

# NORMALIZACIÓN DE ASSETS DE REQUISITOS EN EL CONTEXTO DE LA REUTILIZACIÓN SISTEMÁTICA DEL SOFTWARE\*

Oscar López\*\*  
Instituto Tecnológico de Costa Rica  
olopez@infor.uva.es

Miguel Ángel Laguna  
Universidad de Valladolid  
mlaguna@infor.uva.es

José Manuel Marqués  
Universidad de Valladolid  
jmmc@infor.uva.es

## RESUMEN

*La definición precisa de los requisitos funcionales del sistema juega un papel relevante en los entornos de desarrollo de software con reutilización. El desarrollo mediante elementos reutilizables supone la selección adecuada a partir de un amplio conjunto de estos con lo cual el punto de partida es la funcionalidad del producto que será desarrollado. Por lo anterior se propone un marco de trabajo para la normalización de la captura de requisitos y su inmediata expresión en forma de assets de análisis aptos para la conformación de mecanos en el contexto de la reutilización sistemática. La normalización de los requisitos se realiza mediante un proceso que involucra el uso de workflows y redes de Petri para tratar analíticamente la funcionalidad del sistema expresada inicialmente en términos de casos de uso del negocio (BUC) y de casos de uso (CU) y que luego son traducidos al formato de alguna plantilla particular.*

**PALABRAS CLAVE:** Ingeniería de requisitos, casos de uso, reutilización del software, workflows, redes de Petri.

## 1 INTRODUCCIÓN

Los beneficios de una definición temprana y precisa de los requisitos del sistema han sido ampliamente enfatizados en el contexto del desarrollo de un producto software. Numerosos autores han señalado que la ingeniería de requisitos juega un papel estratégico a la hora de desarrollar software lo cual es más relevante en los entornos de desarrollo mediante la reutilización sistemática de elementos software existentes. Por un lado, el desarrollado con reutilización requiere la selección adecuada de los elementos dentro del previsiblemente amplio conjunto de ellos almacenado en un repositorio. Y por otro lado, la reutilización supone una mejora sustancial en la productividad de sistemas software. De manera que la precisa especificación de los requisitos del software es un elemento clave para aprovechar las ventajas de la reutilización [10][12].

Los requisitos software son el producto final de un proceso que se inicia con la fase de elicitación y que continúa con el análisis de requisitos del software. Este análisis de requisitos trata con dominios específicos organizacionales para indicar en cada dominio cuáles son las características que debe cumplir un producto software en términos de lograr un uso efectivo de la información acorde con los objetivos generales de la organización. En este sentido se tienen dos aspectos relevantes. Por un lado que el análisis de requisitos debe conducir a un documento de especificación de requisitos del software. Y por otro lado, que dichos requisitos deben atender tanto al nivel de cada dominio como al nivel más general de la organización

El documento de especificación define los atributos deseables del software a desarrollar. Dentro de esos requisitos se encuentra la funcionalidad, es decir los requisitos funcionales, que especifica los detalles más relevantes de las funciones que el producto realizará, con independencia de los requisitos no funcionales y de las restricciones de diseño [5]. De acuerdo con Presman [10], la funcionalidad debe cubrir la representación y la comprensión del ambiente específico y debe contener todas las funciones esenciales (funcionalidad delimitada) que el software deberá realizar. En concreto, los requisitos funcionales se refieren a las funciones específicas del software y definen qué es lo que se espera que realice el producto software que se desarrollará.

El principal problema de la especificación de los requisitos funcionales estriba en que esta generalmente consiste en una colección de frases en lenguaje natural con los correspondientes problemas de ambigüedad, escalabilidad y trazabilidad. Los casos de uso (UC) y los casos de uso del

---

\* Este trabajo ha sido parcialmente financiado por la CICYT(TIC97-0593-c05-05) como parte del proyecto MENHIR.

\*\* Oscar López agradece el apoyo financiero de la AECI en España y del MICYT en Costa Rica

negocio (BUC), propuestos por Jacobson [6][7], han surgido como una estrategia de captura de requisitos ampliamente aceptada y que representa los requisitos funcionales como una colección de escenarios de interacción entre el usuario y el sistema. Aunque resuelven las deficiencias de escalabilidad y trazabilidad, y proporcionan facilidad para la descripción, los casos de uso no superan la ambigüedad del lenguaje natural. Se requieren enfoques que permitan derivar los casos de usos en concordancia con algún formalismo sintáctico y semántico, lo cual ya ha sido señalado por Lee *et al* [9].

