

Desarrollo de Aplicaciones usando el enfoque de Sistemas de Información centrado en Procesos y patrones del paradigma Acción Comunicativa

D. Ortega¹, M. Guevara² y I. Durán³

(1, 2, 3)Universidad de Carabobo, Venezuela

RESUMEN

La presente investigación aborda el desarrollo de una Aplicación de flujo de Trabajo bajo el enfoque PAIS (Process-Aware Information Systems), incorporando el uso de patrones del paradigma Acción Comunicativa (en Inglés, Language Action Perspective) con el fin de comprender y mostrar cómo dos o más roles interactúan y establecen acuerdos. En esta investigación se utiliza como estudio de caso el proceso "Registro y Seguimiento del Trabajo Especial de Grado del Departamento de Computación de la Facultad Experimental de Ciencias y Tecnología de la Universidad de Carabobo", el cual inicialmente se ha especificado en BPMN. Para el desarrollo de la Aplicación de flujo de Trabajo se elabora una especificación de procesos empleando YAWL (Yet Another Workflow Language), el cual permite su ejecución posterior, facilitando así la implementación. La metodología de desarrollo de la Aplicación es una instanciación de RUP (Rational Unified Process).

Palabras Claves: Sistemas de Información Centrados en Procesos, Paradigma de Acción Comunicativa, Yet Another Workflow Language (YAWL).

ABSTRACT

This investigation focuses on the development of a workflow application under PAIS (Process-Aware Information Systems) approach, incorporating the use of patterns of communicative action paradigm (Language Action Perspective) to understand and show how two or more roles interact and reach agreements. This research used the process "Registering and Monitoring of the Degree Project in Computer Science Department from the Sciences and Technology Faculty at the University of Carabobo" as case study. For the development of the workflow application is made using a YAWL (Yet Another Workflow Language) process specification, which allows later execution, facilitating the implementation. The development methodology for the application is an instantiation of RUP (Rational Unified Process).

Keywords: Process-Aware Information Systems (PAIS), Language Action Perspective (LAP), Yet Another Workflow Language (YAWL).

1. Introducción

En la actualidad, el uso de tecnologías dentro de las organizaciones juega un papel fundamental para lograr un desempeño deseable. En general, el uso de sistemas de software y Tecnología de Información y Comunicación (TIC's) proporciona un aumento de la eficiencia y productividad de los procesos del negocio.

En este sentido, la construcción de sistemas centrados en procesos (en inglés *Process-Aware Information Systems*, PAIS) proporciona una especificación de fácil comprensión, que representa la realidad de la organización, favoreciendo el desarrollo de sistemas que provean un mayor soporte a sus usuarios [Rus07].

Uno de los aspectos a propiciar, es la comunicación entre los distintos actores involucrados en los procesos del

negocio. Para este fin, existen diversos paradigmas que estudian la identificación de las acciones entre los actores del sistema al momento de describir los procesos, como el paradigma Acción Comunicativa que modela las interacciones comunicativas del negocio [Win87].

Así mismo, con el uso masivo de la Internet, se pone en evidencia la limitación de las Aplicaciones de Flujo de Trabajo en el soporte a la comunicación entre los usuarios, creando una necesidad que ha sido satisfecha con el uso de la Web 2.0, tomando en cuenta, su alta facilidad de integración con los sistemas existentes y la garantía de una comunicación más directa [Gar10].

Adicionalmente, el uso adecuado de un lenguaje de especificación de procesos, es una herramienta esencial para la construcción de PAIS. Entre los lenguajes de especificación se cuenta *Yet Another Workflow Language* (YAWL), el cual ofrece soporte a un mayor número de patrones de

Flujo de Trabajo e información, en comparación con *Business Process Management Notation* (BPMN) y *Unified Modeling Language* (UML), [Bol10, Yaw10].

Por todo lo anterior, se justifica la realización de este trabajo, el cual aborda el desarrollo de una Aplicación de Flujo de Trabajo bajo el enfoque PAIS, mediante una especificación en YAWL del proceso “Registro y Seguimiento del Trabajo Especial de Grado del Departamento de Computación de la Facultad Experimental de Ciencias y Tecnología de la Universidad de Carabobo”, incorporando patrones del paradigma Acción Comunicativa con el fin de comprender y mostrar cómo dos o más roles interactúan y realizan acuerdos. Vale la pena destacar que en esta investigación se partió de una especificación en BPMN del proceso, antes mencionado [Bol10].

