

Validación de Especificaciones Funcionales en el modelado de Esquemas Conceptuales a través de Máquinas Abstractas

Manuel Perez Cota¹, Mario Groppo^{1,2}, Marcelo Marciszack^{1,2}

¹ Facultad de Informática - Universidad de Vigo, España { mpcota@uvigo.es }

² Departamento de Ingeniería en Sistemas de Información -Universidad Tecnológica Nacional, FRC , Argentina { Sistemas@groppo.com.ar, mmarciszack@sistemas.frc.utn.edu.ar }

Abstract

En el presente trabajo se propone una definición metodológica para la especificación y validación de un Modelo Conceptual a través de la transformación de modelos a autómatas finitos y su validación correspondiente. Describe la característica del Proceso de Modelado a través de la aplicación del Desarrollo dirigido por modelos MDD, con utilización de BPMN para el modelado de Procesos de Negocios, su transformación mediante un proceso automatizado a Autómatas Finitos junto con la definición de controles a realizar sobre el modelo abstracto resultante, para de esta manera, validar el modelo de proceso de negocio que representa las especificaciones funcionales en el modelo conceptual del dominio bajo estudio.

Palabras claves: *Modelado Conceptual, Especificaciones funcionales, Validación de modelos, BPMN, Autómatas Finitos, Desarrollo de software Dirigido por Modelos MDD*

Introducción

Muchos son los esfuerzos que se han realizado y se siguen realizando para solucionar los problemas de lo que se ha dado por llamar “la crisis del software” [Gibs 1994], y se evidencia en la gran cantidad de metodologías, métodos y herramientas dedicados a capturar los requerimientos con el fin de obtener un Esquema Conceptual.

Las debilidades de la mayoría de los métodos para la obtención de esquemas conceptuales se reflejan en las primeras etapas del proceso de desarrollo. El principal problema derivado de estas debilidades metodológicas radica en la dificultad en determinar si el modelo conceptual refleja fiel y completamente la esencia del dominio [Insfrán 2002].

Existe una gran variedad de trabajos que evidencian que los errores que se cometen en la etapa de especificación de requerimientos para la obtención de un esquema conceptuales, tienen un costo relativamente alto en relación a su reparación y crecerá en forma exponencial a medida que se avanza en las diferentes etapas del proceso de desarrollo [Boehm 2001]. La preocupación por definir los requisitos de manera adecuada está extensamente tratada en [Sommerville 1997], donde el eje central es la definición de buenas prácticas en el establecimiento de los mismos, ya que plantea que “el éxito de cualquier proyecto de desarrollo está íntimamente relacionado con la calidad de los requisitos.” y que “el proceso de establecimiento de requisitos es mucho menos homogéneo y bien entendido que el proceso de desarrollo de software en su conjunto”.

Los sistemas de software no existen en forma aislada: se utilizan en un contexto social y organizacional y los requerimientos del sistema de software se deben derivar y restringir de acuerdo a ese contexto. A menudo, satisfacer estos requerimientos es crítico para el éxito del sistema a construir. Una razón de por que muchos sistemas de software se entregan, pero que nunca se usan, es porque no se tiene en cuenta en forma adecuada todos los requerimientos del sistema incluidos los de su contexto. [Sommerville 2005]

La tarea del análisis de requisitos es un proceso de descubrimiento, refinamiento, modelado y especificación y, por tanto, el

desarrollador y el cliente tienen un papel activo en la obtención de estas necesidades. Las últimas tecnologías utilizadas para la obtención de requisitos permiten una mejor comprensión de los documentos de especificaciones, que hasta ahora eran demasiado técnicos para la correcta comprensión por parte del usuario. [Sommerville 2011] [Sese 2006]

En este sentido los esquemas conceptuales deben procurar establecer una definición sin ambigüedad de lo que se quiere representar. Es así, que la presente propuesta abarca la definición de una metodología, un conjunto de herramientas de soporte de procesos, y la definición de transformaciones automatizadas entre los modelos intermedios, que posibilita validar y verificar si el modelo conceptual construido representa fielmente el sistema de información a construir.

Elementos de Trabajo y Metodología.

El proceso de creación y mantención de Modelos Conceptuales, 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 y posibilite la validación del Modelo Conceptual resultante.

Modelado Conceptual y Requerimientos Funcionales.

