos días, afectando a unos pocos productos de trabajo y vinculando a unos pocos roles. Es un Elemento de Método que define el trabajo realizado por roles, pero también es una Definición de Trabajo (en procesos). Una Tarea está asociada con (ver ejemplo en Figura 15):

- 1..* Roles, distinguiendo entre:
  - 1 realizador principal obligatorio [responsable]
  - 0..* realizadores adicionales opcionales
- 1..* Productos de Trabajo como:
  - Entradas obligatorias
  - Entradas opcionales
  - Salidas
- 0..* Herramientas, que se recomienda usar.
- 0..* Pasos, que describen de forma secuencial el trabajo a realizar.
- 0..* Habilidades, que se requieren habitualmente para llevar a cabo la tarea.

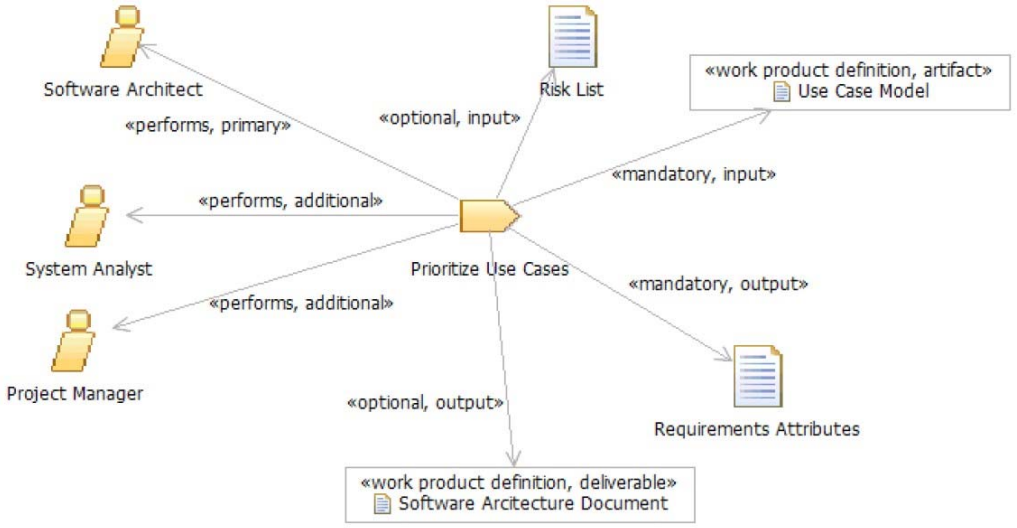


Figura 15. Ejemplo de asociaciones de una tarea.

4.3.1.2 Roles

Un Rol (Role Definition) define un conjunto de habilidades, competencias y responsabilidades relacionadas, de un individuo o de un grupo. No se deben confundir roles con personas, ya que la vinculación entre personas y roles se realiza durante la planificación del proyecto y puede ocurrir que un individuo desempeñe varios roles o que un rol sea desempeñado por varios individuos. Un rol es un Elemento de Método usado en las Definiciones de Tareas para señalar quienes las realizan. Un Rol está asociado con:

- 0..* Productos de Trabajo, de los que es responsable.
- 0..* Habilidades, que el rol típicamente provee.

4.3.1.3 Productos de Trabajo

Un Producto de Trabajo (Work Product Definition) es consumido, producido o modificado por Tareas. Un Producto de Trabajo puede estar asociado con otros 0..* Productos de Trabajo mediante asociaciones de los siguientes tipos:

- Composición (Composition), cuando las instancias de un producto de trabajo sirven para componer instancias de otro producto de trabajo. Ejemplo: "Actores" se emplean para componer "Casos de uso".
- Agregación (Aggregation), si un producto de trabajo está formado por agregación de otros. Ejemplo: el "Manual de usuario" incluye el "Manual de instalación".
- Es impactado por (Impacte by), cuando un producto de trabajo impacta en otro, es decir, si los cambios del primero obligan a cambiar el segundo. Ejemplo: si cambia el "Modelo de casos de uso", es necesario adaptar a dicho cambio la "Realización de casos de uso".

Existen tres tipos predefinidos (especializaciones) de Productos de Trabajo:

- Artefacto (Artifact): de naturaleza tangible (modelo, documento, código, archivos, ..). Un artefacto puede estar formados por otros artefactos más simples.
- Entregable [5] (Deliverable): provee una descripción y definición para empaquetar otros productos de trabajo con fines de entrega a un cliente interno o externo. Representa una salida de un proceso que tiene valor para un usuario, cliente u otro participante

[5] La versión en castellano de EPF Composer le denomina "producto final". Aquí optamos por emplear el término "Entregable", que es el utilizado por los diversos estándares de ingeniería del software y de gestión de proyectos.

(*stakeholder*). Esta asociado con 0.* componentes de entregable (Deliverable Component), que son los productos de trabajo, habitualmente artefactos, que lo forman.

- Resultado (Outcome): un producto de trabajo de naturaleza intangible (resultado o estado), o que no está formalmente definido.

4.3.1.4 Guías

Una guía o instrucción (Guidance) es un elemento de método (o de proceso) que provee información adicional relacionada con otros elementos. Por ejemplo: ayuda o información sobre cómo trabaja un rol, cómo crear un producto de trabajo, cómo usar una herramienta o cómo realizar una tarea.
SPEM 2 tiene predefinidos bastantes tipos de guías:

- Activo Reutilizable (Reusable Asset): Provee una solución a un problema para un contexto dado. Incluye reglas o instrucciones sobre cómo utilizarlo.
- Concepto (Concept): Resumen de ideas clave asociadas con principios básicos subyacentes. Refieren a tópicos más generales que las directrices y abarcan varios productos de trabajo y/o actividades. Ejemplo: "iterativo e incremental".
- Definición de Término (Term Definition): Definición de un término, concepto o idea relevante. Sirve para generar una especie de glosario automático. Se relacionan con Elementos de Contenido mediante su aparición en las descripciones textuales.
- Directriz [6] (Guideline): Provee detalle adicional sobre cómo realizar una tarea o grupo de tareas, o provee información adicional, reglas y recomendaciones sobre productos de trabajo. Entre otros, puede incluir detalles sobre: las mejores prácticas y aproximaciones diferentes para hacer un trabajo; cómo usar cierto tipo de producto de trabajo; los subtipos y variantes de un artefacto y su evolución a lo largo del tiempo; habilidades que los roles deben adquirir o mejorar; medidas del progreso o madurez, etc.
- Documentación [7] (Whitepaper): Versión especial de Concepto que ha sido revisada o publicada externamente y que puede ser leída y comprendida de forma aislada.
- Ejemplo (Example): Ejemplo de una instancia típica, parcialmente completada, de uno o más productos de trabajo o descripción del escenario en que una tarea debe ser realizada.
- Guía de Herramienta (Tool Mentor): Explica el uso de una cierta herramienta en el contexto de cierto trabajo, o de forma independiente.
- Guía para la Estimación [8] (Estimation Considerations): Indicaciones para estimar el esfuerzo asociado con cierto trabajo, incluyendo consideraciones sobre cómo hacer la estimación y las métricas a utilizar.
- Hoja de Ruta [9] (Roadmap): Hoja de ruta que describe, en forma de camino lineal, cómo suele llevarse a cabo una actividad o proceso complejos. Provee información sobre como las actividades y tareas se relacionan entre sí a lo largo del tiempo. Sólo pueden estar asociados a Actividades y Procesos.
- Informe (Report): Plantilla predefinida de un resultado que se obtiene de forma automática mediante alguna herramienta.
- Lista de Comprobación (Checklist): Identifica una serie de ítems que deben ser completados o verificados.

[6] También llamada instrucción en EPF Composer.
[7] La traducción equivalente en castellano sería "libro blanco", pero se deja el término documentación por ser más general.
[8] La versión en castellano de EPF Composer le llama "Consideraciones para el Cálculo", pero es mas correcto denominarle "Guía para la estimación", ya que no es lo mismo estimar (aproximación a un valor futuro, por fuerza inexacto) que calcular (con resultado exacto salvo error).
[9] La versión en castellano de EPF le denomina "mapa". Aquí optamos por emplear el término hoja de ruta, que es como se le conoce habitualmente y cómo figura en el diccionario de la RAE.

- Material de Soporte (Supporting Material): Comodín para utilizar cuando se está en un caso que no encaja en ninguno de los demás tipos de guías.
- Plantilla (Template): Establece la tabla de contenidos, secciones, cabeceras y formato estandarizado predefinido de un artefacto (documentos, modelos, …). Puede incluir descripciones sobre cómo usar y completar cada parte.
- Práctica (Practice): Manera o estrategia predefinida de hacer un trabajo que tiene un impacto positivo sobre la calidad de un producto de trabajo o de un proceso. Son ortogonales a los métodos y procesos, de forma que una práctica resume aspectos que pueden impactar en diferentes partes de un método o proceso. Ejemplos: "gestionar riesgos", "verificación continua de la calidad", "desarrollo centrado en la arquitectura", "desarrollo basado en componentes".

4.3.1.5 Categorías

Una Categoría (Category) es un elemento de contenido, o de proceso, usado para categorizar, es decir, clasificar o agrupar dichos elementos en base a los criterios que desee el ingeniero de procesos. Una categoría puede tener 0..* subcategorías. Esto permite establecer cualquier tipo de jerarquía de agrupamiento de elementos. SPEM 2 distingue dos clases de categorías:

- Estándar (Standard Category): Vienen predefinidas en SPEM. Se comentan después.
- Personalizada (Custom Category): Sirven para que el ingeniero de procesos pueda definir otras categorías nuevas.

La Figura 16 muestra un ejemplo del uso de varias categorías personalizadas y varias estándar para agrupar elementos de contenido: definiciones de método, directrices y artefacto.

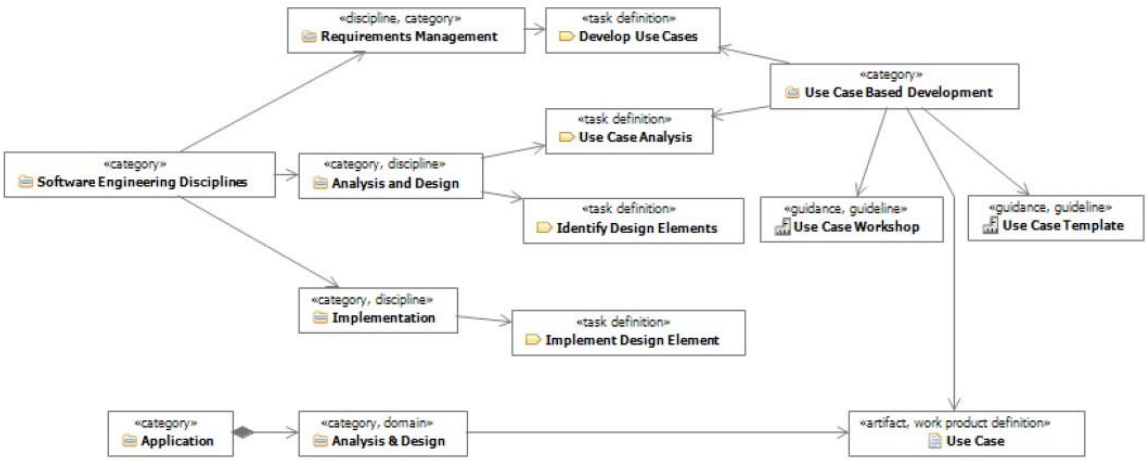


Figura 16. Ejemplo de uso de categorías.

En SPEM 2 se incluyen 5 tipos predefinidos de categorías:

- Conjunto de Roles (Rol Set): Sirven para agrupar roles que tienen algo en común (usan técnicas similares, requieren habilidades parecidas, …). Ejemplo: Analista englobando a Analista de Sistemas e Ingeniero de Requisitos.
- Disciplina (Discipline): Permiten categorizar el trabajo (tareas). Una disciplina es una colección de tareas que están relacionadas con un área principal de esfuerzo dentro de un proyecto completo. Suelen estar basadas en una perspectiva tradicional de proyectos en cascada: requisitos, análisis, diseño, construcción, pruebas, mantenimiento, gestión del proyecto, aseguramiento de calidad, etc.
- Dominio (Domain): Permiten establecer una jerarquía de dominios, para clasificar productos de trabajo, con tantos niveles como se desee. El nivel inferior son productos

de trabajo y el resto de niveles son dominios y subdominios. Al ser una jerarquía, un producto de trabajo sólo puede estar asociado con un único dominio. Ejemplos: "modelo", "código".

- Herramienta (Tool): A pesar del nombre, no sirve para categorizar herramientas sino guías de herramientas. Por tanto, para asociar herramientas con tareas, roles o productos de trabajo deberá emplearse una categoría personalizada.
- Clase de Producto de Trabajo (Work Product Kind): Se incluye por compatibilidad con la versión 1 de SPEM. Se distingue de Dominio en que un producto de trabajo puede pertenecer a varias clases de producto de trabajo distintas. Ejemplo: El mismo artefacto puede incluirse dentro de "Documento de Análisis" y dentro de "Producto Software".

4.3.2 Propiedades Principales

Todos los content elements de todos los tipos anteriores tienen las propiedades ya comentadas de los elementos describibles, incluyendo: name (nombre); presentation name (nombre de presentación); brief description (descripción breve); main description (descripción principal). La última permite utilizar un "rich text editor", pudiendo incluir texto con formatos variados, imágenes, etc., al estilo de HTML.

Otras propiedades específicas de cada clase de content element son:

- Task: purpose (objetivo); key considerations (factores clave); alternatives (alternativas); steps (lista de pasos), que detalla el trabajo a realizar de forma ordenada.
- Role: key considerations (factores clave); skills (habilidades); assignment approaches (propuestas de asignación); synonyms (sinónimos).
- Work Product: unique ID (ID exclusivo); purpose (objetivo); key considerations (factores clave); impact of not having (impacto de no tener); reason for not needing (motivos para no necesitar). Además, los Artifacts tienen brief outline (esquematización breve), y representation options (opciones de representación). Los Deliverables también tienen external description (descripción externa), packaging guidance (guía de empaquetado), y deliverable parts (componentes de entregable), que es una lista de otros work products que forman parte del deliverable.
- Guidance: algunos tipos de guidances tienen otros atributos. Así, Checklist tiene una lista de check ítems (elementos de comprobación), y Practice tiene additional information (información adicional), goals (objetivos), application (aplicación), problem (problema), background (fondo), level of adoption (nivel de adopción), referentes (referencias), que es una lista de content elements referenciados. Por otro lado, un Template tiene una lista de template files (archivos de plantilla).

Cada clase de content element tiene un icono predefinido por el perfil UML de SPEM, que lo caracteriza en los diagramas y documentación utilizados o generados también por las diversas herramientas (como EPF Composer). Con fines de personalización, a cada tipo de work product y guidance se le puede asociar otro icono específico.

4.3.3 Asociaciones

Las asociaciones permitidas en SPEM entre los diversos tipos de content elements se muestran en la Figura 17. De forma resumida, las relaciones que el ingeniero de procesos puede representar con SPEM 2, son las siguientes:

- Task-Steps: lista ordenadas de pasos que se llevan a cabo en una tarea.
- Task-Roles: primary performer (realizador principal, obligatorio); additional performers (realizadores adicionales).
- Task-Work Products: mandatory inputs (entradas obligatorias); optional inputs (entradas opcionales); outputs (salidas).

- Task-Guidances: guidances relacionadas (sólo tipos checklist, concept, estimating guideline, example, guideline, reusable asset, supporting material, y tool mentor).
- Task-Categories: disciplines y custom categories (ver apartado 4.3.1.5) a las que pertenece la task.
- Role-Work Products: responsible for (responsable de); work products that are output of tasks that this role perfoms (productos de trabajo que son salida de tareas que realiza este rol).
- Role-Guidances: guidances relacionadas (sólo tipos checklist, concept, example, guideline, reusable asset, y supporting material).
- Role-Categories: role sets y custom categories (ver apartado 4.3.1.5) a las que pertenece el role.
- Work Product-Guidances: guidances relacionadas (sólo tipos checklist, concept, estimating guideline example, guideline, report, reusable asset, supporting material, template, y tool mentor).
- Work Product-Categories: domains, work product kinds y custom categories (ver apartado 4.3.1.5) a las que pertenece el work product.
- Guidance-Guidances: lista de guidances (de tipo checklist, concept, example, guideline, reusable asset y supporting material) que pertenecen al guidance principal. Todos los tipos de guidance pueden incluir otros guidance, salvo el tipo practice que en lugar de incluir otros guidances está relacionado con diversos content elements.

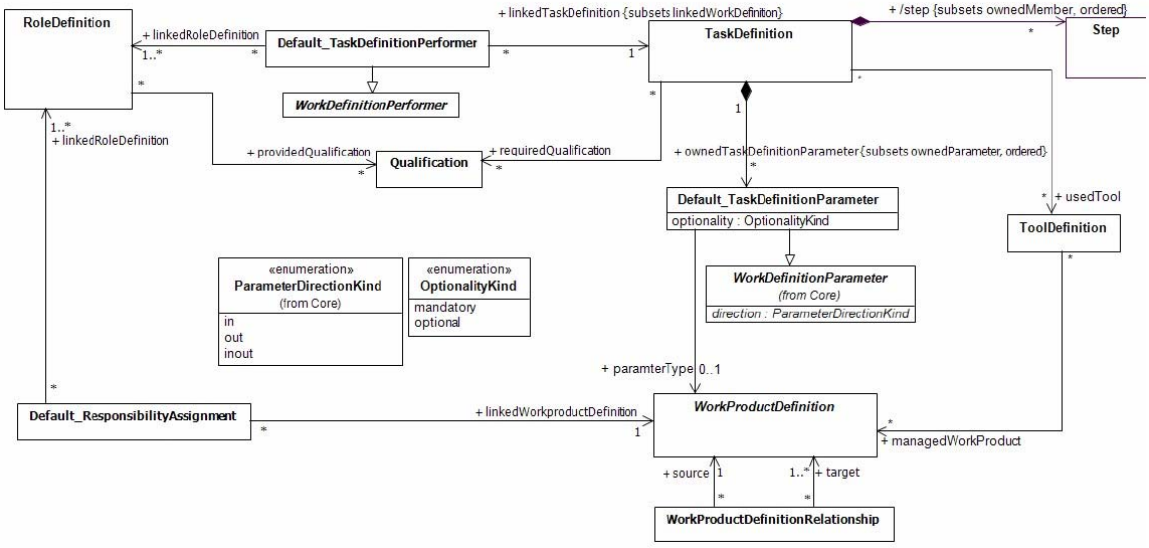


Figura 17. Relaciones entre los Elementos de Contenido de Método.

Otras relaciones secundarias que se pueden representar en SPEM 2 de manera directa o indirecta, según la herramienta editor utilizada (EPF Composer no tiene opciones directas para ello) son:

- Task – Tools: herramientas que se utilizan para hacer una tarea.
- Tool- Work Products: productos de trabajo que se producen utilizando una herramienta.
- Rol – Qualifications: Habilidades que posee un rol.
- Task – Qualifications: Habilidades requeridas para realizar una tarea.
- Work Product – Work Products: Un producto de trabajo (source) es necesario para poder producir otro producto de trabajo (target).

4.4 Procesos

Tal como se muestra en la Figura 7, con SPEM 2 se distinguen dos etapas a la hora de implementar un proceso o metodología: primero se puebla el Contenido de Método, cuyos conceptos han sido presentados anteriormente, y en segundo lugar, se combinan y reutilizan dichos elementos para ensamblar actividades y procesos.
A continuación se presentan los conceptos de SPEM que se manejan para ensamblar patrones de proceso y procesos completos (ver Figura 18). No es una revisión exhaustiva ya que, por motivos de sencillez, se ha optado por no incluir los conceptos de SPEM 2 que quedan "ocultos" al usuario cuando modela procesos con algún editor de SPEM. En los sub-apartados siguientes se presentan conceptos que corresponden, principalmente, con los paquetes "Core", "Process Structure", "Process with Methods" (en parte) y con el "Base Plug-in" (que incluye las especializaciones predefinidas de actividad).

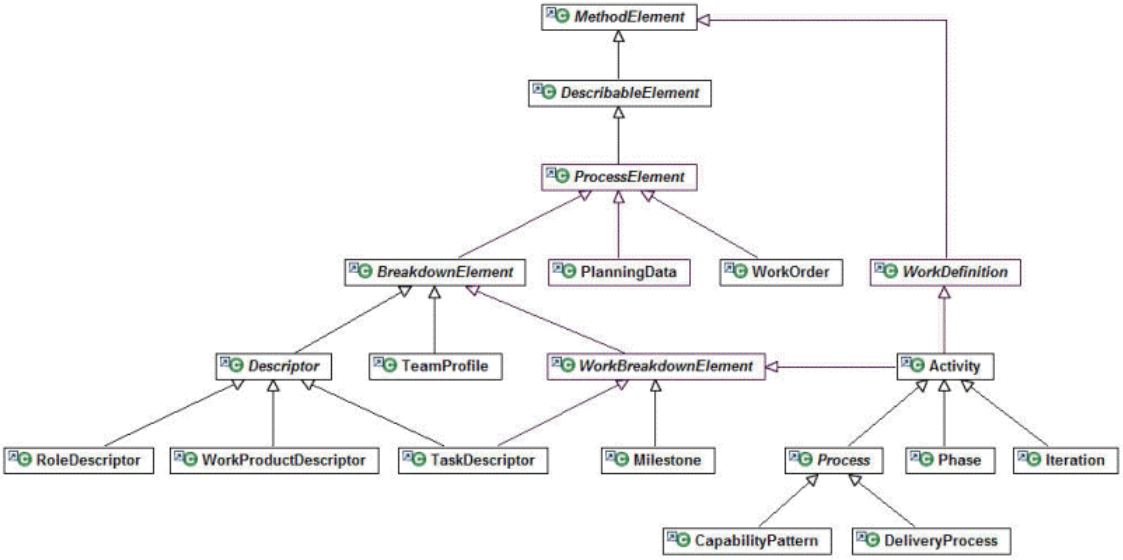


Figura 18. Jerarquía de conceptos de procesos.

4.4.1 Aspectos Generales

Una Definición de Trabajo (Work Definition) es un concepto abstracto que generaliza todos los tipos de definiciones de trabajo en SPEM [se ven después]. Una Definición de Trabajo puede estar asociada con 0..* precondiciones (restricciones que deben cumplirse para que el trabajo pueda comenzar) y con 0..* postcondiciones (restricciones que deben cumplirse para que el trabajo pueda considerarse concluido). Las definiciones de trabajo pueden ser modeladas de forma gráfica mediante diagramas de actividad de UML 2.

Un Elemento de Desglose (Breakdown Element) es una generalización abstracta para cualquier tipo de elemento que aparece en un proceso y es parte de una estructura de desglose. Aporta tres propiedades importantes:

- Admite Varias Apariciones (Has Multiple Ocurrences): al realizar el proceso puede haber mas de una instancia del elemento.
- Es Opcional (Is Optional): no es obligatoria su inclusión cuando se lleva a cabo el proyecto.
- Planeado (Is Planned): El elemento es incluido al generar los planes de proyecto que se exportan a las herramientas de gestión de proyectos.

Los tipos de Elementos de Desglose son:

- Parámetros de Procesos, para asociar productos de entrada/salida.
- Realizadores de Procesos, para asociar roles.
- Secuencias de Trabajo, para las relaciones de precedencia.
- Elementos de Desglose de Trabajo.
- Roles en Uso.
- Productos de Trabajo en Uso.
- Relaciones entre Productos de Trabajo en Uso.
- Asignaciones de Responsabilidad en Procesos.

Cada elemento de desglose tiene asociada una estructura de desglose de trabajo (WBS, Work-Breakdown Structure) que representa su estructura interna y, opcionalmente, un flujo de trabajo (Work Flow). La Figura 19 muestra un ejemplo (instancia) de ambos.

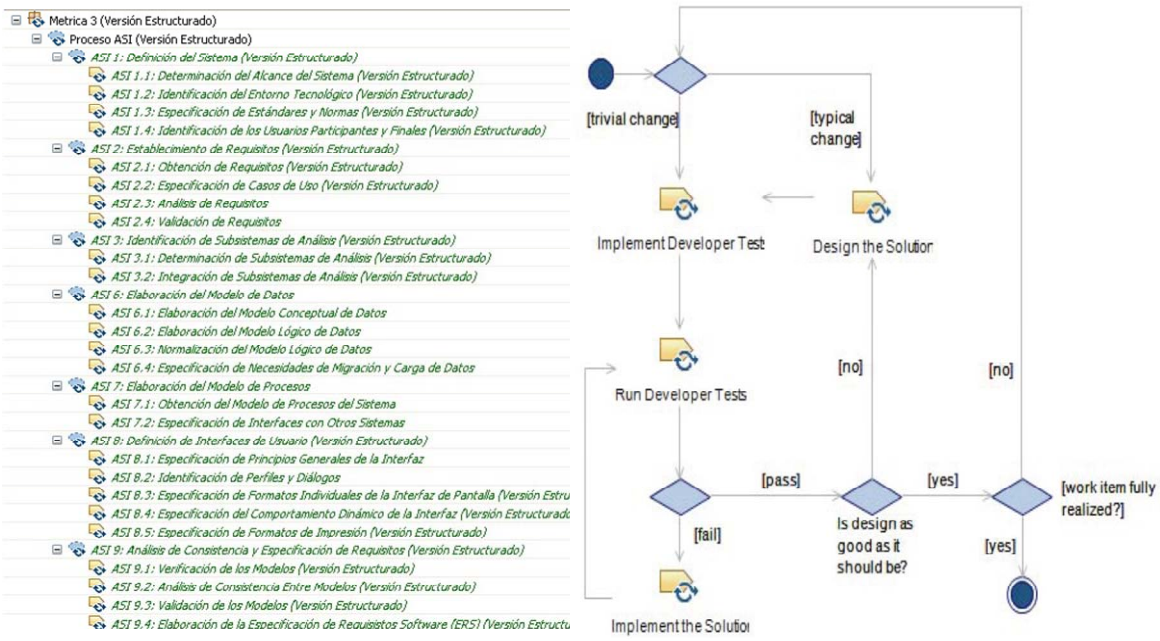


Figura 19. Ejemplos de Estructura de Desglose de Trabajo (izquierda) y de Flujo de Trabajo (derecha).