2. Sistemas de Información Centrados en Procesos

En [vdA09] se propone que la Gestión de Procesos de Negocios (*en inglés, Business Process Management, BPM*) comprende un rango muy amplio de tecnologías que pueden ser aplicadas para la gestión de procesos de negocio y la implementación de Sistemas de Información Centrados en Procesos que incluyen Aplicaciones de Flujo de Trabajo especializadas para la organización.

Los PAIS se pueden definir como sistemas que administran y ejecutan los procesos, involucrando a las personas, aplicaciones y fuentes de información, representados con modelos de procesos [Rus07].

Por último, los PAIS emergen por su flexibilidad al momento de modificar su lógica desde el nivel de especificación hasta la implementación del mismo, y así dar respuestas a la incertidumbre, situaciones excepcionales y los cambios en el contexto de la organización que ocurren comúnmente [vdA09].

3. Yet Another Workflow Language (YAWL)

van der Aalst, y ter Hofstede [Yaw10], desarrollaron un nuevo lenguaje de flujo de trabajo denominado YAWL, basado en Redes de Petri y de característica *Open Source*, especificado a través de una teoría de concurrencia bien establecida, con una representación gráfica simple, y soporte para un conjunto extenso de patrones de flujo de trabajo. En la Figura 1 se visualiza un ejemplo de una red YAWL, correspondientes a un segmento de la especificación del subproceso para el registro de Tutores de un Trabajo Especial de Grado (TEG).

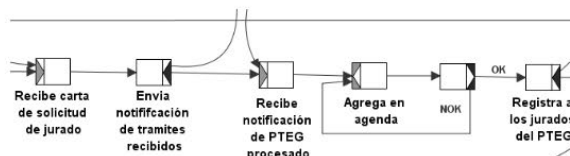


Figura 1: Segmento de una especificación en YAWL para registro de Tutores de un TEG.

En la figura 2, se observa la notación de YAWL con una descripción breve de cada uno de los elementos. Además de ser un lenguaje visual, se resaltan características que permiten realizar el análisis de la especificación de procesos para la corrección y obtención de un buen modelo. Asimismo, la activación de varias instancias en tiempo de ejecución, mediante constructores adicionales a las Redes de Petri y un manejo de archivos XML, *XPath* y *XQuery* para la consulta e implementación preliminar de modelos de negocio.

Símbolo	Nombre	Descripción
	Condición de Entrada	Marca el inicio de un flujo de trabajo.
	Condición de Salida	Marca el fin de un flujo de trabajo.
	Condición	Representa una condición, que es una manera de representar el estado de la red.
	Tarea Atómica	Representa a una sola tarea a realizar, por lo general por un participante humano o una aplicación externa o servicio.
	Tarea Compuesta	Es un contenedor de otra sub-red YAWL, con su propio conjunto de elementos YAWL, restringidos por la misma sintaxis.
	Múltiple Instancia de una Tarea Atómica	Permite ejecutar varias instancias de una tarea al mismo tiempo.
	Múltiples Instancias de una Tarea Compuesta	Permite ejecutar varias instancias de una tarea compuesta al mismo tiempo.
	AND Join	Especifica que la tarea se ejecutará sólo cuando todas sus tareas precedentes se hayan completado.
	XOR Join	Especifica que la tarea se ejecutará una vez que una de las tareas precedentes se complete.
	OR Join	Especifica que la tarea se ejecutará sólo cuando todas las tareas previamente activadas se completen.
	AND Split	Una vez completada la tarea, todas las tareas subsiguientes podrán ejecutarse.
	XOR Split	Consiste en que una vez completada la tarea, sólo una tarea subsiguiente podrá ejecutarse.
	OR Split	Establece que una vez completada la tarea, una o más tareas subsiguientes podrán ejecutarse.

Figura 2: Simbología de los constructos pertenecientes al lenguaje YAWL.

4. Paradigma Acción Comunicativa

Acción Comunicativa (*en inglés, Language Action Perspective, LAP*), es un enfoque de diseño y construcción de sistemas en el cual se asume que la comunicación conlleva una acción, basada en la creación de compromisos entre las partes. Esta acción puede ocurrir a través de actos de habla o comunicativos.

