



Departamento de Informática
Universidad de Valladolid
Valladolid - España

Representación de Requisitos mediante Redes de Petri
Coloreadas

Oscar López¹, Miguel A. Laguna², and Francisco J. García³

¹ Instituto Tecnológico de Costa Rica

olopez@infor.uva.es

² Universidad de Valladolid

mlaguna@infor.uva.es

³ Universidad de Salamanca

fgarcia@usal.es

Resumen La documentación de requisitos juega un papel fundamental en el desarrollo de software de calidad para satisfacer las necesidades de los *stakeholders*. Si los documentos de requisitos se pueden generar a partir de elementos reutilizables (*assets*) que encapsulan el esfuerzo del desarrollo en diferentes niveles de abstracción, se pueden obtener especificaciones de mejor calidad y en un tiempo menor que si se desarrollan desde cero. En este informe técnico se presenta un metamodelo de requisitos y se propone una correspondencia entre ese metamodelo y los elementos de las redes de Petri coloreadas (CPN). De esta manera, con base en el soporte formal dado por las CPN para describir modelos se pretende establecer una técnica para organizar la información de los requisitos del sistema y así abordar su reutilización para el desarrollo de especificaciones de software.

Informe Técnico IT-DI-2002-1

1. Introducción

La documentación de requisitos juega un papel fundamental en el desarrollo de software de calidad para satisfacer las necesidades de los *stakeholders*. Si los documentos de requisitos se pueden generar a partir de elementos reutilizables (*assets de requisitos*) que encapsulan el esfuerzo del desarrollo en diferentes niveles de descripción, se pueden obtener especificaciones de mejor calidad y en un tiempo menor que si se desarrollan desde cero. Sin embargo, las estrategias de reutilización de requisitos deben gestionar la diversidad de técnicas de modelado de los requisitos. Se requiere un lenguaje preciso y, a ser posible estándar, para describir y gestionar los diferentes modelos de requisitos de un dominio.

En el proceso de la ingeniería de requisitos, las necesidades de comunicación entre los «stakeholders» y de integración de los «viewpoints» provocan la proliferación de diversas técnicas de modelado de requisitos de aspecto semiformal [RP99]. Esta diversidad de técnicas de modelado y la existencia de diferentes niveles de descripción de requisitos obstaculizan el establecimiento de un marco de trabajo para reutilización de requisitos. Para afrontar estos obstáculos realizamos dos acciones generales. La primera acción consiste en restringir el número de técnicas de modelado a seis que son ampliamente utilizadas (escenarios, casos de uso, diagramas de actividades, flujos de datos, documentos-tareas y «workflows»). La segunda acción consiste en establecer un marco de reutilización de requisitos conformado por tres etapas: (a) describir los diagramas mediante un modelo común o metamodelo, (b) realizar la proyección de los diagramas integrados por el modelo hacia un lenguaje formal, y (c) organizar la expresión formalizada de los requisitos dentro de un repositorio. En [LLG02a] se cubre la etapa (a) de nuestro marco de reutilización de requisitos. En el presente informe técnico abordamos la etapa (b) para lo cual empleamos las redes de Petri coloreadas (CPN, por sus siglas en inglés) [Jen97a]. Finalmente, la etapa (c) se aborda en [LLG02b].

Con la reutilización se introduce el enfoque de «ingeniería de dominios» como un proceso concurrente con la «ingeniería de aplicaciones» [Kar95,CSPD92]. La ingeniería de dominios permite capturar, organizar y representar la información útil para el desarrollo de sistemas software de manera que esta pueda ser reutilizada para crear nuevos sistemas software. La ingeniería de aplicaciones hace referencia a la información codificada durante la ingeniería del dominio mediante la comparación de situaciones nuevas y previas, y durante la adaptación para dar soporte a nuevos requisitos [PD89]. La producción industrial de software exige que las labores de ingeniería de dominio e ingeniería de la aplicación se realicen con apoyo de herramientas automáticas, lo cual plantea la necesidad de contar con modelos abstractos y precisos que puedan ser analizados sistemáticamente.

La carencia de una base formal suficiente en los modelos de requisitos limita las posibilidades de soporte de herramientas para el análisis sistemático de los modelos de requisitos en el desarrollo con reutilización. Por tanto, es necesario contar con técnicas para expresar los modelos de requisitos en una representación que permita la comprobación automática de ciertas propiedades. En este sentido, el fundamento de nuestra

propuesta es que si el metamodelo [LLG02a] describe diferentes representaciones de requisitos, y ese metamodelo se representa con el formalismo de CPN, se logra una descripción formal de los modelos originales de requisitos como soporte al esquema análisis y organización de requisitos para reutilización que se presenta en [LLG02b].

En este informe técnico se hace una propuesta de correspondencia entre un metamodelo de requisitos y los elementos de las redes de Petri coloreadas. Nosotros sostenemos que las CPN proporcionan un buen soporte para la gestión automática de diagramas de requisitos en un repositorio dado que existen técnicas y herramientas robustas para el análisis de propiedades de modelos de CPN [Jen97b]. El resto del informe se distribuye en 4 secciones. La sección 2 presenta una introducción a los modelos de redes de Petri. La sección 3 trata de la representación de diagramas de requisitos mediante CPN. La sección 4 presenta trabajos relacionados. Finalmente, la sección 5 concluye este informe técnico y propone acciones de trabajo adicionales.

2. Las Redes de Petri y el Modelado de Requisitos

Las redes de Petri (PN, por sus siglas en inglés) [Sil85] han sido ampliamente utilizadas en el modelado de sistemas cuyo comportamiento es guiado por un patrón de reglas complejas como es el caso de los sistemas software. Las PN -también llamadas «redes de Petri clásicas» son grafos orientados que contienen dos tipos de nodos: *lugares* y *transiciones*. Mediante círculos se representan los lugares y mediante rectángulos las transiciones. Los arcos conectan un lugar con una transición o viceversa, pero no a dos lugares o a dos transiciones. Los arcos tienen asociado un peso, es decir una etiqueta con un número entero positivo. Un arco con un peso w equivale a w arcos paralelos, cada uno con peso igual a 1.

Los lugares y las transiciones permiten anotaciones con las que describir estáticamente un sistema. Sin embargo, la asignación de un número entero no negativo a cada lugar, lo que se denomina un número de marcas, permite incluir el modelado de la parte dinámica de un sistema. Una marca se representa como un punto interior en el círculo correspondiente al lugar. Cada lugar de la PN puede contener cero o más marcas. Cada marca posee un valor de 1. El conjunto de marcas en un instante dado se llama un marcado de la PN.

El empleo de las PN para representar sistemas que son activados bajo condiciones explícitas se logra mediante la correspondencia entre los estados y los eventos del sistema con los elementos de la PN. Típicamente, los lugares se corresponden con estados o condiciones. Las transiciones representan los eventos que ocurren en el sistema modelado. Los lugares previos de una transición se corresponden con las pre-condiciones. Los lugares siguientes a una transición se corresponden con post-condiciones. De esta manera, las PN ofrecen una alternativa para la representación de patrones de eventos que suceden en un sistema.

