

**BÚSQUEDA SEMÁNTICA EN UN REPOSITORIO DE PROCESOS
DE NEGOCIO**



**DANIEL FELIPE RIVAS BURBANO
DAVID SANTIAGO CORCHUELO CASTRO**

Director: Dr. JUAN CARLOS CORRALES

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE TELEMÁTICA
POPAYÁN
2011**

**BÚSQUEDA SEMÁNTICA EN UN REPOSITORIO DE PROCESOS
DE NEGOCIO**



**DANIEL FELIPE RIVAS BURBANO
DAVID SANTIAGO CORCHUELO CASTRO**

**Monografía presentada para optar al título de Ingeniero en
Electrónica y Telecomunicaciones**

Director: Dr. JUAN CARLOS CORRALES

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE TELEMÁTICA
POPAYÁN
2011**

NOTA DE ACEPTACIÓN

PRESIDENTE DEL JURADO

JURADO

Popayán, 2011

Agradecimientos

Queremos expresar nuestros más sinceros agradecimientos a Dios, por brindarnos la oportunidad de vivir este desafío y por ser el principal motor de nuestra existencia. A nuestros padres, familiares y amigos por su apoyo y amor incondicional. A nuestros docentes por sus enseñanzas. A nuestro tutor, el Dr. Juan Carlos Corrales por su disposición permanente durante la elaboración de nuestro trabajo de grado. Al Ing. Cristhian Figueroa por su motivación y dirección en la ejecución de este trabajo. A la Dra. Rosalba Giugno por su participación como coautora en la edición de una publicación de la investigación. Al Diseñador Gráfico Andrés Fabián Arias por su participación en el diseño de la aplicación software elaborada y finalmente a la Universidad del Cauca, nuestra Alma Mater. En especial a la Facultad de Ingeniería Electrónica y Telecomunicaciones que en la proximidad de sus 50 años de labores proveyó las circunstancias para nuestra formación tanto personal como profesional.

Atentamente,

Daniel Felipe Rivas Burbano

David Santiago Corchuelo Castro

Tabla de Contenido

1	INTRODUCCIÓN	9
1.1	Contexto General.....	9
1.2	Declaración del Problema	10
1.3	Escenario de motivación	11
1.4	Contribuciones	11
1.5	Alcance	12
1.6	Contenido de la monografía.....	12
2	ESTADO DEL ARTE.....	14
2.1	Contexto General.....	14
2.1.1	Procesos de Negocio	14
2.1.2	Lenguajes de Modelado de Procesos de Negocio.....	15
2.2	Trabajos Relacionados	20
2.2.1	Repositorios de Servicios Web.....	20
2.2.2	Repositorios de Procesos de Negocio	21
2.3	Resumen.....	28
3	REPOSITORIO DE PROCESOS DE NEGOCIO	29
3.1	Conceptos fundamentales de Grafos.....	29
3.1.1	Clasificaciones de Grafos	29
3.1.2	Isomorfismo de subgrafos	31
3.1.3	Método de indexación de grafos	32
3.2	Arquitectura Inicial de Referencia	35
3.2.1	Capa de Presentación.....	36
3.2.2	Capa de transformación de BP	36
3.2.3	Capa de Gestión de BP	42
3.2.4	Capa de Gestión del Repositorio.....	54
3.2.5	Storage Layer	54
3.3	Resumen.....	55
4	BÚSQUEDA SEMÁNTICA.....	56
4.1	Conceptos fundamentales	56
4.1.1	Web semántica	56
4.2	Ontología de Patrones de flujo de Control	58
4.2.1	Ontologías de procesos de negocio	58
4.2.2	Clasificación de patrones.....	59

4.2.3	Relaciones entre patrones.....	60
4.2.4	Definición de la ontología de patrones	65
4.3	Integración de la Ontología de Patrones con la Arquitectura	67
4.3.1	Razonador Semantico	67
4.3.2	Similitud semantica de patrones	68
4.4	Ranking de modelos de procesos recuperados.....	69
4.4.1	Condiciones del ranking semántico	69
4.5	Resumen.....	70
5	PROTOTIPO Y EXPERIMENTACIÓN.....	71
5.1	BPMOONER (BPMO – Ontological Enterprise Repository)	71
5.1.1	Modelo de Casos de Uso del Sistema	71
5.1.2	Software utilizado para la implementación.....	73
5.1.3	Diagramas de Paquetes, subsistemas y capas del Sistema	74
5.1.4	Interfaces gráficas de usuario del Sistema	78
5.2	Experimentación y Pruebas.....	81
5.2.1	Descripción del Benchmark de Referencia.....	82
5.2.2	Medidas de Desempeño.....	82
5.2.3	Plan de Pruebas y Resultados Obtenidos	83
5.3	Resumen.....	93
6	CONCLUSIONES Y TRABAJOS FUTUROS	94
6.1	Contribuciones	94
6.2	Conclusiones.....	95
6.3	Trabajos Futuros	96

Lista de Figuras

Figura 1. El reloj de arena de BPM [17].	16
Figura 2 . Diferentes lenguajes de procesos de negocio.	20
Figura 3. Grafo equivalente a un Mapa de Carreteras.	29
Figura 4. Ejemplo de Grafo con varias asignaciones de vértices.	30
Figura 5. G_1 es un subgrafo de G .	31
Figura 6. Arquitectura propuesta para el repositorio de procesos de negocio.	35
Figura 7. Representación gráfica del Modelo BP de ejemplo “ <i>Develop and Train Workforce</i> ” descrito en WSMML a partir del Modelador BPMO versión 1.4	36
Figura 8. Meta modelo de Referencia.	38
Figura 9. Representación basada en grafos de las estructuras que divergen, convergen y realizan ciclos en el flujo de control de un Modelo BP.	39
Figura 10. Diagrama modular del proceso de transformación de Modelo BP a Grafo BP.	40
Figura 11. (a) Contenido del archivo .dat del Grafo BP y (b) Su representación gráfica.	42
Figura 12. Ejemplo de implementación (STRUCTURED-PARTIAL-JOIN)	46
Figura 13. Resultados del estudio de viabilidad de implementación de los patrones de referencia en la notación BPMO.	47
Figura 14. Almacenamiento de los grafos (Data Storing) y la creación del índice (Index) sobre las tablas de la Base de datos de Berkeley	50
Figura 15. Operación del módulo Búsqueda de Similitud de Patrones.	51
Figura 16. (a) Los Patrones (1) (2) y (3) descritos en Lenguaje GLIDE (b) Consulta exacta de los patrones (1) (2) y (3) utilizando el lenguaje GLIDE.	52
Figura 17. Una consulta inexacta descrita en el lenguaje GLIDE.	52
Figura 18. Abstracción del diagrama de entidad relación de la base de datos de referencias.	55
Figura 19. Relación de especialización entre los patrones Parallel Split y Multiple Choice.	61
Figura 20. Relación de composición entre los patrones Parallel Split, Exclusive Choice y Multiple Choice.	63
Figura 21. Relaciones de especialización y composición entre patrones de flujo de control	64
Figura 22. Relaciones de especialización y composición para los patrones básicos de BPMO.	65
Figura 23. Ontología de patrones de flujo de control básicos.	67
Figura 24. Variantes del lenguaje WSMML (Web Service Modeling Language)	68
Figura 25. Diagrama modular del módulo Búsqueda de Similitud de Patrones con la adaptación de búsqueda semántica.	70
Figura 26. Diagrama de Casos de Uso del Sistema	73
Figura 27. Diagrama de paquetes BPMOONER-client.	75
Figura 28. Diagrama de paquetes BPMOONER-server.	78
Figura 29. Interfaz para el usuario anónimo.	79
Figura 30. Interfaz para el usuario registrado.	80
Figura 31. Interfaz de bienvenida del BPMOONER-server.	80
Figura 32. Interfaz de administración del BPMOONER-server.	81
Figura 33. Gráfica de rendimiento del sistema durante la Publicación de Modelos BP.	85
Figura 34. Gráfica de rendimiento del sistema durante la Recuperación de Modelos BP.	85
Figura 35. Escenario de Pruebas No 1.	87
Figura 36. Escenario de Pruebas No 2.	88
Figura 37. Escenario de Pruebas No 3.	89
Figura 38. Escenario de Pruebas No 4.	90
Figura 39. Escenario de Pruebas No 5.	91

Figura 40. Escenario de Pruebas No 6.....	92
Figura 41. Medidas de Calidad: Precision, Exhaustividad y Promedio de Efectividad.....	92

Lista de Tablas

Tabla 1. Comparación de los repositorios de modelos de procesos de negocio.....	32
Tabla 2. Elementos BPMO y su correspondiente representación en grafos.....	37
Tabla 3. Resultado de la evaluación – 20 Patrones de flujo de control iniciales.....	43
Tabla 4. Patrones adicionales – Total 43 Patrones de flujo de control.....	44
Tabla 5. Patrones de referencia modelados en (a) BPMO y (b) Representación en grafos.	48
Tabla 6. Clasificación según la funcionalidad y comportamiento de los patrones de BPMO.....	60
Tabla 7. Relaciones de especialización entre patrones de flujo de control	62
Tabla 8. Relaciones de composición entre patrones de flujo de control	63
Tabla 9. Plan de Pruebas.	84
Tabla 10. Especificaciones Técnicas del Equipo empleado para las Pruebas del Prototipo.....	84
Tabla 11. Benchmark de referencia para la medida de calidad de <i>exhaustividad</i>	86
Tabla 12. Medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida en el Escenario 1.	87
Tabla 13. Medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida en el Escenario 2.	88
Tabla 14. Medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida en el Escenario 3.	89
Tabla 15. Medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida en el Escenario 4.	90
Tabla 16. Medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida en el Escenario 5.	91
Tabla 17. Medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida en el Escenario 6.	92

Capítulo I

1 INTRODUCCIÓN

1.1 Contexto General

La Web se ha convertido en un medio esencial para el intercambio de información personal, académica y comercial, con la relevancia de ser un medio económico, flexible y adaptativo. Estas características permiten que sea un medio exitoso y ampliamente difundido, sin embargo, las mismas originan inconvenientes como la sobrecarga de información y heterogeneidad de fuentes de información, con el consiguiente problema de interoperabilidad. A través de tecnologías tales como los servicios Web, Web semántica, servicios Web semánticos, ontologías, se intenta dotar a la Web de cierto “nivel de razonamiento” que permita inferir sobre su contenido, para realizar búsquedas de una manera dinámica y eficiente [1].