LAP está basado en las teorías de actos de habla de Searle [Sea69] y actos comunicativos de Habermas [Hab81]. Es el resultado de una serie de contribuciones teóricas de [Flores & Ludlow 1980, Winograd & Flores 1986] citados por [Win87], [Goldkuhl & Lytinen 1982, 1984] citado por [Gol96]; y propone evidenciar las interacciones comunicativas entre diferentes actores de un sistema.

Varios autores han escrito acerca de cómo modelar las interacciones comunicativas [Rit06]. En particular Rittgen [Rit07] propone los conceptos de un lenguaje de modelado genérico que pueden ser utilizados para la extensión de

lenguajes de modelación de sistemas de información, tales como UML (Unified Modeling Language)[MH06, pp. 108-130], el lenguaje de SIMM (Information systems Modeling Method) [Gol96, Rit06], entre otros.

Con el fin de alcanzar una mejor interpretación de la comunicación, existen diversos patrones dentro del enfoque LAP, tal como *Action Workflow* [MWFF92, Sch01], el cual se considera una herramienta del *framework* con el mismo nombre y es utilizado para el registro y análisis de la comunicación, con el propósito de entender y optimizar cómo dos o más partes, transmiten y reciben información, estableciendo la coordinación de sus actividades y creando una realidad común por medio del lenguaje.

En este caso, el patrón *Action Workflow* se ha descrito en el Catálogo de Eriksson & Penker usando diagramas de secuencia del estándar UML [EP00, pp. 329-335]. Dicho patrón se basa en el Modelo de cuatro fases: Preparación, Negociación, Realización y Aceptación [MWFF92]; y establece dos roles fundamentales, el emisor y el receptor. En la fase de **Preparación**, el emisor elabora una petición, para iniciar la comunicación con el receptor. En la fase de **Negociación** el emisor y receptor negocian las condiciones hasta establecer acuerdos. En la fase de **Realización** se realizan el conjunto de actividades que transforman la comunicación en acción. En la fase de **Aceptación** es cuando el emisor y el receptor indican su conformidad con la realización. En este momento, es posible continuar con una nueva instancia del Modelo.

Por tanto, la elaboración de diagramas de secuencia, facilita el análisis de conceptos claves del paradigma como son las interacciones comunicativas entre emisor-receptor, y a qué fase pertenece. En la Figura 3, se muestra la aplicación del patrón al subproceso de revisión del Proyecto de Trabajo Especial de Grado (PTEG) perteneciente al estudio de caso, especificando la comunicación entre un Estudiante (emisor) y su Tutor (receptor).

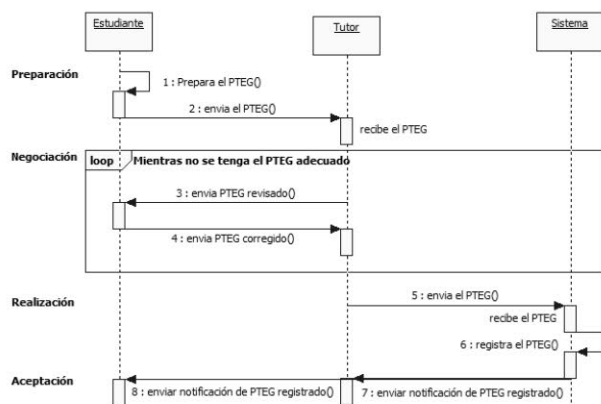


Figura 3: Análisis de las interacciones para un proceso de revisión de PTEG.

5. Metodología de Desarrollo de la Aplicación

Durante la investigación, se utiliza la metodología de desarrollo, el Proceso Unificado de Rational (en inglés, *Rational Unified Process*, RUP) de [Kru00]. RUP, es un proceso de ingeniería de software que integra los aspectos de

gestión y desarrollo a considerar a lo largo del ciclo de vida del software. En particular la disciplina de Análisis y Diseño se enriquece con la utilización del patrón *Action Workflow* del Paradigma LAP, para estudiar la situación actual de los procesos y proponer los cambios que promuevan la acción comunicativa entre los involucrados. La Instanciación de RUP se muestra en la Figura 4.

6. Resultados

En esta sección se describen los resultados más importantes de la aplicación de una instanciación de RUP al estudio de caso, éste es, el proceso: “Registro y Seguimiento del Trabajo Especial de Grado de la Facultad Experimental de Ciencias y Tecnología de la Universidad de Carabobo”, el cual es complejo, extenso, y cuyo funcionamiento está regido por una normativa legal.