4.4.2 Elementos de Desglose de Trabajo

Los Elementos de Desglose de Trabajo (Work Breakdown Element) son el principal tipo de elemento de desglose, ya que representan la descomposición del trabajo. Existen dos tipos: Actividad e Hito (ver Figura 20). Las propiedades comunes a ambos son:

- Se Puede Repetir (Is Repeatable): puede haber varias iteraciones o repeticiones.
- Continuo (Is Ongoing): es un trabajo sin duración fija o estado final. Ejemplo: trabajo de un gestor de proyecto que 1 hora al día se dedica a revisar el estado de avance de las tareas.
- Condicionado por Sucesos (Is Event Driven): su inicio no está determinado por eventos normales (cuando acaba el trabajo que lo precede o cuando se concluye algún producto de trabajo), sino por otro evento especial.

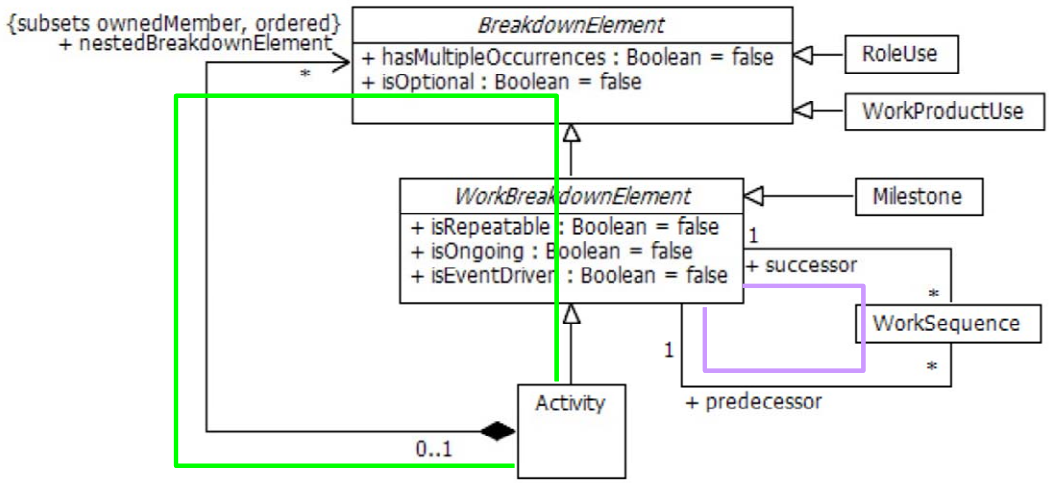


Figura 20. Flujo de trabajo (derecha) y anidamiento de actividades (izquierda) entre elementos de desglose de trabajo.

Como muestra la Figura 20, el flujo entre los Elementos de Desglose de Trabajo se representa por medio de Secuencias de Trabajo (Work Sequence). Cada Secuencia de Trabajo conecta dos Elementos de Desglose de Trabajo, nominados como predecesor y sucesor, mediante alguna de las siguientes clases (P representa al predecesor y S al sucesor) [10]:

- Acabar para Empezar: S no puede empezar hasta que no concluye P.
- Acabar para Acabar: S no puedo acabar mientras no esté acabado P.
- Empezar para Empezar: S no puede comenzar hasta que no lo ha hecho P.
- Empezar para Acabar: S no puede concluir hasta que no se inicia P. Se emplea en Just-in-Time.

Ortogonalmente al flujo de trabajo, en SPEM 2 la descomposición del trabajo en distintos niveles de detalle se realiza mediante el concepto de Actividad (Activity), de forma que cualquier elemento de desglose de trabajo con estructura interna (es decir, que incluye elementos de desglose) recibe el nombre de Actividad, independiente del nivel de desglose, es decir, del tamaño del fragmento de trabajo representado. En consecuencia, una estructura de desglose de trabajo (WBS) se representa (ver Figura 20) mediante una recursividad de agregaciones: una actividad está formada por agregación de diversos elementos de desglose, entre los que se encuentran actividades más pequeñas que a su vez pueden estar formadas de igual manera.

4.4.2.1 Actividades

Una Actividad (Activity) representa una unidad de trabajo general en un proceso. Una actividad puede tener estructura interna formada por agregación de elementos de desglose, que pueden ser de varios tipos, no solo de trabajo: actividades mas simples (anidamiento), elementos de método en uso (roles en uso, productos de trabajo en uso), e hitos. La complejidad de la estructura de desglose de una actividad puede variar entre 0 tareas (pueden estar formadas sólo por productos de trabajo en uso) o todo un proceso completo.
Una Actividad puede estar asociada con varias Actividades en Uso, que reutilizan la primera. Para ello, existen varias maneras de reutilización (herencia), que se presentan en el apartado 4.5.

[10] Son los mismos tipos de precedencias empleados en gestión de proyectos, por ejemplo, en la herramienta MS Project.

Una actividad representa una unidad de trabajo general asignable a realizadores específicos, representados por Roles en Uso. Dicha asignación se hace a través de Realizadores de Proceso (Process Performer). Una actividad puede tener entradas y producir salidas, representadas por Productos de Trabajo en Uso. Para ésta última asociación se utilizan los llamados Parámetros de Proceso (Process Parameter). La Figura 21 muestra estos vínculos entre actividades y otros elementos de proceso.

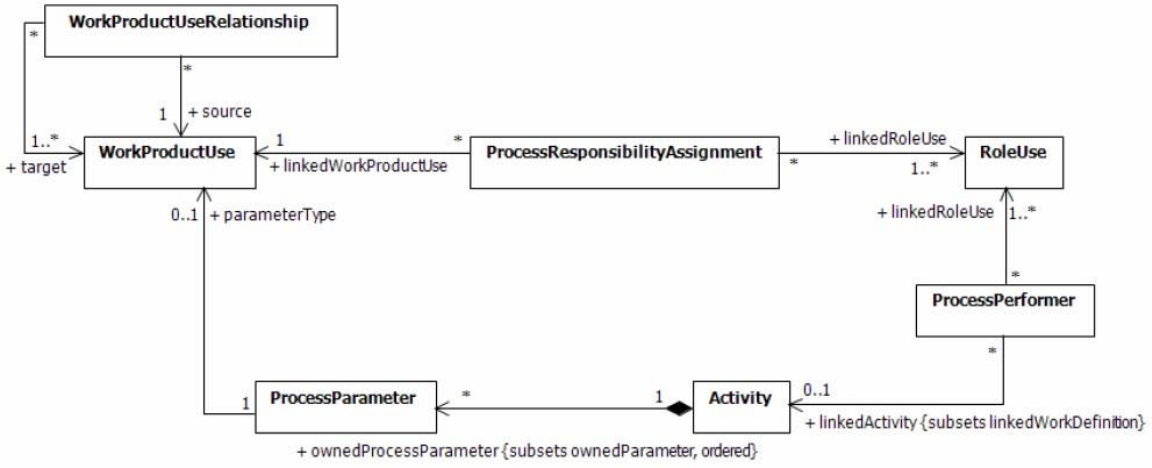


Figura 21. Relación entre actividades y roles y productos de trabajo.

En la Figura 22 se muestra un ejemplo de una actividad (centro de la figura) y sus asociaciones con dos Artefactos en Uso (izquierda y derecha), dos Tareas en uso (abajo, a la izquierda y derecha), un Role en Uso (abajo en el centro) y una Secuencia de Trabajo precediendo a otra Actividad (arriba a la izquierda).

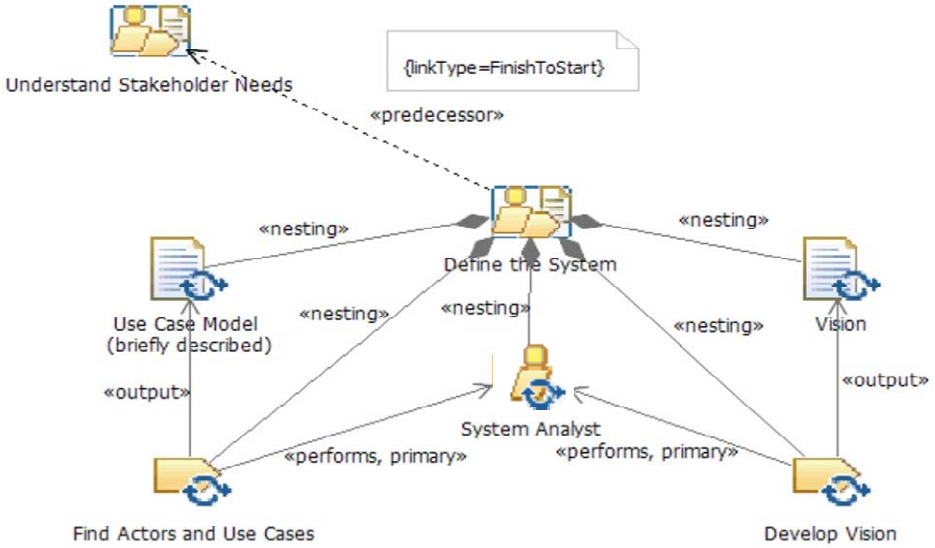


Figura 22. Ejemplo de una actividad y sus asociaciones.

SPEM 2 tiene predefinidos varios tipos especiales de Actividades (ver Figura 18): Iteración, Fase y Proceso. Puesto que Actividad es un concepto genérico, que incluye a proceso como especialización, en SPEM 2 no se respeta la nomenclatura tradicional de los estándares ISO de ingeniería del software. En especial, debe tenerse en cuenta que SPEM no sigue los tres niveles estáticos de descomposición utilizados en el modelo de procesos de ISO/IEC 12207: procesos formado por actividades formadas por tareas.

4.4.2.2 Contenido de Método en Uso

Como ya se comentó, los elementos del contenido de método (tareas, roles y productos de trabajo) reutilizados en los procesos reciben el nombre de elementos en uso. Un elemento en uso es una instancia de un elemento de método particularizada para un contexto de proceso determinado. Por ello, los elementos en uso no son reutilizables (para reuso ya están las descripciones de los elementos en el contenido de método).

Así, una Tarea en Uso (Task Use) representa la ocurrencia de una Definición de Tarea (Contenido de Método) en el contexto de una Actividad. En ella se pueden precisar y modificar, respecto de la tarea original, su documentación, pasos, roles y productos de entrada y de salida. Además, se pueden definir dos asociaciones que no aparecen en las tareas como elementos de contenido de método: los roles que asisten en la tarea y las entradas externas. Existen dos maneras de utilizar tareas en actividades:

a) Reutilizar una descripción de tarea del contenido de método, que pasa a ser una tarea en uso, o
b) Crear directamente en la actividad una instancia de tarea en uso. En este caso no se tendrá descripción asociada, sino sólo el nombre.

Un Producto de Trabajo en Uso (Work Product Use) representa la ocurrencia de un Producto de Trabajo real en el contexto de una Actividad, pudiendo modificar su documentación. Además, se pueden definir dos atributos que no aparecen en la definición de producto de trabajo en el contenido de método: los estados de entrada y de salida del producto en la actividad. Por ejemplo, un producto "Requisitos" puede entrar en una actividad en el estado "identificados" y salir en el estado "refinados". También existen las dos maneras de utilizar productos de trabajo en actividades equivalentes a las dos indicadas para las tareas:

a) Reutilizar una descripción de producto de trabajo del contenido de método, que pasa a ser un producto de trabajo en uso.
b) Crear directamente en la actividad una instancia de producto de trabajo en uso, que sólo tendrá nombre pero no descripción asociada.

De forma similar a las dos anteriores, un Rol en Uso (Role Use) representa la ocurrencia de un Rol real en el contexto de una Actividad, pudiendo particularizar la documentación, productos de los que es responsable o que modifica, y equipos (roles compuestos) a los que pertenece. Igualmente, existen dos maneras de utilizar roles en actividades:

a) Reutilizar una descripción de rol del contenido de método, que pasa a ser un rol en uso. SPEM le llama Realizador (Performer).
b) Crear directamente en la actividad una instancia de rol en uso. En este caso no se tendrá descripción asociada, sino sólo el nombre. SPEM le llama Participante (Participant).

Un Rol Compuesto [11] (Composite Role) es un rol en uso especial, que se corresponde con más de una Definición de Rol del Contenido de Método.

En la Figura 23 se muestra un ejemplo de Contenido de Método en Uso (parte derecha) referenciando a Contenido de Método (parte izquierda). La Figura 24 muestra otro ejemplo de una Tarea en Uso dentro de una Estructura de Desglose de Trabajo.

[11] Llamado "equipo" en EPF Composer en castellano.

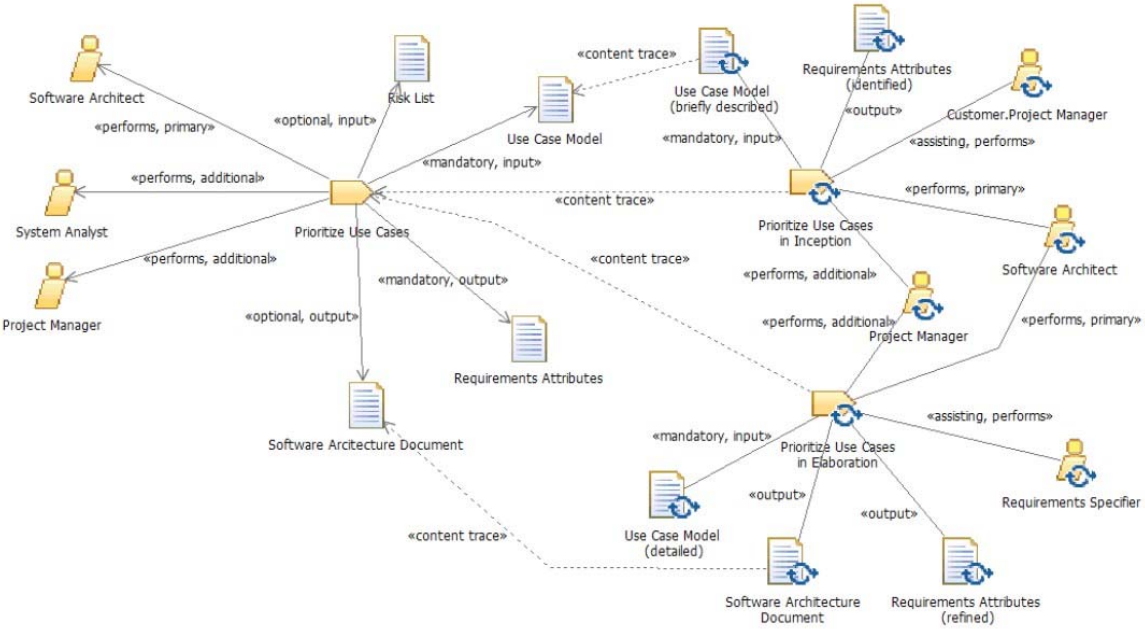


Figura 23. Ejemplo de contenido de método en uso.

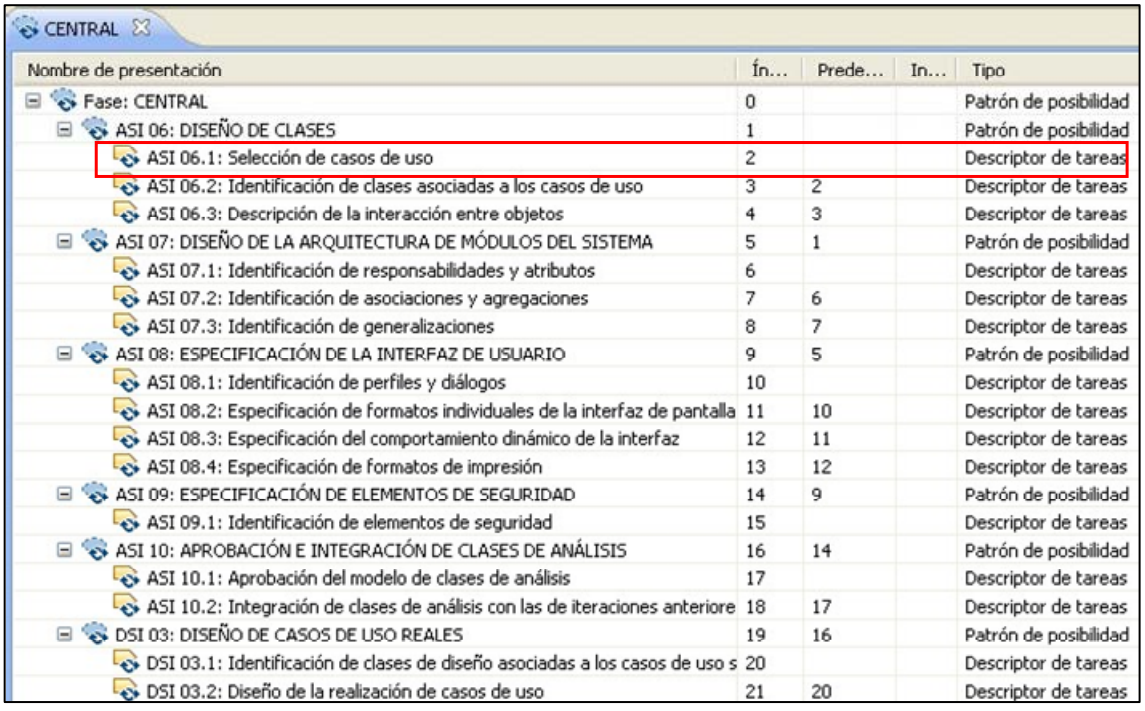


Figura 24. Una tarea en uso dentro de una estructura de desglose de trabajo.

4.4.2.3 Fases, Iteraciones e Hitos

Como ya se ha indicado, las fases e iteraciones son dos tipos especiales de actividades (adicionales a los procesos), mientras que los hitos son un tipo de elemento de desglose de trabajo, distinto de los otros dos, actividades y tareas (ver Figura 18).

Una Fase (Phase) representa un periodo de tiempo que es significativo para un proyecto, y que acaba con un punto de control de gestión importante, un hito o un conjunto de entregables

concluidos. En la práctica, en SPEM 2 una fase es una actividad que tiene el valor falso en la propiedad “Es repetible”.

Por el contrario, una Iteración (Iteration) representa un conjunto de actividades anidadas que se repiten más de una vez. Sirve para organizar ciclos repetitivos de trabajo. En la práctica, en SPEM 2 una iteración es una actividad que tiene el valor verdadero en la propiedad “Es repetible”.

Un Hito [12] (Milestone) representa un evento significativo para el desarrollo de un proyecto:
- Una decisión importante,
- La conclusión de un entregable,
- La conclusión de una fase, etc.

Según se muestra en la Figura 20, los hitos son elementos de desglose del trabajo y, por tanto, aparecen en la estructura de desglose de trabajo y pueden tener relaciones de precedencia.

4.4.3 Tipos de Procesos

En SPEM 2, un Proceso (Process) es un tipo de Actividad que describe una estructura para tipos particulares de proyectos o partes de ellos.

Mediante un proceso de SPEM 2 se pueden representar distintos tipos de métodos de ingeniería del software: un proceso, un modelo de procesos completo, un ciclo de vida con sus diversos procesos, o una metodología completa.

Los procesos software definen cómo serán ejecutados los proyectos software. Por ello, para ayudar a cumplir los requisitos del nivel 3 de CMMI (proceso definido), SPEM 2 provee mecanismos de adaptación y ensamblado dinámico, basados en patrones de proceso reutilizables (se presentan en apartados posteriores):
- Reutilización (Actividad en Uso).
- Particularización (Elementos con Clases - Kinds).
- Especificación descriptiva (Elementos de Desglose).
- Variabilidad (Plugins).

En SPEM 2 existen dos clases principales de procesos:
- Patrón de Proceso (Capability Pattern): Es un “fragmento de proceso” que describe un grupo de actividades reutilizable como solución a algún tipo de problema o situación habitual. Se definen para poder ser empleados más de una vez en uno o varios procesos o con fines de organización. Se pueden almacenar en una jerarquía de Paquetes de Proceso (Process Package). Algunos escenarios de uso de patrones de proceso son:
  - Servir como bloques para construir Procesos para Despliegue o Patrones de Proceso más complejos.
  - Ayudar a la ejecución de proyectos que no siguen un proceso bien definido, sino que trabajan en base a fragmentos de proceso (buenas prácticas) de una manera flexible (métodos ágiles).
  - En formación, para describir el conocimiento de una cierta área clave, buena práctica, disciplina, etc.
- Proceso para Despliegue (Delivery Process): Describe una aproximación completa e integrada para realizar un tipo específico de proyecto, abarcando un ciclo de vida

[12] Aunque EPFC versión en castellano le denomina “objetivo”, se ha optado por hito porque es la palabra que significa exactamente lo mismo que milestone.

completo de desarrollo o mantenimiento. Sirven como plantillas para planificar y ejecutar los proyectos. En un Proceso de Despliegue se ensamblan Patrones de Proceso y Elementos en Uso (tareas, roles y productos de trabajo en uso). Ejemplos: "proceso unificado de Rational (RUP)", "programación extrema (XP)", "mantema, metrica 3", etc.

Existe un tercer tipo de proceso llamado Plantilla para Planificación de Procesos (Process Planning Template), que contiene la misma información que los otros tipos pero ampliada con ciertas decisiones que se refieren a un plan de proyecto concreto, por ejemplo, se incluyen asignaciones de personal o fechas. Por tanto, este tipo de proceso no es realmente un nuevo tipo de proceso sino una adaptación de los anteriores de cara a la exportación a una herramienta de gestión de proyectos, es decir, a poder generar un archivo que sirva de plantilla para un plan de proyecto concreto.

Todos los procesos, independiente del tipo que sean, tienen en común las siguientes propiedades: name (nombre), presentation name (nombre de presentación), brief description (descripción breve), external ID (ID externo), purpose (objetivo), main description (descripción principal), scope (ámbito), usage notes (notas de utilización), alternatives (alternativas), how to staff (cómo proveer de personal), key considerations (factores clave).
Adicionalmente, un proceso para despliegue tiene éstas otras propiedades: scale (escala), project characteristics (características del proyecto), risk level (nivel de riesgo), estimating techniques (técnicas de estimación [13]), project member expertise (especialidad de miembros del proyecto), type of contract (tipo de contrato).

Los patrones de proceso pueden ser incorporados como elementos de desglose a una WBS. Esto implica la reutilización automática de todos sus elementos y estructura (actividades, tareas, roles, productos de trabajo, WBS, etc.).

4.5 Reutilización y Variabilidad

La organización en plugins (Plug-in) permite que se puedan reutilizar los elementos de contenido y los procesos definidos en una Biblioteca (Library). Dicha reutilización se puede realizar de dos maneras:

a) Al crear un plugin nuevo se puede referenciar a otros plugins.
b) Usar de forma directa el contenido de un plugin desde otro plugin diferente.

A veces nos interesa reutilizar el contenido de un plugin con ciertas modificaciones. Para ello existe el mecanismo de variabilidad (Variability), que permite modificar elementos de método o de proceso sin modificar directamente el original. La variabilidad de un elemento de método o de proceso permite definir diferencias (adiciones, cambios, omisiones) con el elemento original. Dichas diferencias afectan a las propiedades, es decir, a los atributos y a las asociaciones con otros elementos.

4.5.1 Variabilidad de Elementos de Contenido

SPEM 2.0 contempla 5 tipos de variabilidad entre elementos de contenido: No Aplicable, Contribuye, Amplía, Reemplaza, y Amplía y Reemplaza. A continuación se describen sus características:

[13] Por ser más correcta, se ha elegido el término "estimación" aunque la versión en castellano de EPFC utiliza "cálculo".

- **No asignado** (not assigned): No existe relación de variabilidad entre el elemento en cuestión y otros. Es el valor por defecto.

- **Contribuye** (contributes): Un elemento E que contribuye a un elemento base EB añade sus valores de atributos e instancias de asociación a EB sin modificar directamente las propiedades que ya tiene dicho elemento base EB (es una adición). Al generar una vista (por ejemplo, publicar en la web), el elemento base EB se muestra combinado con los atributos y relaciones de E, mientras que E queda oculto. Las reglas que rigen dicha combinación son (ver ejemplos en la Figura 25 y resultado en la Figura 26):
  - El valor de los atributos (campos de texto) de E se concatena al final del contenido de los correspondientes atributos de EB.
  - Las instancias de asociaciones de entrada o de salida con cardinalidad "muchos" de E se añaden a EB.
  - Las instancias de asociaciones de entrada y de salida con cardinalidad "1" (como la relación "realizador principal" entre una tarea y un rol) definidas en E se ignoran si ya existen en EB. En caso contrario, se añaden.
  - Un elemento base puede recibir varias contribuciones.
  - La contribución es transitiva.

