

# Comprensión de Especificaciones de Procesos de Negocios Escritas en BPMN

Mario Berón, Carlos Salgado, Mario Peralta, Fernando Saez  
Universidad Nacional de San Luis  
Ejército de los Andes 950  
{mberon,csalgado,mperalta,bfsaez}@unsl.edu.ar

## Resumen

La Comprensión de Programas es una disciplina de la Ingeniería de Software cuyo objetivo es proveer Modelos, Métodos, Técnicas y Herramientas con el propósito de facilitar el estudio y entendimiento de los sistemas de software. Uno de los desafíos más importantes en el contexto de Comprensión de Programas consiste en Relacionar el Dominio del Problema con el Dominio del Programa. El primero hace referencia a la salida del sistema. El segundo a las componentes de software utilizadas para producir dicha salida. Una manera de construir esta relación consiste en definir una representación para cada dominio y luego establecer un procedimiento de vinculación entre ambas representaciones. En este artículo se presenta una línea de investigación que tiene como principal objetivo la elaboración de estrategias para interconectar los dominios del problema y del programa de especificaciones de procesos de negocios escritas en BPMN (Business Process Modeling Notation).

**Palabras Clave:** Comprensión de Programas, Procesos de Negocio, Métricas, BPMN, BPD, XML.

## Contexto

La línea de investigación descrita en este Artículo se encuentra enmarcada en el contexto del proyecto: Ingeniería de Software: Aspectos de alta sensibilidad en el ejercicio de la profesión del Ingeniero de Software de la Universidad Nacional

de San Luis. Dicho proyecto, es reconocido por el programa de incentivos y es la continuación de diferentes proyectos de investigación de gran éxito a nivel nacional e internacional.

## Introducción

La Comprensión de Programas (CP) es una disciplina de la Ingeniería de Software cuyo objetivo es ayudar al programador a entender programas. La CP es útil para Mantenimiento y Evolución de Software (MES), Ingeniería Reversa (IR), Re-Ingeniería (ReI) y Educación. En las tres primeras disciplinas, ayuda a reducir costos y esfuerzos. En la última, asiste al novicio en el proceso de comprensión de algoritmos.

MES es una tarea crítica porque implica tres importantes tareas: perfectiva, correctiva y adaptativa. La primera está relacionada con la incorporación de nuevas funcionalidades. La segunda describe el proceso de eliminar errores de programación. La tercera aborda el problema de adaptar el sistema a nuevos contextos, por ejemplo, usar un nuevo formato de archivo, cambiar la organización del código, etc. Esas tres principales actividades consumen mucho tiempo y recursos. Xi y Hassan [9,10,11] muestran que el 39% de las actividades son perfectivas, 56.7% son correctivas y 2.2% se corresponden con tareas adaptativas. Finalmente, el 2.1% restante representa otras tareas relacionadas con MES.

Por otra parte, la historia del MES muestra un incremento lineal en los costos de proyectos dedicados a MES. Esta información justifica el interés de la comunidad científica por reducir los

esfuerzos dedicados a esas tres principales actividades.

Otra área muy importante es IR. Esta disciplina estudia la creación de estrategias de extracción de la información desde diferentes fuentes (como por ejemplo código fuente, archivos makefiles, documentación, etc.), con el propósito de: Elaborar documentación, Extraer la arquitectura del sistema, etc. IR implica grandes costos y esfuerzos porque generalmente incluye actividades similares a MES. ReI usa técnicas de IR para extraer información y proporciona estrategias para modificar el sistema de acuerdo a los requerimientos del programador. Por esta razón, ReI, de la misma forma que MES e IR, es un área donde se necesitan propuestas para minimizar costos y esfuerzos.

En todas las disciplinas (MES, IR y ReI) mencionadas en los párrafos precedentes, el programador debe entender grandes documentos con diferentes formalismos y metodologías. Además, debe vincular la información recuperada con el código fuente y comprender cómo el sistema lleva a cabo sus funcionalidades en un alto nivel de abstracción. La Comprensión de Programas asiste al programador en esas tres principales áreas. Esta actividad se lleva a cabo por medio de métodos, estrategias y herramientas que disminuyen las tareas operativas y cognitivas del programador [9,10,11,12]. Como un efecto colateral de esta última característica, los costos también se reducen.

Actualmente, existen muchas Herramientas de Comprensión de Programas (HCP) con sofisticadas técnicas de exploración de código. Las mismas funcionan adecuadamente, sin embargo ciertas tareas de comprensión son todavía muy complejas.