En este contexto, Laguna [8] ha propuesto determinar la funcionalidad inicial del sistema software a partir de la descripción de la forma actual del quehacer del usuario. El presente trabajo está orientado a desarrollar dicha propuesta a través de la definición precisa de una herramienta para la determinación de los requisitos funcionales de un sistema de información y almacenarlos en forma de assets de análisis. Como punto de partida se ha escogido utilizar los flujos de trabajo (workflows) modelados con redes de Petri lo que permitirá la generación automática de casos de uso y de otros elementos reutilizables de nivel de análisis (assets de análisis) que pueden ser incluidos en un repositorio para formar mecanos [4]. De esta manera las bondades de los casos de uso se refuerzan mediante el uso de workflows lo que permite la escalabilidad y la trazabilidad. Al mismo tiempo se corrige la informalidad de los casos de uso mediante un formalismo robusto proporcionado por las redes de Petri. En síntesis, se presenta una alternativa para la normalización del proceso de captura de requisitos de usuario que permite modelar la organización como un workflow particular implementado con una red de Petri y que se utiliza para generar casos de uso y assets como componentes de mecanos.

El resto del documento se organiza de la siguiente manera: La sección 2 presenta un marco de referencia para la normalización del proceso de captura de la funcionalidad del sistema. En la sección 3 se especifica el proceso para la utilización de los workflows como punto de partida para automatizar la captura de requisitos. La sección 4 muestra los requisitos capturados expresados en una plantilla de requisitos funcionales. Y por último, la sección 5 concluye el presente artículo y señala aspectos conexos a desarrollar a partir de este trabajo.

## **2 UN MARCO GENERAL PARA LA ELICITACION DE REQUISITOS**

Los problemas para la definición precisa de la funcionalidad se acentúan por el complejo esquema de comunicación en el que interaccionan el usuario y el analista de sistemas o el ingeniero de requisitos. El usuario brinda una concepción de la funcionalidad aunque podría no estar completamente seguro de ello. El analista por su parte deberá especificar correctamente la funcionalidad a partir de esta primera concepción mediante aproximaciones sucesivas. Este ambiente de interacción usuario-analista motiva la búsqueda de estrategias robustas para garantizar que los requisitos del usuario serán descubiertos con precisión y que además serán expresados en una forma correcta y sin ambigüedad, y que sea verificable, trazable y modificable. Todo esto permite concluir que se requiere un marco de trabajo que provea algún grado de formalización para realizar el análisis de los requisitos del usuario [8].

Ante las dificultades para obtener los requisitos correctos del usuario a partir del análisis de requisitos, y en vista de que para obtener los mayores beneficios de la reutilización es muy importante comprender la funcionalidad del sistema, se presentará un modelo formal del proceso de elicitación de requisitos. Este modelo se sitúa en el contexto de los sistemas de eventos discretos. Se prestará atención a la evolución de los estados sin importar cuándo el sistema alcanza un estado particular, ni cuánto tiempo permanece en tal estado. En otras palabras, el modelado de los requisitos en esta etapa está centrado en la secuencia de estados del sistema. Para ello se hará uso de las redes de Petri que proporcionan un soporte formal para la verificación sistemática del comportamiento lógico del sistema [13][15][9].

A partir del modelo de formalización del proceso de elicitación de los requisitos de usuario se tiene un marco de trabajo para tratar analíticamente los requisitos del sistema. El marco general propuesto se presenta en la figura 1.

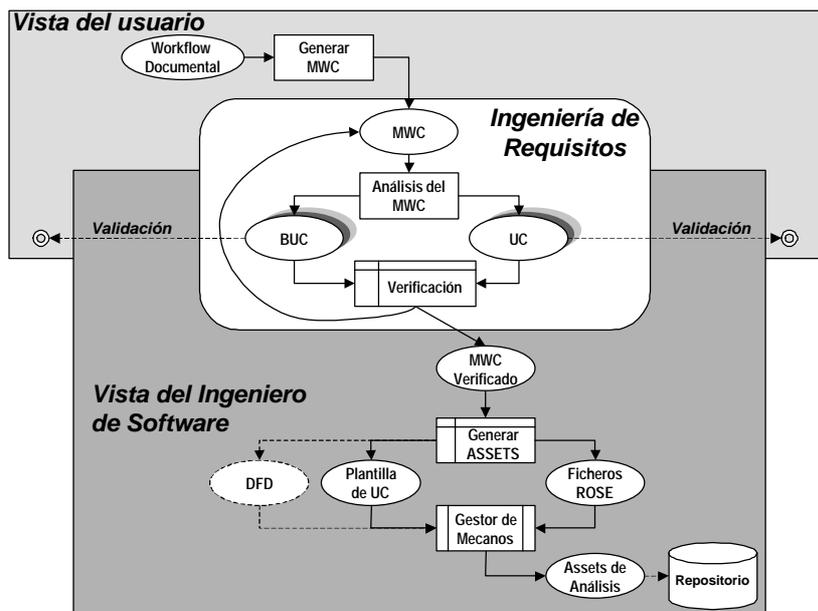


Figura 1. Marco general de trabajo para la normalización de la captura de los requisitos de usuario.