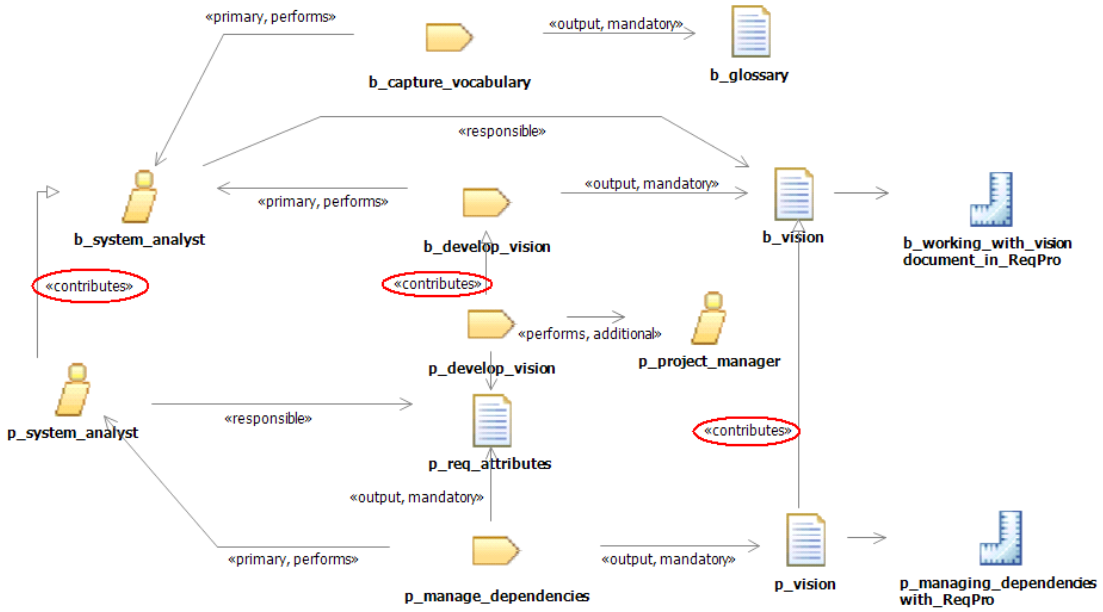


Figura 25. Ejemplos de variabilidad de tipo "contribuye".

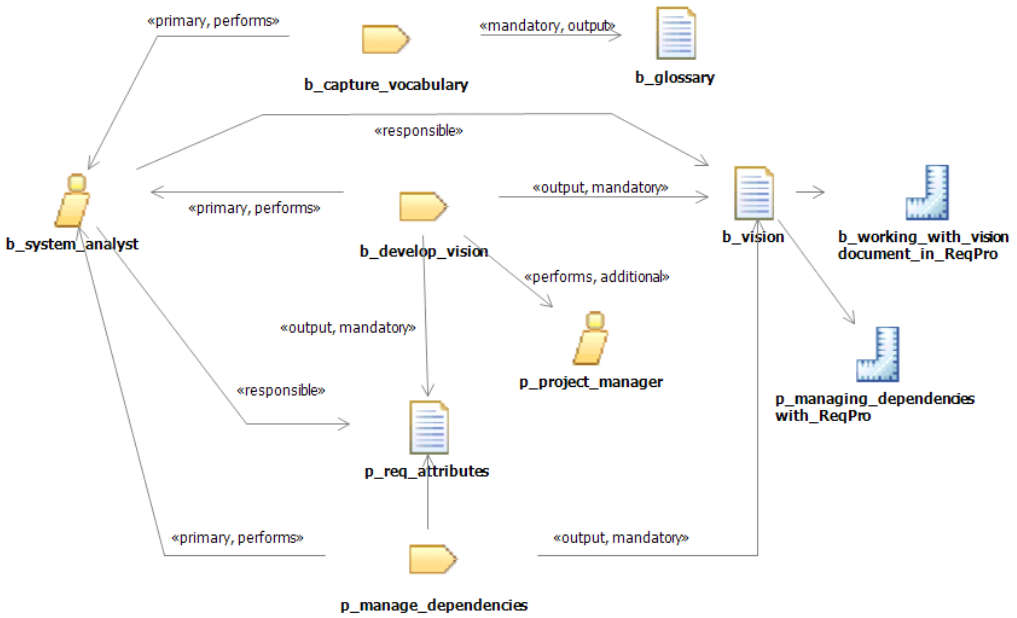


Figura 26. Resultado de la variabilidad de la Figura 25.

- **Reemplaza** (replaces): Mediante este tipo de variabilidad un elemento E puede reemplazar (sustituir) los atributos y las asociaciones de salida de un elemento base EB sin modificar directamente ninguna propiedad de EB. El elemento E que reemplaza se incluirá en las vistas generadas mientras que el reemplazado EB no aparecerá. Las reglas que regulan el reemplazo son (un ejemplo se muestra en la Figura 27 y el resultado en la Figura 28):
  - Los valores de los atributos de EB se reemplazan por los respectivos valores de los atributos no vacíos de E, incluyendo los identificadores.
  - Las instancias de asociación de salida de EB se reemplazan por las de E.
  - Las instancias de asociación de entrada con cardinalidad "muchos" de EB se amplían con las instancias de dichas asociaciones existentes para E.
  - Las instancias de asociación de entrada con cardinalidad "uno" de EB se dejan intactas. Las instancias incluidas en E que no están en EB se añaden a EB.
  - Un elemento base sólo puede ser reemplazado por otro único elemento en una misma configuración. Si se define más de un reemplazo para un mismo elemento base, entonces no se realizará ningún reemplazo.
  - El reemplazo es transitivo.

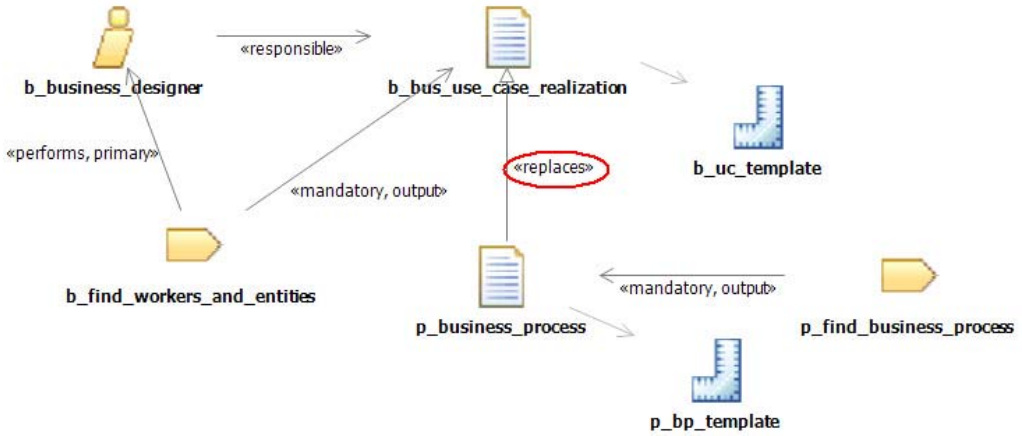


Figura 27. Ejemplo de variabilidad de tipo "reemplaza".

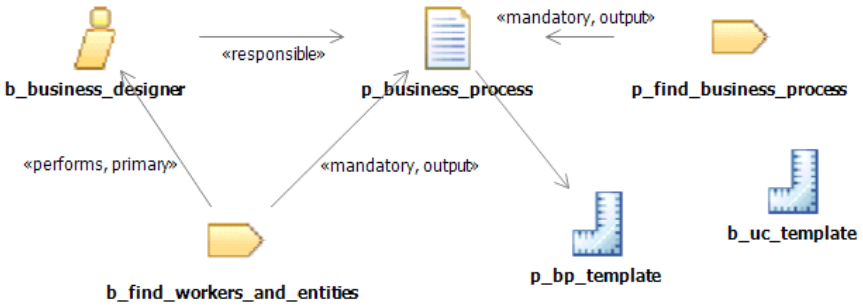


Figura 28. Resultado de la variabilidad de la Figura 27.

- **Amplía** (extends): En este caso, el elemento E hereda las propiedades del elemento base EB que amplía, dejando EB totalmente intacto. El elemento E incluye las propiedades heredadas de EB más las propias (por ejemplo, nuevas asociaciones). Tanto EB como E aparecen en las vistas generadas. La Figura 29 muestra un ejemplo de esta clase variabilidad y la Figura 30 permite comprobar el resultado. Las reglas que la regulan son:
  - Los valores de los atributos de EB son heredados por E en caso de que los equivalentes suyos estén vacíos. Si se definen campos en E, los respectivos campos de EB se ignoran.
  - Las instancias de asociación de salida con cardinalidad "muchos" de EB se añaden a E.
  - En las instancias de asociación de salida con cardinalidad "1" se ignora la instancia de EB si E ha definido la suya propia. En caso contrario, E hereda la de EB.
  - Las asociaciones de entrada de EB se ignoran (no se añaden a E).
  - Las asociaciones de ampliación son transitivas.

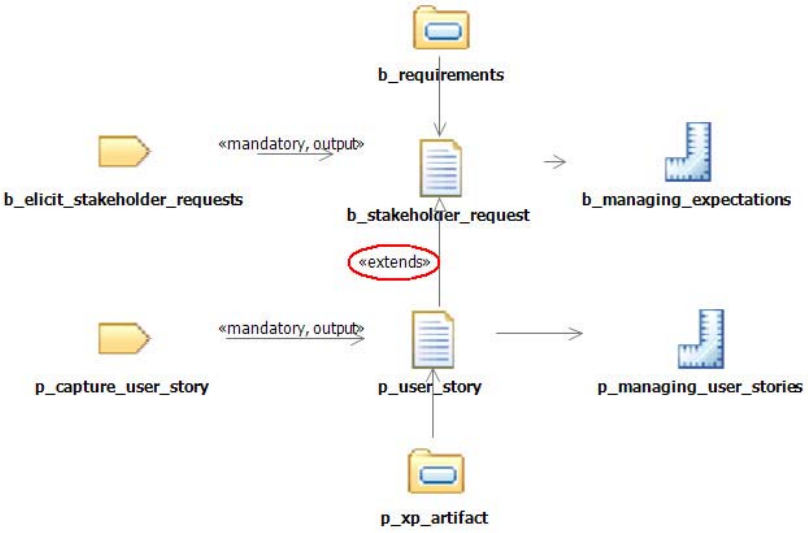


Figura 29. Ejemplo de variabilidad de tipo "amplía".

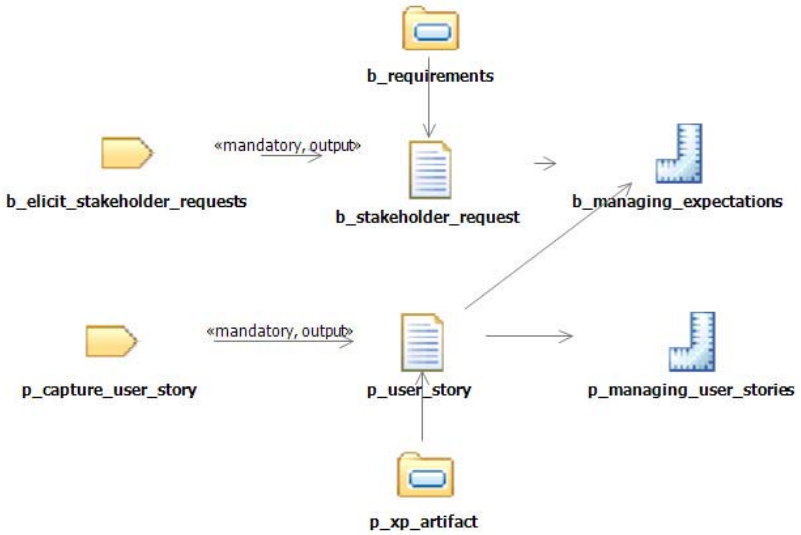


Figura 30. Resultado de la variabilidad de la Figura 29.

- **Amplía y Reemplaza** (extends-replaces): Este tipo de variabilidad combina los efectos de las variabilidades de tipo Amplía y de tipo Reemplaza, ya comentadas. El resultado es que las propiedades que no se han definido en el elemento E que amplía y reemplaza se heredan del elemento base EB. En cambio, las propiedades definidas en E sustituyen a las de EB. E aparece en las vistas que se generan, pero EB no. Este tipo de variabilidad se utiliza al generar plugins que renombran elementos, reemplazan descripciones, etc., sin remodelar completamente todas las relaciones y atributos del plugin base. Las reglas que rigen esta clase de variabilidad son (ver ejemplo en Figura 31 y su resultado en Figura 32):
    - Si el elemento E que amplía y sustituye define sus propias asociaciones de salida, reemplazarán a las del elemento base EB. Las instancias de asociaciones de salida no definidas en E se heredan de EB.
    - Las instancias de asociaciones de entrada de EB se añaden a E.
    - Los valores de los atributos no vacíos de E reemplazan a los valores de los mismos atributos de EB. del elemento base. Si E no define valor para un atributo, el valor se hereda de EB.
    - La relación amplía y reemplaza es transitiva.
    - Un elemento base sólo puede ser reemplazado o ampliado y reemplazado por otro único elemento en una misma configuración. En caso contrario no se realizará ningún reemplazo.

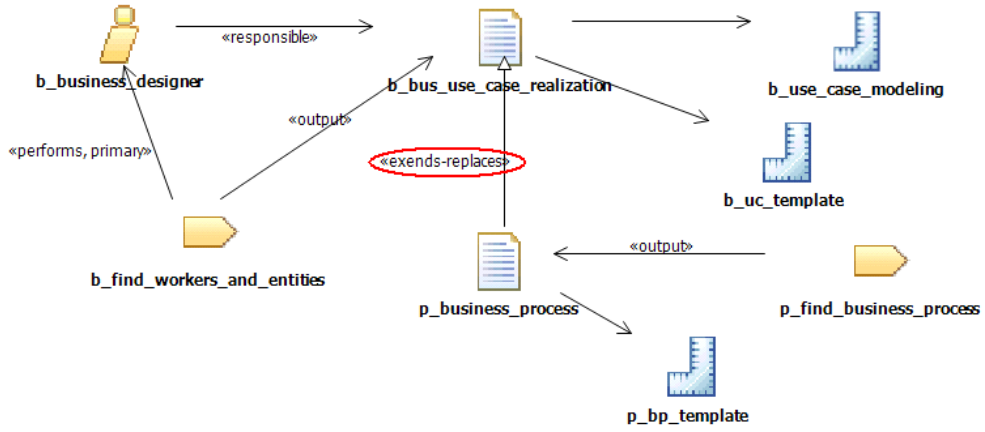


Figura 31. Ejemplo de variabilidad de tipo “amplía y reemplaza”.

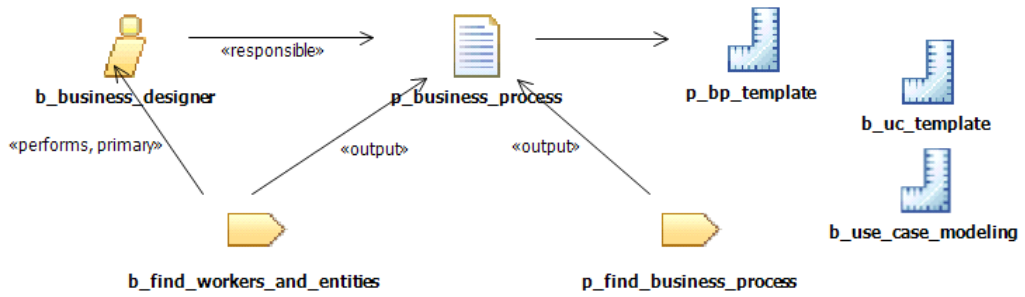


Figura 32. Resultado de la variabilidad de la Figura 31.

Cuando se resuelven varias variabilidades, las reglas de prioridad son las siguientes:
- Primero se realizan las asociaciones de variabilidad de tipo "Contribuye", seguidas de las de tipo "Reemplaza", luego de las de tipo "Amplía" y, por último, las de tipo "Amplía y Reemplaza".
- Entre dos variabilidades del mismo tipo, primero se realiza la que está más arriba en la jerarquía.

4.5.2 Reutilización y Variabilidad de Elementos de Proceso

Una manera de definir procesos es aplicar una estrategia bottom-up desde lo pequeño hacia lo grande. Al seguirla, primero se definen patrones de proceso sencillos, que no incluyen jerarquía de actividades, sino simplemente descriptores de tarea, de rol y de producto). Después se crean patrones más complejos en los que se reutilizan los patrones más sencillos definidos anteriormente. De esta forma se van componiendo los procesos reutilizando bloques de proceso cada vez más complejos, hasta llegar a los procesos para despliegue.
En general, los elementos de proceso de tipo actividad (fase, iteración, actividad, patrón de proceso) se pueden reutilizar en otros elementos de proceso más grandes. En este caso reciben el nombre genérico de Actividad en Uso (Activity Use). Una Actividad en Uso en SPEM define la habilidad para reutilizar las estructuras definidas en los elementos de desglose de una actividad en una segunda actividad, sin necesidad de copiar físicamente dichas estructuras. Por tanto, se trata de una forma para heredar dinámicamente dichas estructuras desde la actividad referenciada.

SPEM 2 establece varios mecanismos de reutilización de una actividad (Activity Use Kind) [14]:

- **No aplicable** (not assigned): Cuando una actividad no tiene asociaciones de Actividad en Uso.

- **Extensión** (extension): Una actividad extendida A hereda la estructura interna (subestructuras e instancias de asociación) de una actividad base AB. Se pueden añadir nuevos elementos de desglose a A, pero no se pueden modificar los elementos de desglose heredados desde AB. Los cambios realizados en A no afectan a AB. En cambio, los cambios realizados en AB con posterioridad se reflejan automáticamente en A y en las demás actividades en uso que extienden AB.

- **Contribución Local** (local contribution): Sirve para definir adiciones locales específicas (llamadas contribuciones) a los elementos de desglose heredados vía la extensión del tipo

[14] El editor EPF Composer emplea unas opciones ligeramente diferentes, llamadas Ampliar, Copiar y Copiar en Profundidad, que son una manera distinta, desde el punto de vista del usuario, de hacer lo mismo.

anterior. Por tanto, para que existe una asociación de este tipo entre una actividad A y otra actividad base AB es necesario que existe previamente la asociación de tipo extensión entre ambas.

- **Reemplazo Local** (local replacement): De forma similar al anterior, sirve para definir sustituciones locales específicas (llamadas reemplazos) a los elementos de desglose heredados vía la extensión. Igualmente, para que existe una asociación de este tipo entre una actividad A y otra actividad base AB es necesario que existe previamente la asociación de tipo extensión entre ambas. Las instancias de asociación de este tipo sustituyen sub-elementos de AB con elementos propios de A.

No existe un tipo "Eliminación Local", es decir, no es posible eliminar en una Actividad en Uso elementos de desglose heredados de otra actividad mediante el tipo "Extensión". Por tanto, al reutilizar una actividad se debe tener en cuenta que se van a reutilizar todos los elementos contenidos en la actividad. Esta restricción en el metamodelo SPEM es debida a razones de integridad y consistencia, pero no supone un problema en la práctica porque existen opciones para no incluir ciertos elementos heredados en las vistas generadas de un proceso (ver apartado 5.2.2).
Debido a lo anterior, el escenario habitual de reutilización de elementos de proceso consiste en reutilizar primero una cierta actividad mediante una asociación de extensión, seguida después de un refinamiento, mediante contribuciones y reemplazos locales, de la actividad en uso resultante.

Por otro lado, los elementos de proceso de tipo actividad (fase, iteración, actividad, patrón de proceso) permiten variabilidad de los tipos siguientes:
- Amplía
- Contribuye
- Reemplaza

Las características de estos tipos son las mismas que las explicadas para la variabilidad en los elementos de contenido de método (ver apartado 4.5.1).

4.6 Configuraciones de Método

Una Configuración de Método (Method Configuration) es una selección de contenidos de los plugins de una Biblioteca de Métodos (Method Library) de forma que se limita el espacio visual de la biblioteca al subconjunto seleccionado. En consecuencia, es el mecanismo ofrecido en SPEM para poder tener diferentes **vistas** de una misma biblioteca de plugins o de un único plugin. La definición de una configuración puede estar basada en las definiciones de otras configuraciones. Por ejemplo, una configuración A podría definirse como el superconjunto de las configuraciones B, C y D más ciertos plugins y paquetes adicionales. En este caso, se dice que B, C y D son las configuraciones base de A. Si alguna de estas configuraciones base cmabian, los cambios serán automáticamente válidos también para A, reduciendo de esta manera el esfuerzo de mantenimiento.

Una configuración de método define un subconjunto lógico de una biblioteca de métodos mediante la selección (filtro) de los paquetes deseados, tanto de contenido de método como de procesos. La Figura 33 muestra un ejemplo, indicando con color rojo (fondo resaltado) las partes de una biblioteca de plugins que se incluye en una cierta configuración de método. La selección de elementos incluidos en una configuración se puede refinar añadiendo o sustrayendo de la configuración todos los elementos asociados con una cierta categoría.

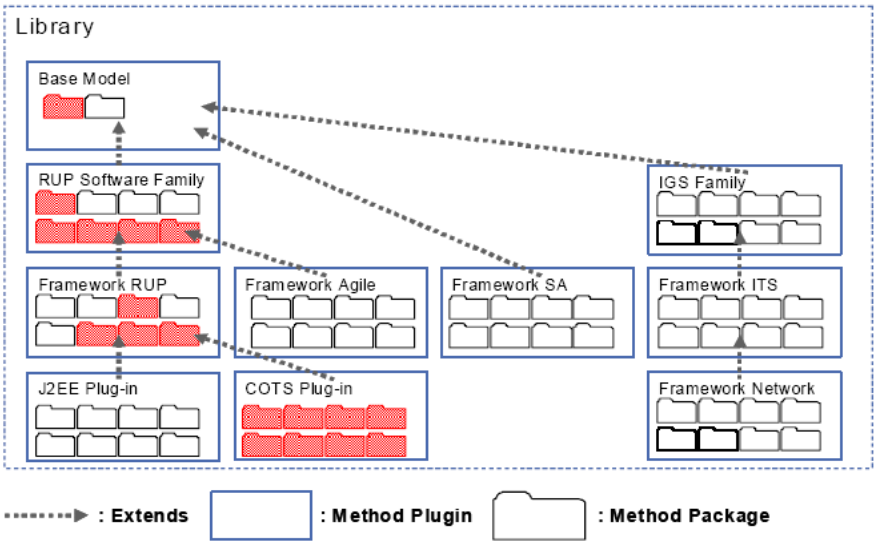


Figura 33. Ejemplo de una configuración de método (elementos en rojo).

Una actividad representando un proceso puede ser valida para diferentes configuraciones de método, pero en cada configuración se incluye o excluye contenido particular para situaciones específicas. Por ejemplo, un proceso definido para desarrollar esquemas de bases de datos puede tener dos configuraciones, una con la vista particular para SGBD (Sistemas de Gestión de Bases de Datos) relacionales y otra para el caso de sistemas orientados a objetos. Por tanto, este mecanismo ofrece una alternativa en la creación de procesos, ya que puede definirse un proceso general que incluya contenidos para varios tipos de proyectos más específicos. Mediante el uso de configuraciones se seleccionan los contenidos de cada uno de esos tipos de proyectos específicos. Crear vistas para los diversos tipos de proyecto es sencillo, ya que basta con seleccionar la configuración adecuada.

5 El Editor EPF Composer

Eclipse Process Framework Composer, en adelante EPFC, es una herramienta gratuita, desarrollada dentro del entorno ECLIPSE, que sirve para editar fragmentos de método, procesos o metodologías, y generar automáticamente la documentación adecuada en formato para la web (ver Figura 34). Dichos fragmentos se almacenan en formato XMI y, al estar basados en el estándar SPEM 2, pueden ser reutilizados por cada vez más herramientas CASE. En realidad, EPFC utiliza la "Unified Method Architecture" (UMA) [15], que a su vez está basada en el metamodelo SPEM 2.
En suma, EPFC es un editor de procesos SPEM 2, que incluye opciones adicionales para publicar de forma automáticamente sitios web.
Además de la herramienta Composer, el Eclipse Process Framework [16] también incluye las implementaciones públicas, abiertas y gratuitas de varias metodologías: OpenUP (proceso unificado abierto), SCRUM y XP.

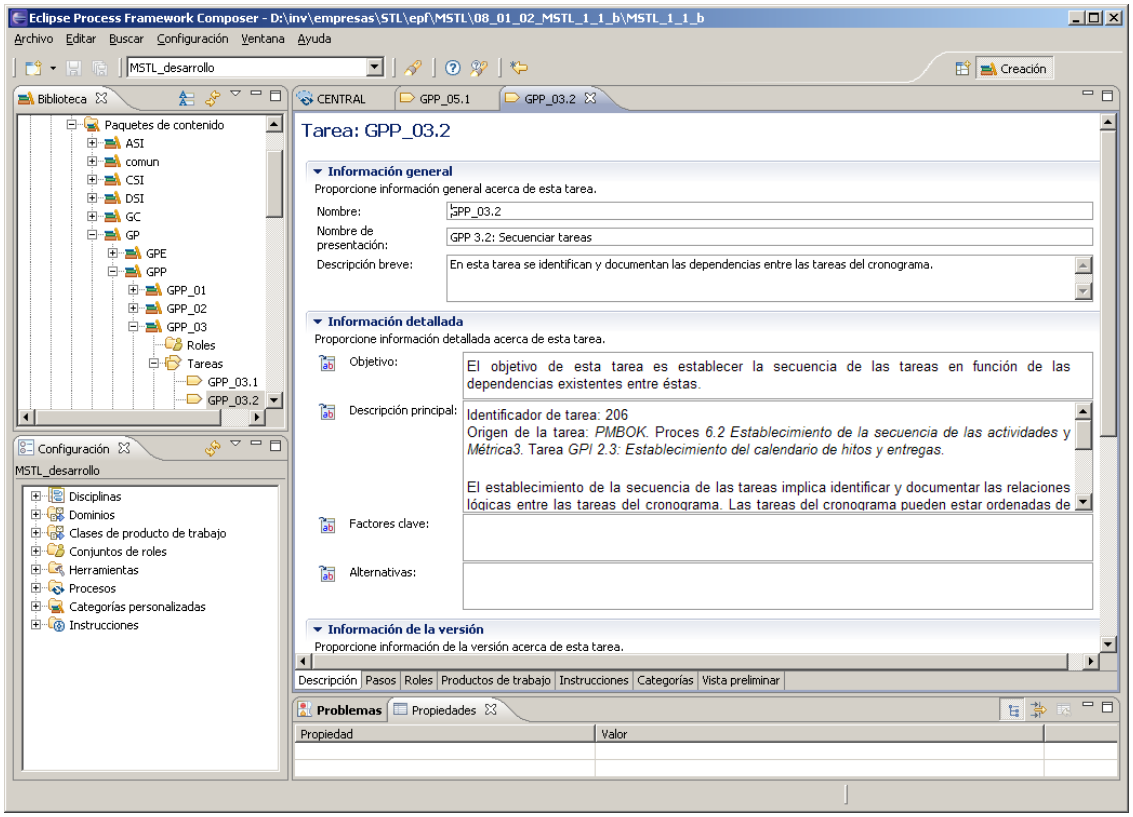


Figura 34. Pantalla de la ficha de una tarea en EPF Composer.