Muchas HCP usan solamente análisis estático para la extracción de información del sistema. Después de eso, esa información se representa visualmente con el objetivo de proveer diferentes vistas. De esta forma, el proceso de comprensión de programas se simplifica porque se puede analizar el sistema desde varios puntos de vista. No obstante, el programador está forzado a encontrar el código fuente usado en ciertas

ejecuciones del sistema. Esta última tarea consume tiempo y esfuerzo. Otra clase de herramientas analizan el flujo de ejecución y proveen otro tipo de visualización del sistema. Esas herramientas, conocidas con el nombre de depuradores, frecuentemente proporcionan información de bajo nivel que es útil en estados avanzados de la CP. Por ejemplo, en la comprensión de rutinas específicas del sistema de estudio.

Finalmente, otras herramientas combinan técnicas de recuperación de información estática y dinámica, pero su salida consiste de interesantes y complejos análisis de cada clase de información.

Generalmente, las HCP no contemplan algunos aspectos importantes de la CP. Un ejemplo de tales aspectos es la relación entre la salida del sistema y las componentes, a nivel programa, usadas para producirla. Esta relación se conoce como: La Interconexión entre los Dominios del Problema y Programa.

La característica mencionada previamente simplifica la exploración porque el programador solamente inspecciona las partes del sistema relacionadas con la funcionalidad de estudio.

En las últimas décadas los procesos de negocio y su modelado [1, 2] han adquirido una gran popularidad. Esto se debe a que posibilitan que las empresas puedan especificar sus tareas y analizar cómo ellas se llevan a cabo. Esta peculiaridad simplifica el proceso de mejora continua debido a que hace más fácil la evaluación y comprensión de cómo las actividades de la empresa se están llevando a cabo.

Los modelos de procesos de negocio y sus lenguajes de ejecución, al igual que el software convencional, se encuentran sujetos a tareas de mantenimiento. Como fue mencionado en los párrafos precedentes dicha tarea consume mucho tiempo y esfuerzo. Es precisamente en este aspecto en que los métodos, técnicas y estrategias proporcionados por la Comprensión de Programas, pueden ayudar a paliar las dificultades proporcionadas por las actividades de mantenimiento.

El proceso para comprender especificaciones de procesos de negocio es similar al de comprender software convencional. Es decir se deben crear

presentaciones de los dominios del problema y programa y luego intentar relacionar ambas representaciones con el propósito de identificar qué partes de la especificación/implementación llevan a cabo tareas específicas del dominio del problema.

Es claro que proporcionar información objetiva acerca de la mantenibilidad de los modelos, facilita la evolución de los procesos de negocio de aquellas empresas que evalúan constantemente sus procesos al estar involucradas en un proceso de mejora continua. Al mismo tiempo se proporciona soporte a la gestión de procesos de negocio al facilitar la evaluación temprana de ciertas propiedades de calidad de sus modelos, con lo cual las organizaciones se ven beneficiadas de dos maneras: 1) Garantizando el entendimiento y la difusión de los procesos, y su evolución, sin afectar su ejecución; 2) Reduciendo el esfuerzo necesario para cambiar los modelos con la consecuente reducción de los esfuerzos de mantenimiento y/o mejora.

## Línea de Investigación

El grupo de investigación centra su trabajo en desarrollar nuevos métodos, técnicas, herramientas y procesos, que ayuden a los ingenieros de software e informáticos a construir/modificar/reconstruir sistemas de mayor calidad. En este sentido son tres las temáticas que actualmente se están abordando. A continuación se describen cada una de ellas:

- **Construcción de Representaciones del Dominio del Programa:** Una forma de llevar a cabo esta tarea consiste en aplicar técnicas de análisis estático y dinámico a las especificaciones/implementaciones de procesos de negocios. El análisis estático posibilita la construcción de una representación fidedigna del proceso de negocio. Esta representación permite, la definición de métricas y su evaluación concreta sobre la representación antes mencionada. Por estas razones, uno de los intereses de la línea de investigación se centra en la definición/utilización de métricas para la evaluación a nivel conceptual y de implementación de los modelos de procesos de negocio. Esto puede ser un aspecto clave para obtener modelos de calidad que sirvan como soporte para mejorar el mantenimiento y la

adaptabilidad de los procesos de negocio a los cambios constantes a los que dichos procesos están sometidos. En función de esto, se propone implementar un conjunto de métricas definidas para modelos conceptuales de procesos de negocio especificados con BPMN [3]. Dichas métricas se basan en el metamodelo del OMG para definir de forma uniforme cada una de ellas en los diferentes modelos del paquete BPDM (BPDM, por su sigla en inglés de Business Process Definition Metamodel [4]).

