Transformación de modelos en el proceso de obtención de Modelos Conceptuales partiendo de BPMN

Fernández Taurant, Juan Pablo – Marciszack, Marcelo Martín
Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract

En éste trabajo se presenta el análisis de una alternativa como soporte al proceso de gestión de requerimientos de software. Esta alternativa consiste en la generación de modelos conceptuales que serán obtenidos mediante la aplicación de un proceso de transformación de modelos a partir del modelo de requerimientos en BPMN. Los modelos resultantes podrán ser utilizados para gestionar y validar los requerimientos de software, y al mismo tiempo para controlar y optimizar los procesos.

Palabras Clave
Transformación de modelos, XSLT, BPMN, Desarrollo de Software dirigido por modelos, validación de requerimientos de software.

Introducción

La gestión de procesos de negocio es actualmente para las organizaciones, un medio para mejorar su eficiencia y competitividad en forma continua.
Para la ingeniería de software, la descripción de estos procesos es una tarea esencial en relación a las actividades de captura de requerimientos.
Existe una variedad de herramientas que tienen por objetivo la captura de requerimientos para la obtención de esquemas conceptuales, cuyas deficiencias pueden detectarse en etapas tempranas del desarrollo de software.
El principal problema de estas herramientas radica no solo en la falta de trazabilidad de los requerimientos a lo largo del ciclo de vida de desarrollo, sino también en el análisis de los modelos generados a los fines de verificar su validez.
En función de estas necesidades, este trabajo se enfoca en el desarrollo de una alternativa como soporte al proceso de gestión de requerimientos que contribuya a obtener modelos coherentes y sin ambigüedades en sus definiciones, permitiendo mantener un control de los procesos, optimizarlos y mejorar su trazabilidad. Asimismo este trabajo servirá como soporte del proyecto "Validación de Requerimientos a través de Modelos Conceptuales", que se encuentra consolidado dentro de la línea de investigación de Sistemas de Información en el Dpto. de Ingeniería en Sistemas de Información de la Universidad Tecnológica Nacional, Facultad Regional Córdoba, marco dentro del cual se desarrolla este trabajo.
Para el análisis de esta alternativa se utilizará un dominio de ejemplo modelado en BPMN[1]. Los modelos creados serán exportados a archivos con formato XPDL[2], a los que luego aplicaremos transformaciones en dos niveles utilizando el lenguaje de transformación XSLT[3] y otro lenguaje de programación comercial con soporte para manejo de archivos XML[4]. Estas transformaciones darán como resultado nuevos archivos con un formato XML específico, que serán utilizados por simuladores de máquinas abstractas para validar los modelos generados.
Actualmente existen trabajos que proponen la transformación de modelos entre distintas metodologías de uso comercial, entre ellos el de "Transformación de modelos conceptuales mediante uso de XSLT"[5], centrado en la trazabilidad y en la pérdida de información asociada a la transformación de modelos entre distintas metodologías. En éste trabajo se propone la transformación de modelos hacia un formato específico que permita

posteriormente, validar los modelos generados.

Elementos del Trabajo y metodología

El proceso de creación y mantención de modelos de requerimientos es una actividad que se realiza generalmente en forma manual, generando con gran frecuencia, inconsistencias entre modelos.
Estas inconsistencias impactan de forma negativa en la trazabilidad de los requerimientos, y adicionalmente, dificultan su análisis para verificar la validez de los mismos.
Por estas razones, se plantea una alternativa que permita mantener la trazabilidad de los requerimientos a través de una serie de transformaciones entre modelos.

- ***Representación de modelos***

El analista es la persona encargada de comprender las necesidades reales de los usuarios y definir el conjunto de requisitos que permiten a los desarrolladores construir el sistema adecuado. Resulta bastante común, que la comunicación entre el analista y los usuarios del sistema se vea dificultada debido a diferencias semánticas que se dan a causa del manejo de distinto vocabulario.
Como solución a estos problemas de comunicación, existen una serie de estándares, entre los más destacados podemos encontrar Business Process Modeling Notation (BPMN) y Unified Modeling Language (UML)[6], donde BPMN destaca por poseer una notación de fácil lectura y entendimiento para las partes involucradas.