5.1 Poblando el Contenido de Método

Típicamente, el primer paso en la edición de procesos es crear los elementos de contenido que posteriormente se reutilizarán en la creación de los procesos. No es imprescindible crear contenido de método que se reutilice al crear procesos, pero es muy aconsejable ya que facilita enormemente la creación y la mantenibilidad de dichos procesos, gracias a la estructuración y la reutilización de los elementos utilizados en ellos.

[15] http://www.eclipse.org/epf/composer_architecture/
[16] Mas información se puede encontrar en: www.eclipse.org/epf/.

Por otra parte, los elementos de contenido sirven como biblioteca de conocimiento en la que se gestiona información interesante en el ámbito de la empresa, como son descripciones de los productos que se generan, descripciones del trabajo que hay que realizar para elaborar dichos productos y guías sobre cómo se realizan, etc. Si interesa, ésta información puede quedar accesible de forma organizada para las personas interesadas en ella, independientemente de que luego se reutilice en la creación de procesos o no.
Los elementos de contenido se pueden organizar y jerarquizar mediante la creación de paquetes de contenido. En cada paquete de contenido se pueden definir los elementos de contenido de método básicos: roles, tareas, productos de trabajo y guías (instrucciones).
La creación de elementos y de paquetes de contenido con EPFC se hace en la vista *Biblioteca*. La especificación de los atributos (descripción breve, objetivos, etc.) se realiza en la pestaña *Descripción* del editor del elemento. Cada tipo de elemento cuenta con una serie de pestañas mediante las que se pueden definir sus asociaciones con otros tipos de elementos. Así, las tareas tienen una pestaña en la que se definen los roles que participan en ellas, otra pestaña en la que se definen los productos de entrada y de salidas; los roles tienen una pestaña en la que se definen los productos de los que son responsables, etc.

5.1.1 Categorización de Elementos de Método

Los elementos del contenido de método se pueden categorizar mediante su asociación con categorías estándar (disciplinas, dominios, clases de productos de trabajo, conjuntos de roles y herramientas), para lo cual existe una pestaña en EPFC.
Además, los elementos del contenido de método (y también los elementos en uso en procesos) se pueden organizar alternativamente mediante categorías personalizadas. Las categorías personalizadas también se crean en EPFC a través de la pestaña correspondiente en la vista *Biblioteca*. Estas categorías personalizadas pueden contener cualquier tipo de elemento, proceso, categoría, etc. Son útiles para organizar de forma lógica elementos que no se puedan organizar mediante las categorías estándar, como por ejemplo, agrupar todas las plantillas de productos, agrupar todas las guías que sean directrices, etc.
Las categorías personalizadas permiten organizar el contenido de acuerdo al esquema que queramos, sirviendo como medio eficaz para organizar el contenido para su publicación.
El contenido de una categoría personalizada se especifica en la pestaña *Asignar* del editor de la categoría.

5.2 Creando Procesos

Una vez se han definido los elementos de contenido de método, éstos se pueden reutilizar para crear procesos. Cabe señalar que no es necesario haber creado elementos de contenido para reutilizarlos en la creación de procesos, pues se pueden definir directamente en un proceso, pero esta opción dificulta mucho la reutilización y actualización posterior de dichos elementos y de sus procesos.

Con EPFC se pueden crear los dos tipos de procesos de SPEM 2:

- Patrón de proceso [17]: Son los patrones de proceso de SPEM, que describen una agrupación reutilizable de tareas o actividades.
- Proceso para despliegue [18]: Describen un enfoque completo e integrado para una metodología completa, un tipo de ciclo de vida, un tipo específico de proyecto, etc.

Al igual que los elementos de contenido, los patrones de proceso y los procesos para despliegue se pueden organizar y jerarquizar mediante paquetes de proceso.

[17] Llamados, de forma incorrecta, patrones de posibilidad en EPFC en castellano.
[18] Llamados procesos para entrega en EPFC en castellano.

La creación de procesos y de paquetes de proceso se hace en la vista *Biblioteca*. En la pestaña *Descripción* del editor se especifican los atributos de los procesos (descripción breve, descripción principal, objetivos, etc.)

5.2.1 Estructuras de Desglose

Para representar los dos tipos de procesos antes comentados (es decir, las llamadas actividades de forma general en SPEM 2), se utilizan estructuras de descomposición, mediante las cuales se representa la jerarquía de actividades (incluyendo fases e iteraciones) e hitos. Además, permiten ordenar la secuencia del trabajo (flujo de trabajo), indicando relaciones de precedencia entre los elementos de la estructura de descomposición.

Para ver y editar desde diferentes puntos de vista la estructura de desglose de un proceso/metodología completo o de un fragmento, EPFC incluye las siguientes pestañas:

a) Estructura de Desglose de Trabajo (Work Breakdown Structure): Corresponde al conocido WBS de gestión de proyectos. En EPFC es una tabla que contiene la descomposición jerárquica de los elementos que definen el trabajo a realizar: actividades (patrones de proceso, fases, iteraciones), hitos, tareas, etc.
b) Asignación de Equipos [19] (Team Allocation): Muestra una jerarquía similar a la anterior, pero mostrando para cada actividad (cada nivel) la lista de roles que participan.
c) Utilización de Productos de Trabajo (Work Product Usage): Esta perspectiva se centra en los productos de trabajo de forma similar a como la anterior se centra en los roles.
d) Vista Consolidada (Consolidated View): Combina las informaciones que se muestran en las tres vistas anteriores. Solo existe con fines informativos, es decir, no permite edición.

Para desarrollar la estructura de desglose de un proceso o metodología se deben crear los elementos de desglose que lo forman: actividades (y sus variantes, fases e iteraciones), hitos (también llamados objetivos) y, en el nivel más bajo de la jerarquía, tareas, roles o productos de trabajo. Éstos últimos dependerán de la perspectiva de desglose (pestaña) en la que estemos trabajando.

Los elementos de desglose se pueden crear directamente en la estructura de desglose o se pueden reutilizar del contenido de método (tareas, roles y productos) o de otros patrones de proceso (actividades). Así, existen varias maneras de abordar la especificación de un proceso:

a) Descendente: Definir la jerarquía del proceso de arriba abajo, de forma que el proceso se desarrolla definiendo de forma descendente la jerarquía de actividades (fases, iteraciones…) e hitos; para después indicar los elementos que forman las actividades del nivel más bajo en la jerarquía.
b) Ascendente: Se define la jerarquía de abajo a arriba, creando primero las actividades de más bajo nivel, que solo contienen elementos y no otras actividades, mediante patrones de proceso. Después se definen patrones más complejos componiendo otros patrones creados. Finalmente se define el proceso para despliegue mediante la composición de patrones complejos.
c) Mixta: Las dos formas anteriores se combinan, según convenga, a la hora de crear procesos complejos.

Para añadir una nueva actividad (fase, iteración) o un hito en un proceso, se hace clic derecho en el elemento padre en el que se quiere incluir, se selecciona *Nuevo hijo* y se elige el elemento que se desea añadir. Las actividades e hitos se pueden añadir en cualquiera de las estructuras de desglose.

[19] Aunque EPFC utiliza la nomenclatura “asignación de equipos”, lo correcto sería hablar de roles o de personal en general.

A la hora de especificar un proceso con EPFC hay que tener en cuenta que esta herramienta no impone restricciones en la jerarquía sobre los tipos de actividad. Por ejemplo, podemos incluir una fase dentro de una iteración, lo cual no tiene sentido. Por ello, es responsabilidad del ingeniero de procesos definir una jerarquía que sea lógica.

5.2.1.1 Estructuras de Desglose de Trabajo

La perspectiva más útil para definir un proceso es el WBS, ya que en ella se establece la jerarquía de todos los esfuerzos o trabajos que deben ser realizados; utilizando las otras dos perspectivas de edición para los detalles de los participantes y de los productos asociados con cada Elemento de Desglose de Trabajo (Work Breakdown Element) o WBE.

Siguiendo el metamodelo SPEM 2, a partir del elemento raíz del WBS (un patrón de proceso o un proceso para despliegue), se añaden los WBE, que pueden ser de los siguientes tipos:

- Actividad (activity): Es el bloque básico para construir el desglose de los procesos. Una actividad representa un grupo de elementos de desglose (otras actividades, tareas, etc.).
- Descriptor de Tarea (Task Descriptor): Corresponde a la Tarea en Uso en SPEM 2. Es la reutilización en el WBS de una tarea. Cada vez que a un WBS se añade como elemento de desglose un elemento de método de tipo tarea, todos los roles y productos de trabajo asociados con ella se incorporan automáticamente al WBS.
- Hito (hito (milestone): Describe un evento significativo para un proyecto, como por ejemplo, una decisión importante o haber completado una fase.

En el nivel más bajo de la jerarquía de un WBS, las actividades están compuestas por tareas. Hay dos formas de añadir una tarea a una actividad: una es reutilizar las tareas definidas en el contenido de método y la otra es definir las tareas directamente en el proceso. EPFC ofrece dos opciones para reutilizar tareas del contenido de método:

- Arrastrar desde la vista *Configuración*.
- Hacer clic derecho en la actividad padre y seleccionar *Añadir desde el contenido de método…*

5.2.1.2 Propiedades de los Elementos de Desglose de Trabajo

En EPFC, un WBS se muestra como una tabla (ver Figura 35) en la que cada fila corresponde a un WBE. Las propiedades (columnas) comunes para todos los tipos de WBE son las siguientes:

- Nombre de presentación (Presentation name).
- Índice (Index): número secuencial de 0 en adelante que sirve para identificar cada elemento (WBE).
- Predecesores (Predecessors): lista de valores index de los WBE que son predecesores inmediatos.
- Información del modelo (Model info): indica si el WBE está basado en un elemento de contenido y el tipo de herencia. En algunos casos indica el subtipo de elemento de contenido, por ejemplo, entrada obligatoria (mandatory input) para un producto de trabajo.
- Tipo (Type): los tipos de WBE son proceso para despliegue (delivery process), patrón de proceso (capability pattern), actividad (activity), iteración (iteration), fase (phase), hito (Milestone) y descriptor de tarea (task descriptor).
- Ocurrencias múltiples (Multiple occurrences) [20]: sí/no permitir más de una ocurrencia (instancia) del proceso.
- Opcional (Optional): si está desactivado significa que se puede quitar dicho WBE del proceso sin problemas.

[20] En la versión en castellano de EPFC se le denomina “varias apariciones”.

- Planificado (Planned): sí/no incluirlo para exportar a una herramienta de gestión de proyectos.
- Suprimido (Suppresed): sí/no queda excluido de la publicación del proceso.

Nombre de presentación	Índice	Predecesores	Tipo
Fase: CENTRAL	0		Patrón de posibilidad
ASI 06: DISEÑO DE CLASES	1		Patrón de posibilidad
ASI 06.1: Selección de casos de uso	2		Descriptor de tareas
ASI 06.2: Identificación de clases asociadas a los casos de uso	3	2	Descriptor de tareas
ASI 06.3: Descripción de la interacción entre objetos	4	3	Descriptor de tareas
ASI 07: DISEÑO DE LA ARQUITECTURA DE MÓDULOS DEL SISTEMA	5	1	Patrón de posibilidad
ASI 07.1: Identificación de responsabilidades y atributos	6		Descriptor de tareas
ASI 07.2: Identificación de asociaciones y agregaciones	7	6	Descriptor de tareas
ASI 07.3: Identificación de generalizaciones	8	7	Descriptor de tareas
ASI 08: ESPECIFICACIÓN DE LA INTERFAZ DE USUARIO	9	5	Patrón de posibilidad
ASI 08.1: Identificación de perfiles y diálogos	10		Descriptor de tareas
ASI 08.2: Especificación de formatos individuales de la interfaz de pantalla	11	10	Descriptor de tareas
ASI 08.3: Especificación del comportamiento dinámico de la interfaz	12	11	Descriptor de tareas
ASI 08.4: Especificación de formatos de impresión	13	12	Descriptor de tareas
ASI 09: ESPECIFICACIÓN DE ELEMENTOS DE SEGURIDAD	14	9	Patrón de posibilidad
ASI 09.1: Identificación de elementos de seguridad	15		Descriptor de tareas
ASI 10: APROBACIÓN E INTEGRACIÓN DE CLASES DE ANÁLISIS	16	14	Patrón de posibilidad
ASI 10.1: Aprobación del modelo de clases de análisis	17		Descriptor de tareas
ASI 10.2: Integración de clases de análisis con las de iteraciones anteriores	18	17	Descriptor de tareas
DSI 03: DISEÑO DE CASOS DE USO REALES	19	16	Patrón de posibilidad
DSI 03.1: Identificación de clases de diseño asociadas a los casos de uso seleccionados	20		Descriptor de tareas
DSI 03.2: Diseño de la realización de casos de uso	21	20	Descriptor de tareas
DSI 04: DISEÑO DE CLASES	22	19	Patrón de posibilidad
DSI 04.1: Identificación de clases adicionales	23		Descriptor de tareas
DSI 04.2: Diseño de asociaciones y agregaciones	24	23	Descriptor de tareas
DSI 04.3: Identificación de atributos de las clases	25	24	Descriptor de tareas
DSI 04.4: Identificación de operaciones de las clases	26	25	Descriptor de tareas

Figura 35. Estructura de Desglose de Trabajo (WBS) en forma de tabla en EPF Composer.

Los WBE de tipo actividad o descriptor de tarea tienen estas otras propiedades adicionales:

- Condicionado por sucesos (Event-driven): activado si la tarea/actividad se inicia cuando ocurre cierto evento; desactivado si se inicia después de otra(s) tareas(s) (relación de precedencia entre ellas).
- Continuo (Ongoing): activado si la tarea/actividad es continua a lo largo de toda la vida de la instancia del proceso; desactivado si tiene un inicio y final en el proceso.
- Repetible (Repeatable) [21]: activado si la tarea/actividad puede iterar para una instancia del proceso; desactivado si puede ocurrir una sola vez cuando el proceso es instanciado (llevado a cabo).

Las anteriores propiedades se pueden seleccionar en las columnas del mismo nombre en las perspectivas de desglose, o también en la pestaña *General* de la vista *Propiedades* para el elemento seleccionado.

En la pestaña *Propiedades* (Properties) de un WBS es posible incluir información más completa y precisa para los WBE del WBS, que completan la heredada desde los elementos reutilizados.

En la pestaña *General* (General), la propiedad Dependencia (Dependency) permite precisar el tipo de las relaciones de precedencia inmediata entre los elementos de un WBS, eligiendo entre:

- Finalizar para Iniciar (Finish To Start), que es la opción por defecto;
- Finalizar para Finalizar (Finish To Finish);
- Iniciar para Finalizar (Start To Finish); o
- Iniciar para Iniciar (Start To Start).

[21] También llamado “se puede repetir”.

Establecer los tipos de dependencias es especialmente útil a la hora de exportar el proceso a una herramienta de gestión de proyectos:

En la pestaña *Documentación* (Documentation) se pueden asignar valores a las propiedades: prefijo (prefix), descripción breve (brief description), guía de utilización (usage guidance), factores clave (key considerations), y descripción perfeccionada (refined description).

En la pestaña *Roles* (Roles) la propiedad "asistido por" (assisted by) sirve para indicar otros roles que apoyan al realizador principal (primary performer).

En la pestaña *Productos de trabajo* (Work products) la propiedad "entradas externas" (external inputs) permite añadir entradas externas al proceso, es decir, que no se generan como salida de ninguna actividad del proceso.

5.2.1.3 Descriptores de Elementos Básicos

En SPEM 2, las tareas que aparecen en el contexto de un proceso son Tareas en Uso. Como ya se ha comentado, en EPFC este mismo concepto recibe el nombre de Descriptor de Tarea (Task Descriptor). Lo mismo ocurre con los roles y los productos de trabajo en el contexto de un proceso: son Descriptores de Rol (Role Descriptor) y Descriptores de Productos de Trabajo (Work Product Descriptor). Los descriptores permiten personalizar y adaptar cada aparición del elemento de contenido a las necesidades del proceso en el contexto en que aparecen, pudiendo refinar los atributos y relaciones del elemento para reflejar su utilización en un punto concreto del proyecto.

Se pueden modificar las propiedades del descriptor que esté seleccionado en el WBS eligiendo la pestaña *Propiedades*. Dependiendo del tipo de descriptor que sea, se podrán modificar unas propiedades u otras. Por ejemplo, las tareas tendrán una pestaña *Roles* mediante la que podrán modificar los roles que participan en el descriptor de tarea, los roles tendrán una pestaña *Productos de trabajo* mediante la que podrán modificarse los productos de los que es responsable ese descriptor de rol, etc.

Los cambios realizados en un elemento de método no se propagan automáticamente a los descriptores que lo refieren. Para que un descriptor refleje los cambios realizados en un elemento de método después de aplicarlo en un proceso, se debe sincronizar el descriptor con el elemento de método. En la **sincronización** se perderán las modificaciones realizadas en el descriptor, ya que se copiarán los valores de los atributos y las relaciones del elemento de método. La sincronización de un descriptor se realiza haciendo clic derecho en el elemento de la estructura (sea la de desglose de trabajo, de asignación de equipos o de utilización de productos) que queremos sincronizar y pulsar *Sincronización predeterminada desde el contenido de método* para sincronizar todos los atributos y relaciones, o pulsar *Personalizar sincronización…* para elegir los atributos y relaciones que queremos sincronizar y no perder los cambios realizados en el descriptor que nos interesa mantener.

También se pueden añadir descriptores de tarea, rol o producto de trabajo a una actividad que no refieran a ningún elemento de método. Para añadir un descriptor a una actividad se hace clic derecho en la actividad a la que se quiere añadir, se selecciona "Nuevo hijo", y se añade el descriptor, que, dependiendo de la estructura en la que estemos trabajando, será de tarea, de rol o de producto de trabajo.

5.2.1.4 Otras Estructuras de Desglose

Como se presentó al comienzo del apartado 5.2.1, además del WBS, EPFC permite otras tres vistas o perspectivas de los elementos de desglose de un elemento de proceso. No se trata de nuevos elementos, nuevos valores de las propiedades o nuevas instancias de asociaciones entre elementos, sino una manera diferente de visualizar la misma información: los elementos y asociaciones incluidos en la estructura de desglose de un cierto elemento de proceso.

La Figura 36 muestra un ejemplo del uso de la perspectiva de Asignación de Equipos (team allocation) para ver o editar la asignación de roles a cada actividad y los productos de trabajo que cada rol tiene asociados.

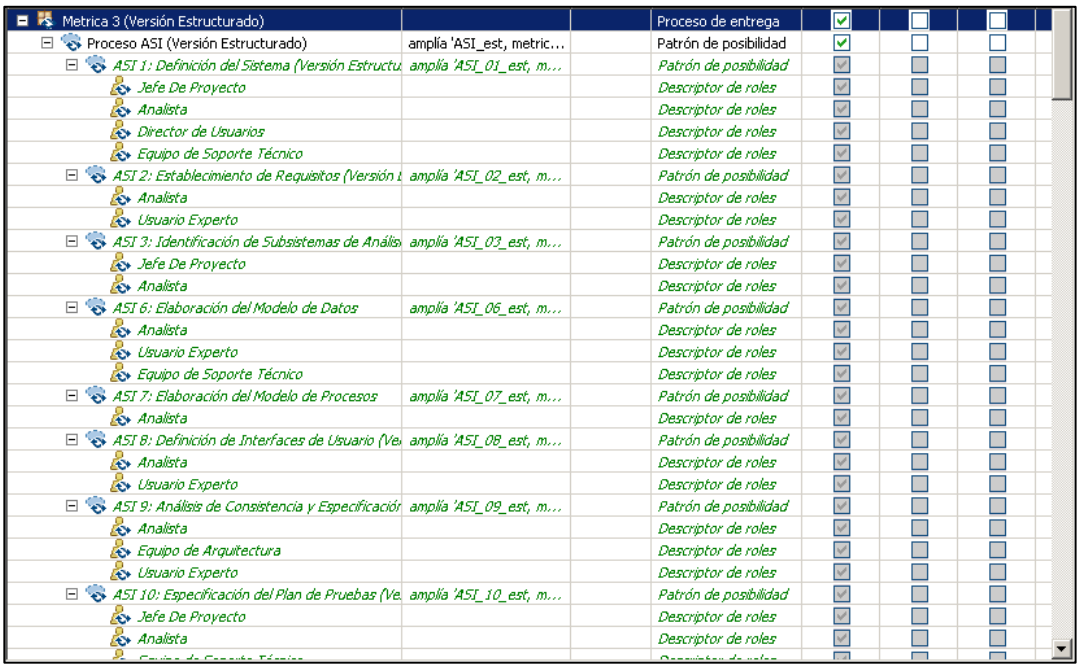
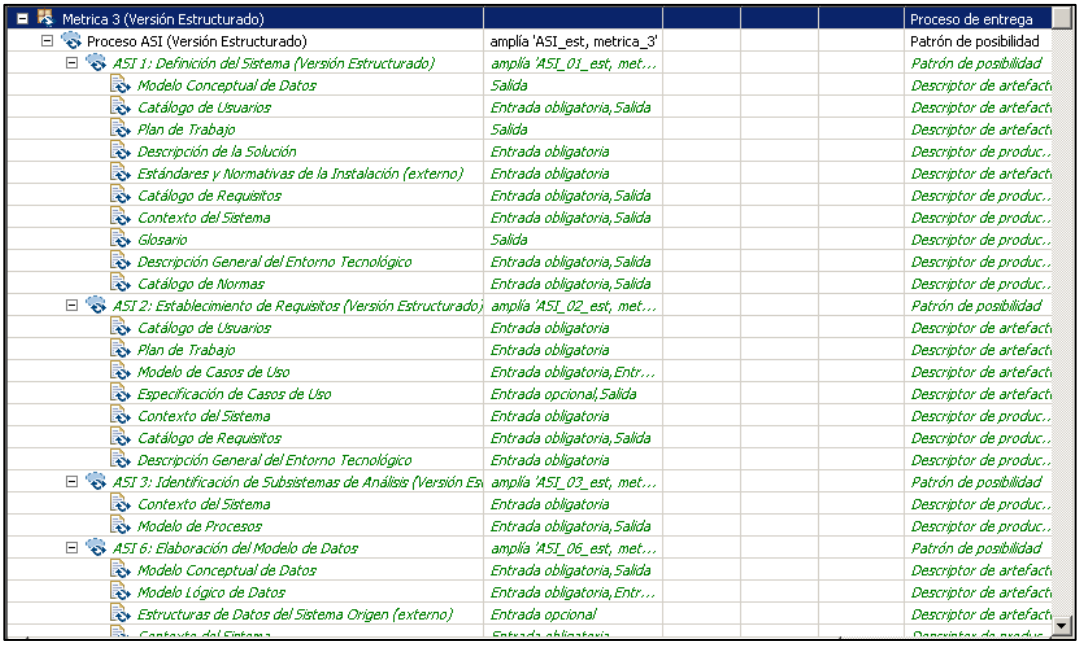


Figura 36. Ventana de la estructura de desglose de Asignación de Equipos.