La figura 1 establece dos niveles relacionados en el proceso de elicitación de los requisitos: El nivel del usuario y el nivel del ingeniero de software. El primero tiene una visión externa del sistema, vista de caja negra, y el segundo corresponde al interior de esa caja negra. Dentro del nivel del Ingeniero del software existe una vista del Ingeniero de Requisitos la cual actúa como una interface entre los dos niveles anteriores. En estos niveles se conforma el marco general para la normalización de la captura de los requisitos.

El nivel del usuario proporciona el contexto inicial de trabajo donde el usuario provee la primera aproximación de los requisitos funcionales del sistema. Se inicia con un workflow documental que según los estándares correspondientes definidos por la WFMC se han utilizado con éxito para representar la logística de los procesos de negocio [15]. En el marco de la captura de requisitos se utilizará un tipo específico de workflow documental, el diagrama Documentos-Tareas [2] que utilizado rigurosamente cumple con los estándares de la WFMC en tanto que especifica qué tareas se deben realizar y en qué orden. El workflow inicial permite modelar el flujo de la información desde que esta ingresa en el sistema hasta que sale debidamente transformada. En esta etapa la propuesta metodológica proporciona una definición preliminar de los requisitos de usuario.

En la vista del Ingeniero de Requisitos, se modela la funcionalidad del sistema a través de un Workflow Case Modificado (MWC) tal y como se explicará en la Sección 3. Mediante el análisis del MWC se obtendrán familias de casos de uso del negocio (BUC) y familias de casos de uso (UC). Las actividades de validación y verificación permitirán corregir la versión inicial del MWC y de ser necesario, el proceso puede solicitar una nueva generación del MWC y en consecuencia de los BUC y los UC.

El MWC debidamente verificado es la fuente para la generación de los assets. El Ingeniero de Software podrá generar los assets ya sea en forma de alguna plantilla específica (por ejemplo, la plantilla propuesta por Durán [3]) o mediante algún formato particular como puede ser el de los ficheros Rose [11]. Estas dos formas alternativas de expresión de los assets de análisis están referidas directamente a las técnicas orientadas al objeto. La expresión de los assets como DFD también puede ser soportada por el marco que se propone.

Finalmente, los assets generados bajo el esquema propuesto están listos para ser utilizados en el desarrollo de productos software en el contexto de la reutilización del software. En consecuencia, el marco general de la solución planteada conduce a assets de análisis aptos para asociar mediante una interfaz de gestión del repositorio con los assets de diseño e implementación correspondientes para formar los mecanos requeridos [4].

### **3 UTILIZACION DE WORKFLOWS Y REDES DE PETRI PARA LA CAPTURA DE REQUISITOS FUNCIONALES**

Un workflow es un modelo de la logística de los procesos del negocio. Un proceso de workflow es realmente un diagrama que especifica el flujo de tareas que se deben ejecutar en un orden determinado y bajo condiciones determinadas. Las redes de Petri se han utilizado para expresar la semántica de los workflows. Un workflow expresado con una red de Petri se ha denominado un workflow net [14].

Una red de Petri permite manipular eventos que son activados bajo condiciones explícitas. Estas redes han sido ampliamente utilizadas en el modelado de sistemas cuyo comportamiento es guiado por un patrón de reglas complejas. Gráficamente, una red de Petri es un grafo orientado que contiene dos tipos de nodos. 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. Cada lugar puede contener cero o más marcas. Al conjunto de marcas en un momento dado se le llama el marcado de la red de Petri [13].

Tomando un dDT como un caso particular de workflow y dado que se han utilizado las redes de Petri para modelar los workflows [14], se puede utilizar las redes de Petri para modelar un dDT. De esta manera las tareas del dDT serán visualizadas como las transiciones de la red Petri. Los documentos son representados por los lugares. El marcado de la red de Petri podría representar la interacción que muestra la secuencia de estados y no la evolución específica de los casos de uso en conjunto.