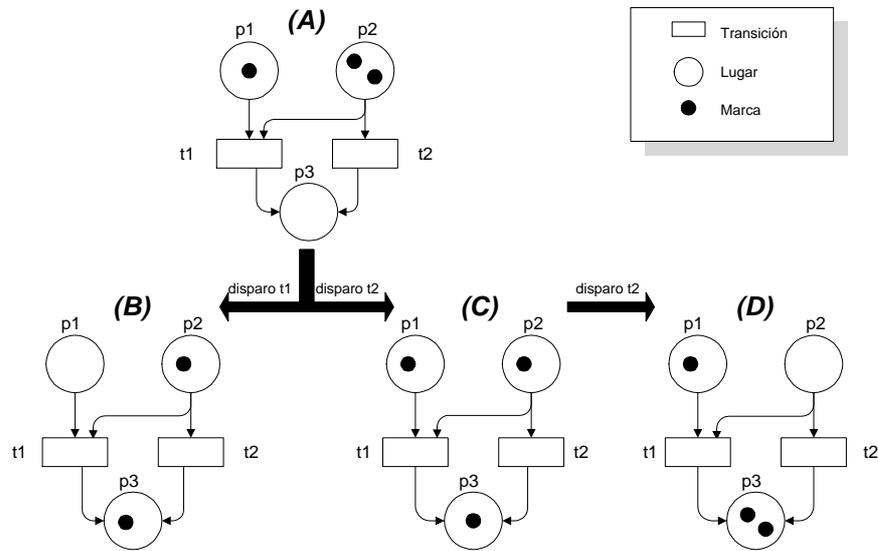


Figura1. El disparo de transiciones en Redes de Petri. A partir de la situación (A), el sistema puede evolucionar a la situación (B) o (C) según la transición que se dispara. A partir de la situación (C) se puede alcanzar la situación (D) mediante el disparo de la transición $t2$.

2.1. Disparo de las Transiciones en Redes de Petri

Uno de los rasgos particulares de las PN es que incluyen condiciones explícitas bajo las cuales se activa un evento, lo que permite representar esquemas complejos de control. Las transiciones se habilitan y se disparan bajo reglas concretas:

- Una transición t se dice que está habilitada bajo un marcado M si cada lugar previo p de t posee un número de marcas no menor al peso del arco de entrada a la transición a partir de dicho lugar p . Una transición debe estar habilitada para poder ser disparada (el disparo de una transición representa la ocurrencia de un evento en el sistema modelado).
- Si dos o más transiciones están habilitadas simultáneamente se dice que están en conflicto. La decisión respecto a cual de las transiciones en conflicto se dispara es no determinista.
- El disparo de una transición implica restar marcas a los lugares de entrada y sumar marcas a los lugares de salida. Se resta un número de marcas igual al peso del arco que conecta cada lugar previo con la transición. Se suma un número de marcas igual al peso del arco que conecta cada lugar de salida con la transición.
- Pueden existir transiciones con cero arcos de entrada (transición fuente) o con cero arcos de salida (transición sumidero). Por tanto, el número total de marcas de la red no es constante a lo largo de la ejecución.

La figura 1 muestra la evolución del marcado de una PN. A partir de un estado inicial (A) se tienen dos opciones de evolución del marcado, dado que ambas transiciones $t1$ y $t2$ están habilitadas (esto es, en conflicto). Si el conflicto se decanta por el

disparo de la transición $t1$, se obtiene el marcado (B) en el que no existen conflictos. Si alternativamente se resolviera el conflicto con el disparo de la transición $t2$, se obtiene el marcado (C) en el cual aún se mantiene el conflicto entre $t1$ y $t2$. El disparo de $t2$ permite evolucionar el sistema desde el marcado (C) hasta el (D).

2.2. Las Redes de Petri Coloreadas

Existen diferentes tipos de redes de Petri entre los cuales se encuentran las Redes de Petri Coloreadas o CPN [Jen97a]. Las CPN se caracterizan principalmente porque las marcas que se ubican en los lugares pueden tener un *tipo* determinado, lo que permite distinguir las marcas unas de otras. A cada tipo de datos se le denomina *conjunto de colores*. Al dato representado en una marca se le denomina *color*. Estos tipos o colores pueden combinarse para dar lugar a tipos compuestos o producto de tipos.

Las CPN poseen tres diferencias fundamentales con respecto a las PN clásicas: (a) los lugares de una CPN pueden contener elementos de un tipo determinado, (b) los arcos de una CPN poseen *expresiones de arco*, y (c) las transiciones poseen una característica adicional que se denomina *guarda*, ver figura 2. La existencia de tipos de datos en las CPN permite que cada lugar pueda contener uno o más valores correspondientes a un tipo básico o a un producto de tipos. Cada dato o color presente en un lugar lleva un indicador de su *multiplicidad*. Es decir, se puede determinar cuántas copias del dato se encuentran en un lugar. En la figura 2, el lugar $p1$ presenta 5 marcas, de las cuales una es el par (0, "cadena A") y los cuatro restantes tienen el valor (18, "cadena B").

Las características particulares de las CPN confieren un mayor potencial para el modelado de sistemas. Las expresiones de arco y las guardas de las transiciones permiten restringir mejor las condiciones de disparo de las transiciones en una CPN con respecto a las PN clásicas. El empleo de colores y de expresiones de arco hace que el modelo que se obtenga con CPN sea más compacto que el modelo equivalente con PN clásicas.

La mejora en la restricción para el disparo de transiciones de una CPN se expresa a través de las siguientes reglas:

- Una transición t en una CPN se dice que está habilitada si, además de tener el número de marcas mínimo según se estipula para las PN clásicas, los valores de esas marcas corresponden con las expresiones de arco de los arcos relacionados.
- Una transición t opcionalmente puede tener una condición de guarda. Una transición t que esté habilitada sólo se disparará si se satisface su condición de guarda.
- En las expresiones de arco y en las condiciones de guarda se pueden utilizar variables, que corresponden a alguno de los tipos válidos, o bien constantes (colores).
- El disparo de una transición coloca en los lugares de salida los valores correspondientes, en cuanto a color y multiplicidad, según la evaluación que se haga de las expresiones de los arcos de salida.

En la figura 2 se puede apreciar las restricciones que existen para el disparo de una transición en una CPN. La transición $t1$ tiene como lugares previos a $p1$ (del tipo

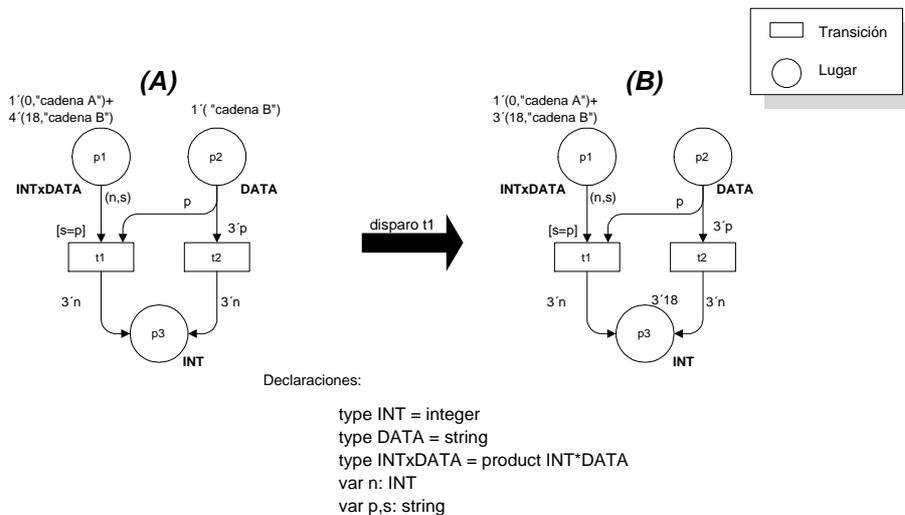


Figura2. El disparo de transiciones en redes de Petri coloreadas (CPN). El contenido de cada lugar está indicado al lado del mismo. A partir de la situación (A), el sistema puede evolucionar a la situación (B) si se dispara la transición $t1$.