Desde la óptica disciplinar de los Sistemas de Información y los sistemas de software asociados a estos, un Esquema Conceptual será definido como un modelo de representación de la realidad, sobre un dominio de problema determinado, el cual deberá incluir además, el lenguaje utilizado en su definición, de manera que no existan

ambigüedades, de esta manera de reducir el “gap” semántico, entre el constructor del modelo y los usuarios del mismo.

En este contexto el presente trabajo, se focaliza con la visión aportada por [Insfrán 2002b] en donde un Esquema Conceptual es interpretado como un refinamiento de los requerimientos de usuario a través de los requisitos funcionales que resultarán en especificaciones más detalladas que constituirán dicho esquema.

En este mismo sentido otro aporte es el desarrollado por [Letelier 1999] en donde el Modelo Conceptual, establece los requisitos funcionales del Software y es uno de los resultados principales de dichas actividades, constituyéndose en una pieza fundamental para posteriores actividades en el desarrollo del Software.

El Modelo Conceptual representando los requisitos funcionales de un sistema de información, es la pieza clave para establecer el vínculo entre el espacio del problema y el espacio de la solución. Las deficiencias del modelo conceptual tienen un impacto considerable en las posteriores actividades en el proceso de desarrollo de software.

Tendencias actuales en la construcción de Modelos.

En los últimos años, el modelado de procesos de negocios, ha despertado especial interés por parte de la Ingeniería de Software, debido a que brinda un punto de partida para la captura de requisitos. Estos modelos se consideran esenciales para conocer las actividades de una organización, permitiendo establecer los fundamentos para la construcción de un sistema de información correcto.

La OMG ha utilizado para representar los modelos de negocio diferentes tipos de notaciones, pero le ha dado principal importancia a Business Process Modeling Notation (BPMN) [OMG 2009], y a Unified Modeling Language (UML) [OMG 2005] (a través de los diagramas de actividad y diagrama de casos de uso). Ambas notaciones ofrecen soluciones similares

para la mayoría de los patrones de flujo de trabajo que soportan. Esto es lógico debido a que ambos estándares fueron diseñadas para satisfacer las mismas necesidades de modelado, pero con objetivos diferentes en diferentes etapas del desarrollo.

El interés de la Ingeniería de Software en el Modelado del Negocio surge porque a partir del estudio de la transformación de modelos, es posible iniciar el modelado de sistemas de información (elicitación de requisitos) que se pueden integrar al proceso de desarrollo del software.

Para el desarrollo de esta propuesta nos centraremos en el Desarrollo de Software Dirigido por Modelos, (MDD) [Pons 2010], las definiciones y documentos emitidos por la OMG, [OMG 2007] , [OMG 2008] que es el organismo que se ha encargado del estudio y definición de los procesos de transformación de los modelos, en forma conjunta con el World Wide Web Consortium [w3 XLST 1999] , [w3 2008].

La transformación de modelos permitiría, además de mejorar los tiempos en los procesos de elicitación de requisitos, otorgar confiabilidad a todo el proceso de obtención del modelo conceptual, lo que se reflejará en la etapa posterior del desarrollo del software. Además como ventaja adicional brinda la posibilidad de ahorrar costos en el desarrollo e implementación de los sistemas, y permite evaluar algunos aspectos de calidad inherentes a los mismos. Estas consideraciones cubren gran parte del proceso de elicitación de los requerimientos funcionales y no funcionales en la construcción de un sistema de información.

Proceso de Captura de requerimientos.

El proceso de captura de requerimientos constituye la primera etapa de la ingeniería de requerimientos (IR), es una etapa esencial para conocer las actividades de la organización y clave para el éxito del sistema de información a desarrollar.

La ingeniería de requerimientos es el proceso que comprende todas las actividades necesarias para crear y

mantener los requerimientos de un sistema, estableciendo los lazos de comunicación entre el dominio del modelado del negocio y el dominio del sistema.

Sin embargo, esta comunicación no siempre es tan sencilla, ya que con frecuencia se ve dificultada por la existencia de una distancia semántica entre los dominios organizacional e informático [Taylor 1998], y por diferencias de formación y vocabulario entre las partes involucradas.

Para cerrar estas brechas de comunicación existen en la industria del modelado de negocio estándares ampliamente utilizados como Unified Modeling Language (UML), y Business Process Modeling Notation (BPMN), siendo BPMN más orientado al dominio organizacional, de fácil lectura, y más entendible para las partes intervinientes.