Para ilustrar la representación de un dDT como una red de Petri considérese un proceso hipotético de pago de salarios. Primero, se recibe un documento con el informe de horas trabajadas por un empleado. Este documento llega a la tarea denominada Registrar Horas. Si hay algo erróneo con el informe se produce una notificación de error para el empleado, de otra manera se registran los datos recibidos y se produce un nuevo documento llamado Registro de Horas el cual a su vez se convierte en la entrada para las tareas Calcular Retenciones y Calcular Pago Bruto. De estas dos últimas tareas surgen los respectivos documentos Retenciones y Salario Bruto que disparan la tarea Procesar Cheque, que a su vez produce dos salidas, el Cheque y el Informe de Salarios. Finalmente, el Informe de Salarios y la Lista de Control son las entradas de la tarea Auditar Nómina que produce la salida Informe de Auditoría. En la figura 2 se muestra gráficamente el proceso descrito de pago de salarios.

El dDT de la figura 2 será modelado como una red de Petri para lo cual se tratará a cada documento como un lugar y a cada tarea como una transición. De esta manera se tiene una red de Petri particular en donde los lugares además de contener el marcado contienen una descripción del documento que representan. Del mismo modo, las transiciones no solo serán las impulsoras de la evolución del marcado sino que también contendrán una especificación de las actividades que conforman la tarea que representan. Y con el propósito de ampliar el potencial descriptivo, el nuevo diagrama es enriquecido en dos sentidos. Por un lado se agregan otros detalles de representación a las transiciones. Y por otro lado, se definen cuatro tipos de transiciones que permiten modelar los diferentes flujos de acción de la empresa.

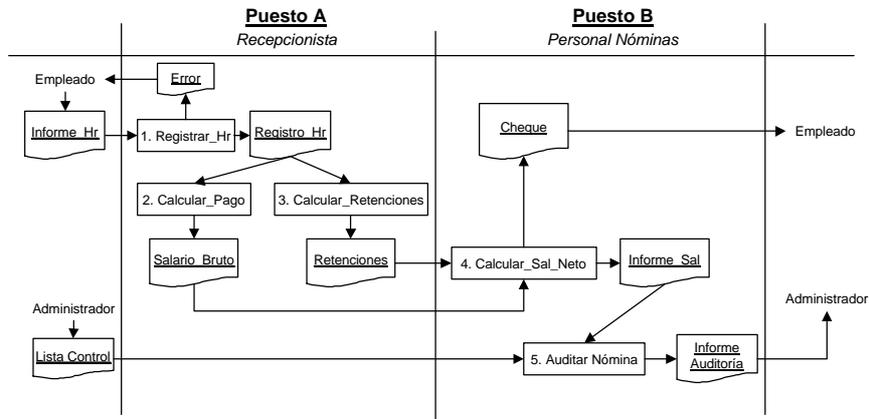


Figura 2. Diagrama dDT para el procesamiento de salarios

Los detalles adicionales de representación de las transiciones hacen que estas sean vistas gráficamente según la figura 3. En el contexto organizacional, para que se ejecute una tarea no solo es necesario que existan los insumos requeridos (en este caso los documentos de entrada a una transición) sino que suelen haber pre-condiciones específicas. Además, la ejecución de una tarea no sólo resulta en la producción de algún documento de salida sino que pueden haber post-condiciones específicas. En algunos casos, aún cuando se cuente con los insumos y las pre-condiciones, las tareas requieren que se produzca un evento temporal para ser ejecutadas. Finalmente, cada tarea tiene un responsable directo dentro de la organización.

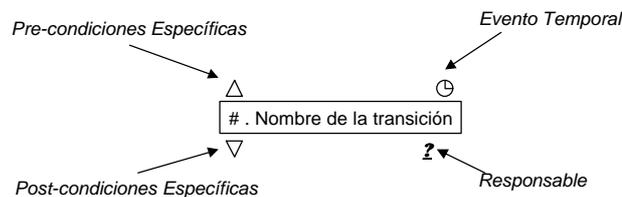


Figure 3. Representación gráfica de una transición genérica en un MWC

Los diferentes tipos de transiciones identificadas en el contexto del negocio se muestran en la figura 4. La transición de tipo AA es la estándar de una red de Petri en donde se requiere la presencia de marcas en cada documento de entrada y el disparo genera las marcas correspondientes en cada documento de salida. El tipo OA se refiere a una transición que se puede disparar con sólo la

presencia del marcado en alguno de sus puntos de entrada y su disparo generaría las marcas para cada uno de sus puntos de salida. El tipo AO es aquella transición que se dispararía ante el marcado que debe estar presente en cada uno de sus lugares de entrada y produciría las marcas correspondientes en alguno de sus lugares de salida. Y por último, la transición de tipo OO es la que se puede disparar con solo las marcas de uno de sus lugares de entrada y genera las marcas para alguno de sus lugares de salida.