La integración de la Web y la semántica producen la Web semántica, la cual propone superar las limitaciones de la Web actual, mediante la introducción de descripciones explícitas del significado en la estructura interna y la estructura global de los contenidos y servicios disponibles en la WWW (World Wide Web) [2]. El W3C (World Wide Web Consortium) define la Web Semántica como “una Web extendida, dotada de mayor significado, en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla, gracias a una información mejor definida a través de lenguaje natural” [3].

Por otra parte, los servicios Web proporcionan un conjunto de protocolos y estándares, que sirven para intercambiar datos entre aplicaciones independientemente del lenguaje de programación y del entorno de ejecución. Del mismo modo, los servicios Web y la Web semántica actúan como tecnologías complementarias para generar los servicios Web semánticos [4].

Las ontologías, desde el punto de vista informático, constituyen estructuras que especifican un vocabulario relativo a un cierto dominio. Este vocabulario define entidades, clases, propiedades, predicados, funciones y las relaciones entre estos [5]. Es importante puntualizar, que las ontologías cumplen un papel importante dentro de la Web Semántica y más aún, en la interoperabilidad semántica entre los sistemas de información, estableciendo correspondencias y relaciones entre los diferentes dominios [6]. La inclusión de ontologías para describir servicios Web semánticos permite la selección, integración e invocación dinámica de servicios, dotándolos así mismo de la capacidad de reconfigurarse dinámicamente para adaptarse a los cambios sin intervención humana [7], dicho de otra manera, en los servicios Web se incorporan componentes semánticos que permiten automatizar el descubrimiento, composición, invocación e interoperabilidad de los mismos [5]. SOA (*Service Oriented Architecture*) promueve las características de encapsulación, reutilización y composición de servicios, generando una gran

interoperabilidad entre sus componentes [8]. Factores que contribuyen a las características de automatización mencionadas anteriormente.

La creación de servicios requiere de la mutua colaboración entre diversas organizaciones especializadas en sus respectivas áreas de negocio y es donde, a través de la arquitectura SOA, es posible crear nuevos servicios más complejos por medio de la composición de servicios Web, esta composición se denomina proceso de negocio. La utilización de los procesos de negocio constituye una alternativa ampliamente adoptada en el contexto empresarial, ya que permite describir las operaciones internas y externas de las organizaciones. Existen herramientas especializadas llamadas BPMS (Business Process Management Systems) que permiten modelar, administrar y organizar estos procesos de una manera ágil y eficiente.

1.2 Declaración del Problema

La globalización de los mercados mundiales y el aumento de la oferta de productos y servicios, han hecho necesaria la incorporación en las empresas de nuevas tecnologías de gestión de procesos de negocio¹. Anteriormente, la mayoría de las empresas se constituían en torno a funciones de negocio tales como: mercadeo, ventas, producción, finanzas y servicio al cliente, donde cada una de ellas se ejecutaban de forma independiente según su propio modelo de negocio. La aparición de los sistemas BPMS (Business Process Management System) permitieron agilizar estas funciones dentro de la empresa facilitando su organización en torno a procesos de negocio [9]. En la actualidad existen varios sistemas BPMS basados en workflows² (soluciones de flujos de trabajo), algunos de ellos se apoyan en los elementos de la arquitectura SOA (Service Oriented Architecture), así como también en la automatización y composición de procesos. Lo anterior, permite coordinar recursos humanos y tecnológicos para llevar a cabo los procesos de la empresa u organización, de acuerdo con la estrategia de negocio definida [10].

En los BPMS se definen los procesos de negocio utilizando un lenguaje de modelado de procesos de negocio [11]. La gestión de estos modelos y la búsqueda de información en sus contenidos es una tarea laboriosa e ineficiente, por tal motivo se hace necesario almacenarlos y organizarlos en un repositorio que permita su recuperación y posterior reutilización [12]. Un repositorio de modelos de procesos de negocio es una unidad central para la recopilación de la información de los mismos. Actualmente, existen varias propuestas de repositorios de modelos de procesos de negocio tales como [13], [14], [15] y que permiten almacenar modelos, cada uno de ellos con sus propias características de almacenamiento y descubrimiento de estos modelos.

¹ Un proceso de negocio se define como un conjunto de procedimientos o actividades relacionadas que buscan un objetivo común de negocio.

² Un workflow es definido como la automatización parcial o total de un proceso de negocio, en donde los documentos, información o tareas son pasados de un participante a otro para realizar una acción, de acuerdo a un conjunto de reglas de procedimiento definidas.

Una de las formas para descubrir procesos se realiza a través de técnicas semánticas aplicando criterios relacionados con ontologías que permiten inferir acerca de conceptos sobre los cuales se realiza la búsqueda [1]. Las propuestas actuales en ésta área, no tienen en cuenta el flujo de control de los procesos, además, las herramientas existentes se basan en arquitecturas propietarias y no permiten adicionar otras características de búsqueda, tales como, búsquedas semánticas basadas en razonamientos ontológicos sobre el flujo de control que componen los procesos. Por tal motivo, este proyecto pretende proponer un método de descubrimiento en un repositorio de procesos de negocio aplicando mecanismos de búsqueda semánticas.

1.3 Escenario de motivación

Considere una organización que modele sus operaciones mediante procesos de negocio. Suponga, que al interior de la misma, se desea modelar nuevos procesos de negocio, que satisfagan necesidades emergentes a partir de los procesos de negocio previamente modelados. En consecuencia, se considera necesario contar con un repositorio de modelos de procesos de negocio, que permita recuperar modelos de procesos de negocio, que puedan adaptarse ó reutilizarse por medio de un mecanismo automático de comparación semántica de sus flujos de control. Este repositorio facilitaría realizar con mayor rapidez las actividades de análisis, gestión y reingeniería de procesos asociados al modelado de los nuevos modelos mencionados.

1.4 Contribuciones

- **Arquitectura para un repositorio de modelos procesos de negocio.** A partir del estudio de diferentes repositorios de modelos de procesos existentes, en el contexto empresarial y académico, fue posible formular una arquitectura especialmente diseñada para el almacenamiento y recuperación de modelos de procesos de negocio.
- **Sistema de Indexación de grafos de modelos de procesos de negocio.** Entre los resultados del estudio anterior, también se encontró que varios de estos repositorios de modelos de procesos, carecen de un método de indexación por lo que ésta contribución constituye un aporte significativo para este trabajo.
- **Mecanismo de búsqueda semántico de modelos de procesos de negocio basado en patrones de flujo de control.** La adición del factor semántico, en las búsquedas al interior del repositorio, permite dotar al sistema de búsqueda la capacidad de inferir sobre las instancias de conceptos pertenecientes a una ontología y calcular la distancia semántica entre procesos asociando patrones de flujo de control.

- **Prototipo de búsqueda semántica de modelos de procesos de negocio, basado en patrones de flujo de control.** Creación de una aplicación Web que permite el almacenamiento y la recuperación de modelos de procesos de negocio, soportado en el mecanismo de búsqueda semántico mencionado anteriormente.
- **Definición de una ontología de patrones de flujo de control.** Esta definición permite realizar un razonamiento semántico y la inferencia sobre las instancias de los conceptos de la ontología de patrones de flujo de control.
- La publicación de 2 artículos, en conferencias internacionales, uno de ellos durante el “1st International Workshop on Reuse in Business Process Management – rBPM2010” realizado en el marco del congreso BPM 2010, y el otro en el congreso “Euro-American Conference on Telematics and Information Systems – EATIS2010”, los cuales abordan el sistema de indexación de modelos de procesos basado en grafos y la arquitectura del repositorio de modelos de procesos de negocio propuesto, respectivamente. Las 2 temáticas se encuentran descritas ampliamente en los Capítulos 3 y 4 de la presente monografía.
- El trabajo desarrollado dentro del presente proyecto, soporta el componente de Analizador de Semántico del Comportamiento del sistema que se propone en la tesis de maestría: **Descubrimiento Automático de Procesos de Negocio Basado en Semántica del Comportamiento** desarrollada dentro del Grupo de Ingeniería Telemática de la Universidad del Cauca.

1.5 Alcance

En este trabajo de grado se analizaron los repositorios de procesos de negocio existentes en el entorno empresarial y académico, y a partir de este análisis se propuso un repositorio en el cual se aplicó y validó el mecanismo de búsqueda semántico para la recuperación de modelos de procesos de negocio. Cabe resaltar, que durante este trabajo se establecieron 7 patrones básicos de grafos, los cuales fundamentan los 12 patrones básicos del lenguaje BPMO, es decir que no están soportados patrones de mayor nivel de complejidad, que resulten de la combinación de los patrones básicos. Además, partiendo de que el proyecto SUPER toma al lenguaje BPMO como punto común entre las notaciones de modelos de procesos más utilizadas, solamente se realizó la implementación de un módulo de transformación (Parser) de BPMO versión 1.4 a la representación formal basada en grafos.

1.6 Contenido de la monografía

Capítulo 2. ESTADO DEL ARTE

Este capítulo presenta un resumen de los principales lenguajes utilizados para el modelado de los procesos de negocio, a partir del cual, es seleccionado un lenguaje que se ajuste a los requerimientos de este proyecto. Adicionalmente, es descrito un estado del

arte referente a los repositorios de servicios Web, repositorios de Procesos de Negocio y trabajos de investigación relacionados con este proyecto.

Capítulo 3. REPOSITORIO DE PROCESOS DE NEGOCIO

Esta sección abarca, principalmente, el Repositorio de Procesos de Negocio propuesto, definiendo una arquitectura de referencia. Posteriormente, es explicado el método por el cual los modelos de procesos de negocio son transformados en una representación formal basada en grafos, describiendo cómo la problemática de la recuperación de procesos de negocio puede convertirse en una problemática de detección de isomorfismo de grafos. Finalmente, es dada una explicación del proceso de ordenamiento y de presentación de resultados al usuario (Ranking de modelos de procesos).

Capítulo 4. BÚSQUEDA SEMÁNTICA

Este capítulo abarca la integración del proceso de recuperación de los modelos de procesos de negocio, junto con una ontología de patrones de flujo de control, que permita adicionar características de búsqueda semántica al repositorio propuesto en el Capítulo 3. Esta integración aborda conceptos relacionados a la búsqueda semántica y al manejo de ontologías que permitirán la posterior formulación de la ontología de patrones de flujo de control. Posteriormente, se explicará la definición de una medida de similitud semántica entre patrones de flujo de control y su aplicación durante el proceso de Ranking de modelos de procesos de negocio recuperados.