Otro aspecto importante en relación a la ingeniería de requerimientos, tiene que ver con la trazabilidad de los requerimientos a lo largo de todo el proceso de desarrollo. La trazabilidad es esencial para la correcta gestión y validación de los requerimientos de software.

Independientemente de la metodología utilizada, mantener la trazabilidad de requerimientos resulta una actividad compleja. Para esto, todos los modelos creados durante el proceso de desarrollo deben ser mantenidos y actualizados. La creación y mantención de modelos se realiza de forma manual, generando con gran frecuencia inconsistencias entre modelos, en detrimento de la trazabilidad de requerimientos.

Resulta entonces, imprescindible la aplicación de transformaciones automatizadas entre modelos, permitiendo que los modelos creados se mantengan actualizados, ya que los cambios realizados en uno de ellos serán propagados a los modelos restantes. De esta forma se contribuye a la trazabilidad de requerimientos, facilitando su gestión y validación.

Modelado Conceptual con BPMN

A través de Business Process Modeling Notation (BPMN), los “Procesos de Negocio” involucran la captura de una secuencia ordenada de las actividades e información de apoyo. Modelar un Proceso de Negocio implica representar cómo una empresa realiza sus objetivos centrales; los objetivos por sí mismo son importantes, pero por el momento no son capturados por la notación. Con BPMN, solo los procesos son modelados.

Las actividades de modelado en BPMN, se pueden percibir a través de distintos niveles de modelado de procesos:

Mapas de Procesos: Simples diagramas de flujo de las actividades; un diagrama de flujo sin más detalles que el nombre de las actividades y tal vez la condiciones de decisión más generales.

Descripción de Procesos: Proporcionan información más extensa del proceso, como las personas involucradas en llevarlo a cabo (roles), los datos, información, etc.

Modelo de Proceso: Diagramas de flujo detallados, con suficiente información como para poder analizar el proceso y simularlo. Además, esta clase de modelo más detallado permite ejecutar directamente el modelo o bien importarlo a herramientas que puedan ejecutar ese proceso.

Business Process Modeling Notation (BPMN) cubre todas estas clases de modelos y soporta cada nivel de detalle. Es una notación basada en diagramas de flujo para definir procesos de negocio, desde lo más simples hasta los más complejos y sofisticados para dar soporte a la ejecución de procesos.

El razonamiento inicial, fue que este enfoque, ayudaría a los usuarios finales dándoles una notación simple y acordada. Esto permitiría un entendimiento apropiado, utilizando cualquier tipo y número de herramientas. Las compañías no deberían re capacitar cada vez que se compre una nueva herramienta o se contrate nuevo personal que haya sido capacitado en otras herramientas y notaciones. En resumen, el

objetivo esencial es que el aprendizaje sea transferible.

BPMN entonces persigue dos objetivos, en principio contradictorios: proporcionar una manera fácil de utilizar la notación de modelado de procesos, accesible a los usuarios empresariales; y proporcionar facilidades para traducir los modelos a otros, mediante un lenguaje de representación definido y estandarizado. Cubrir en simultáneo estos dos objetivos, son las razones principales por las cuales en los procesos de construcción de Modelos Conceptuales, este enfoque metodológico resulta imprescindible.

Lenguaje Soportado por BPMN

El Lenguaje de Soporte es XPD. Tal como es tratado en [White 2011], XPD (XML Process Definition Language) es un lenguaje de la WfMC (Workflow Management Coalition) que es “Una organización sin fines de lucro para desarrolladores, analistas, consultores e investigadores en el campo de la gestión de procesos de negocio”. Fue fundada en 1993 y actualmente es miembro de la OMG siendo uno de los participantes que más influyeron sobre la especificación de UML 2.0.

La versión más reciente de XPD es la 2.0 y mantiene compatibilidad total con las versiones anteriores. Según los propios creadores de XPD, dejando muy claro el propósito de su especificación, las especificaciones XPD y BPMN afrontan el mismo problema de modelado desde diferentes perspectivas. XPD proporciona un formato de fichero XML para ser intercambiado entre aplicaciones. BPMN proporciona una notación gráfica para facilitar la comunicación humana entre usuarios de negocio y usuarios técnicos. Y precisamente esta última versión surge para dotar a XPD de los elementos de BPMN 1.0 que no poseía XPD 1.0.