A partir de los diferentes tipos de transiciones mostrados en la figura 4, se determinan los diferentes tipos de flujos de acción que se pueden encontrar en la organización según se muestra en la figura 5. Las acciones secuenciales y las paralelas pueden ser descritas mediante transiciones estándares del tipo AA. Sin embargo, las acciones condicionales y las iterativas requieren de la adecuada combinación de los tipos AA, OA, AO y OO.

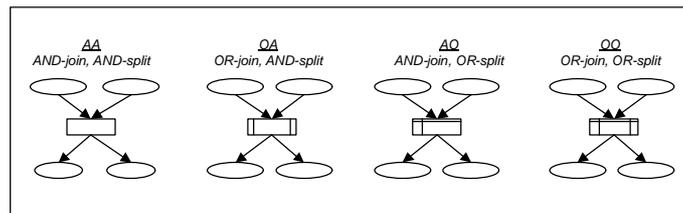


Figura 4. Los cuatro tipos diferentes de transiciones para el modelado de los MWC.

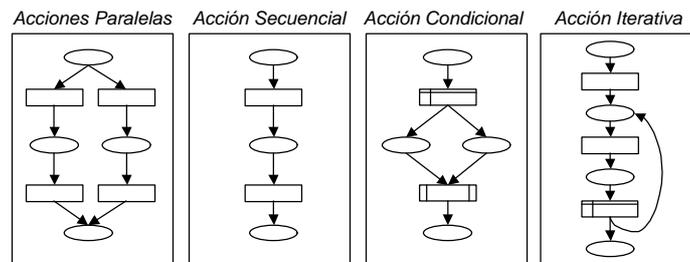


Figura 5. Los tipos de flujo de acción básicos para el modelado de casos de uso con MWC.

Con los detalles adicionales de representación, el dDT de la figura 2 se transforma en un Workflow Case Modificado (MWC) tal y como se representa en la figura 6

El MWC de la figura 6 no se ajusta a la forma de disparo típica de una red de Petri en tanto que la transición 1 no es estándar, (además se le han añadido detalles extras de representación). Se puede solventar el problema de la transición no estándar de dos maneras. Una es definiendo las propias reglas de disparo de las transiciones para los MWC. La otra es expresando las transiciones no estándar en los términos de los estándares de una red de Petri. Dado el interés por realizar un análisis formal de los casos de uso, se escoge la segunda opción. Las transiciones no estándar se podrían re-escribir según se expone en la figura 7. Para los efectos prácticos, lo que se hace es sustituir los tipos OA, AO y OO por la adecuada combinación de transiciones del tipo AA las que a su vez heredan los detalles adicionales de representación.