Capítulo 5. PROTOTIPO Y EVALUACIÓN.

Este capítulo aborda la descripción detallada del prototipo que valida la arquitectura propuesta en los Capítulos 3 y 4, la especificación del plan pruebas realizadas sobre el mismo y los resultados obtenidos a partir de la ejecución de dichas pruebas.

Capítulo 6. CONCLUSIONES Y TRABAJO FUTURO

Por último, se analizan los resultados del trabajo realizado, se detallan las principales contribuciones obtenidas en la ejecución del proyecto y se expone un conjunto de recomendaciones importantes para el desarrollo de trabajos futuros.

Capítulo II

2 ESTADO DEL ARTE

En este capítulo se presenta una síntesis de los principales conceptos tomados como referente para el Repositorio de Procesos de Negocio propuesto. Inicialmente, se señalan las características principales de los procesos de negocio y los lenguajes de modelado de procesos disponibles a nivel empresarial y en el ámbito académico. A partir de ellos, se selecciona el lenguaje que se ajuste a los requerimientos del proyecto. Finalmente, se describe el estado del arte referente a los repositorios de servicios Web y los repositorios de procesos de negocio, así como también los trabajos de investigación relacionados con este proyecto.

2.1 Contexto General

2.1.1 Procesos de Negocio

Un proceso de negocio, según el WfMC³, es un conjunto de uno o más procedimientos o actividades que colectivamente alcanzan un objetivo de negocio o unas políticas, dentro del contexto de una estructura organizacional, definiendo roles y relaciones funcionales [16]. Además, está típicamente asociado con objetivos operacionales y relaciones de negocio, como por ejemplo, el proceso de reclamo en una aseguradora, ó un proceso de desarrollo en ingeniería. Un proceso de negocio podría estar completamente contenido en una sola unidad organizacional o podría extenderse a diferentes organizaciones, tal como un proceso que relaciona a un cliente con los proveedores, inclusive podría involucrar interacciones formales e informales entre los participantes y su duración podría también variar ampliamente. De igual forma, un proceso de negocio puede contener actividades automatizadas, capaces de gestionar workflows⁴ y/o actividades manuales.

A través del modelado de procesos ó BPM (Business Process Modeling), es posible capturar, documentar y diseñar los procesos de negocio en una organización. BPM, además es una forma estructurada, coherente y consistente de entender, documentar, modelar, analizar, simular, ejecutar e intercambiar continuamente procesos de negocio y todos los recursos involucrados, con el propósito de contribuir al mejoramiento de los negocios [17]. En este sentido, BPM busca identificar los principales procesos de negocio dentro de una organización, automatizarlos con una integración eficiente hacia los sistemas y las personas y así maximizar la eficacia de los procesos. El modelado de procesos también se puede entender como una captura de una secuencia ordenada de actividades de negocio e información de apoyo, describiendo cómo una empresa intenta conseguir sus objetivos de negocio [18]. Existen diferentes niveles de modelado de procesos:

³ WfMC (Workflow Management Coalition) crea y contribuye con los estándares relacionados con procesos, es la única organización de estándares que se enfoca solamente en procesos.

⁴ Un workflow es definido como la automatización parcial o total de un proceso de negocio, en donde los documentos, información o tareas son pasados de un participante a otro, para realizar una acción de acuerdo a un conjunto de reglas de procedimiento definidas

- Mapas de Procesos (diagramas de flujo simples de las actividades)
- Descripciones de Procesos (diagramas de flujo extendidos con información adicional, pero insuficientes para definir completamente ciertos desempeños).
- Modelos de Procesos (diagramas de flujo extendidos con suficiente información para que el proceso pueda ser analizado, simulado y/o ejecutado).

A continuación, se consideran algunas de las notaciones más destacadas que permiten modelar procesos de negocio. A partir del análisis presentado en [19] donde se incluye un amplio rango de conceptos, y se propone un metamodelo⁵ para evaluar los diferentes lenguajes de modelado de procesos (BPMLs)⁶, se pretende elegir un lenguaje específico para el modelado de procesos del presente trabajo. Este metamodelo está basado en un marco conceptual de [20], y es un framework⁷ que tiene cuatro perspectivas: organizacional, funcional, de información y de comportamiento; además, el framework se extiende a otra perspectiva denominada contexto de procesos de negocio, que captura información importante como los objetivos de proceso y medidas. Este metamodelo genérico se inspira por la teoría de procesos de negocio, patrones de workflows⁸ y la Workflow Management Coalition (WfMC).

2.1.2 Lenguajes de Modelado de Procesos de Negocio

En esta sección se presentan algunos de los lenguajes de modelado de procesos evaluados en el análisis realizado en [19], considerando algunas ventajas y desventajas inherentes a cada lenguaje y, al final, se selecciona a BPMO como el lenguaje de modelado con las características más adecuadas para este proyecto.

BPEL (Business Process Execution Language)

El lenguaje de ejecución de procesos de negocios para servicios Web, proporciona un lenguaje para la especificación formal de procesos y protocolos de la interacción de negocio, que se extiende al modelo de interacción de los servicios Web, definido por WSDL. Es un lenguaje independiente, en cuanto a la plataforma de ejecución y teóricamente portable. BPEL, describe una capacidad conocida comúnmente como orquestación, que es un componente dominante de las arquitecturas orientadas a servicio (SOA⁹), y que permite a los procesos del negocio estar basados en servicios reutilizables, de manera que ahorra costos y resulta más ágil. Algunas ventajas, inherentes al lenguaje, es que es un estándar de industria para expresar procesos de negocio y está diseñado para interactuar con varios servicios Web. Además de que permite la

⁵ Metamodelo, se refiere a la construcción de una colección de conceptos (cosas, términos, etc.) con un cierto dominio.

⁶ BPML es un metalenguaje para el modelado de los procesos de negocio, así como XML es un metalenguaje para el modelado de los datos de negocio.

⁷ Un Framework, es un conjunto estandarizado de conceptos, prácticas y criterios que se utilizan para enfocar una problemática particular y sirve de referencia para resolver nuevas problemáticas de índole similar.

⁸ Los patrones de workflows, son una iniciativa de la Universidad de Tecnología de Eindhoven y la Universidad de Tecnología de Queensland y buscan proveer una base conceptual para las tecnologías relacionadas con procesos, especialmente tienen en cuenta las perspectivas de flujo de control, datos, recursos y manejos de excepciones.

⁹ SOA, es una arquitectura para componentes de un sistema que define patrones e interacciones entre ellos. Un componente ofrece un servicio que espera a un estado de disponibilidad, mientras que otros componentes podrían invocar el servicio de conformidad con un contrato de servicio.

interoperabilidad entre diferentes procesos, a través de la composición recursiva de los mismos. Una de las desventajas de BPEL, es que no soporta transacciones y no existe una transformación de datos (EDI¹⁰, formatos binarios¹¹), ni tampoco permite la adición de semántica.

BPMN (Business Process Modeling Notation)

BPMN es una notación para definir Procesos de negocio, basado en diagramas de flujo y soporta cada uno de los niveles descritos anteriormente, fue desarrollado por The Business process Management Initiative (BPMI) como una notación estándar para el modelado de Procesos de Negocio (BPMN)¹². El primer objetivo, de los esfuerzos de BPMN, fue proveer una notación fácilmente entendible por todos los usuarios de negocio, desde los analistas de negocio, quienes crean los borradores iniciales de los procesos, incluyendo la responsabilidad técnica de los desarrolladores de la tecnología que realiza estos procesos, y finalmente, hasta las personas de negocio, quienes van a gestionar y monitorear estos procesos (Figura 1). Así, BPMN crea un puente estandarizado para la brecha existente entre el diseño de procesos de negocio y la implementación de procesos.

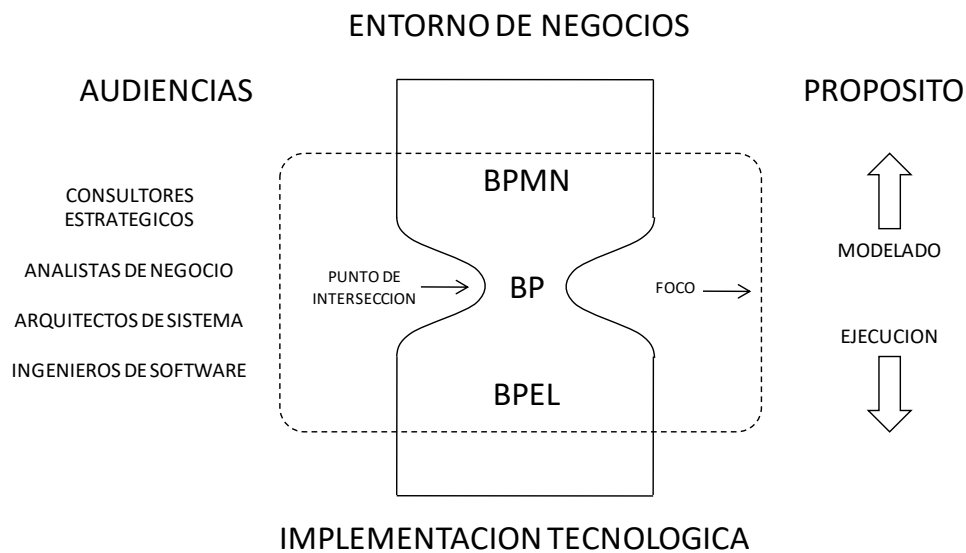


Figura 1. El reloj de arena de BPM [17].

BPMN fue desarrollado para realizar un simple mecanismo de creación de procesos, y al mismo tiempo, posibilitar el manejo de la complejidad inherente a los procesos de negocio. Por lo tanto, para manejar estos dos requerimientos conflictivos, fue necesario organizar los aspectos gráficos de notación en categorías específicas. Algunas de las

¹⁰ EDI (Electronic Data Interchange) ó intercambio electrónico de datos. Un sistema EDI permite el intercambio (envío y recepción) de documentos comerciales por vía telemática.

¹¹ Un formato binario hace referencia a la información que se encuentra digitalizada, representada por cifras "0" y "1", este formato es el utilizado por las computadoras.

¹² Actualmente, BPMI y OMG han fusionado sus actividades de investigación en Gestión de Procesos de Negocio (BPM) para proveer estándares para esta creciente industria. El grupo combinado se ha denominado así mismo: Business Modeling & Integration (BMI) Domain Task Force (DTF).

ventajas de este lenguaje es que define una semántica y se encuentra en un nivel abstracto, lo cual permite que sea posible el mapeo a múltiples especificaciones de lenguajes de bajo nivel (BPEL, BPML), además que gráficamente es muy rico en cuanto al flujo de control. Algunas de las desventajas de este lenguaje es que es débil en el contexto organizacional y existe ambigüedad y confusión cuando se comparte modelos BPMN.