Por lo tanto, XPD es una notación para definir e intercambiar modelos de procesos de negocio. A su vez, XPD puede ser considerado como la notación textual de

BPMN, o al revés, BPMN la notación gráfica de XPDL. Eso al menos para la versión de XPDL 2.0 que, como ya dijimos antes, se modificó precisamente para reflejar todos y cada uno de los elementos de BPMN. Por lo tanto XPDL y BPMN son un binomio a tener muy en cuenta dentro de campo del modelado de procesos de negocio, un campo que cada vez está adquiriendo más importancia para darle mayor efectividad, siempre que mantengan compatibilidad, lo ideal es encontrar una herramienta que nos permita usar ambas especificaciones de la siguiente manera:

Usar BPMN para modelar de manera gráfica los modelos de procesos de negocio (lo cual es más amigable tanto para los ingenieros como para los clientes).

XPDL para guardar los modelos e intercambiarlos entre las diferentes aplicaciones.

Ventajas de adopción de BPMN

Por lo vertido en secciones anteriores arribamos a la conclusión que BPMN es una herramienta muy simple y útil para el modelado de los procesos de negocio, y compitiendo directamente con los diagramas de actividad (UML), presenta ventajas por su simpleza tanto en gráficos, como en la adaptabilidad a las necesidades del usuario. Además, si se lo relaciona con XPDL, forman una pareja ideal para la exportación e importación de los modelos entre diversas herramientas.

Dentro de la propuesta metodológica para la obtención de Modelos Conceptuales, se concluye que la utilización de BPMN y XPDL generan una salida perfecta para construir los Automatas Finitos, que se utilizan como herramientas de validación y verificación de los Modelos.

Máquinas de Estados Finitos o Automata Finito

Un autómata finito (AF) o máquina de estado finito es un modelo computacional que realiza cómputos en forma automática sobre una entrada para producir una salida.

Los autómatas finitos deterministas quedan formalmente definidos mediante una quintupla como sigue:

$$AFD = (\Sigma , Q, q_0, F, f)$$

| | |
|----------|--|
| Σ | Alfabeto de símbolos de entrada. |
| Q | Conjunto finito de estados |
| q_0 | $q_0 \in Q$ – estado inicial previsto |
| | $F \subseteq Q$ - es el conjunto de estado finales de aceptación. |
| F | Función de transición de estados definida como $f: Q \times \Sigma \longrightarrow Q$ |

Transformación del Modelo de Proceso de Negocio a Automata Finito.

Las transformaciones se realizarán a través de la herramienta de transformaciones XLST, a partir del modelado de procesos de negocio en BPMN, en donde el analista deberá seleccionar aquellas actividades que se realizarán en forma manual, las que no serán mapeadas al SI. Para indicar esto utilizaremos el estereotipo “Manual” de BPMN. Cada una de las actividades de Negocio identificadas tendrán un mapeo directo con cada estado identificado del Automata finito. Lo mismo ocurrirá con los estados de Inicio y de finalización, ya sea por el éxito del procedimiento o por el fracaso del mismo. Los arcos del autómata finito surgirán a través de los flujos de trabajo que vinculan las Actividades del proceso de Negocio.

El esquema completo de transformaciones puede verse como un proceso tal cual se grafica a continuación en 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.

La capa siguiente corresponde a la transformación de modelos, mediante la herramienta de transformaciones XLST a partir de la cual obtendremos los archivos XML que es el formato de ingreso a la herramienta de simulación de Autómata finito.

A continuación en la Fig. 1 se explicita en forma gráfica el proceso de transformación automatizada desde el modelo del proceso de negocio en BPMN hasta la construcción del Autómata Finito.