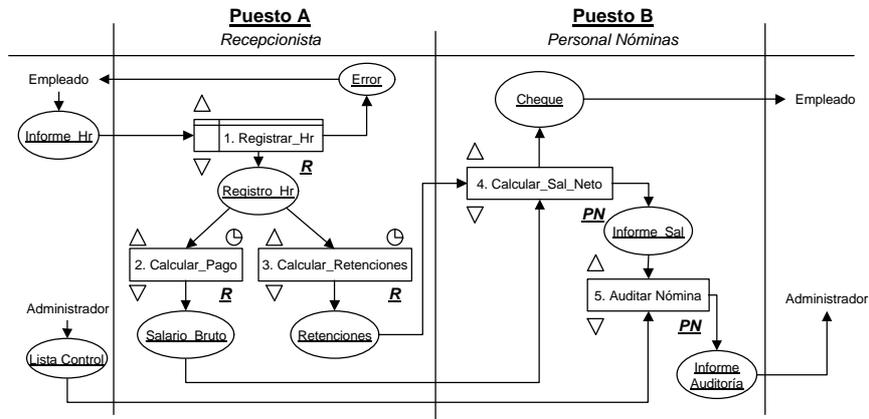


Figura 6. Diagrama del MWC final para el procesamiento de salarios

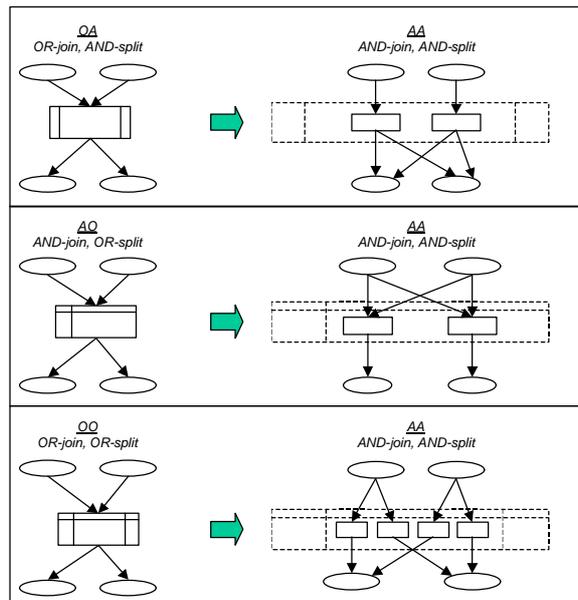


Figure 7. Estandarización de las transiciones OA, AO y OO para el modelado de los MWC.

Dado que la transición 1 de la figura 6 es una del tipo AO, se procederá a transformar ésta incluyendo dos transiciones en su lugar para obtener el workflow expresado como una red de Petri estándar lo cual se refleja en la figura 8.

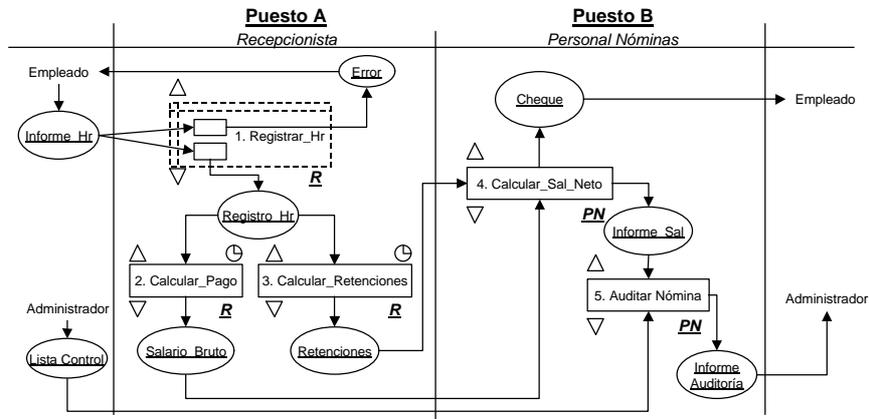


Figura 8. Red de Petri para el diagrama del MWC final para el procesamiento de salarios

A partir de la red de Petri de la figura 8 se obtienen los BUC y los UC de las figuras 9 y 10. Merece ser resaltado que la generación de los BUC y de los UC a partir del MWC es un proceso algorítmico. Es decir, teniendo el MWC debidamente verificado, los BUC y los UC se obtienen automáticamente. Para la obtención de los BUC el procedimiento consiste en tomar cada uno de los documentos de entrada y obtener todos los caminos posibles a través de la red para volver al punto de origen del mismo documento. Para obtener los UC a partir de un BUC se procede a agrupar para cada actor interno las transiciones y los correspondientes documentos relacionados de entrada y salida.

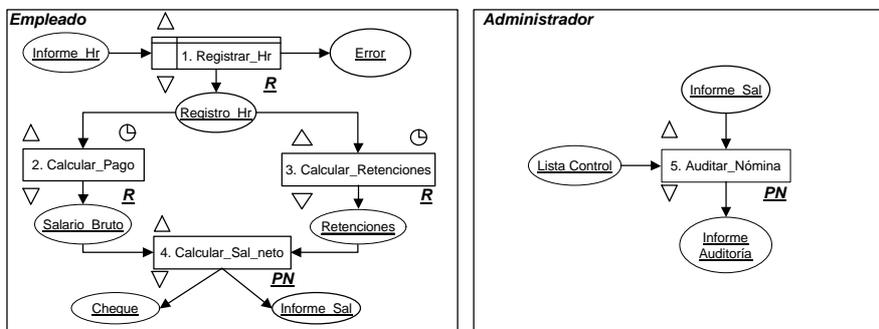


Figure 9. Casos de Uso del Negocio (BUC) para procesamiento de salarios.