EPC (Event-driven Process Chains)

Las cadenas de procesos dirigidas por eventos son un lenguaje gráfico de descripción de procesos, introducido por Keller, Nüttgens y Scheer en 1992, dentro del framework de la arquitectura del Sistema Integrado de Información (ARIS¹³), para modelar procesos de negocio [21]. El lenguaje apunta a la descripción de procesos en el nivel de su lógica de negocio, no necesariamente en un nivel formal de especificación y es fácil de entender y utilizar por las personas de negocio. La metodología hereda el nombre de los tipos de diagramas que muestran la estructura del flujo de control de los procesos, como una cadena de eventos y funciones, esto es, una cadena de procesos dirigida por eventos [22]. Algunas de las ventajas de EPC, es que cubre más aspectos y descripciones detalladas de las unidades de organización de negocio, junto con sus respectivas funciones y recursos materiales. Además es un lenguaje rico en el modelado de dominio, en el modelado organizacional y las perspectivas funcionales y de comportamiento están bien representadas. Algunas desventajas son: ni la sintaxis ni la semántica de un EPC están bien definidas, para tal propósito es necesario realizar un mapeo de EPCs en redes de Petri, además, las perspectivas organizacional y de información son parcialmente soportadas.

Petri Nets

Las redes de Petri se han convertido en una herramienta popular para el modelado y análisis de los sistemas concurrentes. Hay muchos factores que contribuyen a su éxito: naturaleza gráfica, habilidad para modelar procesos paralelos y distribuidos en una manera natural, la simplicidad de modelo y las firmes bases matemáticas sobre las que se fundamentan. Sin embargo, los modelos clásicos de redes de Petri no son adecuadas para modelar varios sistemas encontrados en logística, producción, comunicaciones, manufactura flexible y procesamiento de información. Además, las redes de Petri tienen una semántica formal y proveen unas abundantes técnicas de análisis y descripción de los sistemas en tiempo real, que tienden a ser complejos y extremadamente extensos [23]. A continuación se consideran algunas extensiones de las redes de Petri llamadas Redes de Workflows (Workflows Nets) y Patrones de Workflows (Workflows Patterns).

¹³ ARIS es una plataforma que provee productos software integrados que ayudan a las empresas a mejorar continuamente sus procesos de negocio.

Redes de Workflows (Workflow Nets)

La mayoría de WFMS¹⁴ (Sistemas de Gestión de Workflows) están basados en procedimientos. Un procedimiento, es el método de operación utilizado por un proceso de negocio, para procesar casos. Ejemplos de casos son: órdenes, reclamos, gastos de viaje, declaraciones de impuestos, entre otros. El procedimiento especifica, el conjunto de tareas requeridas para procesar estos casos en forma exitosa; además, especifica la orden (parcial) en que estas tareas tienen que ser ejecutadas. El objetivo de un procedimiento es manejar casos de manera eficiente y apropiada. Algunas de las ventajas de las redes de Petri es que la semántica ha sido formalmente definida a pesar de la naturaleza gráfica y están basadas en estados en lugar de eventos. Algunas desventajas: las redes de workflows no soportan del todo bien la perspectiva organizacional y de información, no hay sistemas con una interfaz robusta y son usualmente modeladas independientemente de los datos.

Patrones de Workflows (Workflow Patterns)

El principal objetivo de los Patrones de workflows, se basa en el diseño de los requerimientos fundamentales que surgen durante el modelado de procesos de negocio. Con el fin de describir estos requerimientos, se adoptó una estrategia basada en patrones, tratando de proveer independencia de la tecnología de implementación y al mismo tiempo, independencia de los requerimientos esenciales del dominio en que ellos se orientaron. Un patrón “es una abstracción de una forma concreta que mantiene la aparición periódica en contextos específicos no arbitrarios” [24]. Algunas ventajas, de los patrones de workflows, es que permiten representar semántica a través de redes de Petri y algunos patrones complejos son fáciles de construir. Las desventajas, de los patrones de workflows, es que consideran una notación independiente del vocabulario común de las expresiones del flujo de control (y luego del flujo de datos) de los lenguajes de workflows, además algunos patrones no se mapean bien sobre redes de Petri de alto nivel.

YAWL (Yet Another Workflow Language)

Los patrones de workflows permitieron el desarrollo de YAWL. A diferencia de otros esfuerzos en el área de BPM, YAWL buscó proveer un lenguaje de modelado comprensible para procesos de negocio, basado en unos fundamentos formales. El contenido del lenguaje YAWL es una adaptación de las redes de Petri, introducidas por los patrones de workflows y una de sus mayores aspiraciones consiste en mostrar que un conjunto relativamente pequeño de constructores podría ser utilizado para soportar directamente la mayoría de los patrones de workflows identificados. Este también intenta ilustrar que podría coexistir dentro de un framework común, y con el propósito de validar que ese lenguaje era capaz de una promulgación directa, se desarrolló el sistema YAWL,

¹⁴ WFMS, Workflow Management Systems, son sistemas que definen, crean y gestionan la ejecución de workflows a través del reuso de software, ejecutándose en uno o más motores de workflows, los cuales interpretan la definición del procesos, interactúan con los participantes de workflows y, donde son requeridos, invocan el uso de herramientas IT y aplicaciones.

y sirve como referencia de implementación del lenguaje. Con el tiempo, el lenguaje YAWL y el sistema YAWL se convierten en sinónimos y genera un amplio interés en los desarrolladores y la comunidad académica [25]. Algunas de las ventajas de YAWL se hallan en el soporte para la adaptación dinámica de modelos de workflows, a través de nociones de worklets¹⁵ y consta de un modelo basado en XML, para la definición y manipulación de datos basado en XML Schema, XPath y XQuery¹⁶. Algunas desventajas derivan de la situación académica para realizar una notación bien definida, que no se trabaja en los Patrones de Workflows y no está ampliamente adoptada, además existe poca semántica implementada.

BPMO (Business Process Modeling Ontology)

BPMO significa Business Process Modeling Ontology [26] y la idea detrás de BPMO consiste, en crear una ontología que sea capaz de representar artefactos de varias metodologías de modelado de procesos de negocio y proponer una representación unificada y una única interpretación. Esta es la principal ontología de modelado del proyecto SUPER¹⁷ (Semantics Utilized for Process Management within and between Enterprises), y unifica dos métodos de modelado: basado en grafos y basado en bloques. El primero es preferido por los usuarios de negocio; el último es necesario para la traducción a BPEL antes de que la ejecución se realice. En el proyecto SUPER, el rol de unificación para varias notaciones de modelado de procesos es provista por BPMO (Business Process Modeling Ontology).

Algunas ventajas, presentes en el lenguaje BPMO, en comparación con los anteriores lenguajes de modelado son las siguientes:

- BPMO, es un destino común de traducción y además, es un recurso para notaciones industriales.
- BPMO es una representación común sin ser de tan bajo nivel y complicado como las redes de workflows.
- Basado en BPMN, en lugar de imponer una nueva notación (así como YAWL).
- BPMO preserva y está fundamentado en la perspectiva organizacional y en las capacidades del modelado de dominio de EPCs, en lugar de concentrarse sólo en el comportamiento.
- BPMO, presenta las bases del razonamiento automatizado basado en ontologías.
- Tiene una conexión formal entre la semántica del comportamiento y los procesos ejecutables, a través del razonamiento basado en ontologías.

¹⁵ Un worklet es un objeto que representa un conjunto de tareas.

¹⁶ XQuery es un lenguaje de consulta diseñado para consultar colecciones de datos XML, similar a SQL pero incluye algunas capacidades de programación.

¹⁷ El proyecto SUPER pretendía llevar la Gestión de Procesos de Negocio (BPM) desde el nivel de las tecnologías de la Información (TI), donde reside ahora, hacia el nivel de negocio donde debe estar. El resultado fue el desarrollo de herramientas que permiten el despliegue de la Gestión de Procesos de Negocio Semánticos (SBPM). Este proyecto fue financiado por la Unión Europea dentro de las prioridades de la Information Society Technologies (IST). Disponible en Internet: <http://www.ip-super.org/>

La Figura 2 ejemplifica algunos de los diferentes lenguajes de procesos de negocio presentados anteriormente, pasando por los lenguajes de modelado, y llegando hasta los lenguajes de implementación o de ejecución de procesos.

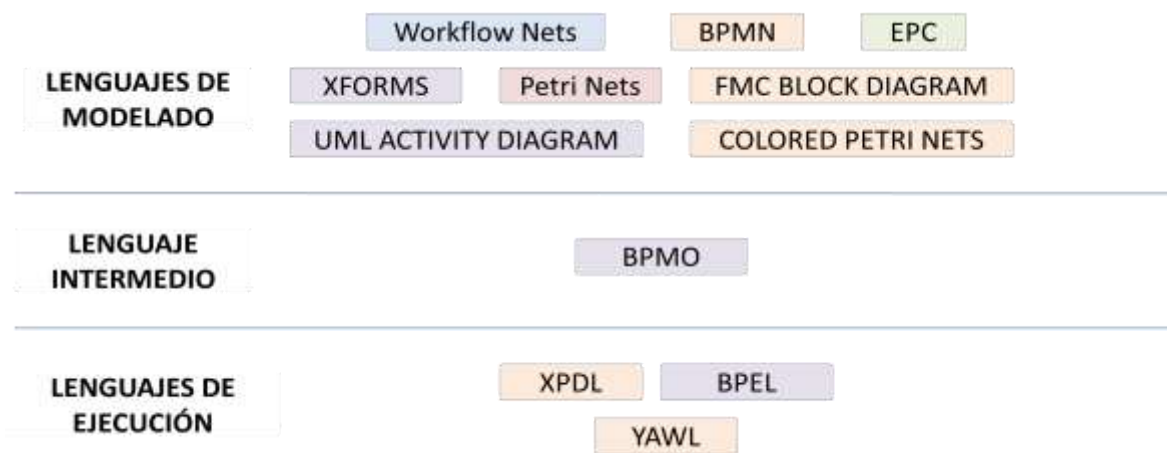


Figura 2. Diferentes lenguajes de procesos de negocio.