El análisis dinámico es tan importante como el análisis estático. Esto se debe a que es posible extraer información relacionada con la ejecución de los procesos de negocio. El análisis estático usa algunas de las técnicas de análisis estático pero con un fin diferente. Dicho fin consiste en incorporar la información necesaria a las especificaciones/implementaciones que permitan transformarlas en otra equivalente pero que realice ciertas tareas adicionales, como por ejemplo: Informar qué partes de ellas fueron utilizadas para llevar a cabo ciertas tareas (las cuales son de interés para la gerencia como para los ingenieros involucrados en el proceso de mejora continua) del dominio de la solución. Este tipo de información, conocida con el nombre de información dinámica, es muy importante porque, además de las características previamente mencionadas, posibilita el cómputo de métricas centradas en el comportamiento en tiempo de ejecución del proceso de negocio. Por otra parte, complementa a la información estática y además puede ser utilizada para decorar la representación estructural del proceso de negocio.

- **Construcción de Representaciones del Dominio del Problema:** El lector puede observar que, hasta este momento, sólo se ha hablado de representaciones del dominio del programa (dominio de especificaciones/implementaciones). No obstante, para facilitar la comprensión de los procesos es necesario construir representaciones del dominio del problema. Es claro que esta tarea no es sencilla, la diversidad de contextos de aplicación

dificulta la elaboración de una representación válida para todos los dominios. No obstante, el grupo de investigación está considerando utilizar formalismos para formalizar dominios específicos. Uno de tales formalismos son las ontologías. Una ontología es una especificación formal de un dominio. Dicha especificación está compuesta por conceptos y relaciones entre ellos. Asociado a cada uno de los componentes antes mencionados se encuentran un conjunto de axiomas que permiten colocar restricciones sobre los conceptos, las relaciones y la ontología en sí. Otras aproximaciones se están analizando (modelos de dominio de UML, especificaciones construidas con métodos formales, etc.) con el objetivo de analizar su aplicabilidad y utilidad para comprender procesos de negocio.

- **Elaboración de Estrategias de Interconexión de Dominios:** Las construcciones de las representaciones fidedignas y útiles de los dos dominios antes mencionados no son suficientes para facilitar la comprensión de los procesos; es necesaria la elaboración de estrategias que permitan vincular ambos dominios. Con respecto a esta temática, el grupo de investigación se encuentra analizando la posibilidad de trasladar estrategias definidas para los lenguajes de programación convencionales al contexto de los procesos de negocio. Las estrategias analizadas hasta el momento se pueden clasificar según el momento en el que se lleve a cabo la vinculación en: Estrategias en Vida y Estrategias Post-mortem. Las primeras muestran la relación entre ambos dominios a medida que se llevan a cabo las tareas del proceso de negocio. Las segundas realizan la misma tarea pero una vez finalizadas las tareas del proceso bajo análisis. También, este tipo de estrategias, se pueden clasificar de acuerdo al tipo de información utilizada para llevar a cabo la relación en: Estáticas, Dinámicas e Híbridas. Básicamente, el equipo de investigación se encuentra dedicado a la elaboración de estrategias que se ubiquen en las clasificaciones antes mencionadas pero que posean las siguientes características: No sean

demasiado complejas de aplicar y el costo computacional y de implementación no sea demasiado alto.

## Resultados y Objetivos

Las tareas realizadas en la línea de investigación objeto de este artículo son las siguientes:

- **Estudio de herramientas de modelado BPMN:** Se ha realizado el estudio de distintas interfaces gráficas de código abierto para el modelado de procesos de negocio en el lenguaje BPMN. Más puntualmente se estudió, aplicando técnicas de Comprensión de Programas, la representación interna de los distintos componentes de los modelos de procesos de negocio, como por ejemplo Tareas, Subprocesos, entre otros. Esto permite tener una trazabilidad de cada uno de los objetos del modelo con su representación en código.
- **Especificación formal de métricas para modelos conceptuales de proceso de negocio:** Se llevó a cabo la especificación utilizando OCL, basados en el estándar de BPD [5], de métricas para modelos de procesos de negocio tomadas de la literatura, como por ejemplo las propuestas en [6, 7]. Para realizar la especificación de dichas métricas se consideraron distintos aspectos que abarca el metamodelo BPD, como por ejemplo: Objetos de Conexión, Carriles y Artefactos; Control de Decisiones de los Objetos; Evento de los Objetos de Flujo, etc.
- **Elaboración e Implementación de Herramientas:** el grupo de investigación se encuentra abocado a la especificación de una herramienta que permita la aplicación de manera automatizada de las métricas al código XPD de los modelos. Esto permitirá un estudio y análisis más confiable de los mismos. Desde esta perspectiva, es necesario tener un medio que permita analizar los códigos generados para los modelos de procesos de negocio y aplicar sobre dichos códigos las métricas propuestas. Esta tarea tiene por finalidad obtener un indicador de la calidad de los modelos estudiados. Desde este punto de vista, y debido a que todo proceso de

desarrollo de Software puede verse como un proceso de negocio, es factible aplicar técnicas de Comprensión de Programas en el análisis del código generado a partir de los modelos antes mencionados. Esto tiene como finalidad determinar la aplicabilidad de las métricas propuestas y así lograr los indicadores de calidad buscados

En la continuidad de este trabajo se prevé: i) Definir nuevas métricas para los modelos conceptuales de procesos de negocio; ii) Definir representaciones de los dominios del problema y programa; iii) Elaborar estrategias para interconectar los dominios antes mencionados y iv) Proponer formas de visualizar la información de cada dominio, como así también la provista por las estrategias de interconexión.

## Formación de Recursos Humanos

Las tareas relacionadas a la presente línea de investigación están siendo realizadas con el objetivo de elaborar tesis de doctorado, maestría, especialización y trabajos de fin de carrera. Se espera a corto plazo poder contar con una o más propuestas de tesis de cada una de las categorías mencionadas previamente. Esta tarea tiene como principal objetivo fomentar el estudio de las temáticas descriptas en esta línea de investigación a través de la captura de alumnos de postgrado y grado que lleven a cabo las tesis antes mencionadas.

## Referencias

- [1] M. A. Rappa, "The utility business model and the future of computing services," IBM Systems Journal, vol. 43, pp. 32-42, 2004.
- [2] J. Becker, M. Rosemann, and C. von Uthmann, "Guidelines of Business Process Modeling," Business Process Management, Models, Techniques and Empirical Studies (BPM'00). Springer, pp. 30-49, 2000.
- [3] OMG, "Business Process Modeling Notation (BPMN)," BPMI - OMG. <http://www.omg.org/spec/BPMN/1.2> 2009.
- [4] OMG, "Business Process Definition MetaModel (BPDM), Beta 1 OMG Adopted Specification," <http://www.omg.org/docs/dtc/07-07-01.pdf>, 2007.
- [5] N. Debnath, C. Salgado, M. Peralta, D. Riesco, and G. Montejano, "Optimization of the Business Process metrics definition according to the BPDM standard and its formal definition in OCL," presented at ACS/IEEE International Conference on Computer Systems and Applications - AICCSA 2010, Tunes, 2010.
- [6] E. Rolon, F. Ruiz, F. Ó. Garcia Rubio, and M. Piattini, "Aplicación de Métricas Software en la Evaluación de Modelos de Procesos de Negocio," Revista Electrónica de la Sociedad Chilena de Ciencia de la Computación, 2005.
- [7] V. Vitolins, "Business Process Measures," presented at Int. Conference on BALTIC DB&IS. Riga, Latvia., 2004.
- [8] OMG, "OMG Object Constraint Language (OCL)," 2012.
- [9] M. Berón, P. Henriques, R. Uzal. Inspección de Programas para Interconectar las Vistas Comportamental y Operacional para la Comprensión de Programas. Tesis Doctoral. Universidad Nacional de San Luis. 2009.
- [10] <http://www3.di.uminho.pt/~gepl/QUIXOTE/>
- [11] E. Oliveira, P. Henriques, M. Pereira. Proposta de um Sistema de Compreensao de Aplicações Web. Tesis de Maestría. Universidade do Minho. 2007.
- [12] S. Rifkin and L. Deimel. Applying Program Comprehension Techniques to Improve Software Inspections. Proceedings of the 19th Annual NASA Software Engineering Laboratory Workshop, Greenbelt, MD, Nov, 1994.