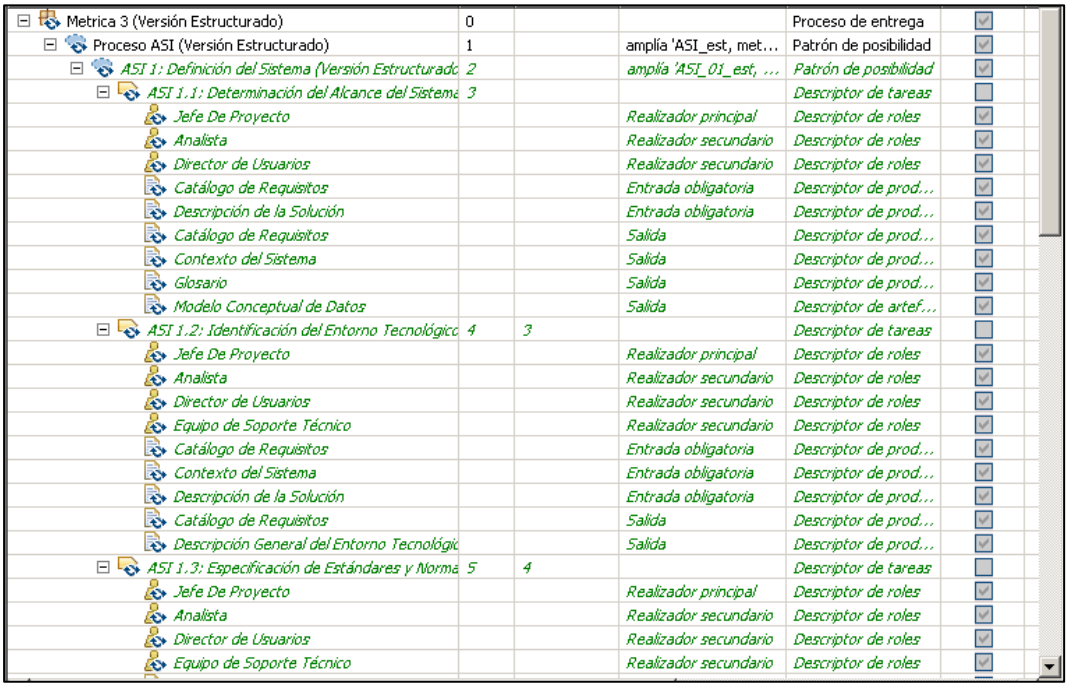
En el ejemplo de vista de "Utilización de Productos de Trabajo" de EPFC mostrada en la Figura 37 se comprueba la manera en que con esta herramienta se pueden establecer los productos de trabajo que serán creados y utilizados en cada actividad del proceso, e identificar los roles responsables de dichos productos de trabajo. Además, para cada producto de trabajo se incluyen las propiedades, en forma de columnas, de: estado de entrada (entry state), estado de salida (exit state) y entregable (deliverable), también llamado producto final.

Por último, la Vista Consolidada (consolidated view) permite ver (pero no editar) de forma conjunta la lista de roles y productos de trabajo asociados con cada actividad (ver ejemplo en Figura 38).



Metrica 3 (Versión Estructurado)					Proceso de entrega
Proceso ASI (Versión Estructurado)	amplía 'ASI_est, metrica_3'				Patrón de posibilidad
ASI 1: Definición del Sistema (Versión Estructurado)	amplía 'ASI_01_est, met...				Patrón de posibilidad
Modelo Conceptual de Datos	Salida				Descriptor de artefacto
Catálogo de Usuarios	Entrada obligatoria, Salida				Descriptor de artefacto
Plan de Trabajo	Salida				Descriptor de artefacto
Descripción de la Solución	Entrada obligatoria				Descriptor de produc...
Estándares y Normativas de la Instalación (externo)	Entrada obligatoria				Descriptor de artefacto
Catálogo de Requisitos	Entrada obligatoria, Salida				Descriptor de produc...
Contexto del Sistema	Entrada obligatoria, Salida				Descriptor de produc...
Glosario	Salida				Descriptor de produc...
Descripción General del Entorno Tecnológico	Entrada obligatoria, Salida				Descriptor de produc...
Catálogo de Normas	Entrada obligatoria, Salida				Descriptor de produc...
ASI 2: Establecimiento de Requisitos (Versión Estructurado)	amplía 'ASI_02_est, met...				Patrón de posibilidad
Catálogo de Usuarios	Entrada obligatoria				Descriptor de artefacto
Plan de Trabajo	Entrada obligatoria				Descriptor de artefacto
Modelo de Casos de Uso	Entrada obligatoria, Entr...				Descriptor de artefacto
Especificación de Casos de Uso	Entrada opcional, Salida				Descriptor de artefacto
Contexto del Sistema	Entrada obligatoria				Descriptor de produc...
Catálogo de Requisitos	Entrada obligatoria, Salida				Descriptor de produc...
Descripción General del Entorno Tecnológico	Entrada obligatoria				Descriptor de produc...
ASI 3: Identificación de Subsistemas de Análisis (Versión Es	amplía 'ASI_03_est, met...				Patrón de posibilidad
Contexto del Sistema	Entrada obligatoria				Descriptor de produc...
Modelo de Procesos	Entrada obligatoria, Salida				Descriptor de produc...
ASI 6: Elaboración del Modelo de Datos	amplía 'ASI_06_est, met...				Patrón de posibilidad
Modelo Conceptual de Datos	Entrada obligatoria, Salida				Descriptor de artefacto
Modelo Lógico de Datos	Entrada obligatoria, Entr...				Descriptor de artefacto
Estructuras de Datos del Sistema Origen (externo)	Entrada opcional				Descriptor de artefacto

Figura 37. Ventana de la estructura de desglose de Utilización de Productos de Trabajo.



Metrica 3 (Versión Estructurado)	0			Proceso de entrega	☑
Proceso ASI (Versión Estructurado)	1		amplía 'ASI_est, met...	Patrón de posibilidad	☑
ASI 1: Definición del Sistema (Versión Estructurado	2		amplía 'ASI_01_est, ...	Patrón de posibilidad	☑
ASI 1.1: Determinación del Alcance del Sistema	3			Descriptor de tareas	☐
Jefe De Proyecto			Realizador principal	Descriptor de roles	☑
Analista			Realizador secundario	Descriptor de roles	☑
Director de Usuarios			Realizador secundario	Descriptor de roles	☑
Catálogo de Requisitos			Entrada obligatoria	Descriptor de prod...	☑
Descripción de la Solución			Entrada obligatoria	Descriptor de prod...	☑
Catálogo de Requisitos			Salida	Descriptor de prod...	☑
Contexto del Sistema			Salida	Descriptor de prod...	☑
Glosario			Salida	Descriptor de prod...	☑
Modelo Conceptual de Datos			Salida	Descriptor de artef...	☑
ASI 1.2: Identificación del Entorno Tecnológico	4	3		Descriptor de tareas	☐
Jefe De Proyecto			Realizador principal	Descriptor de roles	☑
Analista			Realizador secundario	Descriptor de roles	☑
Director de Usuarios			Realizador secundario	Descriptor de roles	☑
Equipo de Soporte Técnico			Realizador secundario	Descriptor de roles	☑
Catálogo de Requisitos			Entrada obligatoria	Descriptor de prod...	☑
Contexto del Sistema			Entrada obligatoria	Descriptor de prod...	☑
Descripción de la Solución			Entrada obligatoria	Descriptor de prod...	☑
Catálogo de Requisitos			Salida	Descriptor de prod...	☑
Descripción General del Entorno Tecnológic			Salida	Descriptor de prod...	☑
ASI 1.3: Especificación de Estándares y Norma	5	4		Descriptor de tareas	☐
Jefe De Proyecto			Realizador principal	Descriptor de roles	☑
Analista			Realizador secundario	Descriptor de roles	☑
Director de Usuarios			Realizador secundario	Descriptor de roles	☑
Equipo de Soporte Técnico			Realizador secundario	Descriptor de roles	☑

Figura 38. Ventana de la Vista Consolidada.

5.2.2 Reutilizando y Adaptando Actividades

Como se mencionó en el punto 5.2.1, una forma de crear procesos es abordar la especificación de un proceso de abajo a arriba. Primero se definen patrones de proceso más sencillos (no incluyendo jerarquía de actividades, sino simplemente descriptores de tarea, de rol y de producto de trabajo). Después, se crean patrones más complejos en los que se reutilizan los patrones más sencillos definidos anteriormente. De esta forma se van componiendo los procesos reutilizando bloques de proceso cada vez más complejos hasta llegar a construir el proceso para despliegue.

EPFC ofrece dos opciones para reutilizar actividades (normalmente del tipo patrón de proceso) previamente creadas:

a) Arrastrar la actividad desde la ventana *Configuración* al elemento padre de la estructura de desglose donde queremos aplicarlo.
b) Hacer clic derecho en el elemento padre y seleccionar *Aplicar patrón*.

En el apartado 4.5.2 se presentaron las opciones que incluye SPEM 2 para la reutilización de elementos de proceso (Actividades en Uso). EPFC incluye unas opciones diferentes para reutilizar actividades, aunque ofrecen al ingeniero de procesos la misma funcionalidad:

- Copiar (Copy): Se copia el contenido de la actividad. La copia se desconecta del original y se puede modificar para personalizarlo según exija el contexto del proceso en el que se aplica. Las modificaciones en la copia no afectan a la actividad original. Las modificaciones en la actividad original no se reflejan en las copias que se hayan hecho de dicha actividad en otros procesos.
- Extender (Extend): Se hereda el contenido de la actividad. Se pueden añadir nuevos elementos a la actividad heredada, pero no se puede modificar el contenido heredado. Los cambios realizados en la actividad extendida no afectan a la. En cambio, los cambios realizados en la actividad original se reflejan en las actividades que la extienden.
- Copia en Profundidad: (Deep Copy): Con la opción anterior se copia una actividad tal cual, es decir, si dentro de ella existe un elemento que fue aplicado utilizando la opción extender, en el sitio donde se copia se respetará la relación de extensión. Mediante la copia en profundidad lo que se consigue es que todo el contenido de la actividad se copie. Incluso los elementos que se aplicaran por extensión en la actividad, al realizar la copia en profundidad serán copiados y desconectados de la actividad original.

Una vez aplicada (reutilizada) una actividad se pueden modificar sus propiedades. Por ejemplo, es posible cambiar su documentación, añadir instrucciones o cambiar el tipo de la actividad para que sea una fase, iteración o actividad. Todo esto se hace en la vista *Propiedades* mediante las pestañas *General*, *Documentación* e *Instrucciones*.

Al aplicar una actividad puede haber elementos que no nos interesen en el contexto en que se va a utilizar dicha actividad. Para ello, en actividades copiadas (primera opción de las tres anteriores) se pueden eliminar los elementos que no nos interesen. En cambio, en actividades extendidas no es posible hacerlo. Pero si queremos que un elemento no aparezca publicado en un proceso completo podemos hacerlo mediante la propiedad *Suprimir* que aparece al hacer clic derecho sobre el elemento. El elemento suprimido no se eliminará del patrón, pero no se publicará.

El tipo de variabilidad (amplía, contribuye o reemplaza) de un elemento de proceso de tipo actividad (fase, iteración, actividad, patrón de capacidad) se especifica en la pestaña *General* de la vista *Propiedades*.

5.2.3 Diagramas

Asociados a un WBS, o a un WBE dentro de un WBS, con EPFC se pueden generar tres de los tipos de diagramas previstos en SPEM 2. Dichos diagramas los genera automáticamente EPFC, que además incluye un editor gráfico para mejorarlos y completarlos. El editor de la tabla de la estructura de desglose y el editor de diagramas son bidireccionales, en el sentido de que los cambios realizados con uno se trasladan automáticamente a la información mostrada por el otro.

Para abrir o crear un diagrama con EPFC, se selecciona un elemento de tipo actividad, se hace clic derecho, se elige la opción *Diagramas* y se selecciona el tipo de diagrama que se quiere crear. Mediante la opción *Diagramas->Opciones de publicación…* se pueden seleccionar los diagramas que se desean publicar.

También se pueden utilizar diagramas diferentes de los tres predefinidos con la herramienta. Para ello, los diagramas realizados con otras herramientas se deben incluir en un tipo de guía llamado *Material de soporte* y seleccionar dicha guía como diagrama mediante la opción *Diagramas->Diagramas definidos por el usuario…*

A continuación se presentan brevemente los tres tipos de diagramas incluidos en EPFC.

5.2.3.1 Diagramas de Actividad

Los Diagramas de Actividad (Activity Diagram) muestran el flujo de control (control flow) entre los diversos elementos de proceso que son desglose del elemento seleccionado; el cual debe tener desglose, es decir, debe ser de tipo actividad (incluidos iteración y fase) o de tipo proceso (patrón o para despliegue). La Figura 39 muestra un diagrama de este tipo junto a la paleta del editor de diagramas del EPFC.

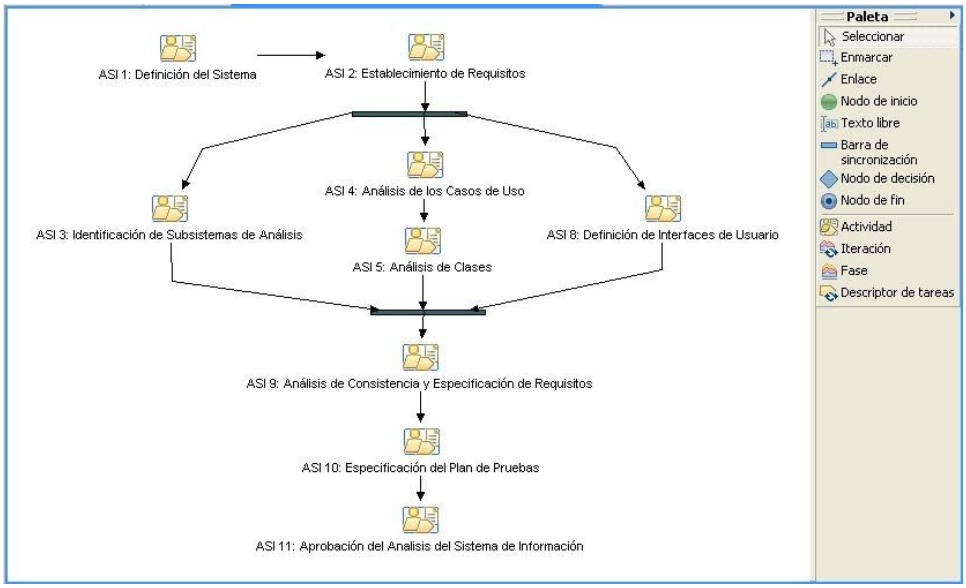


Figura 39. Diagrama de Actividad y paleta de su editor gráfico en EPF Composer.

Estos diagramas suelen utilizarse para mostrar la secuencia de las tareas de una actividad, la secuencia de las actividades de un proceso, etc. EPFC ofrece la opción de generar automáticamente estos diagramas, aunque siempre es posible la posterior edición y retoque (en especial, para resituar los elementos y mejorar la legibilidad).
Las dependencias de precedencia en el orden de las actividades definidas en este diagrama están sincronizadas con las indicadas en el WBS. Ofrecen una visión completa del flujo de una actividad, permitiendo la utilización de calles (verticales) para indicar los roles principales de las tareas de la actividad, nodos de inicio y de fin, nodos de bifurcación y de unión, nodos de decisión, etc. Además, el editor gráfico incluye una serie de botones que permiten mejorar el aspecto visual del diagrama alineando elementos o disponiéndolos automáticamente, entre otras opciones.

5.2.3.2 Diagramas de Detalle de Actividad

Para los elementos de proceso de tipo actividad (sin incluir los subtipos iteración y fase) o de tipo patrón de proceso, los Diagramas de Detalle de Actividad (Activity Detail Diagram) muestran las tareas que forman la actividad, así como los roles que son realizados principales (primary Performer) y los productos de trabajo que son entradas obligatorias (mandatory input) o salidas (output) de cada tarea. La información mostrada en estos diagramas se corresponde con las estructuras de desglose Asignación de equipos (team allocation) y Uso de productosd e trabajo (work product usage). La Figura 40 muestra un diagrama de este tipo y la paleta de su editor gráfico.

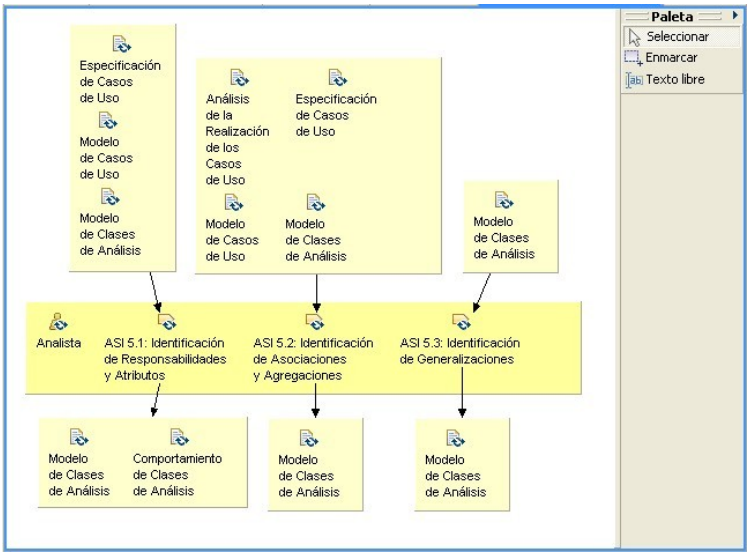


Figura 40. Diagrama de Detalle de Actividad y paleta de su editor gráfico en EPF Composer.

Estos diagramas se generan de forma automática, y en ellos no se pueden modificar los elementos salvo en el aspecto estrictamente visual. Además, se puede incluir texto libre.

5.2.3.3 Diagramas de Dependencias de Producto de Trabajo

Los Diagramas de Dependencias de Producto de Trabajo (Work Product Dependency Diagrams), como el mostrado en la Figura 41, permiten representar cómo uno o más productos de trabajo son usados en la creación o modificación de otros productos de trabajo. Su princicpal utilidad es mostrar la trazabilidad de los productos de trabajo de una actividad o proceso. Estos diagramas existen para todos los WBE de tipo actividad o proceso.

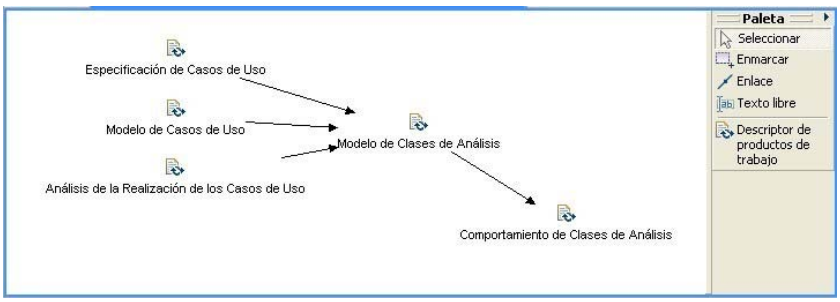


Figura 41. Diagrama de Dependencias de Producto de Trabajo y paleta de su editor gráfico en EPF Composer.

Los productos de trabajo del diagrama son incluidos automáticamente por EPFC pero no así sus relaciones. Por ello, dichas relaciones deben ser dibujadas con el editor gráfico. Dichas relaciones sólo se reflejan en el dibujo (diagrama) y no repercuten fuera de el (en el modelo de proceso en SPEM 2).

5.3 Creando Configuraciones de Método

Una configuración de método es una selección de contenidos de los plug-ins de una librería, de forma que se limita la vista de la librería al subconjunto seleccionado. En una configuración se seleccionan los paquetes y procesos que queremos incluir. La selección de elementos incluidos se puede refinar añadiendo o sustrayendo de la configuración todos los elementos categorizados en una misma categoría.

Las configuraciones de método permiten definir qué elementos aparecerán en la web publicada y cuáles no. También permiten seleccionar los elementos que son visibles para los procesos creados. Ofrecen una alternativa en la creación de procesos, ya que puede definirse un proceso general que incluya contenidos para varios tipos de procesos más específicos. Mediante el uso de configuraciones se seleccionan los contenidos de cada uno de esos tipos de procesos más específicos. Crear los diversos tipos de proyecto es sencillo seleccionando la configuración adecuada para el proceso.

La creación de configuraciones se hace desde la vista *Biblioteca.* Para especificar el contenido de una configuración y refinar dicha selección se siguen los siguientes pasos en la pestaña *Selección de paquete y plug-in* del editor de la configuación:

1. Seleccionar los plug-ins que nos interesen para la configuración.
2. Seleccionar los paquetes de contenido y los procesos de los plug-ins seleccionados que nos interesen en la configuración.
3. Refinar la selección utilizando las categorías que se incluirán y las categorías que se sustraerán de la configuración. De esta forma se pueden incluir o no conjuntos de elementos relacionados de forma lógica en vez de por estar relacionados de forma física (estar incluidos en el mismo paquete).

Al seleccionar los elementos de la configuración, EPF Composer nos avisa si hay elementos referenciados que no se han incluido en la configuración. Se consideran dos tipos de advertencia -errores o avisos- en función del tipo de relación con el elemento ausente.

- Si el elemento ausente es un elemento referenciado por otro elemento que sí pertenece a la configuración (por ejemplo, un producto utilizado en una tarea incluida en la configuración), se indicará con un error.
- Si el elemento ausente es un elemento base para otro elemento que tiene relación de variabilidad amplía, contribuye o reemplaza con dicho elemento ausente, se indicará un aviso.

Dichos avisos y errores los puede solucionar automáticamente EPF calculando el cierre de todos los elementos referenciados e incluyéndolos en la configuración. Ya que el cierre es recursivo, se pueden añadir a la configuración elementos que tal vez no quieran añadirse. La otra opción para solucionar los avisos o errores es seleccionar manualmente los elementos ausentes que se consideren imprescindibles. Hay que tener en cuenta que la existencia de avisos y errores no impide la publicación de la configuración. Lo que ocurrirá es que los elementos ausentes no se publicarán y no aparecerán enlaces a los elementos ausentes referenciados.

En una configuración de método, además de seleccionar los contenidos de la configuración, podemos seleccionar las vistas de la configuración. Las vistas son, básicamente, categorías que

agrupan los contenidos de la configuración. Estas vistas son útiles al publicar la configuración, ya que se generarán árboles de navegación con los elementos definidos en la vista mediante los que explorar el contenido de la configuración. Este concepto no aparece en el estándar SPEM, es un añadido de la herramienta EPF Composer.

Las vistas de una configuración se definen en la pestaña *Vistas* del editor de la configuración. A una configuración se le pueden añadir tantas vistas como se desee. El contenido de las vistas son categorías, por lo que se pueden elegir como vistas las categorías estándar, en las que cada categoría sería una vista, o, preferiblemente, utilizar categorías personalizadas. Ésta segunda opción es más potente, ya que en una categoría personalizada podemos incluir todos los elementos de todos los tipos que queramos. Hay que tener en cuenta que los elementos contenidos en la categoría que se elegirá como vista deben estar incluidos en la configuración, ya que en caso contrario, no se publicarán.

5.4 Publicando Contenidos

Antes de publicar una configuración, es decir, de generar un sitio web basado en dicha configuración, es conveniente comprobar las preferencias generales de publicación en el menú *Ventana->Preferencias->Método->Publicar/Examinar*

Las opciones ofrecidas en dicha opción de menú son las siguientes:

- *Vía de acceso predeterminada*: Seleccionar el directorio donde se guardará la web generada.
- *URL de información de retorno*: Seleccionar la web o dirección e-mail de retorno de información. En todas las páginas publicadas se incluye una dirección cuyo propósito es que se puedan realizar comentarios, informar de errores, etc., sobre el contenido publicado.
- *Incluir el contenido del método en las páginas del descriptor*: Si se marca esta opción, las descripciones realizadas en el contenido de método también se incluirán en las páginas de los descriptores. Si no se marca, cada página de descriptor incluirá un enlace a la página del elemento de método al que referencia para poder consultar sus objetivos, descripciones, etc.
- *Diagramas de roles:* En este apartado se puede ajustar el espaciado horizontal y vertical entre los elementos mostrados en los diagramas de roles y el número de líneas que puede tener el texto de esos elementos. Estos parámetros se deben ajustar en función de la longitud de los nombres de las tareas y los productos con los que están relacionados los roles.
- *Diagramas de actividad:* En este apartado podemos configurar dos opciones relacionadas con la publicación de diagramas de actividad:
    - *Publicar diagramas de actividad para extensiones de actividad que no se hayan modificado:* Si no marcamos esta opción, las páginas de las actividades que extiendan otra actividad y no se hayan modificado, no incluirán el diagrama de actividad. La página cuenta con un enlace a la página de la actividad que extienden, donde se puede consultar el diagrama. Si se marca la casilla, el diagrama de actividad también aparecerá en la página de la actividad que extiende.
    - *Publicar diagramas de detalle de actividad que no se hayan creado en el editor de procesos*: Si marcamos esta opción también se publicarán los diagramas por defecto (que no hayamos creado ex profeso con el editor).

Para publicar una configuración, seleccionamos en el menú principal *Configuración->Publicar*.

Las opciones que podremos elegir sobre la publicación son las siguientes:

1- Seleccionar la configuración que se va a publicar.
2- Seleccionar contenido de la configuración: publicar todos los procesos o sólo los que se seleccionen.
3- Seleccionar opciones de publicación:

- Título y enlaces.
  - *Título:* Título del sitio web que aparecerá en la ventana del navegador web (común para todas las páginas).
  - *Acerca del contenido*: seleccionar una página HTML en la que se muestre información sobre versión de la publicación, autores, copyright, etc. El enlace a dicha página se muestra en la parte superior derecha de las páginas publicadas.
  - *URL de información de retorno*: página web o dirección e-mail de retorno de información. En todas las páginas publicadas se incluye una dirección cuyo propósito es que se puedan realizar comentarios, informar de errores, etc., sobre el contenido publicado. Dicho enlace se muestra en la parte superior derecha de la página publicada.
- Glosario e índice.
  - *Publicar glosario:* Se publicará un glosario con todas las guías de tipo *Definición de término* que se hayan creado.
  - *Publicar índice:* Se publicará un índice similar al de los libros con las palabras marcadas con un tag especial en las descripciones de los elementos, que hayan sido editadas con el editor de texto enriquecido (HTML).
- Aspecto y comportamiento.
  - *Imagen del mensaje de cabecera:* Seleccionar la imagen que aparecerá en la parte superior de las páginas publicadas.
- Validación.
  - *Comprobar hiperenlaces externos*: Comprobar si existen los hiperenlaces externos.
  - *Convertir hiperenlaces rotos a texto sin formato*: Eliminar los hiperenlaces rotos convirtiéndolos en texto plano.
- Diagramas.
  - *Publicar diagramas de detalle de actividad que no se hayan creado manualmente:* Si marcamos esta opción también se publicarán los diagramas por defecto (que no hayamos creado con el editor).
  - *Publicar diagramas de actividad para extensiones de actividad que no se hayan modificado*: Si no marcamos esta opción, las páginas de las actividades que extiendan otra actividad y no se hayan modificado, no incluirán el diagrama de actividad. La página cuenta con un enlace a la página de la actividad que extienden donde se puede consultar el diagrama. Si se marca la casilla, el diagrama de actividad también aparecerá en la página de la actividad que extiende.
- Diseño.
  - *Mostrar la relación de las subcarpetas en los árboles de navegación:* De un nodo del árbol de navegación pueden colgar elementos de diverso tipo: guías, productos, etc. Al marcar esta opción, estos elementos se organizan en carpetas según su tipo.
  - *Mostrar los descriptores de tareas en los árboles de navegación:* Al marcar esta opción, de los nodos del árbol de navegación que sean actividades colgarán los descriptores de tareas incluidos en dichos procesos.
  - *Incluir el contenido del método en las páginas del descriptor:* Si se marca esta opción, las descripciones realizadas en el contenido de método también se incluirán en las páginas de los descriptores. Si no se marca, cada página de descriptor incluirá un enlace a la página del elemento de método al que referencia para poder consultar sus objetivos, descripciones, etc.
  - *Incluir los enlaces a los descriptores relacionados en las páginas de roles, tareas y productos de trabajo:* Si se marca, en las páginas de los elementos

de método aparecen enlaces a todos los descriptores de esos elementos en las actividades en los que se utilicen.