Selección del lenguaje de modelado de procesos de negocio

Por medio de las descripciones anteriores, de los lenguajes de modelado de procesos de negocio más utilizados, y además del análisis presentado en [19], se observa que BPMO es el lenguaje que reúne las mejores características de cada uno de ellos. Además que presenta un enfoque innovador para la representación semántica de procesos de negocio y es un lenguaje que permite realizar anotaciones semánticas, así como también añadir ontologías de dominio para realizar consultas basadas en razonamiento semántico. Así mismo, ofrece los mejores aspectos que se ajustan a los objetivos planteados en este trabajo de grado.

En el Anexo 1, se hace un estudio detallado que justifica la selección del lenguaje de modelado de procesos de negocio a partir de sus ventajas y desventajas, con respecto a los demás lenguajes.

2.2 Trabajos Relacionados

En esta sección se presentan las propuestas de repositorios de procesos disponibles actualmente. Un repositorio digital, según [27], se define como un sitio centralizado donde se almacenan y se mantiene recursos digitales (textuales, imágenes, sonidos). En este contexto, el término repositorio hace referencia a un lugar donde se almacenan servicios Web ó procesos de negocio.

2.2.1 Repositorios de Servicios Web

Actualmente existen varios repositorios de servicios Web, disponibles en línea, presentados como una organización de servicios publicados en páginas Web. Estos,

tienen motores de búsqueda que realizan las búsquedas de acuerdo al nombre del servicio. Mediante la búsqueda de una palabra clave, que retorna el servicio que contiene esa palabra como una sub cadena incluida en los nombres y descripciones de estos servicios Web. La búsqueda también tiene en cuenta los parámetros de entrada y salida de los servicios y los mensajes que se intercambian. Entre los más importantes están: UDDI , Bindingpoint [28], .NET XML Web Services Repertory [29], Woogle [30], eSynaps [31].

Existen otros repositorios que ofrecen categorías de búsqueda, que indican el dominio de los servicios, por ejemplo (Negocios, Comercio, Comunicaciones, Salud, etc.), es decir, permiten realizar las búsquedas de los servicios en forma alfabética, así como también, los nombres y las descripciones de los servicios. Entre estos se encuentran: WebService Resource[32], Web Service List [33], WebserviceX.NET [34].

En [35] se observa un portal que tiene un motor de búsqueda con capacidades de búsqueda semántica y utiliza algoritmos que permiten encontrar los servicios adecuados para una tarea dada, en forma eficiente. Sus resultados proporcionan descripciones más detalladas del servicio, en donde se incluyen: el país de origen, proveedor, disponibilidad, tiempo de respuesta y una breve descripción del servicio. Además permite invocar el servicio y observar sus resultados a través de una interfaz Web.

2.2.2 Repositorios de Procesos de Negocio

A continuación se nombra y se describe algunos de los repositorios de procesos disponibles en el mercado. Algunos autores [36] presentan los resultados del análisis detallado de 16 repositorios de procesos de negocio, en donde se comparan de acuerdo a sus características de datos, funcionalidad y gestión y además se tiene en cuenta los aspectos de almacenamiento, recuperación, integración e indexación. En esta sección se describen estos repositorios.

Business Process Library (BPL) [37], es el resultado del primer diseño y prototipo de la librería de procesos de negocio en el proyecto SUPER. La BPL provee las operaciones estándar de un Sistema de Gestión de Base de Datos, tales como, el soporte de transacciones para la manipulación de datos. Permite crear, leer, actualizar, borrar (CRUD¹⁸) artefactos, utilizando una herramienta de modelado, definida en el proyecto SUPER. Realiza dos tipos de consultas, la primera basada en un motor de base de datos relacionales [38] y la segunda, permite consultas semánticas haciendo inferencia sobre la información contenida en una base de datos relacional a través de una máquina de razonamiento ontológico [39]. Además, almacena modelos de procesos descritos usando las siguientes ontologías: BPMO, sBPEL [38]. BPL es un proyecto muy amplio, que comprende una plataforma para la gestión de procesos de negocio, el inconveniente es que está en fase de desarrollo y en algunos casos no hay acceso para algunas de las

¹⁸ CRUD, iniciales de Create, Read, Update, Delete.

funcionalidades del mismo, además tiene una arquitectura cerrada y utiliza herramientas propietarias del proyecto SUPER.

ebXML (Electronic Business using eXtensible Markup Language) Registry/Repository [40], es un repositorio incluido en la familia de estándares ebXML, el cual empezó como una iniciativa de OASIS (Organization for the Advancement of Structured Information Standards) y UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business). Este repositorio permite almacenar cualquier tipo de dato incluyendo: descripciones de servicios Web, documentos y datos XML, datos binarios (imágenes, archivos de sonido, datos de video, archivos de aplicaciones ejecutables, entre otros). Mediante el registro, se puede adicionar metainformación¹⁹ a los datos ingresados en el repositorio, además se puede hacer búsquedas y clasificaciones mediante un mecanismo avanzado de consultas, que pueden ser filtradas en consultas XML o SQL [15]. Puede interactuar con otros lenguajes de procesos como lo son: WS-Choreography Description Language (WS-CDL) para la coreografía de servicios Web y con WS-Business Process Execution Language (WS-BPEL), para la orquestación de servicios Web [41]. Este repositorio no tiene capacidades de consulta semánticas, solamente SQL y utiliza un lenguaje de definición de procesos (ebBP), propio de los estándares ebXML.

IBM BPEL Repository [13], es una parte del proyecto Business Process Integration and Automation (BPIA), del laboratorio de investigaciones de IBM en Zurich [42]. El repositorio almacena procesos de negocio BPEL y otros documentos XML en sistemas de archivos. Permite hacer consultas a los archivos XML como objetos EMF (Eclipse Modeling Framework), utilizando un lenguaje de consultas orientado a objetos (OCL), así como también, proporciona operaciones básicas como crear, leer, escribir y borrar para manipular los datos almacenados en XML, como objetos Java. Aunque el usuario manipula los datos como objetos, la información es almacenada en forma transparente en el correspondiente documento XML, descrito por lenguajes como: BPEL, WSDL, XSD, entre otros [43]. Es importante resaltar que permite manipular diferentes documentos XML y que la arquitectura del repositorio está creada para ser adaptada permitiendo añadir nuevas funcionalidades, tales como, un nuevo motor de consultas OCL y otras extensiones EMF. Solamente permite almacenar documentos descritos en BPEL y XML y no tiene características de búsqueda semánticas, y las consultas se realizan a través de objetos OCL, necesitando un motor de OCL, muy ligado al IDE Eclipse.

IPM (Integrated Process Management) [44], (Repositorio para la Gestión Integrada de Procesos), gestiona procesos a través del ciclo de vida. Como tal, no sólo tiene funciones básicas de almacenamiento y recuperación de modelos de procesos de negocio, también cuenta con funciones avanzadas en la gestión de configuración y de versiones. La información de los procesos se puede intercambiar con el repositorio a través del lenguaje

¹⁹ La metainformación se define como la adición de información sobre la información ya existente, permite describir características más específicas del contenido de un documento.

IPM Executable Process Definition Language (IPM-EPDL)²⁰ [45]. Internamente, el repositorio tiene conectores para almacenar información de los procesos en una base de datos XML o una base de datos relacional. IPM se centra en el almacenamiento de los modelos de procesos y procedimientos específicos de una compañía, así como también, de la ejecución de las instancias de esos procesos. El repositorio IPM tiene un buen soporte para almacenar y recuperar los modelos de procesos e información relacionada. Para ello, utiliza un lenguaje de consulta, el IPM Proceso Query Language (IPM-PQL), el cual tiene soporte para las consultas específicas en los procesos, como la búsqueda de procesos que tienen una determinada actividad o cierta transición de una actividad a otra. IPM también soporta la búsqueda de procesos en el repositorio utilizando una serie de clasificaciones. Tanto el lenguaje de modelado, como de consulta es propietario y está diseñado para modelar los procesos específicos de las compañías.

RepoX [46] es un repositorio de modelo de proceso basado en XML, el cual es parte del sistema de workflow llamado METEOR²¹. Aunque forma parte del proyecto METEOR, se ha desarrollado específicamente con la intención de estandarizar el intercambio de modelos de procesos entre una herramienta de definición de proceso y un motor de workflows. Los modelos de procesos pueden ser intercambiados usando documentos XML en un formato predefinido. Internamente los modelos se almacenan en una base de datos orientada a objetos. RepoX almacena los aspectos del flujo de control de los procesos junto con los datos que son usados en los procesos y los roles que son autorizados para realizar tareas en los procesos. Desde que RepoX fue definido como un repositorio para motores de workflows, éste almacena modelos de procesos específicos para compañías. No cuenta con una notación específica, la descripción de los procesos se realiza a través de DTD, la cual tiene bastantes limitaciones en la definición de elementos de contenido, por tal motivo surgieron otros lenguajes de esquemas como lo son XMLSchema. Además las consultas están basadas en XQuery, el cual permite extraer información en documentos XML, pero no permite realizar consultas semánticas.

BPMN Repository [47] es la arquitectura de un repositorio, más que una implementación completa. Está dirigido específicamente a repositorios de modelos de procesos que soporten procesos inter organizacionales, esto es, procesos o composiciones de procesos múltiples que abarquen términos organizacionales. Esta arquitectura propone un amplio rango de tipos de procesos y de aspectos que deberían ser soportados para ese propósito, tales como los aspectos de flujo de control, aspectos de datos, la estructura organizacional y la información de autorización, monitoreo y controles. En cuanto a los tipos de procesos que debería soportar, están: modelos de referencia, modelos específicos de compañías, instancias de procesos e información histórica de las instancias de los procesos. Además plantea que los procesos y la información relacionada, deberían ser almacenadas en un formato XML estándar. Solamente describe

²⁰ IPM-EPDL son las siglas de IPM Executable Process Definition Language, el cual es un formato basado en XML que se puede utilizar para almacenar información sobre los modelos de procesos y actividades, el flujo de control, la estructura organizacional, la autorización y asignación de recursos, datos y monitoreo.

²¹ El proyecto (METEOR) fue desarrollado para la gestión de procesos de workflows para gran escala, para aplicaciones de workflows en entornos de computación heterogéneos y multi-empresariales.

una arquitectura, que combina varios trabajos y presenta un repositorio distribuido en capas, el lenguaje de modelado es BPMN, no tiene un lenguaje de consulta definido y no permite consultas semánticas.