INTxDATA) y $p2$ (del tipo DATA). El arco que va de $p1$ a $t1$ posee la expresión de arco (n,s) . El arco que va de $p2$ a $t1$ posee la expresión de arco p . Estas expresiones indican que el disparo de $t1$ extrae un par de la forma (n,s) desde $p1$ y extrae un valor p desde $p2$. La expresión $3'n$ del arco que va de $t1$ a $p3$ indica que el disparo agrega 3 veces el valor n en el lugar $p3$. La transición $t1$ requiere que $s = p$ (condición de guarda) para ser disparada.

2.3. Jerarquía de Redes de Petri

El modelado de sistemas complejos mediante redes de Petri se puede abordar mediante la composición de modelos más simples [Jen97a]. La composicionalidad de redes de Petri da origen al concepto de redes jerárquicas, cuyo proceso de construcción se puede realizar en dirección top-down, bottom-up, o mixta. La jerarquía de redes de Petri se puede lograr mediante dos tipos de construcciones: (a) transiciones de sustitución y (b) fusión de lugares.

Las *transiciones de sustitución* se han utilizado como mecanismo para dotar de modularidad a las descripciones de sistemas. Una transición de sustitución actúa de manera análoga al concepto de módulo de los lenguajes de programación y a la idea de descripción jerárquica de los diagramas de flujos de datos. La idea básica de las transiciones de sustitución es relacionar una transición y sus arcos correspondientes con una red que representa a la actividad correspondiente con mayor precisión y mediante una descripción más detallada.

La *fusión de lugares* permite especificar que un conjunto de lugares, llamado un *conjunto de fusión*, representa un único lugar conceptual aunque se hallen dibujados como lugares independientes entre sí. De esta manera al agregar o extraer una marca de uno de estos lugares en realidad se agrega o se extrae una marca idéntica de todos los restantes miembros del conjunto de fusión.

Las redes de Petri jerárquicas permiten tener diferentes niveles de descripción del sistema. En un nivel se puede describir al sistema global de manera simple y en niveles adicionales se proporcionan los detalles del comportamiento. En la figura 9 se muestra un ejemplo de una red de Petri jerárquica que emplea transiciones de sustitución y fusión de lugares. La transición de sustitución *Caso de Uso 1* de la situación (B) se desglosa en la situación (C) como una red de Petri que aporta mayor nivel de detalle. Los lugares *O* de las situaciones (C) y (E) representan un conjunto de fusión.

3. Descripción de Diagramas de Requisitos con Redes de Petri Coloreadas

En la sección 2 se ha mostrado que las CPN poseen características que las hacen útiles para el modelado de sistemas gobernados por reglas complejas. Como hemos visto, las CPN constituyen un formalismo para describir un sistema en su dimensión estática y dinámica. Existen tres características fundamentales de las CPN que las hacen apropiadas para el modelado de sistemas complejos:

- La inclusión de anotaciones en lugares, arcos y transiciones.
- La existencia de *colores* como marcas de una CPN.
- El mecanismo de composicionalidad o anidamiento que proporcionan las transiciones de sustitución.

Los diagramas de requisitos pretenden mejorar los aspectos de comunicación entre las personas involucradas en la elicitación y negociación de los requisitos. Por esa razón, tales diagramas potencian los aspectos gráficos e intuitivos aunque ello pueda implicar el menoscabo de la formalidad en la expresión de los requisitos [RP99]. Sin embargo, el carácter semi-formal limita las posibilidades de investigar sistemáticamente las propiedades de los diagramas de requisitos, lo cual es crítico para la reutilización en entornos reales. En consecuencia, se debe buscar la forma de traducir los modelos de requisitos a una expresión formal para definir una estrategia de reutilización de requisitos que permita aumentar la calidad y reducir el tiempo en el desarrollo de especificaciones.

En esta sección se aborda la descripción de diagramas de requisitos (escenarios, casos de uso, workflows, diagramas de actividad) con el formalismo de las CPN. Dos razones centrales nos impulsan a proponer el uso de las CPN como técnica de descripción de requisitos. Una, los requisitos del software actúan como un patrón complejo de reglas que refleja aspectos estáticos y dinámicos del sistema [Poh96]. Ambos aspectos pueden modelarse adecuadamente con CPN. Dos, las redes de Petri coloreadas ofrecen un sólido soporte formal y se han utilizado con éxito en el modelado de diferentes productos del ciclo de vida del software [Jen97c].

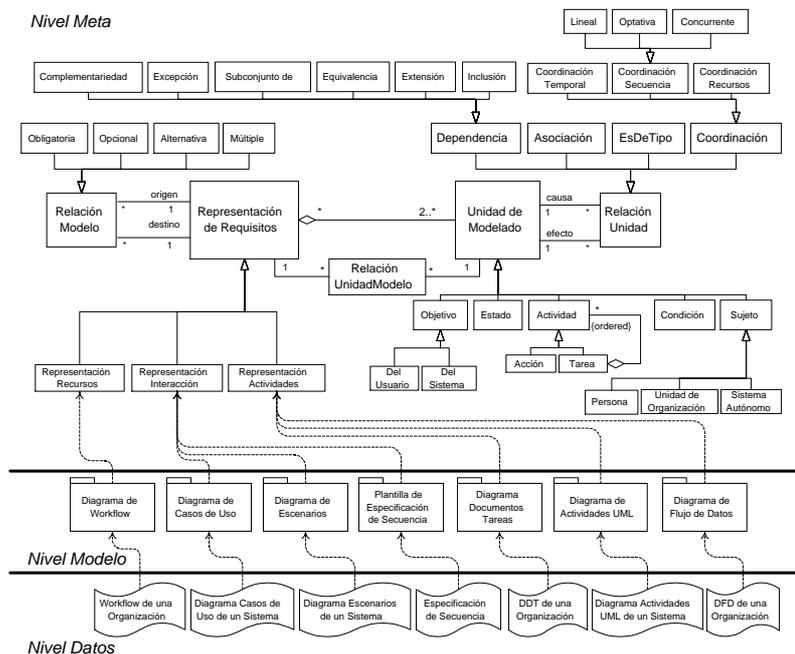


Figura3. Metamodelo de Representaciones de Requisitos del Software, expresado en el lenguaje MOF. Los requisitos se representan mediante Unidades de Modelado que se relacionan entre sí y con Representaciones de Requisitos que también se relacionan entre sí.

3.1. El Metamodelo de Diagramas de Requisitos

En la figura 3 se propone un metamodelo de requisitos que describe diagramas de requisitos que representan el *Universo del Discurso* en el contexto de las organizaciones y los sistemas cuando se instancian en el Nivel Datos de dicha figura. Los elementos centrales de este metamodelo son la *Representación de Requisitos*, utilizada para describir los diferentes diagramas, y la *Unidad de Modelado*, utilizada para describir las unidades contenidas en los diagramas. Las Representaciones de Requisitos consideradas en nuestro estudio se clasifican en tres categorías mientras que las Unidades de Modelado, las relaciones entre Unidades de Modelado y las relaciones entre Representaciones de Requisitos, en seis, cuatro y cuatro categorías, respectivamente.

Las relaciones entre Unidades de Modelado se describen en el metamodelo como elementos de la clase *Relación Unidad*. Estas relaciones representan las formas en que se enlazan los elementos dentro de cada diagrama de requisitos. El metamodelo debe contemplar estas relaciones entre Unidades de Modelado para poder describir el contenido de los diferentes diagramas y así integrarlos en una estrategia de reutilización.