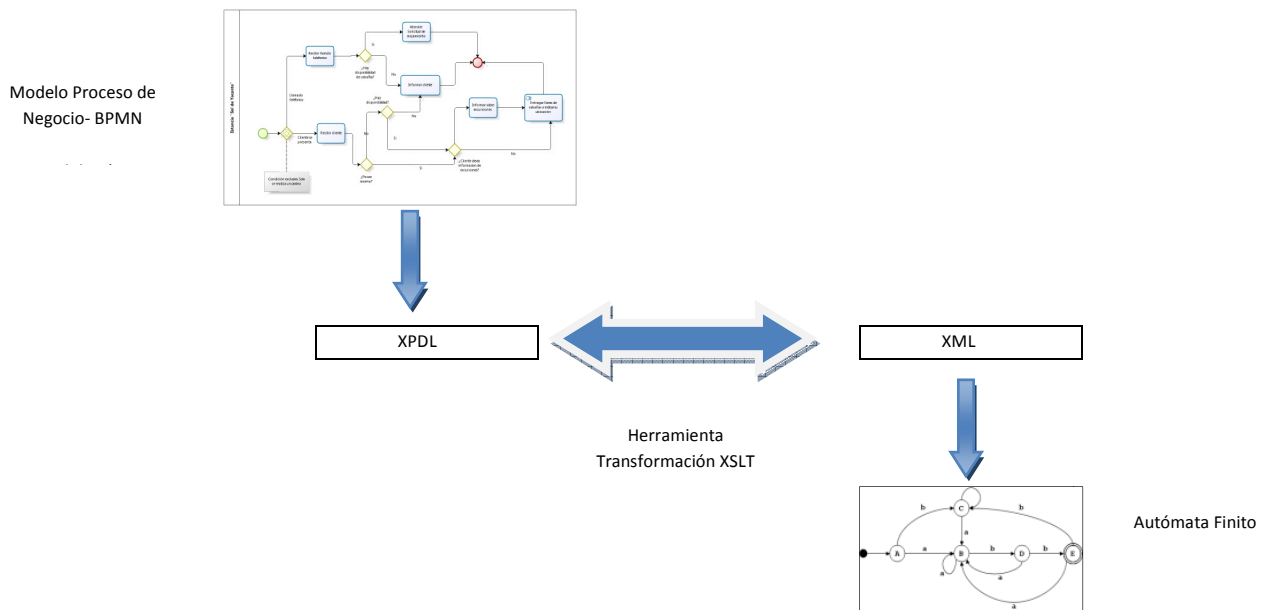


Figura 1

Construcción del Autómata Finito

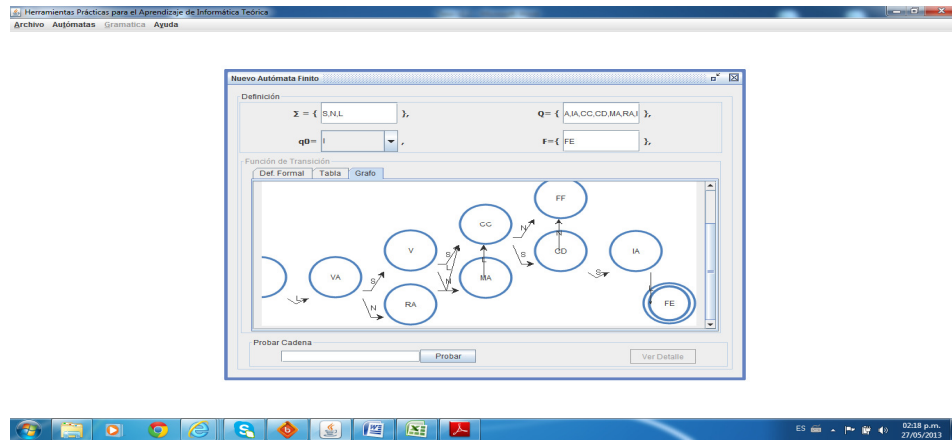
El Autómata Finito surge de un proceso de construcción sobre una abstracción realizada a partir de un proceso de Negocio, en donde las actividades de Negocio serán los estados del AF y los flujos entre los mismos serán las transiciones entre estados, contando de un estado inicial y uno de finalización por caso de éxito donde se logra el objetivo del Proceso de Negocio.

Adicionalmente, resulta sumamente útil contar con una representación gráfica de los mismos.

En la pantalla 1 se muestra la ejecución de la herramienta donde se realiza la representación de un grafo del Autómata

Finito en la herramienta de validación, en donde se pueden visualizar los estados y transiciones entre estados, y el estado inicial y de finalización. Otra forma es por medio de la definición formal de sus componentes, de todos los estados y las transiciones que se producen entre ellos, y además se puede disponer de una resumen de los mismos (Tabla) que relaciona estados / entradas.

A continuación se presenta la pantalla 1 en donde se muestra el grafo que representa el autómata finito generado a partir de un proceso de negocio en BPMN.