6.1 Fase Inicio: Actividades de Análisis y Diseño

En esta investigación se parte de una especificación de procesos realizada en BPMN, por lo cual se necesita realizar una transformación previa al lenguaje de especificación y ejecución YAWL, aplicando de forma manual el proceso descrito en [YS10], obteniendo así, la Red YAWL inicial.

Dicha Red es analizada aplicando el patrón *Action Workflow* del Paradigma LAP, con el fin de identificar las interacciones comunicativas entre los actores involucrados candidatas a ser automatizadas y redefinidas para mostrar con mayor exactitud las interacciones que suceden en la realidad. De esta manera, la especificación inicial del proceso puede presentar variaciones para reflejar estos aspectos. en una red YAWL posterior.

En este sentido, se procede a identificar Subprocesos, tomando en cuenta las siguientes consideraciones:

Usualmente en una Red YAWL se aprecia que si un flujo de trabajo parte desde una actividad, recorre varias actividades y culmina en otra actividad que en conjunto representan un aspecto u objetivo del negocio, entonces se trata de un objetivo o necesidad que debe ser satisfecha por el uso del sistema. Eso corresponde con un Subproceso.

Cuando se observa la especificación YAWL del proceso Registro y Seguimiento del Trabajo Especial de Grado del Departamento de Computación de la Facultad Experimental de Ciencias y Tecnología de la Universidad de Carabobo; y por ejemplo, si se analiza el segmento de la red que parte desde la actividad “Realizar solicitud de tutores” por parte del estudiante, recorre otras actividades y culmina en “Registra tutores”, se concluye que la solicitud de tutores se ha logrado, y por tanto ese segmento de la Red corresponde al Subproceso Solicitud de Tutores (Ver Figura 5).

Este razonamiento se aplica a los distintos segmentos de la especificación que constituyen Subprocesos, y se obtiene un conjunto de objetivos o necesidades que deben ser satisfechos por el uso de la aplicación. Este resultado puede ser un insumo para contribuir a la actividad de encontrar los requisitos funcionales.

INICIO		ELABORACIÓN		CONSTRUCCIÓN		TRANSICIÓN	
MODELACIÓN DE NEGOCIO		ANÁLISIS Y DISEÑO		IMPLEMENTACIÓN		PRUEBA	
Actividades	Artefactos	Actividades	Artefactos	Actividades	Artefactos	Actividades	Artefactos
Describir el Contexto del Negocio	Contexto del Negocio	Especificar Red YAWL, basado en el Análisis LAP	Red YAWL definitiva del proceso	Implementar Perspectiva de Datos	Archivo XML con la especificación de las variables	Elaborar Manual	Manual de Usuario
Estudiar la especificación de procesos en BPMN	Lista de Actividades y Roles del Negocio	Modelar Datos	Variables de Tipo Complejo	Implementar expresiones XQuery	Archivo XML con las entradas y salidas de cada Tarea	Realizar pruebas	Evaluación de la flexibilidad
	Diagrama BPMN coloreado		Variables de la Red				
REQUISITOS		MODELACIÓN DE SERVICIOS		Implementar flujo de control	Archivo XML con la lógica de decisiones		
Actividades	Artefactos	Actividades	Artefactos				
Elaborar el Documento de Requisitos	Documento textual con los requisitos	Elaborar Documento de Arquitectura de Software	Documento con las características principales del diseño del software	Parametrización de las formas dinámicas	Archivo XML con modificación de las variables y tareas		
				Implementar Página Web	Archivo HTML para la interfaz de usuario		
ANÁLISIS Y DISEÑO							
Actividades	Artefactos						
Transformar especificación BPMN a YAWL	Red YAWL del proceso actual						
Aplicar Análisis LAP al Diagrama BPMN	Diagramas de Secuencias de las interacciones comunicativas del proceso						

Figura 4: Instanciación de RUP aplicada en la investigación.