Todas las relaciones entre Unidades de Modelado poseen un carácter de *causalidad* que establece que unos elementos actúan como causa y otros como efecto. Esta carac-

terización de causalidad se instancia en los diagramas de requisitos de diversas formas, como por ejemplo «requiere», «produce», «inicia», «ejecuta», «realiza», etc. Gráficamente, la causalidad se distingue en los modelos de requisitos porque el sentido de la asociación apunta hacia el elemento que actúa como efecto.

Las relaciones entre Representaciones de Requisitos se describen en el metamodelo como elementos de la clase *Relación Modelo*. Estas relaciones son la base para integrar diferentes niveles de descripción, Requisitos-C y Requisitos-D [Rom90], dentro de una estrategia de reutilización de requisitos. Por tanto, estas relaciones hacen referencia a la forma en que se combinan los diagramas de requisitos. La Relación Modelo permite enlazar Requisitos-C, por ejemplo un Diagrama de Casos de Uso o un Diagrama de Actividades, con Requisitos-D, por ejemplo un Diagrama de Clases o un Diagrama de Secuencia.

La Relación Modelo tiene un carácter estructural que determina el grado de asociación entre dos o más modelos enlazados. Una Relación Modelo puede ser de tipo obligatorio, opcional, alternativo o múltiple.

La complejidad de algunas unidades de modelado puede motivar la necesidad de representar su especificación en una representación completa de requisitos. Por ejemplo, un proceso dentro de un Diagrama de Flujo de Datos puede ser explotado en otro Diagrama de Flujo de Datos, o un caso de uso puede ser expresado mediante una Plantilla de Especificación de Secuencia. Este tipo de relaciones se describen en el metamodelo como Relación Unidad Modelo.

3.2. Correspondencia entre el Metamodelo de Requisitos y las CPN

Para aprovechar el soporte formal de las CPN en el modelado de requisitos se debe establecer la correspondencia entre las metaclases que conforman el metamodelo de requisitos y los elementos de una CPN. Nuestra propuesta es que dicha correspondencia se establezca en los siguientes términos (ver figura 4):

1. Unidades de Modelado:
 - a) Estado, objetivo, condición y sujeto se corresponden con marcas de CPN.
 - b) Tarea y acción se corresponden con transiciones de CPN.
2. Las Representaciones de Requisitos se corresponden con transiciones de CPN.
3. Relaciones:
 - a) Relaciones entre Unidades de Modelado:
 - 1) Dependencia y Asociación se corresponden con un conjunto de elementos de CPN formado por arcos, lugares, marcas y expresiones de arco.
 - 2) Relación EsDeTipo se ignora porque es sólo descriptiva dentro de los modelos de requisitos estudiados.
 - 3) Relaciones de Coordinación se corresponden con modelos de CPN. Existe la restricción de que las relaciones de coordinación sólo se establecen entre unidades de modelado que son representadas como transiciones de CPN.

Traducción de Relaciones de Coordinación. En la figura 5.A se muestra la representación en CPN de las relaciones de Coordinación entre unidades de modelado. La secuencia *lineal, optativa y concurrente*, así como las *coordinación de recursos* y la *coordinación temporal*, se restringen a aquellas unidades de modelado representadas como transiciones. De esta manera, los diferentes modelos de CPN mostrados en la figura 5.A representan relaciones de coordinación entre transiciones, de acuerdo con la regla 3a3.

Las relaciones de coordinación se traducen en una transición de coordinación, un lugar coordinador (LC), un lugar de fusión (en líneas punteadas) y los arcos y expresiones de arco correspondientes. La transición de coordinación se denomina Secuencia Optativa, Secuencia Concurrente, Secuencia Lineal, Coordinación Temporal o Coordinación Recursos según el tipo de coordinación entre las transiciones denotadas por ActA, ActB y ActN en la figura 5.A. El lugar de fusión se denota por O, C, L, T o R dependiendo del tipo de relación de coordinación. El disparo de la transición de coordinación produce una lista de las actividades bajo esa relación. Esta lista se deposita en el lugar coordinador para que se activen las transiciones relacionadas dependiendo de si el nombre de la actividad está incluido en la lista de actividades.

La lista de actividades también se envía, junto con el nombre de la página de CPN y el tipo de relación de coordinación, al lugar de fusión correspondiente. Esta información será utilizada dentro del entorno de reutilización de requisitos para efectos de organización de los elementos reutilizables.

Algunas estructuras de CPN para las relaciones de coordinación son similares mientras que otras son diferentes. Las secuencias optativa y concurrente, así como la coordinación de recursos, presentan una estructura similar de CPN que sólo se diferencia en cuanto al lugar de fusión. La secuencia lineal y la coordinación de recursos presentan estructuras particulares. La secuencia lineal obliga a que la ActA sea la primera en ejecutarse, y luego la ActB hasta la ActN. La coordinación de recursos incluye un lugar para el recurso compartido.

Traducción de Relaciones de Asociación y Dependencia. Las relaciones de Asociación y Dependencia se representan en la figura 5.B como arcos, lugares, marcas y expresiones de arco tal como se establece en la regla 3a1. Estos tipos de relaciones se pueden manifestar entre diferentes combinaciones de transiciones y lugares. Una representación de requisitos (regla 3a1), así como las unidades de modelado de tipo Tarea y Acción se representan como transiciones de sustitución (regla 1b). Mientras, las unidades de modelado de los tipos Objetivo, Sujeto, Condición y Estado se representan mediante colores o constantes (es decir, marcas de un tipo de color específico), según la regla 1a. Por tanto, para aquellos elementos participantes en una relación de Dependencia o Asociación, y que se corresponden con un lugar o con una transición, la relación correspondiente se representa mediante las combinaciones de arcos y colores mostrados en la figura 5.B. Según la correspondencia del metamodelo con elementos CPN las relaciones de Dependencia, Asociación no se presentan entre lugares, sin embargo por

razones de ortogonalidad en la figura se muestra la forma de modelar esas relaciones entre lugares.

Las relaciones de Asociación y Dependencia también utilizan un lugar de fusión para depositar la información necesaria para el entorno de reutilización. El lugar de fusión se etiqueta con A o D según sea la relación de asociación o dependencia. Una de las transiciones de CPN participantes en la relación se enlaza al lugar de fusión mediante un arco cuya expresión consiste en una tripleta con el tipo de relación, y las etiquetas de los dos elementos enlazados.

Traducción de la Relación Modelo. En la figura 5.C se muestra cómo representar las relaciones entre Representaciones de Requisitos. Estas representaciones se corresponden con transiciones de CPN por lo que las relaciones entre modelos de requisitos se manifiestan como arcos, lugares, marcas y expresiones de arco, según las reglas 2 y 3c.

La relación entre una Representación de Requisitos D1 y un conjunto de representaciones D2, D3,...,DN se traduce en un lugar de relación (LR) al cual se enlazan cada una de las transiciones D2, D3,...,DN. La transición D1 se enlaza al lugar LR a un lugar de fusión denotado por M (en líneas punteadas). El lugar LR recibe la lista de las transiciones relacionadas con D1. El lugar de fusión M recibe el tipo de relación modelo (obligatoria, alternativa, opcional o múltiple) así como la lista de transiciones enlazadas mediante esa relación. Esta información será utilizada dentro del entorno de reutilización de requisitos para efectos de organización de los elementos reutilizables.