- ***Herramientas para transformación de modelos***

Igualmente, podemos encontrar distintos estándares disponibles para realizar transformaciones de modelos, entre ellos Query View Transformation Language (QVT)[7], cuyo principal problema es la falta de soporte de modelado de las herramientas que contienen este procesador, en las que también se destacan su incompatibilidad con BPMN y su propia forma interna de representación de modelos.
Otro tipo de herramientas alternativas, y quizás mayormente utilizadas, son las basadas en el estádar Extensible Markup Language (XML), que nos permite representar modelos por medio de los estándares Metadata Interchange (XMI) [8] de la OMG, y Process Definition Language (XPDL), y transformarlos por medio del estándar Extensible Stylesheet Language Transformations (XSLT).

- ***Transformación de modelos***

Las transformaciones se realizarán a partir del modelado de procesos de negocio en BPMN sobre un dominio de ejemplo, en donde el analista deberá seleccionar aquellas actividades que se realizarán en forma manual que no serán mapeadas al SI. Para indicar esto utilizaremos el estereotipo "Manual" de BPMN.
Luego, se realizarán las transformaciones de los modelos BPMN para la obtención de archivos en formato XML que serán utilizados por los simuladores de autómatas finitos.
El esquema completo de transformaciones puede verse como un proceso en cuatro capas, como lo indica la figura 1.
La primera capa corresponde al modelo conceptual, que contiene en este caso, los modelos BPMN que luego serán exportados a archivos en formato XPDL para poder realizar las transformaciones.
Las dos capas siguientes corresponden a la transformación de modelos, son las más importantes y a partir de las cuales obtendremos los archivos XML para los simuladores de máquinas abstractas.
En la primera de estas capas intermedias se prepara el archivo XPDL para aplicar las transformaciones XSLT, este proceso de

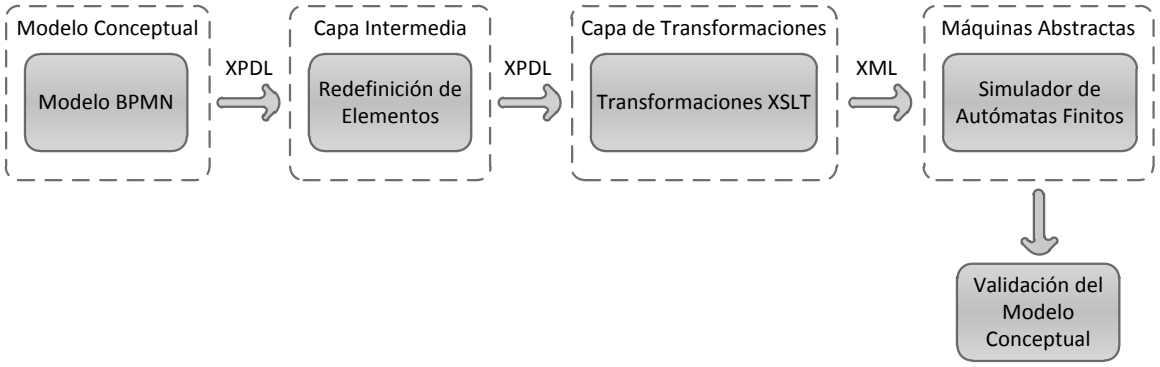

Figura 1

preparación debe realizarse debido a ciertas limitaciones técnicas que posee el lenguaje XSLT, en el que se incluye tareas tales como redefiniciones de elementos, reestructuraciones, entre otras.

La segunda capa intermedia es la que realiza la transformación de los modelos aplicando las distintas reglas definidas en lenguaje XSLT.

Para llevar a cabo tanto el proceso de preparación del archivo XML como el de transformación se puede utilizar cualquier lenguaje de programación disponible en el mercado con capacidades de manejo de archivos XML. Ambos procesos se ilustran en la Figura 2 y Figura 3 respectivamente.