Pantalla 1

Validación efectuadas a los Automatas Finitos

Analizaremos ahora las validaciones y verificaciones a efectuar sobre los Automatas Finitos, teniendo en cuenta que cada uno de estos se obtiene a partir de una transformación directa de la representación del modelo de Proceso de Negocio que debe ser validado:

Conjunto Conexo y accesibilidad de estados: Esta verificaciones resultan fundamentales para verificar que todos y cada una de las abstracciones de estados por los que transita en AF tienen correlación con el planteo del mismo, ya que si un estado definido en el AF no es accesible desde el estado inicial, significa que el modelo que está siendo representado por el autómata no está correctamente planteado y debe necesariamente ser reformulado.

Autómata Finito Determinista: Esta validación que se efectúa sobre el Autómata Finito, no siendo la misma concluyente de su aplicación taxativa. La forma de definir tanto los modelos de procesos pueden resultar caminos o procesos paralelos o simultáneos, los cuales se traducen en no determinismo dentro de los Automatas Finitos, los cuales merecen una especial atención de su conveniencia en mantenerlos en los

modelos. La herramienta propone convertir el AF No determinista en uno Determinista equivalente, de manera de brindar al analista la posibilidad de analizar si se reformula el modelo o se mantiene tal como está definido.

Minimización del Autómata Finito: Al igual que el punto anterior la herramienta de validación de Máquinas abstractas, informará si el Autómata finito es posible minimizarlo con una cantidad mínima de estados. Este informe y la propuesta de minimización de la herramienta, deberá ser utilizado para analizar nuevamente el modelo y verificar la conveniencia o no de una nueva reformulación del mismo.

Un AF no mínimo significa la presencia de estados equivalentes, los cuales pueden ser identificados y reemplazados, y de esta manera simplificar el Modelo que representa al Proceso de Negocio. (En el proceso de Negocio dos estados equivalentes del AF equivale a la existencia de una reinvocación de una acción que puede ser eliminada)

Simulación de Ejecución de Automatas Finitos: Esta característica, es la que proporciona a la herramienta de validación, de una versatilidad y funcionalidad propia de un esquema de validación dinámica, ya que para cada modelo de proceso de Negocio y su

correspondiente representación del Autómata Finito, pueden establecerse un conjunto de entradas para los cuales simular las mismas en la herramienta su ejecución y verificar si produce los resultados esperados por el modelo.

Resultados

Este proceso de modelado y transformación entre modelos, resulta imprescindible para verificar y validar un Modelo Conceptual. Los Procesos de Negocios definidos a partir de BPMN y luego transformados a Autómatas Finitos respectivos, los cuales son sometidos a diferentes procesos de validación y simulación mediante una herramienta de validación de Autómatas Finitos, donde a partir de diferentes estímulos externos representados con diferentes entradas se utiliza para analizar la respuesta del modelo ante los mismos y de esta manera verificar si los procedimientos están representados por el modelo de manera correcta.

Para cada uno de los Procesos de Negocios representados por un Autómata Finito resultante del proceso de transformación se seleccionará un conjunto de entradas posibles y se verificará de acuerdo al proceso de Negocio que representa si cada una de estas debe producir la aceptación o no de la cadena de entrada en el Autómata Finito, en función a la respuesta esperada en el proceso de Negocio. Una correcta selección de las entradas a verificar serán las que conduzcan a validar que: toda secuencia de estímulos del exterior produzcan en el sistema una respuesta satisfactoria que se plasmará en el AF ya que al concluir en un estado de aceptación por parte de la máquina abstracta, y por otro lado, que toda entrada que lleve a un estado de finalización en el AF pero que no sea definido en el mismo como de no aceptación, será un conjunto de estímulos

no válidos y no soportados por el proceso de Negocio que representa al modelo Conceptual.

Discusión

Nos centraremos en analizar ahora la completitud del Proceso en la Validación de Modelos, a través de la definición del proceso de Negocios con BPMN, su transformación automatizada a máquina de estado, más precisamente Autómata finito y como los chequeos automatizados aplicados a este, contribuyen a la validación del modelo del proceso de negocio.