Traducción de la Relación Unidad Modelo. En la figura 5 no se muestra como expresar la relación Unidad Modelo mediante CPN. Esto se debe a que este tipo de relación sólo se establece entre transiciones y confiere el carácter de transición de sustitución a la unidad de modelado. Es decir que esta relación establece que una transición adquiere el carácter de transición de sustitución. El contenido de esta transición de sustitución está determinado por los restantes elementos del metamodelo que se traducen en estructuras CPN según lo establecido en la presente subsección.

3.4. Algoritmo para Traducir Diagramas de Requisitos a CPN

En la sección 3.2 se ha establecido la correspondencia entre los elementos del metamodelo de requisitos y las CPN. Para obtener la proyección de los distintos diagramas de requisitos descritos según el metamodelo, se deben aplicar las reglas de correspondencia en un orden determinado. En esta sección presentamos un algoritmo para traducir un diagrama de requisitos (como un Diagrama de Casos de Uso, de Workflow, o de Actividades) en un modelo de CPN siguiendo una dirección top-down.

1. Cada representación de requisitos se hace corresponder con una transición de sustitución.

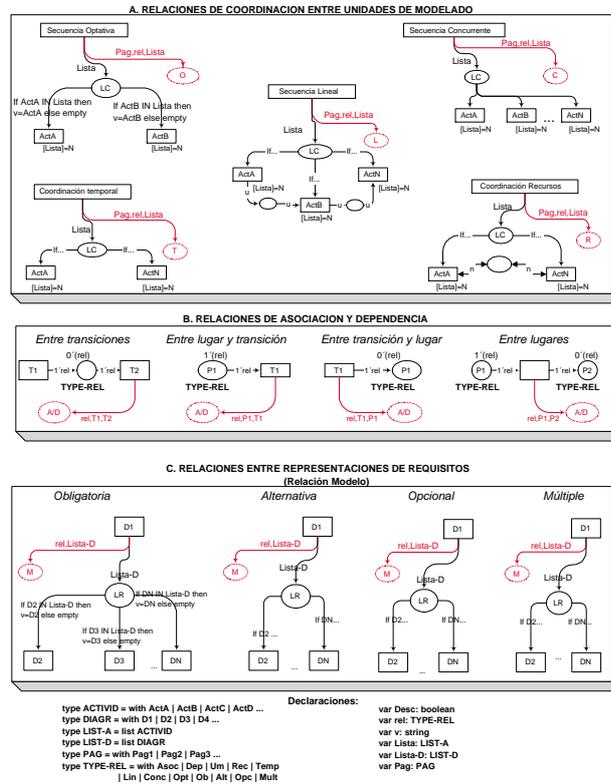


Figura5. Representación de elementos del metamodelo de requisitos mediante estructuras compuestas de CPN. Las relaciones de coordinación constituyen el detalle de una transición de sustitución. Las elipses de líneas punteadas representan lugares de fusión.

2. Representar las relaciones entre las representaciones de requisitos (instancias de Relación Modelo), siguiendo la regla 3c. (Comentario: Las relaciones de tipo obligatoria, alternativa, opcional y múltiple se representan como modelos de CPN, según se muestra gráficamente en la figura 5.C).
3. Mientras haya transiciones de sustitución representar el contenido de cada una de ellas. Las relaciones se representan como sigue:
 - a) Representar relaciones de dependencia y coordinación entre instancias de las metaclasses Tarea y Acción.
 - b) Representar relaciones de Asociación y Dependencia en unidades de modelado de tipo Estado, Objetivo, Condición y Sujeto.
 - 1) El rol de Causa en Estados, Objetivos, Condiciones y Sujetos se representa como marcas en los lugares previos correspondientes y expresiones de arco.
 - 2) El rol de Efecto en Estados, Objetivos, Condiciones y Sujetos se representa como marcas en los lugares posteriores correspondientes y expresiones de arco.

La aplicación de este algoritmo permite obtener una expresión formalizada de los diagramas de requisitos descritos mediante el metamodelo. Esta expresión formalizada con redes de Petri coloreadas permite el tratamiento analítico de los requisitos con el soporte de las técnicas de análisis de redes de Petri y las herramientas disponibles. Además, con base en la descripción de requisitos de sistema mediante redes de Petri, propondremos un esquema de organización de requisitos para reutilización. Este esquema de organización será abordado como parte de nuestro trabajo futuro.

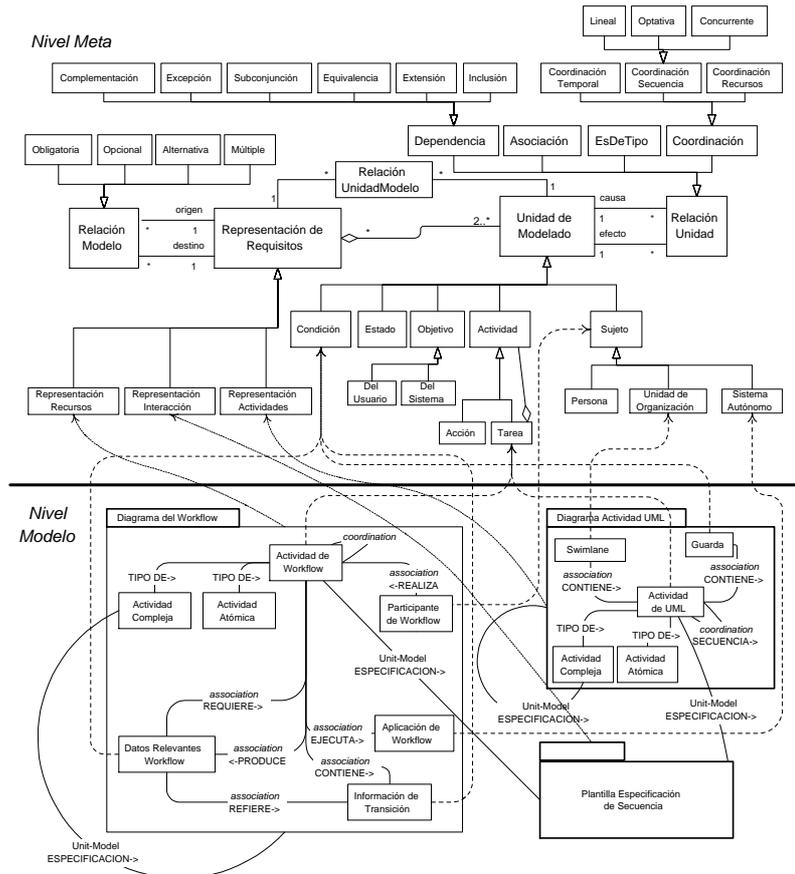


Figura6. Representación de Workflows y Diagrama de Actividades bajo el metamodelo de diagramas de requisitos.

3.5. Representación de Workflows y Diagramas de Actividades

En la figura 6 se muestra la forma en que se representa el Diagrama de Workflow [WfM] y el Diagrama de Actividades [JBR99] según el metamodelo de requisitos. El

paquete del Workflow (instancia de la metaclassa Representación Recursos) está formado por elementos de la clase Actividad de Workflow que se enlazan mediante relaciones de Coordinación. Una Actividad de Workflow puede ser de tipo Complejo o Atómico y estar especificada mediante una Plantilla de Especificación de Secuencia. El tipo complejo se puede explotar mediante otro Diagrama de Workflow. Una Actividad de Workflow posee relaciones de Asociación con elementos de la clase Participante del Workflow, Aplicación de Workflow, Información de Transición y Datos Relevantes del Workflow. Los elementos de la clase Información de Transición se asocian con elementos de la clase Datos Relevantes del Workflow. Por su parte, el Diagrama de Actividades está formado también por instancias de las metaclases Tarea, Condición y Unidad de Organización e incluye relaciones de tipo Asociación y Coordinación.