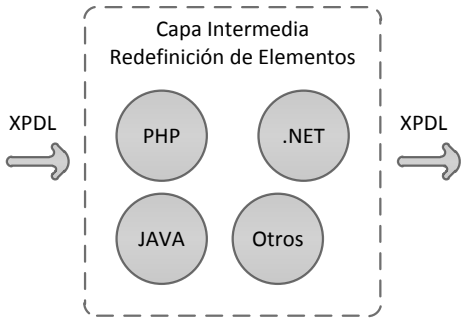

Figura 2

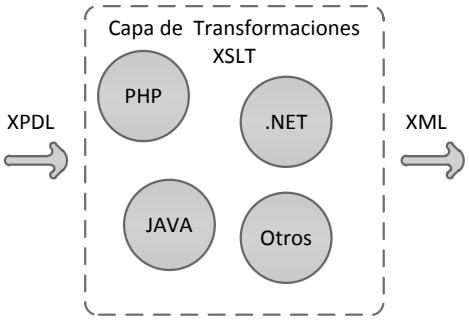

Figura 3

Finalmente la última capa es la que contiene los archivos XML que fueron generados como resultado de la aplicación de las transformaciones anteriores.

Por cada archivo XPDL generado a partir del proceso modelado en BPMN, obtendremos un nuevo archivo XML para simuladores de autómatas finitos, con el que se validarán los modelos conceptuales iniciales.

- ***Aplicación en un caso práctico***

Para el análisis de esta alternativa sobre un caso práctico utilizaremos un dominio de ejemplo modelado en BPMN, del cual tomaremos el proceso que indica la figura 4, sin tener en cuenta las actividades seleccionadas por el analista con el estereotipo "Manual".

Luego se exporta el proceso utilizando alguna herramienta como BIZAGI, Enterprise Architect, etc., a un archivo con formato XPDL que representará la salida de la primera capa del modelo conceptual.

En la figura 5 se puede observar una actividad en formato XPDL, en este caso de la actividad "Registrar Alumno".

El siguiente paso consiste en realizar las transformaciones de modelos en dos capas intermedias.

En la primera capa intermedia se prepara el archivo XPDL redefiniendo elementos para aplicar las transformaciones en la siguiente capa. Estas redefiniciones consisten principalmente en la reestructuración del archivo XPDL reasignando nombres de actividades, relaciones y

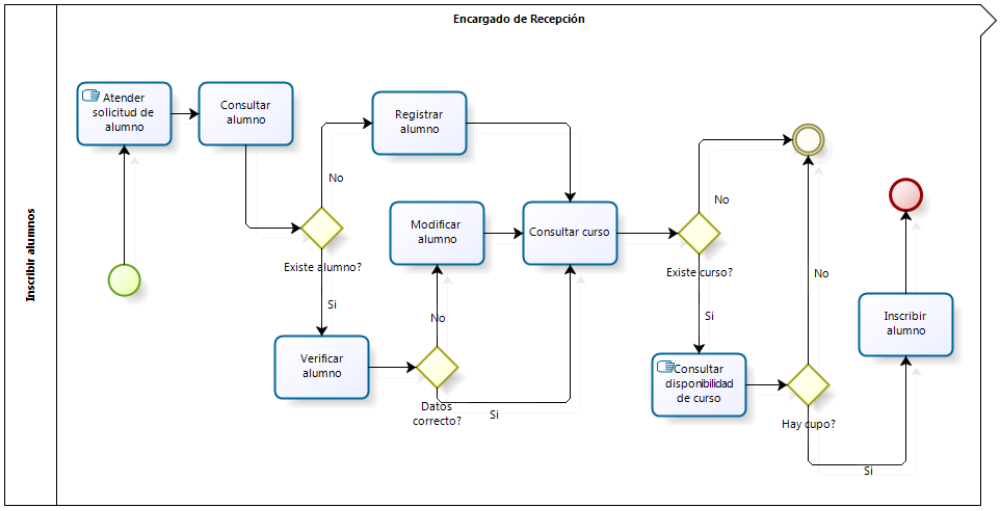

Figura 4

transiciones. En la figura 6 se puede observar la actividad "Registrar Alumno" redefinida manteniendo el formato XPDL.