Las propiedades definidas, sobre la validación de Máquinas Abstractas: Autómatas Finitos, son consideradas completas en la validación de los mismos, ya que el único punto que no está presente en comparación de otras herramientas de model Checker, es el control de deadlock, o bloqueo mutuo, el cual no es necesario considerarlo ya que por la particularidad de la elección dentro de las máquinas Abstractas, como son los Autómatas Finitos, su definición, configuración y forma de funcionamiento no permiten que esta situación se plantee ya que las entradas son secuenciales y procesadas únicamente con desplazamiento en un solo sentido.

Conclusiones

Las validaciones efectuadas sobre los autómatas finitos que representan a un determinado modelo, a partir de procesos de negocio, resultan por demás ventajosas para validar aspectos imprescindibles de los mismos, que ayudarán al analista a alertar sobre posibles defectos en la construcción de los modelos.

Existe, una gran variedad de herramientas para validar modelos representados a través de abstracciones con máquinas de estados, pero las mismas no resultan de

aplicación directa, ya que o necesitan de una formalización a través de un lenguaje matemático que resulta poco comprensible por los analistas, expertos en el dominio del problema y posteriormente los desarrolladores, o no permiten en su aplicación la posibilidad de importar y construir los Autómatas finitos directamente de los modelos que se va a representar.

Es por tal motivo que la herramienta de Validación de Autómatas Finitos resulta fundamental en el proceso de validación de los modelos construidos y que forman parte del modelo conceptual.

Referencias:

[Gibs 1994] Gibs Wait. "La crisis crónica de la programación" Revista Investigación y ciencia N° 218 Dialnet 1994. págs. 72-81.

[Insfrán 2002] E. Insfrán, I. Díaz, M. Burbano, *Modelado de Requisitos para la Obtención de esquemas conceptuales*. Disponible en: <http://www.dsic.upv.es/~einsfran/papers/39-ideas2002.pdf> Fecha Consulta: 02/10/06

[Insfrán 2002b] E. Insfrán, E. Tejadillos, S. Marti, M. Burbano, *Transformación de Especificación de requisitos en esquemas conceptuales usando Diagramas de Interacción*. Disponible en: [www.inf.puc-rio.br/~wer02/zip/Transformacion_Espec\(7\).pdf](http://www.inf.puc-rio.br/~wer02/zip/Transformacion_Espec(7).pdf) Fecha Consulta: 04/12/07

[Boehm 2001] B. Boehm, V.R. Basili, *Software defect reduction top 10 list*. IEEE Computer, 01/01/01

[Sesé 2006] F. Sesé Muniátegui, Tesis Doctoral: *Propuesta de un método de validación de esquemas conceptuales y análisis comparativo de la noción de información en los métodos de desarrollo de Sistemas de información* Disponible en: www.tesisenxarxa.net/TDX-0517107-131929/ Fecha consulta: 20/05/08

[Sommerville 1997] I. Sommerville, P. Sawyer, *Requirements Engineering: A Good Practice Guide*. Computing Department, Lancaster University, John Wiley & Sons Ltd. ISBN 0 471 974444 7, 1997.

[Sommerville 2005] I. Sommerville, *Ingeniería del Software*. ISBN 9788478290741, Pearson Educación.

[Sommerville 2011] Ian Sommerville *Ingeniería de Software* Editado por Pearson Educación – México 2011 Versión impresa ISBN 978-607-32-0603-7

[Letelier 1999] P. Letelier, P. Sanchez, I. Ramos http://www.researchgate.net/publication/36720988_Un_ambiente_para_especificaciones_incremental_y_validacin_de_modelos_conceptuales/file/d912f50ca20c33f5e5.pdf.

Fecha de consulta web: 12 de marzo de 2013.

[OMG 2005] Object Management Group. Unified Modelling Language: Superstructure Version 2.0 (online), Julio 2005, <http://www.omg.org>

[OMG 2007] 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.

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

[OMG 2009] 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.

[Pons 2010] C. Pons, R. Giandini, G. Pérez. *Desarrollo de Software dirigido por modelos – Conceptos Teóricos y su aplicación práctica* Editorial Universidad Nacional de la Plata 1ra edición 2010.

[Taylor 1998] Taylor-Cummings, A., 1998. Bridging the user-IS gap: a study of major information systems projects. *Journal of Information Technology*, 13, 29-54.

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

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

[White 2011] Stephen A. White, PHD, *Derek Wiers, BPMN Guía de Referencia y Modelado - BPMN 2.0 Handbook* Copyright © 2011 by Future Strategies Inc ISBN-13: 978-0-9819870-7-1 (Digital Edition).