En la figura 7 se muestra gráficamente la traducción a CPN del Diagrama de Workflow (situación A1) y el Diagrama de Actividades (situación A2). Ambos diagramas se representan como una transición de sustitución que, aunque no se muestra en la figura, deben estar dentro de una CPN que permita relacionarlos con otras representaciones de requisitos, tal y como estipulan los pasos 1 y 2 del algoritmo de la sección 3.4.

El contenido del Diagrama de Workflow está dominado por instancias de la metaclassa Tarea que presentan relaciones de Coordinación. No existen relaciones de Dependencia entre instancias de la metaclassa Tarea. Las relaciones de Coordinación entre actividades del workflow permite obtener una red de Petri como la que se propone en la situación B1 de la figura 7 y como resultado de la aplicación del paso 3. Esta situación B1 muestra actividades de workflow en secuencia optativa y de coordinación de recursos. Las actividades complejas (las que son denominadas *procesos de workflow*) se desglosan según su contenido de (sub)actividades y relaciones de coordinación. Por ejemplo en la situación C1 se describe una actividad compleja de workflow con tres (sub)actividades optativas. Las (sub)actividades atómicas se pueden considerar formadas por etapas que se relacionan entre sí mediante coordinación y por tanto se expresan como una transición de sustitución. En la situación D1 se muestra una actividad atómica de workflow formada por tres etapas secuenciales. Si cada una de estas etapas se puede especificar como una acción y una excepción entonces la situación D1 toma la forma de una Especificación de Secuencia.

La traducción del Diagrama de Actividades se logra aplicando el paso 3a para pasar de la situación A2 a la B2 de la figura 7. Esta situación representa un diagrama de actividades con tres actividades concurrentes. Estas actividades pueden ser complejas o atómicas por lo que se desglosan según corresponda de manera idéntica a lo explicado para el Diagrama de Workflow.

La aplicación del paso 3b del algoritmo de la sección 3.4 en la traducción del Diagrama de Workflow y de Actividades permite situar la información correspondiente a las relaciones de asociación y dependencia entre unidades de modelado, tal como se muestra en las situaciones B1, C1, D1 y B2.

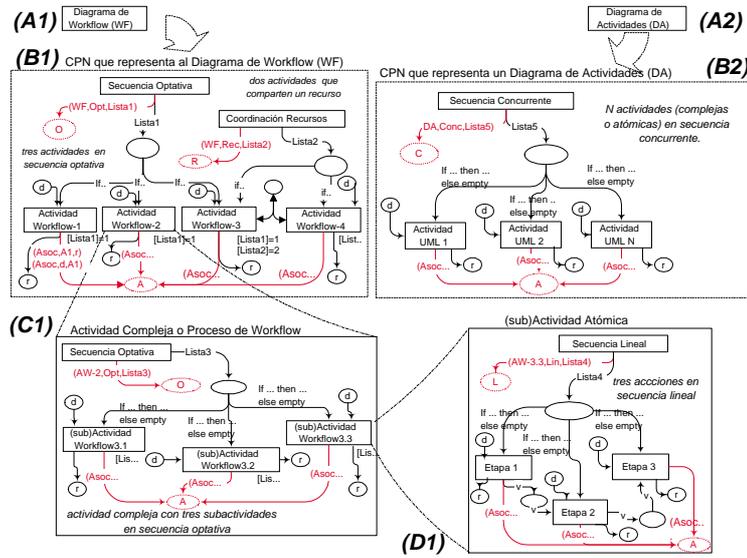


Figura7. Traducción de Workflows y Diagramas de Actividad a modelos de CPN.

3.6. Traducción del Diagrama de Casos de Uso

En la figura 8 se muestra la forma en que se representa el Diagrama de Casos de Uso según el metamodelo de requisitos. El Diagrama de Casos de Uso se representa como un paquete que instancia a la metaclass Representación Interacción. Dentro de este paquete se instancia la metaclass Tarea como Caso de Uso. Diferentes subtipos de unidades de modelado están relacionados al caso de uso mediante instancias que reflejan un carácter de causalidad. Además, un Caso de Uso muestra relaciones de Dependencia con otro Caso de Uso, estas relaciones son del tipo Inclusión o Extensión. Un Caso de Uso se especifica mediante actividades de satisfacción y de excepción. Estas actividades se expresan mediante la Plantilla de Especificación de Secuencia que es una instancia de Representación Interacción. De acuerdo al metamodelo, la relación entre un Caso de Uso y la Plantilla es una instancia de Relación Unidad Modelo.

En la figura 9 se muestra un esquema de la traducción de un Diagrama de Casos de Uso a un modelo de CPN. El diagrama de la situación A se traduce en una transición de sustitución que puede formar parte de una CPN para relacionar el diagrama con otras representaciones de requisitos tal y como estipulan los pasos 1 y 2 del algoritmo de la sección 3.4. Cada caso de uso se corresponde con una transición y al aplicar los pasos 3a y 3b, dado que las relaciones *incluye* y *extiende* son relaciones de Dependencia entre instancias de la metaclass Tarea, se obtiene un modelo CPN como el de la situación B. Un caso de uso se representa con una transición con un lugar previo y uno posterior. En el lugar previo se tienen marcas que representan Sujeto, Precondición y Evento disparador. Las marcas del lugar posterior representan Meta, Post-condición y Resultado.

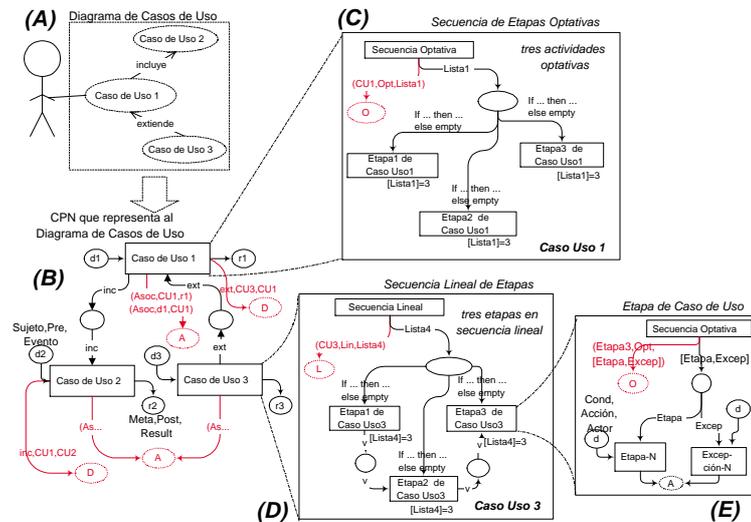


Figura9. Representación del Diagrama de Casos de Uso mediante elementos de CPN.

casos de uso es su bajo nivel de formalidad [LCK98]. Por tanto, se justifica la búsqueda de soluciones a la escasa formalidad de los casos de uso. Siguiendo el algoritmo propuesto en la sección 3.4, se puede obtener una expresión de un Diagrama de Casos de Uso mediante el formalismo de las redes de Petri coloreadas, según se muestra en la figura 9. Esencialmente un caso de uso se representa con una transición de sustitución con un lugar previo y uno posterior. En el interior de la transición de sustitución se encuentra la secuencia optativa entre transiciones que representan la Descripción de Satisfacción y Excepción. Cada una de estas transiciones internas contiene a su vez una secuencia lineal de transiciones (etapas).