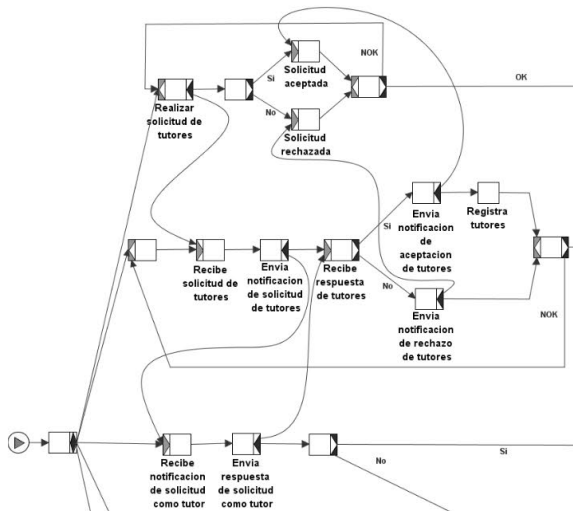


Figura 5: Especificación YAWL de las actividades (tareas) relacionadas al Subproceso de Solicitud de Tutores.

Para aplicar el enfoque LAP en el análisis y diseño de Aplicación de Flujo de Trabajo del caso en cuestión, primero se elabora un diagrama de secuencia de la situación actual del Subproceso, luego se procede a especificar las diversas interacciones entre los actores involucrados realizando otro diagrama de secuencia, siguiendo las fases de preparación, negociación, realización y aceptación de las interacciones comunicativas entre los involucrados, conforme a lo indicado en el patrón *Action Workflow*. En el

ejemplo del Subproceso antes mencionado, se elaboran manualmente un diagrama de secuencia del Subproceso de Solicitud de Tutores para la situación actual (ver Figura 6) y otro diagrama después de aplicar el patrón (ver Figura 7).

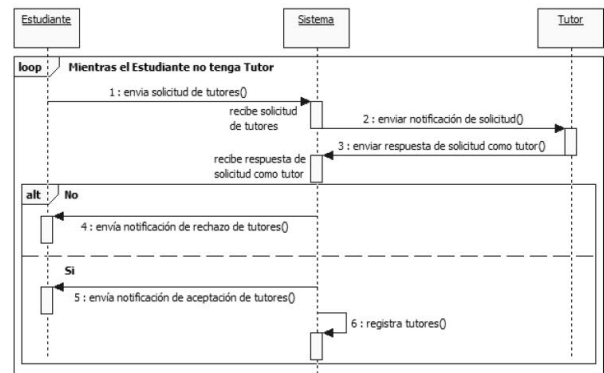


Figura 6: Caso actual del Subproceso de Solicitud de Tutores.

Como resultado del análisis LAP al Subproceso Solicitud de Tutores, se identifica a través de la fase negociación, un conjunto de interacciones entre el estudiante y el profesor (posible tutor), que no es considerado en la situación actual, pero juega un papel crucial y de gran importancia para la solicitud de tutores.

Para casos más complejos, a través de las cuatro fases del patrón, no solamente se verifica que se especifican las actividades correspondientes, sino también permite diseñar una aplicación más cercana a la realidad de la organización, al tomar la comunicación entre actores.

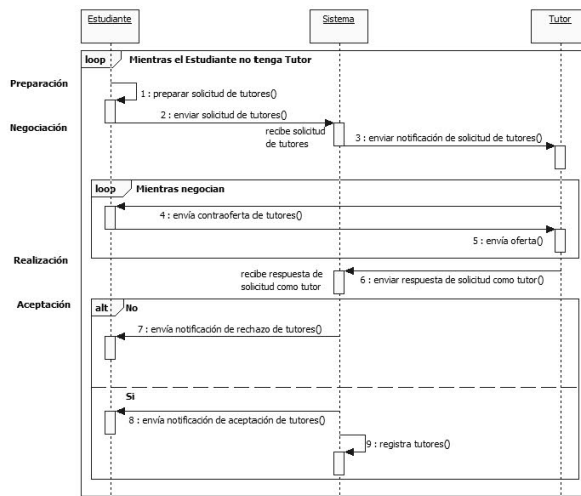


Figura 7: Subproceso de Solicitud de Tutores luego de aplicar el patrón Action Workflow.

6.2 Fase Elaboración: Actividades de Análisis y Diseño

Posterior a la obtención de los diagramas de secuencia después de aplicar el análisis LAP usando el patrón Action Workflow, se procede a especificar la red YAWL definitiva que permitirá la implementación de la aplicación. Para esto, adicionalmente se utilizan la red de YAWL inicial resultante de la transformación del diagrama BPMN a YAWL.