En la segunda capa de transformaciones se procesa el archivo XPDL aplicando diferentes reglas XSLT dando como resultado un nuevo archivo con un formato XML específico que utilizarán los simuladores de máquinas abstractas.

La figura 7 muestra un ejemplo del formato de una transición de estados entre la actividad "Registrar Alumno" y la actividad siguiente "Consultar Curso".

```
<Activity
Id="2fb63c16-30bb-4bc6-8900-18d8e04aca95"
Name="Registrar alumno">
  <Description />
  <Implementation>
    <Task />
  </Implementation>
  <Performers />
  <Documentation />
  <Loop LoopType="None" />
</Activity>
```

Figura 5

```
<Activity Id="2"
Name="Registrar alumno">
  <Description />
  <Implementation>
    <Task />
  </Implementation>
  <Performers />
  <Documentation />
  <Loop LoopType="None" />
</Activity>
```

Figura 6

```
<transicion>
  <simbolo>
    <simbolo>L</simbolo>
  </simbolo>
  <estado-inicio>
    <denominacion>5</denominacion>
  </estado-inicio>
  <estado-fin>
    <denominacion>14</denominacion>
  </estado-fin>
  <denominacion>f( 5, L) = 14</denominacion>
</transicion>
```

Figura 7

Un ejemplo de reglas de transformación XSLT para la generación de estados se indica en la figura 8.

```
<xsl:template
  match="Activities"
  name="conjunto_estados">
<xsl:variable name="espace_name_2">
    java:dominio.automatas.Estado
</xsl:variable >
  <conjunto-estados>
    <xsl:for-each
    select="//Activities/Activity">
      <xsl:if test="(not(Event) and Loop)
        or Event/EndEvent
        or Event/StartEvent
        or Event/IntermediateEvent">
          <estado-automata>
            <xsl:attribute name="xsi:type">
              <xsl:value-of
              select="$espace_name_2"/>
            </xsl:attribute>
            <denominacion>
              <xsl:value-of select="@Id"/>
            </denominacion>
          </estado-automata>
      </xsl:if>
    </xsl:for-each>
  </conjunto-estados>
</xsl:template>
```

Figura 8

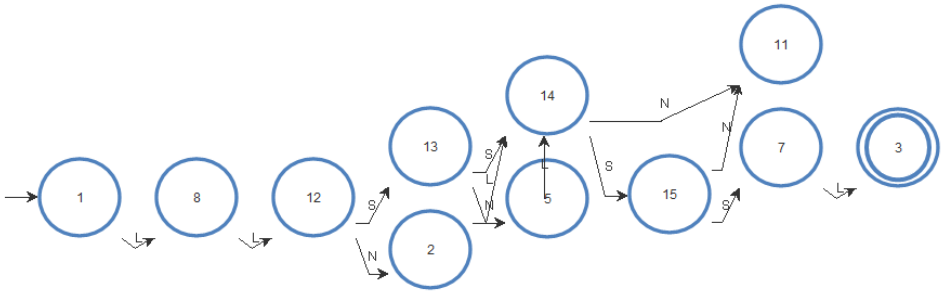

Figura 9

Por último, el archivo XML resultante contiene las definiciones necesarias de la máquina abstracta que podrá utilizarse para validar los modelos iniciales. El grafo correspondiente a la máquina abstracta generada se ilustra en la figura 9.