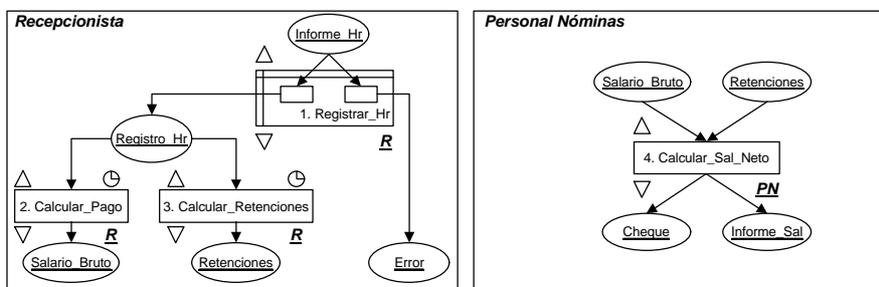


Figura 10. Red de Petri para los casos de uso derivados del BUC del Empleado

De acuerdo con el marco de trabajo expuesto en la sección 2, la funcionalidad del sistema expresada mediante redes de Petri y workflows se puede presentar en términos de alguna plantilla de expresión de requisitos como puede ser la plantilla propuesta por Durán, este paso será explicado en la sección 4. La expresión de la funcionalidad en formato XML será objeto de trabajo futuro.

También se pueden realizar análisis para establecer familias de BUC y de UC de manera que se puedan realizar análisis de dominios conducentes a establecer combinaciones de requisitos y estimaciones de desarrollo lo que será una acción futura a partir de este trabajo.

#### 4 REQUISITOS FUNCIONALES EXPRESADOS EN PLANTILLAS

A partir del sistema modelado a través de un MWC, tal y como se ha descrito en el presente trabajo, los requisitos funcionales pueden ser expresados en algún formato particular. Si se requiere obtener una especificación de los casos de uso en el formato de alguna plantilla se debería realizar en forma automática. Para lo anterior se requiere especificar –lo que debiera hacerse en algún lenguaje específico- cada una de las tareas representadas en las transiciones. Para efectos del ejemplo del pago de salarios, considérese el cuadro siguiente:

<i>TAREA</i>	<i>DESCRIPCION</i>
1. Registrar_Hr	a. Recibir Informe de Horas b. Validar datos del Informe de Horas c. Si (No._Hr en rango-válido) Registrar datos del Informe de Horas, sino generar notificación de error
2. Calcular-Pago	a. Realizar y registrar los cálculos del Salario Bruto en respuesta a un evento temporal
3. Calcular_Retenciones	a. Realizar y registrar los cálculos de las Retenciones en respuesta a un evento temporal
4. Calcular_Sal_Neto	a. Localizar y recuperar los datos de Salario Bruto y Retenciones b. Realizar y registrar los cálculos del Salario Neto e imprimir el cheque c. Registrar datos del Informe de Salarios
5. Auditar_Nómina	a. Recibir Lista de Control b. Localizar y recuperar los datos del Informe de Salarios c. Procesar Informe de Auditoría

Con base en las descripciones de las tareas del procesamiento de salarios, la expresión de los casos de uso según la plantilla propuesta por Durán [3] puede ser como sigue:

<b>RF-01</b>	<b>Procesamiento de Horas</b>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso concreto cuando un empleado envíe su informe de horas trabajadas	
Precondición	El empleado es un trabajador de la empresa	
Secuencia normal	Paso	Acción
	1	El recepcionista solicita al sistema iniciar el proceso del Puesto A.
	2	El sistema solicita los datos del Informe de Horas
	3	El empleado proporciona los datos del Informe de Horas
	4	El sistema valida los datos del Informe de Horas
	5	Si los datos del Informe de Horas cumplen con las validaciones el sistema registra los datos del Informe de Horas.
	6	El sistema realiza y registra los cálculos del Salario Bruto
Excepciones	7	El sistema realiza y registra los cálculos de las Retenciones
	5	Si el sistema detecta algún error en el Informe de Horas entonces se envía una notificación de error al Empleado
Poscondición	La información de Salario Bruto y Retenciones está actualizada	
Desempeño	Por definir	
Importancia	Por definir	
Frecuencia	Por definir	
Urgencia	Por definir	
Comentarios		

<b>RF-02</b>	<b>Procesamiento de Cheques</b>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso concreto cuando el personal de nóminas lo solicite	
Precondición	Se hallan registrados los datos de Salario Bruto y Retenciones	
Secuencia normal	Paso	Acción
	1	El personal de nóminas solicita al sistema iniciar el proceso del Puesto B.
	2	El sistema localiza y recupera los datos de Salario Bruto y Retenciones
	3	El sistema realiza y registra los cálculos del Salario Neto e imprime el cheque
	4	El sistema registra los datos del Informe de Salarios
Excepciones		
Poscondición	Se imprimió el cheque y se generó el Informe de Salarios	
Desempeño	Por definir	
Importancia	Por definir	
Frecuencia	Por definir	
Urgencia	Por definir	
Comentarios		

Los assets expresados mediante la Plantilla pueden ser enviados a un gestor de mecanos para finalmente ajustar los assets bajo la semántica específica del repositorio y así poder recuperar los mecanos necesarios o formarlos en tiempo de reutilización [4].