6.3 Fase Construcción: Actividades de Implementación de la Aplicación

A continuación, se describen las actividades a realizar para el funcionamiento de la aplicación mediante YAWL.

Perspectiva de Datos

Esta perspectiva se basa en los estándares XML, XQuery y XPath. En particular, los datos están representados como documentos XML donde para cada variable de tipo complejo, se especifica su estructura. En la Figura 8, se muestra el código para definir la variable Profesor.

```
<xs:complexType name="Profesor">
  <xs:sequence>
    <xs:element name="Nombre" type="xs:string" />
    <xs:element name="Institucion" type="xs:string" />
    <xs:element name="TelF" type="xs:string" />
    <xs:element name="Email" type="xs:string" />
  </xs:sequence>
</xs:complexType>
```

Figura 8: Código XML para la variable Profesor.

Expresiones XQuery

Para el funcionamiento de cada tarea y de su interfaz, es esencial programar las rutas de acceso y salida (Input y Output) de cada una de las variables que participan a través

de la definición de expresiones XQuery. En la Figura 9 se muestra la definición para la tarea Realiza solicitud de Tutores.

```
Input
<Tutores>{/Red/Tutores/*}</Tutores>
<Area>{/Red/Area/text ()}</Area>

Output
<Area>
{/Realiza_solicitud_de_tutores_/Area/text ()}
</Area>
<Comentario>
{/Realiza_solicitud_de_tutores_/Comentario/text ()}
</Comentario>
<Tutores>
{/Realiza_solicitud_de_tutores_/Tutores/*}
</Tutores>
```

Figura 9: Expresiones XQuery para la tarea Realiza solicitud de Tutores.

Flujo de Control

Existen diversas actividades con constructos Split-XOR expresando la posibilidad de tener más de un flujo de trabajo, que dependiendo del valor o estado de una variable se ejecuta una sola ramificación. Para identificar el flujo a ejecutar, se necesita establecer una condición XPath por cada flujo candidato. En la Figura 10 se muestran las condiciones XPath para la tarea Recibe evaluación del TEG.

```
Destino 1
(Envía notificación de corrección de TEG)
#{/Red/Evaluacion-TEG/TEG-Aprobado/text ()='false'}

Destino 2
(Envía evaluación de TEG)
#{/Red/Evaluacion-TEG/TEG-Aprobado/text ()='true'}
```

Figura 10: Condiciones XPath para la tarea Recibe evaluación del TEG.

Parametrización de las formas dinámicas

La composición y el diseño de los formularios de interfaces de usuario en YAWL son básicos. Sin embargo, con el uso de atributos extendidos se puede desarrollar una versión personalizada de la interfaz, ofreciendo algunas facilidades.

En este sentido, es posible con YAWL realizar la modificación de atributos relacionados para cada formulario web. En la Figura 11 se muestra la parametrización de la tarea Recibe PTEG revisado del Estudiante.

Página Web del Sistema

Para el desarrollo de las páginas web, se utiliza JSP (Java Server Pages) y CSS (Cascading Style Sheets). Desde la página *index.jsp*, se permite el ingreso de un usuario al sistema y desde la página *proceso.jsp* se muestra la lista de trabajo, la cual contiene todas las tareas pendientes.

```

background-alt-color: #3366FF
font-color: #333333
header-font-color: #FFFFFF
hideBanner: true
label: "ATENCIÓN: Usted ha recibido una notificación"
justify: left
page-background-image:
http://localhost:8080/images/fextest.jpg
title: null
Atributos de la Tarea

hide: true
Atributos de variables
Correcciones
T1
hide: true
text-below: ">>Respecto al Planteamiento:
#{/Recibe_PTEG_revisado_JE/Correcciones/Planteamiento/text()}"
T2
hide: true
text-below: ">>Respecto a los Antecedentes:#{
{/Recibe_PTEG_revisado_JE/Correcciones/Antecedentes/text()}"

```

Figura 11: Parametrización de la tarea Recibe PTEG revisado.

6.4 Fase Transición: Actividad Pruebas de Flexibilidad

En el área de modelado y especificación de procesos de negocios, se describe la flexibilidad como la capacidad de un proceso de negocio de mantener el objetivo para el cual se diseñó, a pesar de los cambios que puedan ocurrir, los cuales pueden impactar la estructura o secuencia del mismo. Inclusive, se considera la posibilidad de realizar cambios en su especificación, si es necesario [DN07].