- *Separador predeterminado para páginas de actividad:* Seleccionar la pestaña que se mostrará por defecto en la página de una actividad: Descripción, Estructura de desglose de trabajo, Asignación de equipos o Utilización del producto de trabajo.

4- Seleccionar formato del sitio web:
   - *Sitio web estático*: páginas html organizadas en carpetas.
   - *Aplicación web EE Java empaquetada en un archivo WAR:* Aplicación Java para desplegar en un servidor que soporte Servlets.

En la Figura 42 se muestra un ejemplo de sitio web generado con EPFC.

Figura 42. Ejemplo de sitio generado automáticamente con EPF Composer.

5.4.1 Personalización de la Web.

EPF Composer no ofrece muchas opciones para la personalización de la apariencia de las páginas generadas. De las opciones de publicación comentadas en el apartado anterior, las únicas relacionadas sólo con la apariencia de la web (que no tengan que ver con cómo se publican los elementos de la metodología) son *Título*, *Acerca del contenido*, *URL de información de retorno* e *Imagen del mensaje de cabecera*.

6 Ejemplo de Aplicación – METRICA 3

Como ejemplo de uso de SPEM 2 mediante el empleo de la herramienta EPFC, a continuación se presenta, primero, la manera en qué se ha aplicado SPEM 2 para representar la metodología METRICA 3 [22], en adelante, M3, y después, ejemplos de resultados para los principales tipos de elementos de método y proceso.

6.1 Reglas de Aplicación

En este apartado se introducen las reglas que establecen cómo representar cada aspecto de METRICA 3 (M3) con EPFC.

6.1.1 Aspectos Generales

Primero se dan las reglas sobre los conceptos y aspectos incluidos en el documento "Introducción" de M3.

En cuanto al contenido del **method content** las reglas son:

- Procesos y subprocesos: Cada proceso principal de M3 (PSI, DSI y MSI) se estructura en un content package dentro del method content. La estructura interna del content package se organiza en sucesivos packages, uno por cada subproceso (EVS, ASI, DSI, CSI e IAS para el proceso de desarrollo), y actividad. De esta manera, el method content de M3 tiene una estructura de packages basada en la jerarquía proceso-subproceso-actividad-tarea.
- Roles: como los roles son generales y comunes a toda M3, se definen en un package "comun" fuera de los packages de los procesos.
- Productos: se procede igual que con los roles, utilizando el package "comun" para guardarlos como work products.
- Productos finales: se implementan como deliverables. M3 no especifica los productos finales; por ello se consideran como tales aquellos que se citan como salidas de cada proceso en el documento "Introducción" o en los diagramas resumen que aparecen al inicio del documento de cada proceso.
- Agregación de productos finales: los productos finales formados por otros más simples se representan como deliverables con deliverable parts. En este caso los deliverable parts se implementan como artifacts salvo que a su vez estén formados por otros más simples. Ejemplo: Producto software (salida del proceso CSI) es un deliverable cuyos deliverable parts son los artifacts código fuente de los componentes, procedimientos de operación y administración del sistema, manuales de usuario, …
- Instrucciones: se procede igual que con los roles y productos, utilizando el package "comun".
- Propósito y descripción de los procesos: el propósito se pone como purpose y el texto descriptivo adicional se añade como main description.
- Clasificación de productos: los productos se clasifican de dos maneras en paralelo; primero se emplean domains (standard categories) para distinguir entre documentos, modelos, código, datos, entornos e intangibles; y segundo, se emplean work product kinds para establecer grupos de productos varios: documentos de pruebas, documentos de análisis, etc.
- Interfaces a otros procesos: las interfaces con GP, SEG, CAL y GC se almacenan como subprocesos dentro del package "interfaces" en el method content.

[22] Esta metodología fue elaborada por el Ministerio de Administraciones Públicas de España. Más información está disponible en www.csi.map.es/csi/metrica3/.

- Tareas: cada tarea se implementa como una task. Con fines organizativos, cada tarea se adscribe a una de las siguientes disciplines según el proceso o subproceso al que pertenece: “planificación”, “viabilidad”, “análisis”, “diseño”, “construcción”, “implantación”, “mantenimiento”, “gestión del proyecto”, “seguridad”, “gestión de configuración” y “calidad”.

En cuanto al contenido del **process** las reglas son:
- Capability patterns: cada proceso, subproceso o interfaz tendrá un capability pattern. Si de un proceso existieran varias versiones diferentes para diferentes tipos de proyectos, cada versión será un capability pattern.
- Delivery processes: cada versión de la metodología para un cierto tipo de proyecto será un delivery process diferente.

6.1.2 Aspectos sobre Participantes

El documento “Participantes” de M3 se carga en EPFC siguiendo las siguientes reglas:
- Perfiles: los perfiles generales de M3 (Directivo, Jefe de Proyecto, Consultor, Analista y Programador) se implementan como rol sets.
- Participantes: cada participante se implementa como un role del package “comun”. Del texto descriptivo se pueden extraer las propiedades brief description, main description, key considerations y skills (ver ejemplo en apartado 6.2.1).
- Equipos: los equipos formados por varios roles y que no aparecen como participantes en M3, se implementan como role sets. De esta manera, cada role miembro se puede incluir en el equipo.

6.1.3 Aspectos sobre Técnicas y Prácticas

El documento “Técnicas y Prácticas” de M3 se centra en los aspectos que tienen que ver con las guidances en EPFC, aunque también tiene implicaciones sobre los work products.

En M3 se hace una distinción entre técnicas y prácticas en función del propósito al que responden. Una técnica es un “conjunto de heurísticas y procedimientos que se apoyan en estándares, es decir, que utilizan una o varias notaciones específicas en términos de sintaxis y semántica y cumplen unos criterios de calidad en cuanto a la forma de obtención del producto asociado”. Por contra, una práctica es un “medio para la consecución de unos objetivos específicos de manera rápida, segura y precisa, sin necesidad de cumplir unos criterios o reglas preestablecidas”. A partir de la anterior diferenciación, cabe establecer que, desde el punto de vista de su implementación en EPFC, lo que distingue a las técnicas de las prácticas de M3 es que las primeras tienen asociado uno o varios guidances complementarios: whitepaper de una norma o estándar, template de un work product, etc.

Teniendo en cuenta todo lo anterior, las reglas a aplicar son:
- Técnicas: cada técnica de M3 se corresponde con una guideline (directriz). Por su naturaleza algunas técnicas pueden encajar mejor en otro tipo de guidances, por ejemplo, el “Método Albrecht para el Análisis de los Puntos Función” se puede considerar una estimating guideline; pero, por sencillez, todas las técnicas se anotan como guidelines en EPFC. Del texto descriptivo de una técnica se pueden extraer otros guidances asociados con dicha técnica (ver ejemplo en apartado 6.2.2).
- Prácticas: aunque coincide en nombre con el guidance de tipo practice, en realidad la noción de práctica de M3 corresponde a una guía (guideline). Por ello, se implementan en EPFC como guidelines. Para distinguir en la documentación las técnicas y prácticas de M3, su brief description y main description se inicia con “Técnica …” o “Práctica …”. Además, se

crean dos custom categories con las que se asocian, llamadas “Técnicas de M3” y “Prácticas de M3”. Un ejemplo se puede consultar en el apartado 6.2.3.

- Grupos de Técnicas: M3 establece varios grupos de técnicas (técnicas de desarrollo, diagramas de interacción, modelado de procesos de la organización, técnicas de gesti´n de proyectos, etc.). Cada grupo se implementa como una guideline que incluye a otras guidelines, de forma que se respeta la jerarquía de técnicas del índice del documento.
- Grupos de Prácticas: los grupos de prácticas de M3 (pruebas, sesiones de trabajo) se implementan como practice cuyas references incluyen a sus prácticas individuales.

6.1.4 Aspectos sobre Procesos, Actividades y Tareas

A continuación se presentan las reglas relacionadas con los procesos, y con las actividades y tareas de cada proceso de M3, complementarias a las ya indicados en el apartado 6.1.1. Éstas reglas se extraen del análisis de los documentos de los procesos. A fines de ejemplo, se ha trabajado con el documento de “Análisis del Sistema de Información”.

Las reglas a nivel de **proceso** son:

- Proceso: del apartado inicial de descripción y objetivos se derivan los valores de varias propiedades del capability pattern correspondiente al proceso: name, presentation name, brief description, purpose, y main description (ver ejemplo en apartado 6.2.4).
- Desglose del trabajo: el WBS del capability pattern de cada proceso está formado, en su primer nivel de desglose, por los capability patterns de las actividades que forman el proceso.
- Versiones de proceso: los procesos que tienen diferentes actividades en el desarrollo estructurado y en el desarrollo OO tienen dos capability pattern, uno para cada versión, que se distinguen terminando el nombre en “-est” o “-oo”. Igualmente, cada proceso con las dos versiones, tiene un delivery process acabado en “-est” y otro acabado en “-oo”.
- Flujo del proceso: los diagramas que muestran el flujo de actividades (como el de la figura de la página 4), se implementan en forma de diagramas de actividades del capability pattern del proceso. En la Figura 39 se muestra el equivalente al proceso ASI en versión OO.

Las reglas a nivel de **actividad** son:

- Actividad: una actividad se representa en EPFC como un capability pattern (ver ejemplo en apartado 6.2.5). Las actividades de un mismo proceso se crearán en un paquete de procesos con el nombre del proceo al que pertenencen (por ejemplo, Actividades ASI).
- Desglose del trabajo: el WBS del capability pattern de cada actividad está formado por los task descriptors de las tareas que forman el proceso. Dichos task descriptors se obtienen por herencia de las tasks correspondientes del method content.
- Flujo de trabajo: el flujo de trabajo de las actividades se representa mediante el activity diagram. También se utiliza el activity detail diagram para mostrar las correspondencias entre participantes responsables de cada tarea y los productos de entrada y de salida. En la Tabla 7 se muestran ejemplos de ambos tipos.

Las reglas sobre aspectos del nivel de **tarea** son:

- Tarea: cada tarea se implementa como una task en el content package, dentro de la jerarquía del method content, que corresponde a la actividad a la que pertenece la actividad.
- Versiones de tarea: las tareas que tienen variación entre estructurado y OO se implementan creando una task general de la que heredan la task para estructurado (cuyo nombre acaba en –est) y la task para OO (con nombre acabado en –oo).
- Pasos de tarea: si en el texto descriptivo de la tarea se indica una lista de acciones o pasos a realizar, éstos se implementan en EPFC como steps. En otro caso, se pueden indicar como pasos la realización ordenada de los diferentes productos de salida. Para ahorrar la

descripción del paso, a ser posible, el nombre del paso debe indicar quien hace qué o quien produce qué producto.

- Productos de entrada: se debe decidir cuales son obligatorios, cuyos work products se asocian a la task como mandatory input, y cuales opcionales, cuyos work products se asocian como optional input.
- Productos de salida: sus work products se asocian a la task como outputs.
- Técnicas: se asocian a la task como guidelines.
- Prácticas: se asocian a la task como guidelines.
- Participantes: se asocian como roles a la task. Se debe elegir uno como primary performer y el resto como additional performers.

6.1.5 Aspectos sobre Productos

Al contrario que EPFC, en M3 no se incluye ningún documento o apartado que especifique los tipos de productos y sus características. Por ello, la única información sobre los productos de trabajo que se puede extraer es:

a) El nombre de los productos se obtiene de los productos de entrada y de salida de las tareas.
b) En algunas técnicas, se dan detalles sobre el contenido y forma de algún producto (especificación de un caso de uso, modelo de clases, etc.).

Del análisis de las citadas fuentes de M3, se han extraído las siguientes reglas:

- Productos: cada producto de trabajo que se emplea como entrada o salida en alguna tarea se implementa como un work product dentro del content package “comun”. Por defecto, el tipo del work product será artifact salvo en los casos siguientes:
  - Si es un producto formado por otros productos tangibles más simples, será un deliverable, cuyos deliverable parts son dichos componentes más simples.
  - Si el producto es intangible o es un entorno de trabajo formado por herramientas varias instaladas y configuradas, entonces será un outcome.
- Naturaleza de productos: la naturaleza de un producto puede ser: documento, modelo [23], código, datos (resultados, normalmente contenidos en un archivo), entorno (colección de herramientas adecuadamente instaladas y configuradas de pruebas, de construcción, etc.), o intangible. Dicha naturaleza se especifica como domain (standard category) asociado al work product correspondiente.
- Clasificación de productos: los productos se pueden clasificar de manera opcional usando work product kinds cuyo nombre está formado por la concatenación de la naturaleza del producto y del proceso en el que se produce. Ejemplos: documento de análisis, documento de diseño, datos de gestión del proyecto, etc.

6.2 Ejemplos de Resultados

En esta segunda parte del caso de uso se presentan algunos ejemplos concretos del resultado de aplicar las anteriores reglas.

6.2.1 Participante

El participante “administrador de bases de datos”, del perfil “analista” (página 5 del documento “Participantes” de M3) se implementa como el siguiente rol:

[23] Un modelo es en realidad un tipo especial de documento, pero se distingue como modelo cuando está formado por uno o varios diagramas basados en una cierta notación estándar. Ejemplo: modelo de casos de uso.

Propiedad	Valor
Description:	
Name:	administrador_bd
Presentation name:	Administrador de Bases de Datos
Brief description:	Colabora en el diseño físico de las bases de datos y es responsable del funcionamiento de las bases de datos y sus sistemas gestores.
Main description:	El Administrador de Bases de Datos participa en la obtención del diseño físico de datos, definiendo la estructura física de datos que utilizará el sistema a partir del modelo lógico de datos normalizado o del modelo de clases, teniendo presentes las características específicas del sistema de gestión de base de datos concreto a utilizar, los requisitos establecidos para el sistema de información, y las particularidades del entorno tecnológico, se consiga una mayor eficiencia en el tratamiento de los datos. Si se va a realizar una migración de datos colabora con el equipo de proyecto estimando los volúmenes de las estructuras de datos implicadas, definiendo los mecanismos de migración y carga inicial de datos y participando activamente en su realización. Una vez que el sistema está en producción se ocupa de la gestión y operativa asociada a las bases de datos y al software en el que están implementadas.
Work Products:	
Responsable for	
Work Products that are output …	Diseño físico de datos; mecanismos de migración y carga inicial de datos
Guidances:	
Guidance:	
Categories:	
Role sets:	Analista
Custom categories:	

Tabla 2. Ejemplo de participante implementado como role.

6.2.2 Técnica

La técnica “Casos de uso” (página 8 del documento “Técnicas y Prácticas” de M3) se implementa con la siguiente guidance:

Propiedad	Valor
Description:	
Name:	casos_uso
Presentation name:	Casos de uso
Type:	guideline
Brief description:	Técnica cuyos objetivos son los siguientes: – Capturar los requisitos funcionales del sistema y expresarlos desde el punto de vista del usuario. – Guiar todo el proceso de desarrollo del sistema de información.

Propiedad	Valor
Main description:	La técnica de casos de uso proporcionan, por tanto, un modo claro y preciso de comunicación entre cliente y desarrollador. Desde el punto de vista del cliente proporcionan una visión de "caja negra" del sistema, esto es, cómo aparece el sistema desde el exterior sin necesidad de entrar en los detalles de su construcción. Para los desarrolladores, suponen el punto de partida y el eje sobre el que se apoya todo el desarrollo del sistema en sus procesos de análisis y diseño. Un caso de uso es una secuencia de acciones realizadas por el sistema, que producen un resultado observable y valioso para un usuario en particular, es decir, representa el comportamiento del sistema con el fin de dar respuestas a los usuarios. Aquellos casos de uso que resulten demasiado complejos se pueden descomponer en un segundo nivel, en el que los nuevos casos de uso que intervengan resulten más sencillos y manejables. Para especificar este comportamiento existen una serie de recomendaciones o técnicas que se aplican dependiendo del momento del desarrollo que se esté y de la complejidad del caso de uso. Puede ser desde una simple descripción textual que recoja un requisito funcional a una especificación del caso de uso, e incluso un conjunto de diagramas: - Especificación de un caso de uso - Diagrama de casos de uso
Guidances:	
Guidance:	especificacion_caso_uso; diagrama_caso_uso

Tabla 3. Ejemplo de técnica implementada como guideline.

Además de lo mostrado en la tabla anterior, para la completa implementación de la técnica de "Casos de uso" es necesario tener en cuenta las dos técnicas más concretas que la forman. Así, antes de añadir la guideline anterior se deben añadir las guidelines "Especificación de un caso de uso" y "Diagrama de casos de uso". Ésta última tiene un apartado de notación estándar que se puede añadir dentro del main description del guideline (opción sencilla) o se puede considerar como norma estándar (opción más completa), lo que se traduciría en un whitepaper asociado. La manera de referir que es una Técnica es asociando el guideline a la custom category "Técnicas de M3".

6.2.3 Práctica

Para este ejemplo se ha elegido la práctica de "Pruebas de integración" (página 146 del documento "Técnicas y Prácticas" de M3). Su implementación podría ser con la guideline mostrada en la Tabla 4. Como se ha dicho en las reglas, se implementan con un guideline igual que las técnicas, pero se manifiesta que es una práctica asociándo el guideline a la custom category "Prácticas de M3".

Propiedad	Valor
Description:	
Name:	pruebas_integracion
Presentation name:	Pruebas de integración
Type:	guideline
Brief description:	El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probados unitariamente.
Main description:	En las pruebas de integración se examinan las interfaces entre grupos de componentes o subsistemas para asegurar que son llamados cuando es necesario y que los datos o mensajes que se transmiten son los requeridos. Debido a que en las pruebas unitarias es necesario crear módulos auxiliares que simulen las acciones de los componentes invocados por el que se está probando y a

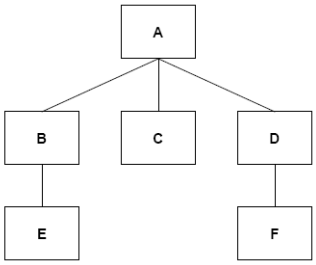
Propiedad	Valor
	que se han de crear componentes "conductores" para establecer las precondiciones necesarias, llamar al componente objeto de la prueba y examinar los resultados de la prueba, a menudo se combinan los tipos de prueba unitarias y de integración. Los **tipos fundamentales de integración** son los siguientes: ▪ *Integración incremental*: Se combina el siguiente componente que se debe probar con el conjunto de componentes que ya están probados y se va incrementando progresivamente el número de componentes a probar. Con el tipo de prueba incremental lo más probable es que los problemas que surjan al incorporar un nuevo componente o un grupo de componentes previamente probado, sean debidos a este último o a las interfaces entre éste y los otros componentes. ▪ *Integración no incremental*: Se prueba cada componente por separado y posteriormente se integran todos de una vez realizando las pruebas pertinentes. Este tipo de integración se denomina también Big-Bang (gran explosión).  En la integración elemental de varios componentes (ver figura anterior), se distinguen las siguientes estrategias de integración: – De arriba abajo (*top-down*). El primer componente que se desarrolla y prueba es el primero de la jerarquía (A). Los componentes de nivel más bajo se sustituyen por componentes auxiliares para simular a los componentes invocados. En este caso no son necesarios componentes conductores. Una de las ventajas de aplicar esta estrategia es que las interfaces entre los distintos componentes se prueban en una fase temprana y con frecuencia. – De abajo arriba (*bottom-up*). En este caso se crean primero los componentes de más bajo nivel (E, F) y se crean componentes conductores para simular a los componentes que los llaman. A continuación se desarrollan los componentes de más alto nivel (B, C, D) y se prueban. Por último dichos componentes se combinan con el que los llama (A). Los componentes auxiliares son necesarios en raras ocasiones. Este tipo de enfoque permite un desarrollo más en paralelo que el enfoque de arriba abajo, pero presenta mayores dificultades a la hora de planificar y de gestionar. - Estrategias combinadas. A menudo es útil aplicar las estrategias anteriores conjuntamente. De este modo, se desarrollan partes del sistema con un enfoque "top-down", mientras que los componentes más críticos en el nivel más bajo se desarrollan siguiendo un enfoque "bottom-up". En este caso es necesaria una planificación cuidadosa y coordinada de modo que los componentes individuales se "encuentren" en el centro.
Guidances:	
Guidance:	

Tabla 4. Ejemplo de práctica implementada como guideline.

6.2.4 Proceso

Para este ejemplo se ha elegido el subproceso de "Análisis del Sistema de Información" (página 1 del documento de igual nombre de M3) en la versión para desarrollo OO. Su implementación se realiza con el capability pattern siguiente:

Propiedad	**Valor**
Description:	
Name:	asi-oo
Presentation name:	ASI: Análisis del Sistema de Información (versión OO)
Brief description:	A partir de los resultados del proceso de estudio de viabilidad y de otras entradas externas se obtienen los requisitos y modelos de análisis necesarios para comenzar el proceso de diseño del sistema.
External ID:	
Purpose:	El objetivo de este proceso es la obtención de una especificación detallada del sistema de información que satisfaga las necesidades de información de los usuarios y sirva de base para el posterior diseño del sistema.
Main description:	En la primera actividad, Definición del Sistema (ASI 1), se lleva a cabo la descripción inicial del sistema de información, a partir de los productos generados en el proceso Estudio de Viabilidad del Sistema (EVS). Se delimita el alcance del sistema, se genera un catálogo de requisitos generales y se describe el sistema mediante unos modelos iniciales de alto nivel. También se identifican los usuarios que participan en el proceso de análisis, determinando sus perfiles, responsabilidades y dedicaciones necesarias. Así mismo se elabora el plan de trabajo a seguir. La definición de requisitos del nuevo sistema se realiza principalmente en la actividad Establecimiento de Requisitos (ASI 2). El objetivo de esta actividad es elaborar un catálogo de requisitos detallado, que permita describir con precisión el sistema de información, y que además sirva de base para comprobar que es completa la especificación de los modelos obtenidos en las actividades Identificación de Subsistemas de Análisis (ASI 3), Análisis de Casos de Uso (ASI 4), Análisis de Clases (ASI 5) y Definición de Interfaces de Usuario (ASI 8). Hay que hacer constar que estas actividades pueden provocar la actualización del catálogo, aunque no se refleja como producto de salida en las tareas de dichas actividades, ya que el objetivo de las mismas no es crear el catálogo sino definir modelos que soporten los requisitos. Para la obtención de requisitos en la actividad Establecimiento de Requisitos (ASI 2) se toman como punto de partida el catálogo de requisitos y los modelos elaborados en la actividad Definición del Sistema (ASI 1), completándolos mediante sesiones de trabajo con los usuarios. Estas sesiones de trabajo tienen como objetivo reunir la información necesaria para obtener la especificación detallada del nuevo sistema. Las técnicas que ayudan a la recopilación de esta información pueden variar en función de las características del proyecto y los tipos de usuario a entrevistar. Entre ellas podemos citar las reuniones, entrevistas, Joint Application Design (JAD), etc. Durante estas sesiones de trabajo se propone utilizar la especificación de los casos de uso como ayuda y guía en el establecimiento de requisitos. Esta técnica facilita la comunicación con los usuarios y en el análisis orientado a objetos constituye la base de la especificación. A continuación se identifican las facilidades que ha de proporcionar el sistema, y las restricciones a que está sometido en cuanto a rendimiento, frecuencia de tratamiento, seguridad y control de accesos, etc. Toda esta información se incorpora al catálogo de requisitos. En la actividad Identificación de Subsistemas de Análisis (ASI 3), se estructura el sistema de información en subsistemas de análisis, para facilitar la especificación de los distintos modelos y la traza de requisitos. En paralelo, se generan los distintos modelos que sirven de base para el diseño (modelo de clases y modelo de interacción de objetos), mediante el análisis de los casos de uso. Se especifican, asimismo, todas las interfaces entre el sistema y el usuario, tales como formatos de pantallas, diálogos, formatos de informes y

Propiedad	Valor
	formularios de entrada. En la actividad Análisis de Consistencia y Especificación de Requisitos (ASI 9), se realiza la verificación y validación de los modelos, con el fin de asegurar que son: – Completos, puesto que cada modelo obtenido contiene toda la información necesaria recogida en el catálogo de requisitos. – Consistentes, ya que cada modelo es coherente con el resto de los modelos. – Correctos, dado que cada modelo sigue unos criterios de calidad predeterminados en relación a la técnica utilizada, calidad de diagramas, elección de nombres, normas de calidad, etc.). En la actividad Especificación del Plan de Pruebas (ASI 10), se establece el marco general del plan de pruebas, iniciándose su especificación, que se completará en el proceso Diseño del Sistema de Información (DSI). La participación activa de los usuarios es una condición imprescindible para el análisis del sistema de información, ya que dicha participación constituye una garantía de que los requisitos identificados son comprendidos e incorporados al sistema y, por tanto, de que éste será aceptado. Para facilitar la colaboración de los usuarios, se pueden utilizar técnicas interactivas, como diseño de diálogos y prototipos, que permiten al usuario familiarizarse con el nuevo sistema y colaborar en la construcción y perfeccionamiento del mismo.
Work Breakdown Structure:	
Breakdown elements:	< lista de capabilities patterns correspondientes a las actividades del proceso: asi-01-oo, asi-02-oo, asi-03-oo, asi-04, asi-05, asi-08-oo, asi-09-oo, asi-10-oo, asi-11 >
Activity diagram:	Se muestra en la Figura 39.