ORYX [48] es una herramienta de modelado de procesos basada en Web, que soporta búsqueda de usuarios, creación, almacenamiento y actualización de modelos de procesos en línea. La herramienta utiliza un repositorio para almacenar los modelos de procesos de negocio que son creados en él. ORYX se enfoca principalmente en el aspecto del flujo de control y almacena modelos de procesos específicos de compañías, además soporta muchas notaciones de modelado de procesos, incluyendo: BPMN, EPC, redes de Petri, redes de workflows, FMC Block Diagrams²² y XForms²³. Internamente, los procesos son almacenados en una base de datos, externamente representa modelos de procesos en un formato RDF²⁴. Además no solamente se enfoca en las funciones básicas de almacenamiento (CRUD), sino que también implementa funciones de importación y exportación de modelos en formatos ERDF²⁵, JSON²⁶, PNML²⁷, XPDL y el formato XHTML se puede transformar en redes de Petri. Para la consulta de procesos, ORYX integra BPMN-Q²⁸. El almacenamiento y recuperación de procesos se hace a través de RDF y no soporta consultas semánticas, se hace necesario combinar RDFSchema y OWL²⁹ para añadir semántica al contenido.

BP-Suite [49] es un conjunto de herramientas basadas en la ejecución de procesos especificadas en BPEL. Consta de BP-QL (un lenguaje para las definiciones de procesos), BP-Mon (una herramienta para monitorear las instancias de procesos de negocio en ejecución) y BP-Ex (una herramienta para analizar las anotaciones de las ejecuciones). BP-Suite soporta los aspectos de actividades, flujo de control, datos, recursos, estructura organizacional, monitoreo y autorización, además del almacenamiento de modelos de procesos, instancias de procesos e información histórica acerca de las instancias. Los modelos de procesos son almacenados en formato BPEL. Además de las funcionalidades estándar de una base de datos, para el almacenamiento y recuperación de modelos de procesos, BP-Suite soporta consultas sobre los 3 tipos de procesos que puede almacenar (definiciones, instancias en ejecución, anotaciones). Solamente utiliza BPEL como lenguaje de modelado de procesos, BP-QL como lenguaje propietario para consultas, instancias y manejo de historial a través del IDE Eclipse, necesita del IDE Eclipse para ser ejecutado y tampoco permite realizar consultas semánticas.

²² Los diagramas de bloques FMC, son utilizados para realizar bosquejos de composiciones estructurales de sistemas, a través de una notación gráfica que pueda ser fácil de entender.

²³ XForms, es un formato XML diseñado por el W3C para definir interfaces de usuario, principalmente formularios Web.

²⁴ RDF, es un framework para metadatos desarrollado por la W3C y se basa en la idea de convertir las declaraciones de recursos en expresiones con la forma sujeto-predicado-objeto, conocidas en RDF como tripletas.

²⁵ ERDF, es una sintaxis para escribir HRML de una manera que la información en un documento HTML extraída en RDF.

²⁶ JSON, es el acrónimo de JavaScript Object Notation, el cual es un formato ligero para el intercambio de datos.

²⁷ PNML, acrónimo de Petri Net Markup Language, es un formato de intercambio estándar basado en XML para redes de Petri.

²⁸ BPMN-Q, es un lenguaje de consultas visual para modelos de procesos de negocio, que extiende los elementos del lenguaje BPMN para soportar consultas.

²⁹ OWL (Ontology Web Language), es un lenguaje de marcado para publicar y compartir datos utilizando ontologías sobre RDF.

Process Handbook [50] (El manual de procesos) es una base de conocimiento de descripciones de procesos; este consta de: Un esquema para organizar modelos de procesos y sus actividades, una colección de modelos de procesos que está organizado de acuerdo a ésta clasificación y una colección de herramientas para la gestión de la base de conocimiento. El enfoque principal del Process Handbook, es organizar el conocimiento relacionado a los procesos y no proveer información relacionada a los modelos de procesos. Consecuentemente, el repositorio se basa en texto más que en modelos y el metamodelo principalmente contiene clases que almacenan modelos de procesos y sus descripciones para relacionar los modelos de procesos uno a uno y con sus respectivos elementos. Los modelos de procesos se almacenan a través de herramientas propietarias, en las cuales la información se ingresa en forma de texto. Internamente los modelos de procesos son almacenados en una base de datos genérica. Solamente se enfoca en almacenar las referencias a los procesos y tenerlos disponibles como una base de conocimiento generado, además que solamente se hace una descripción de los procesos en lenguaje natural (texto) y no especifica un tipo de almacenamiento determinado.

Process Reuse Architecture [51] (La arquitectura de reutilización de Procesos), es una arquitectura y una herramienta para almacenar modelos de procesos reutilizables, ya sea en parte o en su totalidad. El principal tipo de proceso de la arquitectura consiste en procesos de referencia configurables, llamados procesos del framework. Para configurar un proceso de referencia, la arquitectura también almacena patrones de procesos y modelos de procesos, específicos de una compañía. Estos pueden ser reutilizados en parte o totalmente, para adaptar un proceso de referencia configurable, a los requerimientos de una organización en particular. Así como The Process Handbook, esta arquitectura se enfoca en describir los procesos más que en el modelado de los mismos. Sin embargo, esta arquitectura tiene un conjunto más elaborado de conceptos para la creación de estas descripciones, cubriendo aspectos como: características del proceso, organización, estructura y relación de los modelos de procesos a controles y objetivos estratégicos. Los diferentes tipos de procesos pueden ser almacenados en formato XML, la búsqueda de procesos se realiza basada en texto. Al igual que el anterior, solamente hace descripciones de los procesos en lenguaje natural (texto) y no se enfoca en el modelado.

Library for Process Programming [52] (La librería para la programación de procesos), es una arquitectura para organizar modelos de procesos de negocio en librerías, en los cuales, los modelos de procesos de negocio individuales pueden ser reutilizados. Esta librería se enfoca en la definición de procesos, representación de instancias de procesos y adaptación de los procesos para su reutilización. Con el propósito de definir procesos, el sistema introduce un lenguaje llamado "P-language", que se parece a un lenguaje de programación textual. Este se enfoca en la definición de procesos y actividades, flujo de control y datos utilizados por los procesos. La notación de los modelos, así como la recuperación de los mismos, utiliza un lenguaje propietario denominado "P-language", para el almacenamiento utiliza sistemas de archivos.

Repository for workflows systems (El repositorio para sistemas de workflows) [53], es un proyecto que provee: un modelo conceptual para almacenar modelos de workflows, los requerimientos para gestionar repositorios de modelos de workflows y una arquitectura diseñada para implementar un gestor de repositorio. Este repositorio soporta los aspectos de actividades, flujo de control, datos, recursos y monitoreo, además almacena workflows en una base de datos orientada a objetos. Esta base de datos provee las funcionalidades básicas para almacenar y recuperar workflows. No especifica un lenguaje de modelado, ni tampoco un lenguaje para el almacenamiento y recuperación de los workflows.

ProcessGene [54] es un proyecto que provee una herramienta para la consulta de modelos de procesos de negocio. Consta de 4 partes: Ámbito asistencial (SA), Interface de especificación de consultas (QSI), un intérprete de consultas (QI) y un empaquetador de resultados de consultas (QRP). Los usuarios de ProcessGene, proveen los alcances de las preguntas y las especificaciones a través de los módulos SA y QSI, entonces el módulo QI compila la especificación a los requerimientos de consulta y el módulo QRP retorna los resultados de la consulta. ProcessGene soporta las funcionalidades estándar de almacenar y recuperar modelos de procesos. No utiliza un lenguaje específico de modelado y las consultas se realizan a través de SQL, por lo tanto no permite realizar consultas semánticas.

Process Variant Repository (PVR) [55] provee mecanismos para tratar con diferentes variantes de un modelo de procesos de negocio, en tiempo de ejecución. Para este propósito, esta provee funcionalidades para almacenar un modelo de proceso de negocio (específico para una compañía), junto con restricciones variaciones permitidas de ese modelo de proceso de negocio. La información histórica acerca de las variaciones son realizadas en tiempo de ejecución y también son almacenadas, de tal forma que pueden ser utilizadas para mejorar el modelo de proceso de negocio. PVR soporta las actividades, el flujo de control, los datos, los recursos, aspectos de monitoreo y autorización y almacena procesos específicos de compañías, instancias de procesos e información histórica acerca de las instancias. Aunque PVR es diseñado para la consulta de variantes de procesos (registros), también provee soporte para consultar definiciones de procesos, además, provee funcionalidades estándar para almacenar y recuperar modelos de procesos y sus variantes. El lenguaje de modelado es propietario y utiliza un lenguaje de consultas de definiciones e instancias, pero no utiliza un lenguaje específico para el almacenamiento y recuperación.

Querying Framework [56] es un framework para el desarrollo de mecanismos avanzados de búsqueda para un repositorio de procesos de negocio. Este framework soporta los aspectos de actividades, flujo de control, datos, objetivos, recursos, incluyendo información de estos recursos y autorización. De esta forma, el aspecto de datos es soportado en un alto nivel de abstracción. Además, almacena información acerca de estos aspectos, basándose en un repositorio de ontologías RDF, el lenguaje necesario para ingresar esta información es WSML y la información puede ser almacenada en patrones de procesos, fragmentos de procesos, procesos específicos de compañías y, aunque

estos no están explícitamente mencionados, referencias de procesos. El repositorio puede ser consultado utilizando expresiones lógicas WSML, y no tiene un lenguaje de modelado definido. Es un framework que no está disponible en línea y es resultado de un proyecto de investigación sin aplicaciones disponibles.

Prosero [57] es un proyecto que combina la gestión de procesos de negocio y la arquitectura orientada a servicios (SOA), que soporta outsourcing³⁰ de procesos de negocio basado en servicios Web. La semántica de este repositorio consta de 4 componentes: la terminología, el repositorio de modelos de referencia (RMR), el repositorio de servicios Web (WSR) y el repositorio de modelos del Usuario (CMR). Estos componentes pueden ser utilizados para almacenar modelos de procesos de referencia y modelos de procesos específicos de compañías y las instancias de esos procesos. Prosero tiene una interface externa para herramientas de modelado (estructura organizacional del modelado, modelado de datos y modelado de procesos de negocio) y el motor de ActiveBPEL4People. Los aspectos soportados por Prosero son: actividad, flujo de control, datos, recursos, autorización y estructura organizacional; además almacena procesos descritos en BPMN y ejecuta procesos BPEL, debido a que tiene un generador de BPEL que puede transformar BPMN en BPEL. No tiene un lenguaje definido para la recuperación de procesos y no se permite realizar consultas semánticas.