Los cambios en la estructura se refieren a las variaciones y eventualidades observadas posterior al diseño/del modelo del proceso. Es decir, mientras el proceso está en ejecución se detecta una variación no esperada, por lo que se hace necesario para satisfacer las necesidades, un rediseño u optimización del mismo. Esto se conoce como Flexibilidad en tiempo de ejecución.

En cuanto a los cambios en la secuencia, ocurre particularmente en procesos donde se conoce con anterioridad cuáles son las variaciones que podrían afectarlo durante su ejecución, por lo que se tiene la previsión de especificar en tiempo de diseño los distintos caminos alternativos para responder y terminar su flujo de manera satisfactoria y evitar la finalización abrupta del proceso. Este concepto se denomina Flexibilidad en tiempo de diseño.

Basados en los conceptos de tipos de flexibilidad anteriormente planteados, en [SMR*08] se presenta una taxonomía más detallada para evaluar la flexibilidad de un proceso: diseño, desviación, bajo especificación y cambios. La Flexibilidad por diseño, evalúa la capacidad de incorporar caminos alternativos de ejecución en un modelo de proceso en tiempo de diseño. La Flexibilidad por desviación, evalúa la Posibilidad de que una instancia de proceso se desvíe en tiempo de ejecución de la ruta de flujo de trabajo prevista por el proceso original, sin alterar su modelo. La Flexibilidad bajo especificación, mide la capacidad de ejecutar un modelo de proceso incompleto en tiempo de ejecución. La Flexibilidad por cambio, mide la posibilidad de modificar un modelo de proceso en tiempo de ejecución de tal manera que una o todas las instancias del

mismo, emigran al nuevo modelo.

Después de la evaluación de la Aplicación con la taxonomía presentada en [SMR*08], se observa en la Figura 12, cómo la Aplicación satisface un 100% de la Flexibilidad por diseño, un cumplimiento parcial de 62,5% de la Flexibilidad por cambio, un 50% para la Flexibilidad bajo especificación. Sin embargo, no se promueve la Flexibilidad por desviación, debido a que YAWL no soporta la realización de operaciones fuera de lo establecido por la especificación.

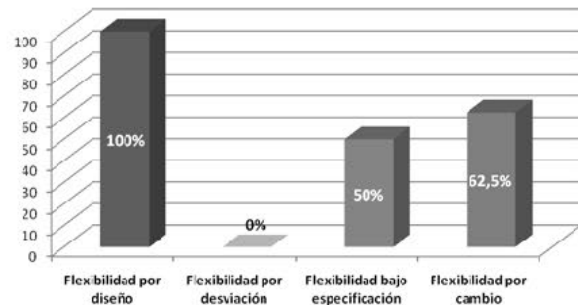


Figura 12: Porcentaje de Flexibilidad por enfoques.

En la Figura 13 se observa como el 67% de los aspectos de flexibilidad evaluados son promovidos por la aplicación, en contraste con un 33% no encontrado.

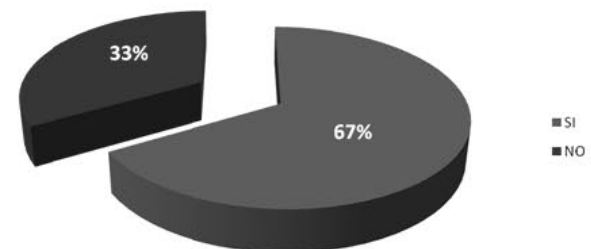


Figura 13: Porcentaje de criterios flexibles de la taxonomía.

7. Conclusiones

La Aplicación de Flujo de Trabajo desarrollada en esta investigación, automatiza el proceso "Registro y Seguimiento del Trabajo Especial de Grado del Departamento de Computación de la Facultad Experimental de Ciencias y Tecnología de la Universidad de Carabobo". En este trabajo, se utiliza el patrón *Action Workflow* del Paradigma *Language Action Perspective* para el análisis y diseño de Aplicaciones de Flujo de Trabajo, permitiendo incorporar a la especificación de procesos, aspectos de soporte a la comunicación entre los usuarios involucrados, tomando en cuenta sus interacciones comunicativas.