Tabla 5. Ejemplo de proceso implementado como capability pattern.

6.2.5 Actividad

La actividad elegida es “ASI 1 – Definición del Sistema” (página 5 del documento “Análisis del Sistema de Información” de M3) en la versión para desarrollo OO. Su implementación se realiza con el capability pattern de la Tabla 6. Puesto que la mayoría de la información incluida en M3 es a nivel de tarea, la información a nivel de actividad es poca. En este sentido, debe tenerse en cuenta que toda la información mostrada en las tablas resumen de actividad, como la de la página 5, es generada automáticamente por EPFC a partir de las asociaciones a nivel de tarea entre tasks, roles, work products y guidances.

Propiedad	Valor
Description:	
Name:	asi-01-oo
Presentation name:	ASI 1: Definición del Sistema (versión OO)
Brief description:	Determinar el alcance del sistema e identificar el entorno tecnológico, las normas y estándares a aplicar y los usuarios afectados o interesados.
External ID:	
Purpose:	Esta actividad tiene como objetivo efectuar una descripción del sistema, delimitando su alcance, estableciendo las interfaces con otros sistemas e identificando a los usuarios representativos.
Main description:	Las tareas de esta actividad se pueden haber desarrollado ya en parte en el proceso de Estudio de Viabilidad del Sistema (EVS), de modo que se parte de los productos obtenidos en dicho proceso para proceder a su adecuación como punto de partida para definir el sistema de información.
Work Breakdown Structure:	
Breakdown elements:	< lista de tasks heredadas del method content: asi-01.1, asi-01.2, asi-01.3, asi-01.4 >

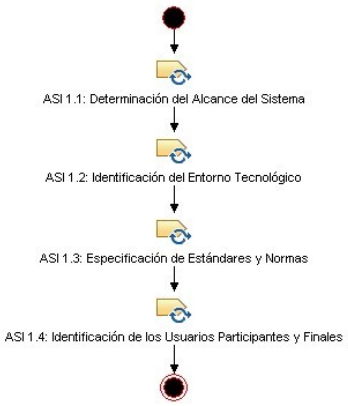
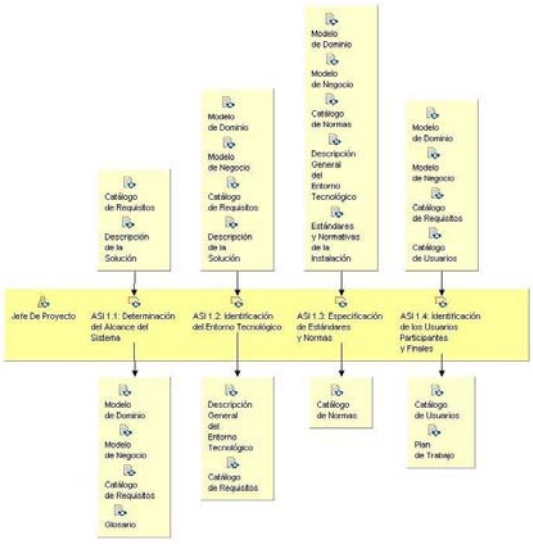
Propiedad	Valor
Activity diagram:	
Activity detail diagram:	

Tabla 6. Ejemplo de actividad implementada como capability pattern.

6.2.6 Tarea

La tarea que se muestra de ejemplo es “ASI 1.1: Determinación del Alcance del Sistema” (página 5 del documento “Análisis del Sistema de Información” de M3) en la versión para desarrollo OO. La Tabla 7 muestra su implementación como un content element de tipo task.

Propiedad	Valor
Description:	
Name:	asi-01-1-oo
Presentation name:	ASI 1.1: Determinación del Alcance del Sistema (versión OO)
Brief description:	En esta tarea se delimita el sistema de información, utilizando como punto de partida el modelo de procesos especificado en la descripción de la solución del proceso Estudio de Viabilidad del Sistema (EVS).
Purpose:	Establecer el alcance del sistema.
Main description:	Antes de la captura de requisitos empleando los casos de uso, puede ser conveniente establecer el contexto del sistema a partir del modelo de negocio obtenido en el proceso Estudio de Viabilidad del Sistema (EVS), y además, opcionalmente, del modelo de dominio. El modelo de negocio especifica los procesos a los que se quiere dar respuesta en el sistema de información, en forma de casos de uso de alto nivel, y el subconjunto de objetos del dominio requerido para ello. En esta tarea se realiza, también, la definición del catálogo de requisitos del sistema

Propiedad	Valor
	a partir del catálogo de requisitos generado en el proceso Estudio de Viabilidad del Sistema (EVS). A medida que se van generando los productos anteriores, se recomienda la definición de un glosario de términos del ámbito de negocio, con el fin de conseguir una mayor precisión en la especificación del sistema de información. El glosario es un catálogo de términos general y común a todos los procesos, y susceptible de ser entrada o salida en cualquier tarea, de modo que por sencillez en las restantes tareas se omite la referencia al mismo. Para obtener esta información es necesario llevar a cabo sesiones de trabajo con los usuarios responsables del sistema de información que se está analizando.
Key considerations:	-
Alternatives:	-
Steps:	
Steps:	▪ Establecer el contexto del sistema en forma de modelos de negocio y de dominio. ▪ Completar el catálogo de requisitos obtenido en EVS. ▪ Elaborar glosario de términos.
Roles:	
Primary performer:	Jefe de Proyecto
Additional performers:	Analistas, Directores de Usuarios
Work Products:	
Mandatory inputs:	Descripción de la solución, Catálogo de requisitos
Optional inputs:	-
Outputs:	Catálogo de requisitos, Glosario, Modelo de negocio, Modelo de dominio
Guidances:	
Guidances:	Diagrama de flujo de datos (guideline), Modelo entidad relación extendido (guideline), Diagrama de clases (guideline), Casos de uso (guideline), Sesiones de trabajo (practice), Catalogación (practice)
Outputs:	Catálogo de requisitos, Glosario, Modelo de negocio, Modelo de dominio
Categories:	
Disciplines:	Análisis
Custom categories:	

Tabla 7. Ejemplo de tarea implementada como task.

Anexo A: Términos e Iconos de SPEM 2 y EPF Composer

Entre corchetes se indican las partes de un término opcionales. Entre paréntesis se representan los términos alternativos.

A.1: Español -> Inglés

Término	Original (Inglés)	Icono
Actividad	Activity	
Activo Reutilizable	Reusable Asset	
Artefacto	Artifact	
Biblioteca [de Métodos] (Repositorio)	[Method] Library	
Categoría	Category	
Categoría Estándar	Standard Category	
Categoría Personalizada	Custom Category	
Clase de Producto de Trabajo	Work Product Kind	
Componente de Entregable	Deliverable Component	
Componente de Proceso	Process Component	
Concepto	Concept	
Configuración [de Método]	[Method] Configuration	
Conjunto de Roles	Role Set	
Consideraciones para el Cálculo	Estimation Considerations	
Contenido de Método	Method Content	
Definición de Término	Term Definition	
Definición de Trabajo	Work Definition	
Descripción de Contenido	Content Description	
Diagrama de Actividad	Activity Diagram	
Diagrama de Dependencias de Producto de Trabajo	Work Product Dependency Diagram	
Diagrama de Detalle de Actividad	Activity Detail Diagram	
Directriz	Guideline	
Disciplina	Discipline	
Documentación	Whitepaper	
Dominio	Domain	
Ejemplo	Example	
Elemento de Contenido	Content Element	
Elemento de Desglose (de Descomposición)	Breakdown Element	

Término	Original (Inglés)	Icono
Elemento de Desglose de Trabajo	Work Breakdown Element	
Elemento de Método	Method Element	
Elemento de Proceso	Process Element	
Elemento de Variabilidad	Variability Element	
Elemento Describible	Describable Element	
Entregable	Deliverable	
Estructura de Asignación de Equipos (de Personal)	Team Allocation Structure	
Estructura de Desglose de Trabajo	Work Breakdown Structure (WBS)	
Estructura de Utilización de Productos de Trabajo	Work Product Usage Structure	
Fase	Phase	
Grupo de Disciplinas	Discipline Group	
Guía (Instrucción)	Guidance	
Guía de Herramienta	Tool Mentor	
Guía para la Estimación (Consideraciones para el Cálculo)	Estimation Considerations	
Herramienta	Tool	
Hito (Objetivo)	Milestone	
Hoja de Ruta (Mapa)	Roadmap	
Informe	Report	
Iteración	Iteration	
Lista de Comprobación	Checklist	
Material de Soporte	Supporting Material	
Paquete de Categorías	Category Package	
Paquete de Contenido	Content Package	
Paquete de Métodos	Method Package	
Paquete de Proceso	Process Package	
Parámetro de Proceso	Process Parameter	
Participante [de Proceso]	[Process] Participant	
Paso	Step	
Patrón de Proceso (patrón de capacidad)	Capability pattern	
Plantilla	Template	
Plantilla para Planificación de Proceso	Process Planning Template	
Plugin [de Método]	[Method] Plugin	
Práctica	Practice	

Término	Original (Inglés)	Icono
Proceso	Process	
Proceso para Despliegue (de entrega)	Delivery Process	
Producto de Trabajo [Definición de]	Work Product [Definition]	
Producto de Trabajo en Uso (Descriptor de Producto de Trabajo)	Work Product Use (Work Product Descriptor)	
Realizador [de Proceso]	[Process] Performer	
Resultado	Outcome	
Rol [Definición de]	Role [Definition]	
Rol Compuesto (Equipo)	Composite Role	
Rol en Uso (Descriptor de Rol)	Role Use (Role Descriptor)	
Sección	Section	
Secuencia de Trabajo	Work Sequence	
Tarea [Definición de]	Task [Definition]	
Tarea en Uso (Descriptor de Tarea)	Task Use (Task Descriptor)	
Tipo de Reutilización de Actividad	Activity Use Kind	

A.2: Inglés -> Español

Término Original	Traducción (español)	Icono
Activity	Actividad	
Activity Detail Diagram	Diagrama de Detalle de Actividad	
Activity Diagram	Diagrama de Actividad	
Activity Use Kind	Tipo de Reutilización de Actividad	
Artifact	Artefacto	
Breakdown Element	Elemento de Desglose (de Descomposición)	
Capability pattern	Patrón de Proceso (patrón de capacidad)	
Category	Categoría	
Category Package	Paquete de Categorías	
Checklist	Lista de Comprobación	
Composite Role	Rol Compuesto (Equipo)	
Concept	Concepto	
Configuration [Method]	Configuración [de Método]	
Content Description	Descripción de Contenido	
Content Element	Elemento de Contenido	
Content Package	Paquete de Contenido	
Custom Category	Categoría Personalizada	

Término Original	Traducción (español)	Icono
Deliverable	Entregable	
Deliverable Component	Componente de Entregable	
Delivery Process	Proceso para Despliegue (de entrega)	
Describable Element	Elemento Describible	
Discipline	Disciplina	
Discipline Group	Grupo de Disciplinas	
Domain	Dominio	
Estimation Considerations	Consideraciones para el Cálculo	
Estimation Considerations	Guía para la Estimación (Consideraciones para el Cálculo)	
Example	Ejemplo	
Guidance	Guía (Instrucción)	
Guideline	Directriz	
Iteration	Iteración	
Library [Method]	Biblioteca [de Métodos] (Repositorio)	
Method Content	Contenido de Método	
Method Element	Elemento de Método	
Method Package	Paquete de Métodos	
Milestone	Hito (Objetivo)	
Outcome	Resultado	
Participant [Process]	Participante [de Proceso]	
Performer [Process]	Realizador [de Proceso]	
Phase	Fase	
Plugin [Method]	Plugin [de Método]	
Practice	Práctica	
Process	Proceso	
Process Component	Componente de Proceso	
Process Element	Elemento de Proceso	
Process Package	Paquete de Proceso	
Process Parameter	Parámetro de Proceso	
Process Planning Template	Plantilla para Planificación de Proceso	
Report	Informe	
Reusable Asset	Activo Reutilizable	
Roadmap	Hoja de Ruta (Mapa)	

Término Original	Traducción (español)	Icono
Role [Definition]	Rol [Definición de]	
Role Set	Conjunto de Roles	
Role Use (Role Descriptor)	Rol en Uso (Descriptor de Rol)	
Section	Sección	
Standard Category	Categoría Estándar	
Step	Paso	
Supporting Material	Material de Soporte	
Task [Definition]	Tarea [Definición de]	
Task Use (Task Descriptor)	Tarea en Uso (Descriptor de Tarea)	
Team Allocation Structure	Estructura de Asignación de Equipos (de Personal)	
Template	Plantilla	
Term Definition	Definición de Término	
Tool	Herramienta	
Tool Mentor	Guía de Herramienta	
Variability Element	Elemento de Variabilidad	
Whitepaper	Documentación	
Work Breakdown Element	Elemento de Desglose de Trabajo	
Work Breakdown Structure (WBS)	Estructura de Desglose de Trabajo	
Work Definition	Definición de Trabajo	
Work Product [Definition]	Producto de Trabajo [Definición de]	
Work Product Dependency Diagram	Diagrama de Dependencias de Producto de Trabajo	
Work Product Kind	Clase de Producto de Trabajo	
Work Product Usage Structure	Estructura de Utilización de Productos de Trabajo	
Work Product Use (Work Product Descriptor)	Producto de Trabajo en Uso (Descriptor de Producto de Trabajo)	
Work Sequence	Secuencia de Trabajo	

Anexo B: Pasos Recomendados en la Creación de Procesos con EPF Composer

El estándar SPEM, y por tanto la herramienta EPF Composer, ofrecen gran flexibilidad a la hora de crear Métodos (o Metodologías). Por ejemplo, se pueden crear repositorios de elementos de método sin la necesidad de crear ningún proceso, o se puede dar el caso contrario: crear un proceso definiendo sus elementos directamente sin haber definido antes ningún repositorio de elementos de método. A la hora de crear procesos también existe flexibilidad ya que, como se comenta en el apartado 5.2.1, éstos se pueden definir mediante un enfoque ascendente o descendente. En el enfoque descendente se crea la estructura de desglose desde el elemento más general (el proceso completo) hacia los elementos más específicos (por ejemplo: fases, las cuales contienen actividades, que, a su vez, contienen tareas). En el enfoque ascendente se crean primero los elementos más sencillos (actividades básicas, las cuales sólo contienen tareas) que se van componiendo para formar elementos de proceso más complejos hasta, finalmente, definir la estructura completa del proceso.

El párrafo anterior ilustra que existen diferentes opciones a la hora de definir un Método con SPEM y EPFC, cada una con sus ventajas y desventajas, especialmente en cuanto a la mantenibilidad, reusabilidad y escalabilidad del Método definido. A continuación se explica la manera que los autores consideran aconsejable de crear un Método desde cero con EPFC, atendiendo a que dichas propiedades de mantenibilidad, reusabilidad y escalabilidad sean las mejores posibles.

Paso 1

El primer paso es **crear una biblioteca de métodos** *(Archivo->Nuevo->Biblioteca de métodos)* que albergue todos los plugins y configuraciones de la metodología que va a definirse. Más información en el apartado 4.2.

Paso 2

El segundo paso es **crear un plugin de método** (*Archivo->Nuevo->Plug-in de método,* o *clic derecho en la vista biblioteca->Nuevo plug-in de método)*. La estructuración de la metodología mediante plugins facilita la modularización y la organización del contenido de ésta. Un plugin contendrá los elementos necesarios para la definición de un proceso (o varios si tienen una relación estrecha). Más información en el apartado 4.2.

Paso 3

El siguiente paso es **crear los elementos de método** (guías, productos de trabajo, roles y tareas, ver más información sobre estos tipos en el apartado 4.3 y sus subapartados). Se aboga por crear primeramente los elementos de método que se van a utilizar posteriormente al definir los procesos, ya que la reusabilidad y mantenibilidad mejoran notablemente. Dichos elementos se podrán reutilizar tantas veces como sea necesario en los procesos, y si hay que hacer cambios en dichos elementos habrá que hacerlos únicamente en un sitio (en la definición del elemento de método), y no en cada una de las ocurrencias del elemento en los procesos.

Subpaso 3.1

Para crear elementos de método, previamente se deben **crear paquetes de contenido** (como mínimo uno) que servirán para organizar el contenido del plugin, creando jerarquías de paquetes. Para crear un paquete de contenido se hace: *clic derecho sobre Paquetes de contenido->Nuevo->Paquete de contenido* (para acceder a Paquetes de contenido en la vista

Biblioteca: *nombre_del_plug-in->Contenido de método->Paquetes de contenido).* También se pueden crear paquetes de contenido dentro de otros paquetes de contenido. Dentro de cada paquete de contenido se pueden crear los elementos de método de cualquier tipo que se consideren oportunos.

Subpaso 3.2

En cada paquete se crean de forma automática cuatro carpetas para almacenar las definiciones de cada tipo de elemento. Para crear un elemento de método se hace: *clic derecho sobre una de dichas carpetas (la del tipo que nos interese)->Nuevo->Tipo_del_elemento.*

El orden en la creación de los elementos de método no es relevante, aunque por comodidad se aconseja seguir el siguiente orden:

1) **Crear las guías**, ya que pueden ser referenciadas desde cualquier otro elemento de método.
2) **Crear los productos de trabajo**. Al crear cada producto, asociarle las guías convenientes.
3) **Crear los roles**. Al crear cada rol, indicar los productos de trabajo de los que son responsables y asociarle las guías convenientes.
4) **Crear las tareas**. Al crear cada tarea, indicar los roles que participan en la tarea, los productos de trabajo que son entrada o salida de la tarea y las guías convenientes.

La ventaja de respetar este orden es que cuando se crea un elemento ya existen todos los elementos con los que está relacionado y, por tanto, su edición es más cómoda. Las descripciones de los elementos de método y sus relaciones con otros elementos se indican en las distintas pestañas de la vista de edición de cada elemento. Se puede consultar información adicional sobre la creación de elementos de método en el apartado 5.1.

Paso 4

Una vez se han creado los elementos de método, el siguiente paso es **crear las categorías estándar**, las cuales permiten categorizar los elementos de método creados. Existen diferentes tipos de categorías estándar: disciplinas para categorizar tareas, dominios y clases de productos de trabajo para categorizar productos de trabajo, conjuntos de roles para categorizar roles y herramientas para categorizar guías de herramienta (un tipo específico de guía) de una herramienta específica (más información en los apartados 4.3.1.5 y 5.1.1). Además, las disciplinas y los conjuntos de roles se pueden agrupar mediante agrupaciones de disciplinas y agrupamientos de conjuntos de roles, respectivamente. Para crear una categoría estándar se utiliza el método habitual: *clic derecho sobre el tipo de categoría que se quiere crear->Nuevo->Tipo_de_categoría* (para acceder a las categorías estándar en la vista Biblioteca: *nombre_del_plug-in->Contenido del Método->Categorías estándar).* La asignación de elementos a una categoría se hace desde una pestaña específica de la vista de edición de la categoría.

Paso 5

Una vez se han creado los elementos de método y las categorías estándar, y antes de poder crear un proceso que hace uso de dichos elementos (sea patrón de proceso o proceso para despliegue), se debe **crear una configuración**. Esto es necesario porque cada elemento de proceso debe especificar su configuración (o configuraciones, ya que un proceso puede estar asociado con varias configuraciones). Las configuraciones son subconjuntos de la biblioteca de métodos que nos permiten restringir la vista de todos los elementos de la biblioteca a únicamente el conjunto de elementos que nos interese. Para crear una configuración: *Archivo->Nuevo->Configuración de método,* o desde la vista biblioteca: *clic derecho sobre Configuraciones->Nuevo-*

>*Configuración de método.* La selección de elementos de la configuración se hace desde la pestaña *Selección de paquete y plug-in* de la vista de edición de la configuración. Más información sobre configuraciones en los apartados 4.6 y 5.3.

Paso 6

Una vez que se ha creado al menos una configuración, se pueden empezar a **crear los patrones de procesos** (más información sobre procesos en los apartados 4.4 y, especialmente, 4.4.3). A la hora de crear procesos se recomienda utilizar un enfoque ascendente, ya que facilita la reutilización de los procesos definidos.

Subpaso 6.1

Para aplicar dicho enfoque ascendente, lo primero es **crear los patrones de proceso básicos** en los que se reutilizan las tareas definidas en el contenido de método. Para crear un patrón de proceso desde la vista de biblioteca: *hacer clic derecho sobre Patrones de proceso->Nuevo->Patrón de Proceso (*se accede a Patrones de proceso desde la vista Biblioteca: *nombre_del_plug-in->Contenido de método->Procesos->Paquetes de proceso).* La descripción y la estructura del proceso se definen en las distintas pestañas de la vista de edición del proceso. Para ver más detalles sobre cómo se reutilizan elementos de método en las estructuras de desglose del proceso, ir al apartado 5.2.1.1. También cabe señalar que los procesos se pueden organizar y modularizar mediante paquetes de proceso.

Subpaso 6.2

Una vez se ha definido la estructura del proceso, se pueden **definir los diagramas** de actividad, de detalle de actividad y de dependencias de productos de trabajo para cada patrón de proceso (más información sobre diagramas en el apartado 5.2.3).

Subpaso 6.3

A partir de los patrones básicos se pueden **crear patrones de proceso más complejos**. En estos patrones se reutilizan los patrones creados con anterioridad. De esta forma se pueden crear jerarquías de patrones que reflejen la estructura de fases, iteraciones, actividades, etc. de un proceso. A la hora de aplicar un patrón en otro existen dos opciones: copiarlo o ampliarlo. Si el patrón que se va a aplicar no se piensa personalizar dentro del patrón aplicado, se recomienda utilizar la opción *ampliar*, ya que resulta mucho más fácil de mantener. Las modificaciones posteriores sólo hace falta hacerlas en el patrón original, y los cambios se reflejarán en todos los patrones donde se aplicó dicho patrón mediante ampliación. Sin embargo, si el patrón debe personalizarse dentro del proceso en que se aplica, la aplicación mediante ampliación no permite hacerlo (permite añadir nuevos elementos, pero no eliminar elementos existentes que no sean útiles en el contexto en el que se aplica el patrón), por lo que habría que aplicarlo mediante copia. Un patrón copiado sí puede personalizarse para adaptarse al contexto en el que se aplica, sin embargo, su mantenibilidad disminuye, ya que los cambios posteriores deben hacerse sobre todas las copias manualmente. Se puede encontrar más información sobre la reutilización de patrones de proceso en el apartado 5.2.2. Una vez se han aplicado los patrones se pueden definir los diagramas para el patrón de proceso creado según se indica en el subpaso 6.2.

Paso 7

Una vez se han creado los patrones de proceso necesarios se pueden **crear los procesos para despliegue** reutilizando dichos patrones de proceso. La reutilización de patrones de proceso en procesos para despliegue se hace de igual forma que la reutilización de patrones de proceso

dentro de otros patrones comentada en el punto anterior. También se pueden definir diagramas para el proceso para despliegue.

Con lo indicado en los pasos 6 y 7 se puede implementar una *jerarquía tarea-actividad-subproceso-proceso* (típica de los estándares ISO) de la siguiente forma: Se crearían los patrones básicos de las actividades que sólo incluyen tareas (subpaso 6.1). Después se crearían los patrones de los subprocesos en los que se reutilizan los patrones de las actividades (subpaso 6.3). Finalmente se crearían los procesos como procesos para despliegue en los que se reutilizan los patrones de los subprocesos (paso 7).

Paso 8

Una vez que tenemos todos los procesos y todos los elementos de método definidos podemos publicarlos. Como se ha comentado en el apartado de crear una configuración, cuando se publica un método lo que realmente se hace es publicar una configuración de método, la cual puede corresponderse con el contenido de toda la biblioteca, el de un plugin o cualquier otro subconjunto de elementos de la biblioteca que sea conveniente. Por ello, antes de publicar la configuración hay que **definir la vista de la configuración** (o vistas, ya que se pueden definir varias). Una vista de la configuración es una estructura jerárquica que sirve para navegar por los contenidos publicados de forma más accesible, ya que dicha estructura se publica como un árbol de navegación (menú desplegable que aparece a la izquierda) en el sitio web generado en la publicación. Los contenidos de dicha vista se pueden personalizar completamente, ya que las vistas se definen mediante categorías personalizadas, las cuales pueden contener los elementos que queramos con la estructura jerárquica que queramos. Más información sobre categorías personalizadas y vistas de configuración en los apartados 5.1.1 y 5.3.

Paso 9

Una vez se ha definido la vista (o vistas) de la configuración se puede **publicar la configuración**. La publicación de contenidos y las opciones disponibles se comentan en detalle en el apartado 5.4.