3.7. Traducción del Diagrama de Escenarios

El Diagrama de Escenarios se representa según el metamodelo de requisitos tal como se muestra en la figura 10. Este diagrama se representa como un paquete que instancia a la metaclassa Representación Interacción. Dentro de este paquete se instancia la metaclassa Tarea como Escenario. Otros subtípos de unidades de modelado están enlazados con el Escenario mediante instancias de relaciones que reflejan un carácter de causalidad. Además, un Escenario presenta relaciones de Dependencia y de Coordinación con otro Escenario. Un Escenario se especifica mediante la Plantilla de Especificación de Secuencia (instancia de Representación Interacción). La relación entre un Escenario y la Plantilla es una instancia de Relación Unidad Modelo.

La figura 11 muestra un esquema de la traducción de un Diagrama de Escenarios a un modelo de CPN. El diagrama de la situación A se traduce en una transición de sustitución dentro de una CPN que permita relacionar el Diagrama de Escenarios de

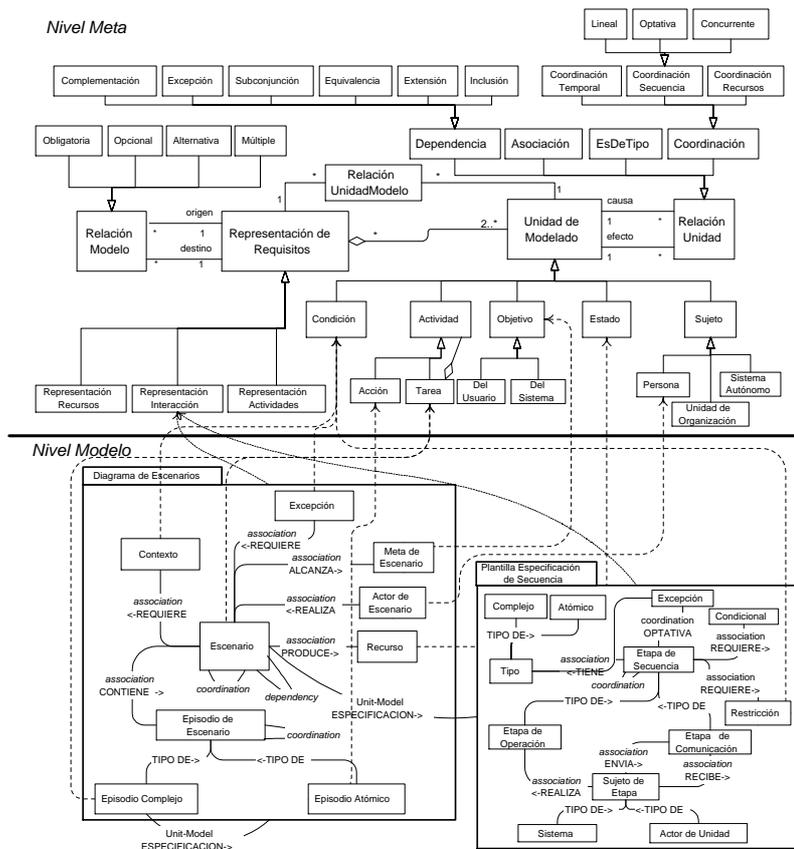


Figura10. Representación del Diagrama de Escenarios según el metamodelo de diagramas de requisitos.

manera opcional, obligatoria, alternativa o múltiple con otras representaciones de requisitos según lo estipulado en los pasos 1 y 2 del algoritmo de la sección 3.4. Cada Escenario se corresponde con una transición y al aplicar los pasos 3a y 3b, dado que las relaciones *complementa* y *extiende* son relaciones de Dependencia entre instancias de la metaclass Tarea, se obtiene el modelo CPN de la situación B. Un escenario se representa con una transición de sustitución con un lugar previo y uno posterior. En el lugar previo se tienen marcas que representan Contexto, Excepción y Actor. Las marcas del lugar posterior representan Meta, y Recurso.

Cada transición de la situación B representa un escenario que está formado por episodios que guardan relaciones de coordinación entre ellos. Por ejemplo, asumiendo que un escenario está formado por tres episodios optativos, se desglosa cada escenario como se muestra en la situación C. Un episodio complejo se puede explotar según el modelo de CPN de la situación D que muestra una secuencia de (sub)episodios. Cada

episodio atómico está formado por una secuencia optativa entre una etapa y una excepción. La situación E muestra la secuencia optativa de un episodio atómico.

Los escenarios constituyen una técnica de modelado ampliamente utilizada para la captura de requisitos [SMMM98]. Sus fortalezas se pueden complementar agregando un mayor nivel de formalidad a las descripciones de sistemas mediante escenarios. Siguiendo el algoritmo propuesto en la sección 3.4, se puede obtener una expresión de un Diagrama de Escenarios mediante el formalismo de las redes de Petri coloreadas, según se muestra en la figura 11. Esencialmente un escenario se representa con una transición de sustitución con un lugar previo y uno posterior. En el interior de la transición de sustitución se encuentra la secuencia optativa entre transiciones que representan los episodios. Cada una de estas transiciones internas puede contener a su vez una secuencia de transiciones (subepisodios).

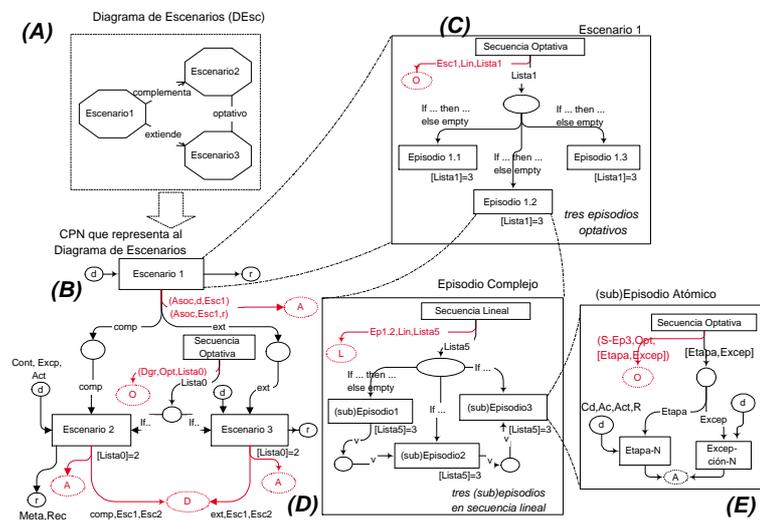


Figura 11. Representación del Diagrama de Escenarios mediante elementos de CPN.

4. Trabajos Relacionados

En nuestro trabajo previo [LL01] hemos comprobado que escenarios, casos de uso y workflows son técnicas complementarias de documentación de requisitos. Los workflows proporcionan una descripción de mayor nivel de abstracción para los procesos gestionados en una organización. Por su parte, los escenarios y casos de uso permiten describir con mayor detalle las actividades que componen los procesos del workflow. Por tanto, la combinación de esas técnicas de modelado permite obtener una descrip-

ción detallada de la gestión de procesos y actividades de una organización.

Escenarios, casos de uso y workflows son técnicas intuitivas que facilitan la comunicación entre los *stakeholders*. Sin embargo, estas técnicas carecen de sintaxis y semántica claras que prevengan la ambigüedad. Existen varios intentos por formalizar los diagramas de requisitos mencionados. En [LCK98] se propone una extensión de las PN clásicas para formalizar los escenarios y casos de uso. Las PN clásicas también se han propuesto como soporte formal de los workflows [dA97]. Estos trabajos muestran el potencial de las redes de Petri para el soporte de la verificación del comportamiento lógico de sistemas que presentan concurrencia, paralelismo, sincronización, no-determinismo y exclusión mutua. Nosotros hemos utilizado PN clásicas para soportar la obtención de escenarios a partir de workflows [LLM00]. Sin embargo, a pesar del sólido soporte formal ofrecido por las PN clásicas, la necesidad de contemplar diferentes tipos de secuenciación y de relaciones semánticas y estructurales entre los elementos de modelado hace que el modelo del sistema que se obtiene con el uso de PN clásicas sea poco compacto.