Se valida el uso de YAWL como lenguaje de especificación y ejecución de procesos de negocio. En un primer aspecto soporta la mayoría de los Patrones de Workflow. En un segundo aspecto, permite la elaboración de diagramas de procesos con menor cantidad de elementos. Y por

último, no se requiere de la utilización de otros lenguajes de programación para el desarrollo de aplicaciones. Sin embargo, se reconoce a YAWL como un lenguaje, que está más dirigido a los programadores y analistas del sistema que a los diseñadores o modeladores de procesos de negocios.

Por último, la Aplicación de Flujo de Trabajo desarrollada muestra los aspectos de flexibilidad propiciados por el uso de YAWL, favoreciendo la modificación de componentes de software, con poco esfuerzo.

Agradecimientos

Los autores agradecen al Consejo de Desarrollo Científico y Humanístico Universidad de Carabobo (CDCH-UC) el financiamiento de esta investigación, según subvención Nro.0388 del 2010.

Referencias

- [Bol10] BOLAÑOS L. *Modelación de Procesos de Negocio Orientada a Servicios Web. Caso de estudio: Registro y Seguimiento del Trabajo Especial de Grado del Departamento de Computación - FACYT*. Universidad de Carabobo, Noviembre 2010.
- [DN07] DAODI F. & NURCAN S. A benchmarking framework for methods to design flexible business processes. *Software Process: Improvement and Practice* 12, 1 (2007), 51-63.
- [EP00] ERIKSSON H. E. & PENKER M. *Business Modeling with UML: Business Patterns at Work*. New York: John Wiley & Sons, Inc. 2000.
- [Gar10] GARTNER. Gartner Says Social-Networking Services to Replace E-Mail as the Primary Vehicle for Interpersonal Communications for 20 Percent of Business Users by 2014. Retrieved July 02, 2011, from Gartner: <http://www.gartner.com/it/page.jsp?id=1467313>, November 11 2010.
- [Gol96] GOLDKUHL G. Generic Business Frameworks and Action Modelling. *International workshop on the Language Action Perspective on Communication Modelling*, (1996), 1-17.
- [Hab81] HABERMAS J. *The theory of communicative action*. Book (Vol. 1). Beacon Press, (1981).
- [Kru00] KRUCHTEN P. *The Rational Unified Process - An Introduction*. Addison-Wesley- Longman, 2000.
- [MWFF92] MEDINA-MORA R., WINOGRAD T., FLORES R., & FLORES F. The action workflow approach to workflow management technology. *The Information Society*, 9, 4, (1992), 391-404.
- [MH06] MILES R. & HAMILTON K. *Learning UML 2.0*. O'Reilly Sebastopol CA, 2006.
- [Rit06] RITTGEN P. Co-Evolution of Business Action Theories and Languages. In *Proc. of the 4th International Conference in Action in Language, Organisations and Information Systems-ALOIS*, (2006), 3-16.
- [Rit07] RITTGEN P. Towards a Language for Business Action Theory. In *Proc. European Conference on Information Systems- ECIS*, (2007), 1977-1989.
- [Rus07] RUSSELL N. C. *Foundations of process-aware information systems. Language. PHD Thesis*, Queensland University of Technology. Retrieved June 14, 2011, from <http://eprints.qut.edu.au/16592/>. December 2007.
- [SMR*08] SCHONENBERG H., MANS R., RUSSELL N., MULYAR N., & VAN DER AALST W. Process flexibility: A survey of contemporary approaches. *Advances in Enterprise Engineering I*, (2008), 16-30.
- [Sea69] SEARLE J. Speech Act Theory. *Philosophy* (1969), 221- 239.
- [Sch01] SCHOOP M. An Introduction to the Language-Action Perspective. *ACM SIGGROUP Bulletin* 22, 2 (2001), 3-8.
- [vdA09] VAN DER AALST W.M.P. Process-aware information systems: Lessons to be Learned from Process Mining. *Transactions on Petri Nets and Other Models of Concurrency II*, (2009), 1-26.
- [Win87] WINOGRAD T. A language/action perspective on the design of cooperative work. *Human-Computer Interaction* 3, 1 (1987), 3-30.
- [Yaw10] YAWL FOUNDATION. *YAWL - User Manual*. Version 2.1, 2010.
- [YS10] YE J. & SONG W. Transformation of BPMN Diagrams to YAWL Nets. *Journal of Software* 5, 4 (2010), 396-404.