OSIRIS (Open Service Infrastructure for Reliable and Integrated process Support) ha sido propuesto para la ejecución de procesos peer-to-peer. Este repositorio se enfoca en el almacenamiento de modelos de procesos de negocio, especificaciones de servicios, así como éstas sean provistas o utilizadas por los procesos de negocio e instancias de modelos de procesos de negocio ejecutables. Además, OSIRIS provee funciones típicas peer-to-peer, así como el control de concurrencia³¹ y el balance de carga³². De igual forma, OSIRIS soporta los aspectos de actividad, flujo de control y datos y almacena modelos de procesos (o servicios) específicos de compañías e instancias de procesos en ejecución. Internamente, los modelos de procesos y las especificaciones de los servicios son almacenados en una base de datos. Utiliza una notación propietaria para el modelado de procesos y tiene funcionalidades básicas para el almacenamiento y recuperación de los modelos de procesos, aunque no especifica el lenguaje empleado.

APROMORE [58] es un repositorio avanzado que mantiene, analiza y reutiliza grandes colecciones de modelos de procesos; además es una plataforma de código abierto, implementado de acuerdo a la arquitectura SOA y se presenta al usuario a través de servicios Web. La representación de los procesos se basa en un formato canónico y utiliza EPC y BPMN como lenguajes de modelado. Actualmente hay una versión de un prototipo disponible en la red y contiene unas funcionalidades básicas: importar y exportar modelos, búsqueda y clasificación de modelos, así como funcionalidades de comparación

³⁰ El outsourcing o subcontratación, se refiere a un sistema utilizado por grandes compañías que rentan los servicios de pequeñas empresas para efectuar proyectos.

³¹ El control de concurrencia se asegura que los resultados correctos para operaciones concurrentes estén generados, mientras que se consigue esos resultados en el menor tiempo posible.

³² El balance de carga, es un término en informática, que se refiere a la técnica utilizada para compartir el trabajo entre varios procesos, equipos, discos u otros recursos.

(búsqueda de similitud de procesos y funcionalidad de gestión de procesos). No proporciona consultas semánticas.

2.3 Resumen

De esta manera se configura el contexto general de las tecnologías que están alrededor de los procesos de negocio, incluye algunos conceptos relacionados a los servicios Web, vinculados con los procesos de negocio, y los diferentes lenguajes que permiten la descripción, anotación y modelado de estos procesos, basándose en el trabajo expuesto en [19], donde se realiza un análisis a través de un metamodelo que incluye un amplio rango de conceptos de evaluación, explica algunas de las ventajas inherentes al lenguaje BPMO, con respecto a los otros lenguajes y que, finalmente, conduce a seleccionarlo para el modelado de los procesos de negocio de este trabajo.

La revisión de los repositorios de servicios Web y de procesos de negocio, disponibles en el entorno académico y empresarial, junto con la descripción de algunos de ellos, tomando como referencia la forma de almacenamiento, el tipo de consultas y la recuperación de los procesos almacenados, y la evaluación de los diferentes repositorios de procesos de negocio, tomando como referencia el trabajo presentado en [36], (donde se evalúan 16 repositorios de procesos), permitió conformar la base de conocimiento necesaria para avanzar en la propuesta del Repositorio de Procesos de Negocio, aplicado en este proyecto, el cual se explicará en detalle en el siguiente Capítulo.

Capítulo III

3 REPOSITORIO DE PROCESOS DE NEGOCIO

Esta sección trata principalmente el Repositorio de Procesos de Negocio propuesto. Inicialmente, se define una arquitectura de referencia basada en [36], y posteriormente se explican los módulos que la componen y sus respectivas funcionalidades. Hay que resaltar que los modelos de procesos de negocio, que se almacenan en el repositorio, son transformados en una representación formal basada en grafos; por tal motivo, se presentan algunos conceptos fundamentales de la teoría de grafos, para comprender este proceso de transformación. Además, se describe cómo la problemática de la recuperación de procesos de negocio, puede convertirse en una problemática de detección de isomorfismo de grafos. Finalmente, se explica la detección de patrones de flujo de control, como subestructuras de grafos almacenados en el repositorio, y el posterior proceso de ordenamiento y presentación de resultados al usuario (Ranking de modelos de procesos).

3.1 Conceptos fundamentales de Grafos

La teoría de grafos tiene aplicación en campos tan diversos como las ciencias sociales, la lingüística, ciencias físicas, ingeniería de la comunicación, entre otros. Entre las aplicaciones más destacables se pueden considerar las siguientes: Juego de raciocinio, generación de ambientes de realidad virtual, Mapas, GPS, rutas entre ciudades, determinar máximos y mínimos en un proceso, flujo y control en un programa, entre otros.

3.1.1 Clasificaciones de Grafos

Un Grafo en matemáticas y ciencias de la computación, se representa con un conjunto de puntos llamados vértices o nodos, unidos a través de líneas denominadas aristas o arcos. Dependiendo de la aplicación, las aristas pueden ser dirigidas o no dirigidas. En la Figura 3 se ilustra el ejemplo de un grafo equivalente a un Mapa de Carreteras entre diferentes ciudades.

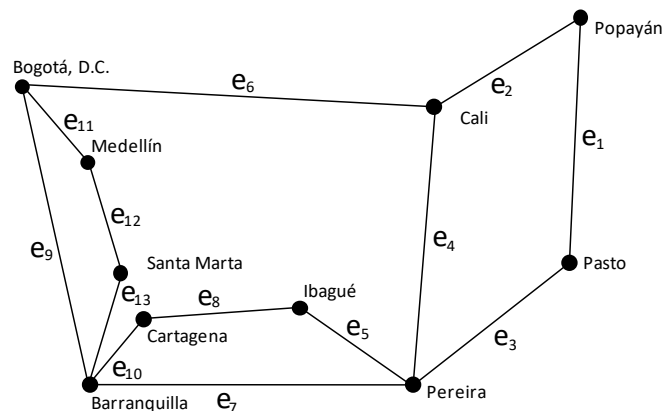


Figura 3. Grafo equivalente a un Mapa de Carreteras

Algunas de las definiciones matemáticas utilizadas, en el área de los grafos, son presentadas a continuación:

Sea V un conjunto finito no vacío a cuyos elementos se les conoce como vértices y sea E un conjunto de pares no ordenados de V a cuyos elementos se les conoce como aristas, al par (V, E) se le llama grafo no dirigido.

Si una arista $e \in E$ está asociada a los vértices v y w escribiremos $e = \{v, w\}$. Podría ocurrir que $v = w$. Un vértice puede estar asociado a 0 aristas, pero toda arista une uno o dos vértices. Cuando dos vértices están asociados a una arista se dice que son adyacentes, y a ellos se les llama extremos de la arista. En el ejemplo anterior el conjunto de vértices sería:

$V = \{\text{Pasto, Popayán, Santa Marta, Pereira, Ibagué, Bogotá D.C., Barranquilla, Cartagena, Medellín, Cali}\}$

Y el conjunto de aristas de la Figura 2 queda descrito a continuación:

- | | |
|--|---|
| $e_1 = \{\text{Pasto, Popayán}\}$ | $e_8 = \{\text{Cartagena, Ibagué}\}$ |
| $e_2 = \{\text{Popayán, Cali}\}$ | $e_9 = \{\text{Bogotá, D.C., Barranquilla}\}$ |
| $e_3 = \{\text{Pereira, Pasto}\}$ | $e_{10} = \{\text{Cartagena, Barranquilla}\}$ |
| $e_4 = \{\text{Cali, Pereira}\}$ | $e_{11} = \{\text{Bogotá, D.C., Medellín}\}$ |
| $e_5 = \{\text{Ibagué, Pereira}\}$ | $e_{12} = \{\text{Medellín, Santa Marta}\}$ |
| $e_6 = \{\text{Bogotá, D.C., Cali}\}$ | $e_{13} = \{\text{Santa Marta, Barranquilla}\}$ |
| $e_7 = \{\text{Barranquilla, Pereira}\}$ | |

A partir de la Figura 4, se hacen otras definiciones básicas de grafos. La arista e_1 está asociada al par ordenado (v_2, v_1) , en donde v_2 es el origen y v_1 el extremo. La arista e_7 se asocia al par ordenado (v_6, v_6) y en este caso, el origen y el extremo coinciden. Se llama lazo o también loop ó bucle.

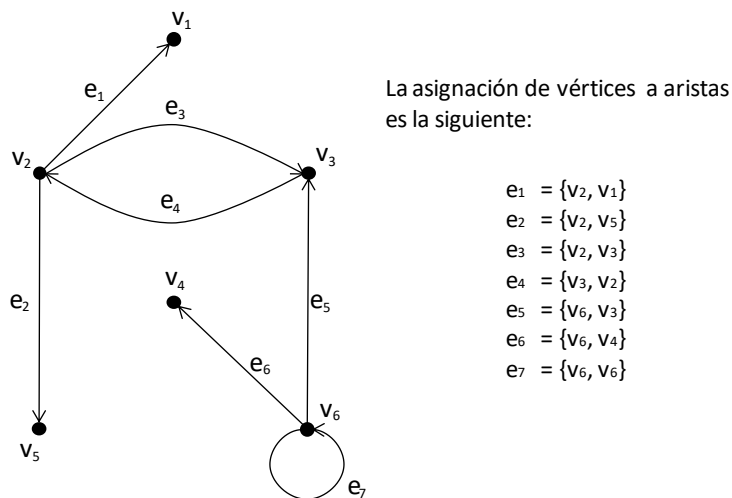


Figura 4. Ejemplo de Grafo con varias asignaciones de vértices.

Los grafos (dirigidos o no) que no tienen lazos ni más de una arista adyacente al mismo par de vértices, se llaman grafos simples. Un grafo completo, K_n , con n vértices es un grafo simple no dirigido, en el que existe una arista uniendo cada par de vértices distintos. Un grafo con n vértices y dirigido se dice grafo dirigido completo cuando es simple y para cada par de vértices u, v existe exactamente una de las aristas (u, v) ó (v, u) . A dichos gráficos se les denota por K_n^* .

Es importante resaltar que los procesos de negocio modelados en BPMO tienen un flujo de control entre tareas, eventos y compuertas, es decir, tienen una dirección y un solo sentido entre cada uno de los elementos del flujo de control. Por ejemplo en la Figura 3, se observa que cada una de las aristas tiene un nodo de origen y un nodo de destino: e_1 se asocia al par ordenado (v_2, v_1) , donde v_2 es el origen y v_1 es el destino.