## 5 CONCLUSIONES Y TAREAS INMEDIATAS

El presente trabajo utiliza las redes de Petri y los workflows como herramientas útiles para el modelado del análisis de dominios en el contexto de la reutilización del software. Se ha propuesto una forma normalizada para realizar el proceso de captura de los requisitos de usuario. Este proceso permite obtener los requisitos funcionales tanto a nivel de organización como a nivel de dominio del usuario. Con la funcionalidad capturada se podrían realizar análisis diversos que permitan establecer familias de casos de uso para efectuar estimaciones de costos para implementación de los diferentes conjuntos de casos de uso.

Los MWC como sistemas de funcionalidad paralela fundamentados en la teoría de redes de Petri que permiten capturar los requisitos funcionales en dominios específicos, deberán ser probados y validados para mejorar la elicitación de requisitos. Los MWC proporcionan una forma clara y no ambigua para modelar la dinámica del negocio con la información mínima requerida. De esta manera, los MWC facilitan la representación de los flujos de actividad de las organizaciones mediante un esquema de control complejo y robusto para modelar la captura de requisitos que se realiza en el contexto del negocio.

Las tareas inmediatas son:

- Proporcionar una herramienta completamente automatizada para traducir un workflow a assets de requisitos funcionales
- Fundamentar formalmente el análisis de los BUC y UC generados a partir de esta propuesta.
- Proponer estimaciones plausibles para el desarrollo de software a partir de las familias de BUC y UC generadas.

## 6 REFERENCIAS

- [1] Booch, G., Rumbaugh, J., Jacobson, I., “El lenguaje unificado de modelado”, Addison Wesley, 1999.
- [2] Collongues, Hugues and Laroche. “Merise. Methode de conception”, Dunod, 1987.
- [3] Durán, A. *et al.*, “Una propuesta metodológica para la recolección de requisitos de un sistema software”, III Jornadas de Trabajo MENHIR, 1998.
- [4] García, F.J., Modelo de Reutilización Soportado por Estructuras Complejas de Reutilización Denominadas Mecanos. TESIS DOCTORAL, Universidad de Salamanca, 2000.
- [5] IEEE, “Recommended Practice for Software Requirements Specifications”, Std 830-1993
- [6] Jacobson, I., *et al.*, Object-Oriented Software Engineering – A use case driven approach. Addison-Wesley, 1992.
- [7] Jacobson, I., Griss, M., Jonsson, P., Software Reuse, Architecture, Process and Organization for Business Success. ACM, 1997
- [8] Laguna, M.A., Marqués, J.M., García, F.J., “A user requirements elicitation tool”. IV Jornadas de Trabajo MENHIR, Sedano, 1999.
- [9] Lee, W.J., Cha, S.D., Kwon, Y.R., “Integration and Analysis of Use Cases Using Modular Petri Nets in Requirement Engineering”, IEEE Transactions on software engineering, 24:12, diciembre 1998.
- [10] Pressman, R., “Ingeniería del Software”, 4ta. Edición, Madrid, McGraw-Hill, 1998.
- [11] Rational Software, “A Rational Approach to Software Development Using Rational Rose 4.0”, 1998.
- [12] Rumbaugh, *et al.*, “Modelado y diseño orientados a objetos”. Madrid, Prentice-Hall: 1991.
- [13] Silva, M., “Las redes de Petri en la automática y la informática”, Madrid, 1985.
- [14] Van der Aalst, W.M.P., “The application of Petri Nets to Workflow Management”, Eindhoven University of Technology, Netherland, 1997. **URL:** [www.wis.win.tue.nl/~wsinwa/jcsc/jcsc.html](http://www.wis.win.tue.nl/~wsinwa/jcsc/jcsc.html)
- [15] WFMC. Workflow Management Coalition Terminology and Glosary (WFMC-TC-1011). Technical report, Workflow Management Coalition, Brussels, 1996. <http://www.aiim.org/wfmc/standards/docs/glossy3.pdf>

## AGRADECIMIENTOS

*La realización de este artículo no habría sido posible sin la participación crítica y constructiva de los miembros del grupo GIRO de la Universidad de Valladolid. Los autores desean manifestar su agradecimiento por todas las aportaciones recibidas.*