Las redes de Petri coloreadas, empleadas para modelar diferentes tipos de sistemas [Jen97a,Jen97c], presentan las mismas ventajas de las PN clásicas (carácter gráfico, soporte formal y la existencia de herramientas automatizadas para su análisis). Sin embargo, como hemos visto, las CPN permiten obtener modelos más compactos debido a que proporcionan un mayor potencial de abstracción que las PN clásicas.

Además de las redes de Petri se han empleado otras técnicas para solventar la falta de formalidad de los diagramas de requisitos del sistema. En [Gli95,RG99] se propone el uso de diagramas de estados, en [BPP99] se ofrece el cálculo de contratos, en [HSG⁺94] se propone una gramática de escenarios. Lamentablemente, para estas propuestas no existe la misma disponibilidad de herramientas y técnicas de análisis que existe para las redes de Petri.

5. Conclusiones y Trabajo Siguiete

La labor desarrollada en este informe técnico se enmarca dentro de una línea de trabajo para la definición de una estrategia de reutilización de requisitos. Se ha presentado la correspondencia entre un metamodelo de requisitos y los elementos de las redes de Petri coloreadas. El metamodelo proporciona una descripción de alto nivel de abstracción para los diagramas de requisitos del sistema. Las redes de Petri coloreadas permiten modelar de forma compacta los aspectos estáticos y dinámicos del sistema y con un sólido soporte formal.

La descripción de modelos de requisitos mediante redes de Petri coloreadas proporciona un mecanismo para tratar analíticamente los diagramas de requisitos en el contexto de la reutilización del software. La correspondencia planteada entre las diferentes metaclases y los elementos de redes de Petri coloreadas permiten que los diversos

diagramas descritos con el metamodelo se puedan traducir en su conjunto a un modelo red de Petri. De esta manera se consigue una descripción rigurosa de distintas técnicas de modelado de requisitos.

Nuestra intención no es proponer una técnica sustituta de los diferentes diagramas de requisitos estudiados sino obtener una descripción formalizada de tales diagramas con el propósito de integrarlos en una estrategia de reutilización de requisitos. Nuestro trabajo siguiente consiste en establecer un proceso asistido para la descripción de requisitos con modelos CPN y la correspondiente extracción de información reutilizable.

Agradecimientos

Este trabajo se desarrolla dentro del Grupo de Investigación en Reutilización y Orientación al Objeto (GIRO) de la Universidad de Valladolid y es parcialmente financiado por el proyecto DOLMEN (CICYT-TIC2000-1673-C06-05).

Oscar López agradece a la Agencia Española de Cooperación Internacional (AECI) y al Ministerio de Ciencia y Tecnología de Costa Rica.

Referencias

- [BPP99] R. Back, L. Petre, and I. P. Paltor. Formalizing UML use cases in the refinement calculus. Technical Report TUCS-279, Turku Centre for Computer Science, Department of Computer Science, Lemminkäisenkatu 14, FIN-20520 Turku, Finland, May 1999.
- [CSPD92] Richard E. Creps, Mark A. Simos, and Rubén Prieto-Díaz. The STARS conceptual framework for reuse processes. In *Proceedings of STARS'92*, November 1992.
- [dA97] V. d. Aalst. The Application of Petri Nets to Workflow Management. Eindhoven, University of Technology, Netherland, URL: www.wis.win.tue.nl/~wsinwa/jcsc/jcsc.html, 1997.
- [Gli95] M. Glinz. An integrated formal model of scenarios based on statecharts. In W. Schäfer and P. Botella, editors, *Proceedings of the Fifth European Software Engineering Conference*, pages 254–271. Springer-Verlag, 1995.
- [HSG⁺94] Pei Hsia, Jayarajan Samuel, Jerry Gao, David Kung, Yasufumi Toyoshima, and Cris Chen. Formal approach to scenario analysis. *IEEE Software*, 11(2):33–41, March 1994.
- [JBR99] Ivar Jacobson, Grady Booch, and James Rumbaugh. *The Unified Software Development Process*. Object Technology Series. Addison-Wesley, 1999.
- [Jen97a] Kurt Jensen. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use, Volume 1*. Springer-Verlag, Berlin, Germany, 2nd. edition, 1997.
- [Jen97b] Kurt Jensen. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use, Volume 2*. Springer-Verlag, Berlin, Germany, 2nd. edition, 1997.
- [Jen97c] Kurt Jensen. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use, Volume 3*. Springer-Verlag, Berlin, Germany, 2nd. edition, 1997.
- [Kar95] Even-André Karlsson, editor. *Software Reuse. A Holistic Approach*. Wiley Series in Software Based Systems. John Wiley and Sons Ltd, 1995.
- [LCK98] Woo Jin Lee, Sung Deok Cha, and Yong Rae Kwon. Integration and analysis of use cases using modular Petri nets in requirements engineering. *IEEE Transactions on Software Engineering*, 24(12):1115–1130, December 1998. Special Issue: Scenario Management.

- [LL01] O. López and M. A. Laguna. Requirements reuse for software development. In *Proceedings of the RE'01 Doctoral Workshop. Fifth IEEE International Symposium on Requirements Engineering*, Toronto, Canada, August 2001.
- [LLG02a] O. López, M.A. Laguna, and F.J. García. Requirements modeling for reuse. In *Actas de Las II Jornadas de Trabajo DOLMEN*, pages 105–114, Valencia, España, Marzo 2002.
- [LLG02b] O. López, M.A. Laguna, and F.J. García. Reuse based requirements clustering. In *Actas de Las II Jornadas de Trabajo DOLMEN*, pages 129–138, Valencia, España, Marzo 2002.
- [LLM00] Óscar López, Miguel Ángel Laguna, and José Manuel Marqués. Generación automática de casos de uso para desarrollo de software con reutilización. In *Actas de Las V Jornadas de Ingeniería Del Software y Bases de Datos (JISBD'2000)*, pages 89–101, Spain, Noviembre 2000.
- [PD89] Rubén Prieto-Díaz. Classification of reusable modules. *Software Reusability, Vol 1. Concepts and Models*, 1:99–123, 1989.
- [Poh96] K. Pohl. Requirements engineering, an overview. *Encyclopedia of Computer Science and Technology*, Vol. 36, Marcel Deccer Inc., 1996.
- [RG99] J. Ryser and M. Glinz. A practical approach to validating and testing software systems using scenarios, 1999.
- [Rom90] H. Dieter Rombach. Software specifications: A framework. SEI curriculum module. Technical Report SEI-CM-11-2.1, Software Engineering Institute, Carnegie Mellon University, January 1990.
- [RP99] Colette Rolland and Naveen Prakash. From conceptual modelling to requirements engineering. Technical Report Series 99-11, CREWS, 1999.
- [Sil85] Manuel Silva. *Las Redes de Petri en la Automática y la Informática*. Editorial AC, Madrid, España, 1985.
- [SMMM98] A. Sutcliffe, N. Maiden, S. Minocha, and D. Manuel. Supporting scenario-based requirements engineering. *IEEE Transactions on Software Engineering*, 24(12), December 1998.
- [WfM] WfMC. The workflow management coalition. Terminology and Glossary. Document Number WFMC-TC-1011. United Kingdom, 65 pages. feb 99.