Resultados

Partiendo del modelo conceptual de un dominio de ejemplo modelado en BPMN, fue posible exportarlo a archivos con formato XPDL, aplicando posteriormente las transformaciones descriptas en cada una de las capas intermedias.
En la primera capa intermedia donde se preparan y limpian los archivos XPDL se utilizó como herramienta con soporte XML la plataforma .NET de Microsoft. También se utilizó esta plataforma para realizar las transformaciones XSLT de la segunda capa.
Como resultado de las transformaciones fue posible generar los archivos XML correspondientes a las máquinas abstractas que serán soportadas por los simuladores, para posteriormente poder analizar el modelo conceptual y chequear su validez.

Discusión

Los resultados obtenidos indican que es posible construir una herramienta que permita realizar transformaciones entre modelos. Los nuevos modelos generados se podrán utilizar para verificar la validez los modelos conceptuales iniciales.

En un futuro se seguirán realizando transformaciones y optimizando el proceso para tratar en lo posible, de reducir la cantidad de capas intermedias, evitando de esta manera excesivas manipulaciones de los archivos con el objetivo de minimizar tanto los niveles de procesamiento como la probabilidad de introducir errores entre las transformaciones intermedias.
Por otro lado se plantea la posibilidad de realizar transformaciones partiendo de diferentes niveles de abstracción de modelos conceptuales.
En relación a otros trabajos dentro de la misma línea de investigación, específicamente el de “Transformación de modelos conceptuales mediante uso de XSLT”, se evidencia en los resultados obtenidos la posibilidad de transformar modelos de diferentes metodologías con cierta pérdida de información asociada. En este trabajo se plantea un esquema de transformaciones entre modelos sin pérdida de información. Adicionalmente, los modelos correspondientes a las maquinas abstractas resultantes de las transformaciones podrán utilizarse para validar los modelos creados.

Conclusión

A través de este proceso de transformaciones y sus mejoras, será posible generar nuevos modelos que sirvan para representar las maquinas abstractas necesarias para la validación de los modelos conceptuales creados inicialmente, y en todo caso, optimizarlos

mediante la combinación con otra técnica o metodología, logrando de esta forma que los modelos representen la realidad sin ambigüedades, manteniéndose coherentes, y por consiguiente, aportando gran valor al proceso de gestión de requerimientos en cuanto a su definición y trazabilidad.

Esta herramienta se integra también como soporte en el marco de desarrollo del proyecto "Validación de Requerimientos a través de Modelos Conceptuales".

Adicionalmente, podrá ser utilizada como elemento de comparación con otras metodologías que arrojen resultados similares y hacer un análisis de los resultados obtenidos.

Referencias

[1] Object Management Group. Business Process Modeling Notation (BPMN). http://www.omg.org/technology/documents/br_pm_spec_catalog.htm, version 1.2, 3 January 2009.

[2] http://www.wfmc.org/xpdl.html.

[3] World Wide Web Consortium: XSL Transformations (XSLT). version 1.0, 16 November 1999. http://www.w3.org/TR/xslt

[4] World Wide Web Consortium: Extensible Markup Language (XML). Version 1.0(fifth edition), 26 November 2008. http://www.w3.org/XML/

[5] http://hdl.handle.net/10915/27229.

[6] OMG. Unified Modelling Language: Superstructure Version 2.0 (online), Julio 2005, http://www.omg.org.

[7] Object Management Group: MOF Query / Views / Transformations. Version 1.0, April 2008. http://www.omg.org/technology/documents/modeling_spec_catalog.htm.

[8] Object Management Group: XML Metadata Interchange (XMI). version 2.1.1, 1 December 2007. http://www.omg.org/technology/documents/modeling_spec_catalog.htm#XMI.

Datos de Contacto

Juan Pablo Fernández Taurant
jtaurant@gmail.com
Universidad Tecnológica Nacional – Facultad Regional Córdoba.
Maestro M. López esquina Cruz Roja Argentina.
X5016ZAA – Córdoba
República Argentina

Marcelo Martín Marciszack
marciszack@gmail.com
Universidad Tecnológica Nacional – Facultad Regional Córdoba.
Maestro M. López esquina Cruz Roja Argentina.
X5016ZAA – Córdoba
República Argentina