Anexo C: Opciones de Exportación

EPF Composer ofrece varias opciones para exportar el contenido definido con la herramienta las cuales tienen diferentes finalidades. En la Figura 43 se muestran las opciones de exportación que se comentarán a continuación.

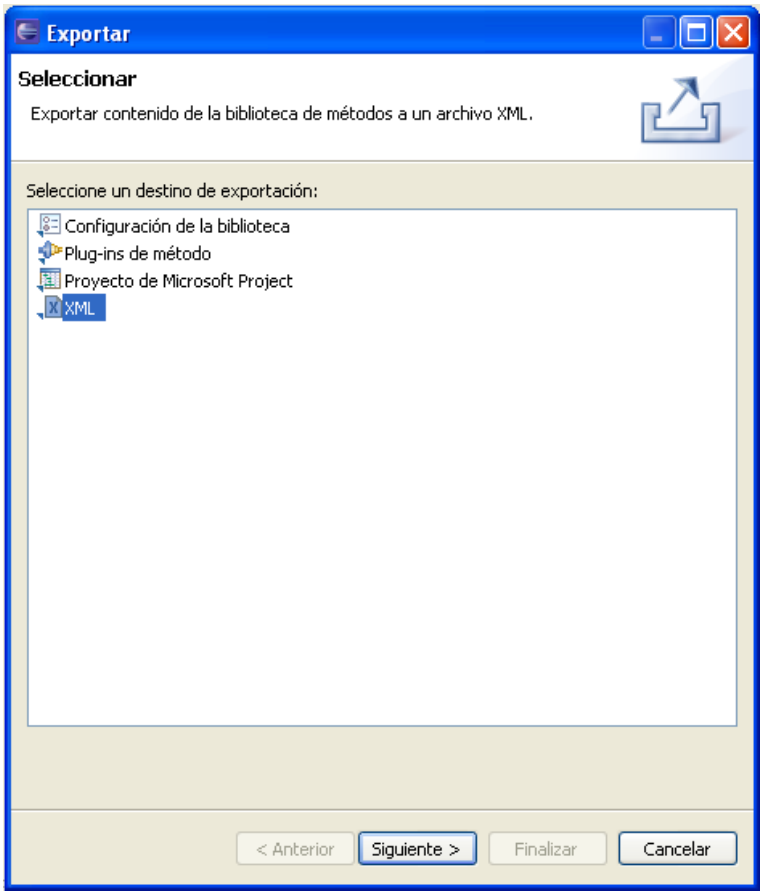


Figura 43. Opciones de exportación de EPF Composer.

C.1: Exportar Configuraciones y Plugins de Método

Mediante las opciones de exportación “Configuración de la biblioteca” y “Plug-ins de método” se pueden exportar configuraciones y plugins respectivamente, para posteriormente importarlos desde otra biblioteca de método. Esta opción es útil cuando una biblioteca necesita reutilizar elementos definidos en otra biblioteca. Puesto que los elementos de una biblioteca no pueden hacer referencia a los elementos de otra biblioteca, la solución pasa por exportar los elementos necesarios y posteriormente importarlos en la biblioteca que los necesite. Hay que tener en cuenta que al exportar se crea una copia del contenido, y por tanto, los cambios posteriores a la exportación sobre dichos elementos de la biblioteca original no se reflejarán en las bibliotecas que los hayan importado.

La mayoría de las veces lo que interesa es exportar plugins de método, pero la opción de exportación de configuraciones permite definir con la granularidad deseada la selección de elementos que se exportarán. Esta granularidad puede ir desde unos pocos elementos hasta una biblioteca de métodos entera.

Estos dos tipos de exportación se hacen en un formato interno, el cual no está pensado para reutilizarse por otras herramientas.

C.2: Exportar a Microsoft Project

Mediante la opción de exportación “Proyecto de Microsoft Project” se pueden exportar procesos a un formato procesable por la herramienta Microsoft Project. En este apartado se explica la relación de esta opción con las Plantillas para Planificación de Procesos (Process Planning Templates) de SPEM 2 y cómo se realiza la exportación a Microsoft Project.

EPFC implementa ampliamente el estándar SPEM 2, aunque no al cien por cien. Uno de los elementos del estándar SPEM 2 que EPF Composer no implementa son las Plantillas para Planificación de Procesos. Como se explica en el apartado 4.4.3, SPEM define las Plantillas para Planificación de Procesos como un tipo especial de Proceso el cual se ha preparado para su instanciación en una herramienta de planificación de proyectos. Aunque EPFC no permite definir Plantillas para Planificación de Procesos, sí permite exportar los otros dos tipos de Procesos (Patrones de Proceso y Procesos para Despliegue) a herramientas de planificación de proyectos, en concreto a Microsoft Project.

EPF Composer no permite añadir información de instanciación de la realización (fecha de comienzo, fecha de fin, duración, etc.) a los elementos de un proceso. Lo que sí se puede hacer de interés para los plan del proyectos es definir las relaciones de precedencia entre elementos y el tipo de dependencia (ver los apartados 4.4.2 y 5.2.1.2). Estas opciones son muy útiles a la hora de exportar el proceso a una herramienta de planificación de proyectos.

Antes de exportar un proceso a un proyecto de MS-Project hay que comprobar en la pestaña “Estructura de desglose de trabajo” de la vista de edición del proceso que los elementos que queremos exportar tienen activada la casilla “Planeado”. También se debe comprobar que los tipos de dependencias entre los elementos de desglose son los adecuados. Cuando se crea una dependencia de precedencia entre dos elementos de desglose, el tipo de dependencia por defecto es “Finalización a inicio”. El tipo de dependencia de un elemento se puede modificar en la pestaña “General” de la vista “Propiedades” cuando se tiene seleccionado dicho elemento en la vista de edición del proceso.

Para exportar a MS-Project se siguen los siguientes pasos:

1- Seleccionar la opción *Archivo->Exportar*.
2- Seleccionar “Proyecto de Microsoft Project”.
3- Seleccionar el proceso (patrón de proceso o proceso para despliegue) que se quiere exportar.
4- Seleccionar la configuración asociada con el proceso con la que se quiere hacer la exportación, ya que un proceso puede tener varias configuraciones asociadas. En función de la configuración seleccionada, habrá elementos que no se exporten (los que no pertenezcan a la configuración seleccionada). Al exportar el proceso existe la opción de publicar el sitio web del proceso, publicándose todos los elementos de la configuración o sólo aquellos relacionados con el proceso. También existe una opción para ignorar la propiedad “Planeado”, definida en la estructura de desglose de trabajo, y así exportar todos los elementos del proceso, tengan activada o no la opción “Planeado”.
5- Si se ha seleccionado la opción para publicar el sitio web del proceso, seleccionar las opciones de publicación del mismo. Dichas opciones son las mismas que cuando se publica una configuración normalmente (ver las opciones de publicación – punto 3- en el apartado 5.4).
6- Seleccionar el nombre y el directorio del archivo de MS-Project generado.

En la exportación se genera un archivo XML que se puede abrir con MS-Project.

El archivo generado contiene la estructura de tareas y actividades (tareas resumen) definidas en el proceso exportado, es decir, el WBS. Las dependencias entre tareas también se reflejan en el archivo XML generado. Los roles se exportan como recursos de Microsoft Project. Dichos roles se conectan con las tareas o tareas resumen correspondientes. En el caso de elementos no exportados (al no marcar su propiedad "Planeable"), los roles definidos se asociarán a la tarea o tarea resumen correspondiente al elemento padre de dichos elementos no exportados. Las fechas del proyecto serán unas fechas por defecto que deberán modificarse desde MS-Project, así como el resto de elementos del plan de proyecto que no se pudieron definir y exportar desde EPF Composer, como los recursos no humanos, costes, esfuerzos, etc.

C.3: Exportar a XML

EPF ofrece la posibilidad de exportar a formato XML uno o varios plugins o una biblioteca de métodos entera. El archivo XML generado contendrá toda la información de los plugins exportados, excepto los diagramas de los procesos, los cuales se mantendrán como archivos XMI con la estructura de directorios habitual en las bibliotecas de métodos de EPF.

Con la exportación a XML se consigue tener toda la información de varios plugins o una biblioteca de métodos en un único archivo (sin tener en cuenta los diagramas y otros recursos complementarios como imágenes, archivos, etc.). Hay que tener en cuenta que si se exporta una biblioteca de métodos con muchos plugins y muchos elementos, el archivo XML resultante puede ser muy grande.

Al igual que mediante las formas de exportación comentadas antes (ver apartado C.1: Exportar Configuraciones y Plugins de Método), los elementos exportados en un archivo XML se pueden importar posteriormente desde una biblioteca de métodos.

El archivo XML generado sigue un esquema XSD. Dicho XSD puede descargarse de la siguiente dirección:
http://www.eclipse.org/epf/tool_component/xml_schema.php.
Al usar un XSD/XML estándar, otras herramientas que permiten trabajar con esquemas XML, también podrán leer y procesar los archivos XML exportados.

Anexo D: Otras Preguntas Frecuentes

En este anexo se dan respuestas a preguntas relacionadas con la creación de metodologías, edición y publicación de contenidos, etc., que aparecen frecuentemente al trabajar con SPEM 2 y EPFC.

D.1: ¿Cómo generar distintas vistas de una metodología?

Para generar distintas vistas de una metodología se utilizan las **configuraciones**. Se pueden crear varias configuraciones en las que se incluyan sólo los elementos que se consideren necesarios. Así, por ejemplo, se podría crear una vista para desarrolladores en la que no aparezcan actividades, tareas, productos, etc. que no tengan relación directa y estricta con el desarrollo del código (como por ejemplo gestión del proyecto, implantación,…) Hay que tener en cuenta que cuando se publica contenido, lo que se publica es una configuración. Por tanto, si queremos publicar varias vistas habrá que hacerlo por separado, dando como resultado varios sitios web distintos. Para más información sobre configuraciones, ver los apartados 4.6 y 5.3.

D.2: ¿Cómo adecuar una metodología a proyectos específicos?

Para adecuar una metodología a proyectos específicos se aprovecha la capacidad de **reutilización** y la **distinción entre elementos de método y elementos de método en uso (descriptores de elemento)** ofrecida por SPEM y EPFC. Mediante dicha distinción, cada aparición de un elemento de método en un proceso se convierte en un descriptor de ese elemento, el cual se puede modificar para adaptarlo al contexto específico en el que aparece (es decir, las características del proceso o proyecto específico). Para más información sobre elementos de método en uso (descriptores) ver los apartados 4.4.2.2 y 5.2.1.3.

Al crear un tipo de proyecto específico se pueden reutilizar los patrones de proceso creados para la metodología general. Para facilitar el mantenimiento, los patrones del proyecto específico que no necesitan ninguna modificación respecto al patrón definido de forma general se pueden aplicar utilizando el tipo de variabilidad “amplía”. Sin embargo, es frecuente que el proyecto específico requiera modificaciones respecto al proceso general como, por ejemplo, que en una actividad no se realicen todas las tareas o que en una tarea no se utilicen todos los productos indicados, que no se realice alguno de los pasos, no participe algún rol, etc. En caso de que los patrones del proceso específico requieran **modificaciones con respecto al patrón general**, caben dos opciones:

a) **Aplicar mediante copia** el patrón definido de forma general **y realizar las modificaciones** necesarias. Al aplicar el patrón mediante copia, se pueden modificar sus elementos para adaptarlos a las necesidades del proceso en el que se han aplicado. Este caso es útil y adecuado si los cambios que hay que hacer son mínimos, ya que el esfuerzo es menor que en la otra opción. Tiene como desventaja la mantenibilidad, ya que los cambios posteriores que se realicen sobre el patrón original no se reflejarán automáticamente en las copias y habrá que realizarlos manualmente.
b) En caso de que las diferencias entre el proyecto específico y el proceso general sean significativas es preferible **crear patrones de proceso específicos**, ya que modificar los patrones copiados puede llevar más tiempo que crear un nuevo patrón de proceso que contenga los elementos necesarios. En estos patrones se reutilizarán los elementos de método generales y se modificarán para adaptarse al contexto específico del proyecto. Los patrones creados se pueden aplicar mediante ampliación, con lo que se facilita su mantenimiento.

Es importante respetar lo máximo posible las capacidades ofrecidas por SPEM 2. Por ello, ya que SPEM ofrece los descriptores de elementos para poder adaptar un mismo elemento de método a varios contextos (en este caso el de cada proyecto específico), en lugar de crear nuevos elementos de método con el contenido específico del contexto del proyecto en el que se aplican, es mejor utilizar dicho mecanismo de herencia y copia para adaptar los elementos (tareas, productos, etc.) definidos para el proceso general a un tipo específico de proyecto. Un caso diferente es que el proyecto específico requiera elementos que no están contemplados en la metodología general. En este caso sí que es inevitable tener que crear dichos elementos.

D.3: ¿Cómo preparar versiones reducidas para entregar a terceros?

Para entregar versiones reducidas a terceros también se utilizan las **configuraciones** y la **capacidad de exportarlas** que ofrece EPF. Así, se incluirán en la configuración todos los elementos que deban entregarse a terceros. Se exportará la configuración y posteriormente se importará en una nueva biblioteca de métodos vacía que únicamente contendrá la configuración exportada y todos los elementos seleccionados en dicha configuración. De esta forma, se puede entrega a terceros una biblioteca de métodos que únicamente contiene los elementos que se desean entregar. Hay que tener en cuenta que las referencias a los elementos que no se incluyen en la configuración exportada se pierden, y por tanto, hay que tener cuidado ante posibles pérdidas de información.

D.4: ¿Cómo incorporar documentos y enlaces externos en las descripciones de elementos y procesos?

Se pueden incorporar archivos o enlaces externos en la página de cualquier elemento o proceso publicado. Ambos se pueden añadir desde la vista de edición del elemento o proceso, en cualquiera de los campos de texto que permitan formato RTF (típicamente se haría en la Descripción principal). Para ello se abre el editor RTF pulsando el icono al lado de Descripción principal (u otro campo si se prefiere) y se pulsa el icono que aparece en la barra superior del editor de RTF. Entonces aparecerá la siguiente ventana:

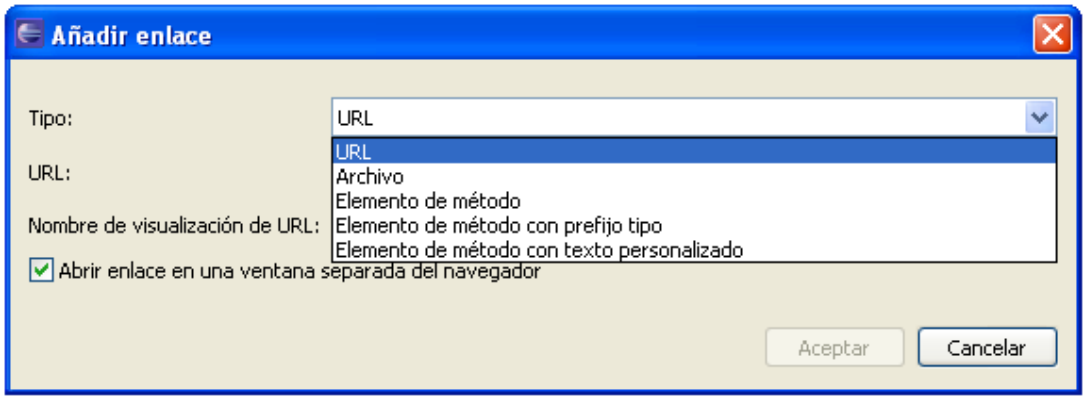


Figura 44. Diálogo para añadir un enlace a una URL externa, un Archivo o un Elemento de método.

Desde ella se puede añadir un enlace a cualquier otra página (por ejemplo, que contenga documentación) indicando su URL yo se puede añadir un archivo, el cual se copiará automáticamente en la estructura del sitio web generado sin tener que preocuparnos nosotros de su ubicación.

Puede ser interesante que, por cuestiones de mantenibilidad, en vez de incluir los archivos y enlaces a otras páginas directamente en la descripción de los elementos, se cree una guía de tipo "material de soporte" (supporting material) y sea en la descripción de ésta donde se inserte el archivo o el enlace a otra página, acompañado de una descripción breve del contenido del archivo o de la página enlazada. Este material de soporte se podrá añadir como guía a cualquier elemento en el que se quiera hacer referencia a dicho archivo o enlace externo. De esta forma si la página o archivo que se quería enlazar cambia de ubicación sólo habrá que cambiar ésta en un único elemento (el material de soporte definido) en lugar de en todos los elementos donde aparezca dicho archivo o enlace.

Por ejemplo, si en nuestra metodología publicada se quieren añadir enlaces a páginas externas con contenido interesante sobre definición de casos de uso en los elementos relacionados con casos de uso (por ejemplo, los productos "Modelo de casos de uso" y "Especificaciones de casos de uso", las tareas "Identificar casos de uso" y "Especificar casos de uso", etc.), en lugar de incluir los enlaces a dichas páginas en la descripción principal de cada uno de los elementos, se podría crear un material de soporte llamado "Información sobre casos de uso" en el que se comenten las fuentes de información y se incluyan los enlaces a esas páginas (ver Figura 45). Los elementos interesados incluirán como guía ese material de soporte. Si en algún momento una de las páginas enlazadas deja de existir o cambia su URL, no será necesario borrar o modificar su enlace en todas las descripciones de los elementos en las que aparecía dicho enlace, ya que será suficiente con modificarlo en el material de soporte que se ha definido.

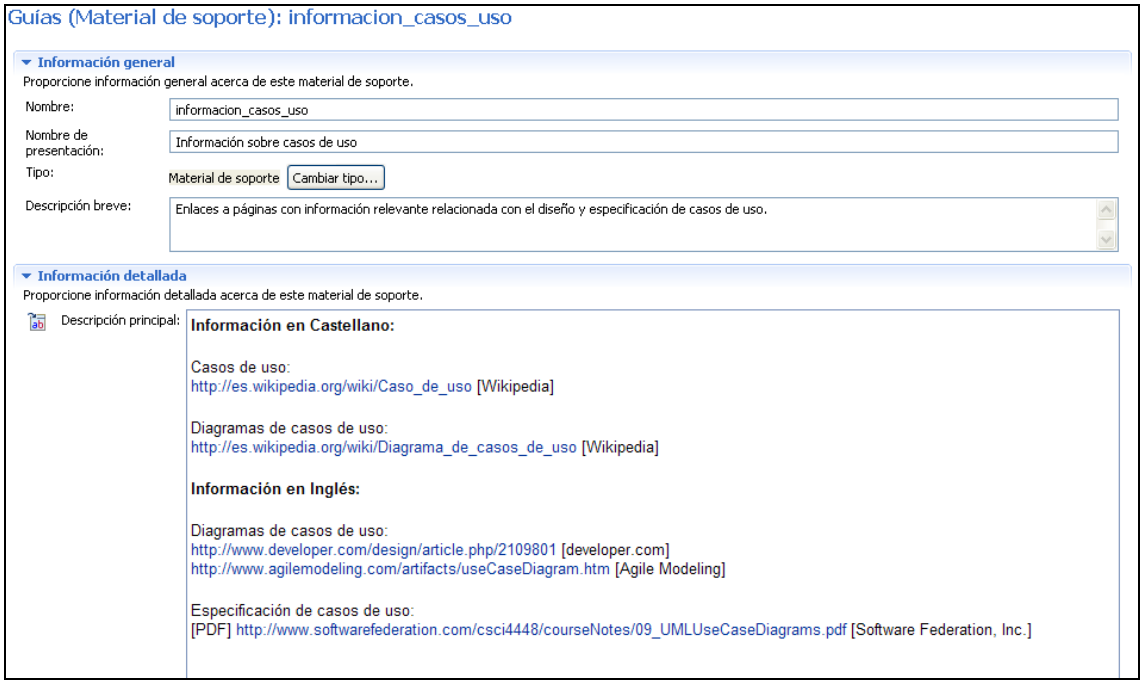


Figura 45. Ejemplo de material de soporte con enlaces a páginas externas.

En la Figura 46 se muestra un ejemplo del artefacto "Modelo de casos de uso" que incluye como material de soporte la guía "Información de casos de uso" mostrada en la Figura 45.

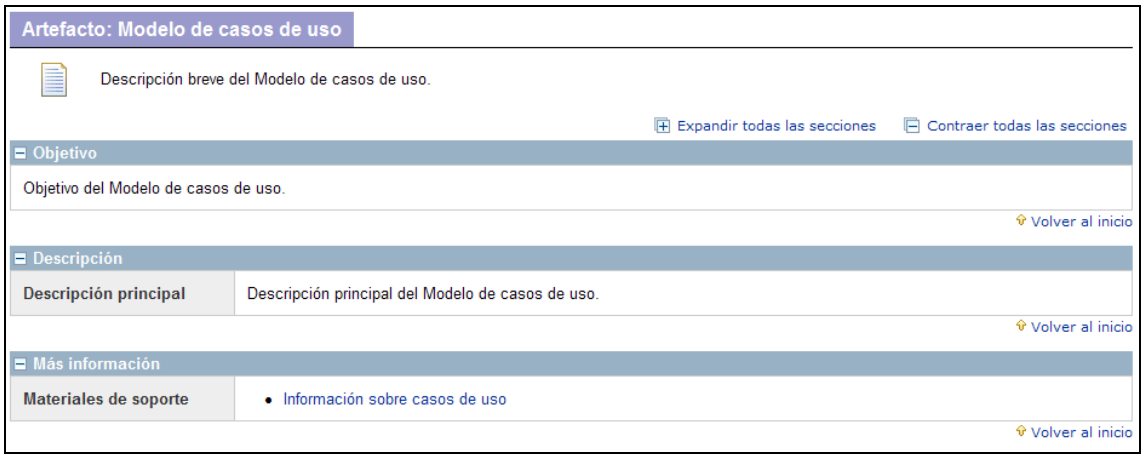


Figura 46. Elemento publicado que incluye el material de soporte de la Figura 45 como guía.

El caso de las plantillas (templates) es un poco especial. Puesto que el objetivo de dicho elemento es incluir un enlace a un documento que sirva como plantilla para un producto de trabajo, en la ventana de edición aparecen las opciones para incluir directamente el documento plantilla, sea como archivo incluido en la web generada o como un enlace externo a un sitio remoto en el que se encuentre dicha plantilla. Así, no es necesario incluir el archivo o un enlace a él de la forma que se ha comentado anteriormente. En la siguiente figura se han resaltado las dos opciones para adjuntar archivos a una plantilla:

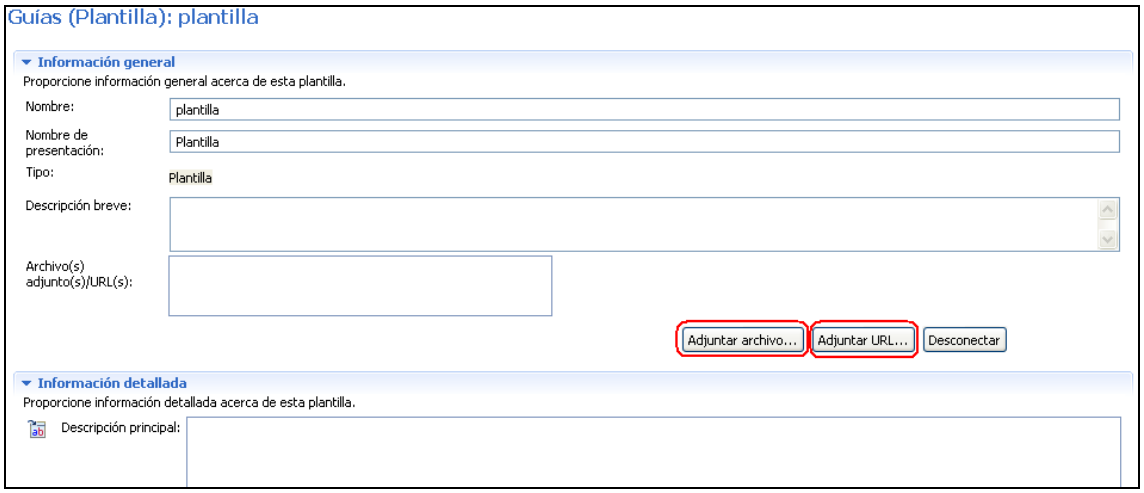


Figura 47. Opciones disponibles para adjuntar archivos a una plantilla.

D.5: ¿Cómo definir la estructura de la Web publicada?

Una de las ventajas de EPF Composer es la generación automática de un sitio web en el que se publiquen todos los elementos que nos interesen. Toda la estructura de la web publicada se genera automáticamente, no siendo necesario crear ningún directorio ni archivo manualmente. Tras la publicación, el sitio web generado está listo para desplegarlo en un servidor sin tener que modificar nada. Como contrapunto a la sencillez en la generación del sitio web se puede señalar la desventaja de la poca flexibilidad ofrecida por la herramienta para personalizar la web generada.

Al publicar contenidos (ver el apartado 5.4) se debe seleccionar el formato de la web publicada, para el que existen dos opciones:

a) Una es que el sitio web generado sea estático, es decir, todos los archivos html y sus recursos (imágenes, hojas de estilo, etc.) están organizados en carpetas que únicamente hay que desplegar en cualquier servidor http.
b) La otra opción es que el sitio generado sea una aplicación web Java empaquetada en un archivo WAR. Dicho archivo WAR hay que desplegarlo en un servidor web que soporte Servlets (como por ejemplo, Apache Tomcat). La ventaja de generar el sitio web en este formato es que se incluye capacidad para realizar búsquedas de contenido en todo el sitio web. La desventaja es que se debe contar con dicho servidor que soporte Java Servlets 2.3 o superior. Este tipo de servidores con capacidades dinámicas son algo más lentos que los servidores http sencillos.