Dado que el mecanismo de búsqueda propuesto, está basado en el razonamiento sobre las actividades que componen el flujo de control, una representación formal basada en grafos del flujo de control de los procesos de negocio, permite utilizar algoritmos de detección de subestructuras de grafos, para detectar los patrones de flujo de control que caracterizan dichos modelos de procesos. Una vez que los procesos tienen su representación formal basada en grafos, se procede a la detección de subestructuras basada en isomorfismo de subgrafos. En la sección 3.1.2 se presenta una definición matemática del isomorfismo de subgrafos, mientras que las subestructuras, denominadas patrones de flujo de control, se explican más adelante en la sección 3.2.3.

3.1.2 Isomorfismo de subgrafos

Considérese dos grafos G y G' ; un isomorfismo de grafos entre G and G' es un mapeo biyectivo $f: V \rightarrow V'$ tal que $\alpha(v) = \alpha'(f(v)) \forall v \in V$. Para cualquier arista $e = (u, v) \in E$ existe una arista $e' = (f(u), f(v)) \in E'$ tal que $\beta(e) = \beta'(e')$ y para cualquier arista $e' = (u', v') \in E'$ existe una arista $e = (f^{-1}(u'), f^{-1}(v')) \in E$ tal que $\beta(e) = \beta'(e')$. Si $f: V \rightarrow V'$ es un isomorfismo entre los grafos G y G' , y G' es un subgrafo de otro grafo G'' , es decir $G' \subset G''$, entonces la función f es llamada un isomorfismo de subgrafos de $G \subset G''$. Por ejemplo, la Figura 5 presenta un grafo G y un subgrafo suyo G_1 .

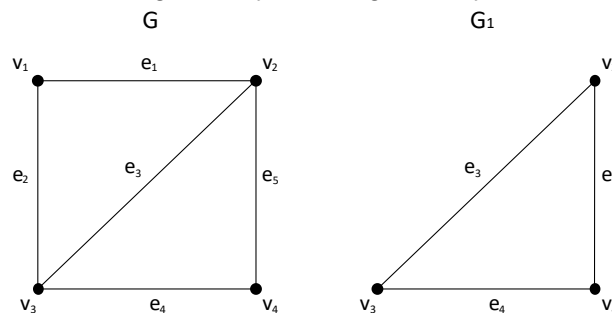


Figura 5. G_1 es un subgrafo de G.

3.1.3 Método de indexación de grafos

A partir de los lenguajes de modelado de procesos de negocio, disponibles en el entorno académico y empresarial mencionados en el capítulo anterior, y el análisis fundamentado en [19], se concluye que el lenguaje más apropiado para esta propuesta es BPMO. De igual manera, en el trabajo señalado en [36], los resultados obtenidos sobre las características de búsqueda y almacenamiento en repositorios de procesos de negocio, incluyen el análisis sobre la aplicación de métodos de indexación sobre estos repositorios, llegando a las conclusiones mostradas en la Tabla 1.

	Process meta model			Process storage model			Process index model	
	aspect	type	notation	external	internal	related	classifications	others
MIT Process Handbook	A	Re	Structured natural language	Not specified	Database		Part-Whole, Generalization-Specialization	
Process Reuse Architecture	A, CF, D, R, Au, G, O, MC, M	Re, C, P	Structured natural language	XML	Database		Facets, Patterns	
Process library	A, CF, D	C	P language	P language	P language		Part-Whole, Generalization-Specialization	
IPM	A, CF, D, R, O, M, Au	C	Proprietary (graphical)	IPM-EPDL	Database		Categories	
RepoX	A, CF, D	C	Not specified	XML	Database (tables/objects)			
Workflow repository	A, CF, D, R, M	C, I	Not specified	Not specified	Database			
BPMN Repository	A, CF, D, R, MC, O, M, Au	Re, C, I, H	BPMN	XML	Repository objects	Ontology		
Oryx	A, CF	C	BPMN, EPC, Petri nets	RDF	Database			
BP-Suite	A, CF, D, R, O, M, Au	C, I, H	BPEL	BPEL	Database			
ProcessGene	A, CF, Au	Re, C	Not specified	Not specified	Database		Categories	
PVR	A, CF, D, R, M, Au	C, I, H	Proprietary (graphical)	Not specified	Database			
Querying Framework	A, CF, D, G, R, Au	Re, C, P	Not specified	WSML	Repository Objects (RDF-like)	Ontology		

Tabla 1. Comparación de los repositorios de modelos de procesos de negocio [36].

	Process meta model			Process storage model			Process index model	
	aspect	type	notation	external	internal	related	classifications	others
SBPR	A, CF, D, Au, R	C	BPMO, sBPMN, sEPC, sBPEL	WSML	Database (tables)			
IBM BPEL Repository	A, CF, Au	C	BPEL	BPEL	Repository Objects (EMF)	WSDL		
Prosero	A, CF, D, R, O, Au	Re, C, I	BPMN, BPEL	XML	Database	Terminology, WSDL		
OSIRIS	A, CF, D	C, I	Proprietary	Not specified	Database			

Tabla 1 (cont.). Comparación de los repositorios de modelos de procesos de negocio [36].

La Tabla 1 presenta, entre otros resultados, el análisis basado en 3 características principales: meta modelos del proceso, almacenamiento del modelo del proceso e indexación del modelo de proceso. En este último, es posible apreciar que las técnicas de indexación aplicadas son escasas (recuadros rojos de la Tabla 1), solamente 5 de ellas utilizan como índice común, una clasificación de modelos de procesos, en términos de las funciones de negocio disponibles.

La indexación de procesos de negocio define los índices que identifican los modelos almacenados en el repositorio, y de esta forma, permiten tanto para el sistema como para el usuario, realizar búsquedas rápidas de los procesos o de colecciones de los mismos. Un índice comúnmente usado es la clasificación de modelos de procesos en términos de sus funciones de negocio. Por ejemplo, clasificar los procesos en categorías: ventas, adquisiciones, producción, finanzas, entre otros.

En este caso se utiliza una técnica de indexación de grafos que permita recuperar las representaciones formales basada en grafos, del flujo de control de los procesos de negocio, y de esta manera, transformar el problema de la recuperación de procesos de negocio en un problema de recuperación de grafos. Para entender este proceso, es necesario conocer un poco más sobre algunos trabajos orientados a la perspectiva de la búsqueda, recuperación y minería de grafos³³.

Trabajos tales como [59] y [60] se enfocan en la recuperación de grafos basada en minería de patrones de grafos frecuentes. Para el caso un grafo se considera frecuente, si su frecuencia de ocurrencia³⁴ en un conjunto de datos dado no es menor que el umbral mínimo de ocurrencia. Estas técnicas están basadas en algoritmos de búsqueda en profundidad (DFS)³⁵ y descomposición de subestructuras de patrones. En [61] se

³³ La minería de grafos se podría describir como la extracción no trivial de información que reside de manera implícita en los datos. En otras palabras, la minería de grafos prepara, sondea y explora los datos para sacar información oculta de ellos.

³⁴ Frecuencia de ocurrencia puede entenderse como el número de veces que ocurre un evento determinado.

³⁵ DFS es el acrónimo de Depth First Search se define como un algoritmo que permite recorrer todos los nodos de un grafo o árbol de una manera ordenada, pero no uniforme.

describen las técnicas de minería de patrones de grafos basadas en restricciones. Estas restricciones, reducen el espacio de búsqueda³⁶ y pueden clasificarse en diferentes categorías.

Los autores [62], [63], [64] y [65] definen métodos de indexación de grafos, donde dada una base de datos de grafos y un grafo de consulta, se encuentra todos los grafos que contienen este grafo de consulta, excluyendo a los grafos que no contienen la subestructura del grafo de consulta. El proceso de búsqueda se realiza a través de un algoritmo de isomorfismo de grafos y toma aspectos relacionados a la búsqueda de grafos en una base de datos de grafos vista desde 2 perspectivas:

- i) Búsqueda de grafos: Encuentra todos los grafos contenidos en el grafo de consulta.
- ii) Búsqueda de grafos contenidos: Encuentra todos los grafos contenidos por el grafo de consulta.

Estas 2 técnicas permiten hacer una búsqueda exacta e inexacta de patrones predefinidos dentro del grafo de consulta a través de la búsqueda de subestructuras de grafos. Un trabajo realizado en el 2010 integra algunas de las características de los repositorios de procesos analizados en el Capítulo 2, junto con la búsqueda y minería de grafos descritos anteriormente. Este trabajo se encuentra en [66] y propone un método eficiente para la consulta de modelos de procesos de negocio en repositorios, teniendo un fragmento del modelo de proceso (ó un modelo de consulta). Por este método se encuentran todos los modelos de procesos en el repositorio que contienen ese fragmento. Los resultados se filtran a través del uso de índices, obteniendo un conjunto de modelos de procesos candidatos. Luego, se aplica una detección de isomorfismo de grafos sobre este conjunto de procesos candidatos, a través de una adaptación del algoritmo de Ullman [67]. Cabe mencionar que este trabajo propuesto carece de una ontología que permita realizar razonamiento semántico.

De los anteriores trabajos relacionados, se seleccionó a GraphBlast [63] como el método de indexación de grafos, que permite realizar una búsqueda de patrones de manera exacta e inexacta dentro de una base de datos de grafos, el cual incluye un algoritmo de isomorfismo de grafos llamado VF2 [68], que se explica de forma más detallada en la sección 3.2. Este algoritmo presenta mayor rendimiento en el tiempo de respuesta en comparación con el algoritmo de Ullman presentado en [66]; además, cabe resaltar que en el presente trabajo se toman como referencia algunas características del repositorio de procesos APROMORE [58], con el fin de participar en una representación común en cuanto al formato canónico de los modelos de procesos. En ambos casos, esta corresponde con una representación formal basada en grafos.

³⁶ Un espacio de búsqueda se refiere al dominio de la función a ser optimizada. En el caso de algoritmos, que manejan espacios discretos, se refiere al conjunto de todas las posibles soluciones candidatas a un problema.

3.2 Arquitectura Inicial de Referencia

A continuación, se presenta la arquitectura del Repositorio de Procesos de Negocio propuesto a partir del trabajo desarrollado en [36], que en adelante se denominará *Repositorio de Modelos BP* (Figura 6), así como también se presentan los modelos de procesos de negocio almacenados en el repositorio, que en adelante se denominarán *Modelos BP*. La arquitectura está compuesta por 5 capas: *Capa de Presentación*, *Capa de Transformación de BP*, *Capa de Gestión de BP*, *Capa de Gestión de Repositorio* y *Capa de Almacenamiento*.

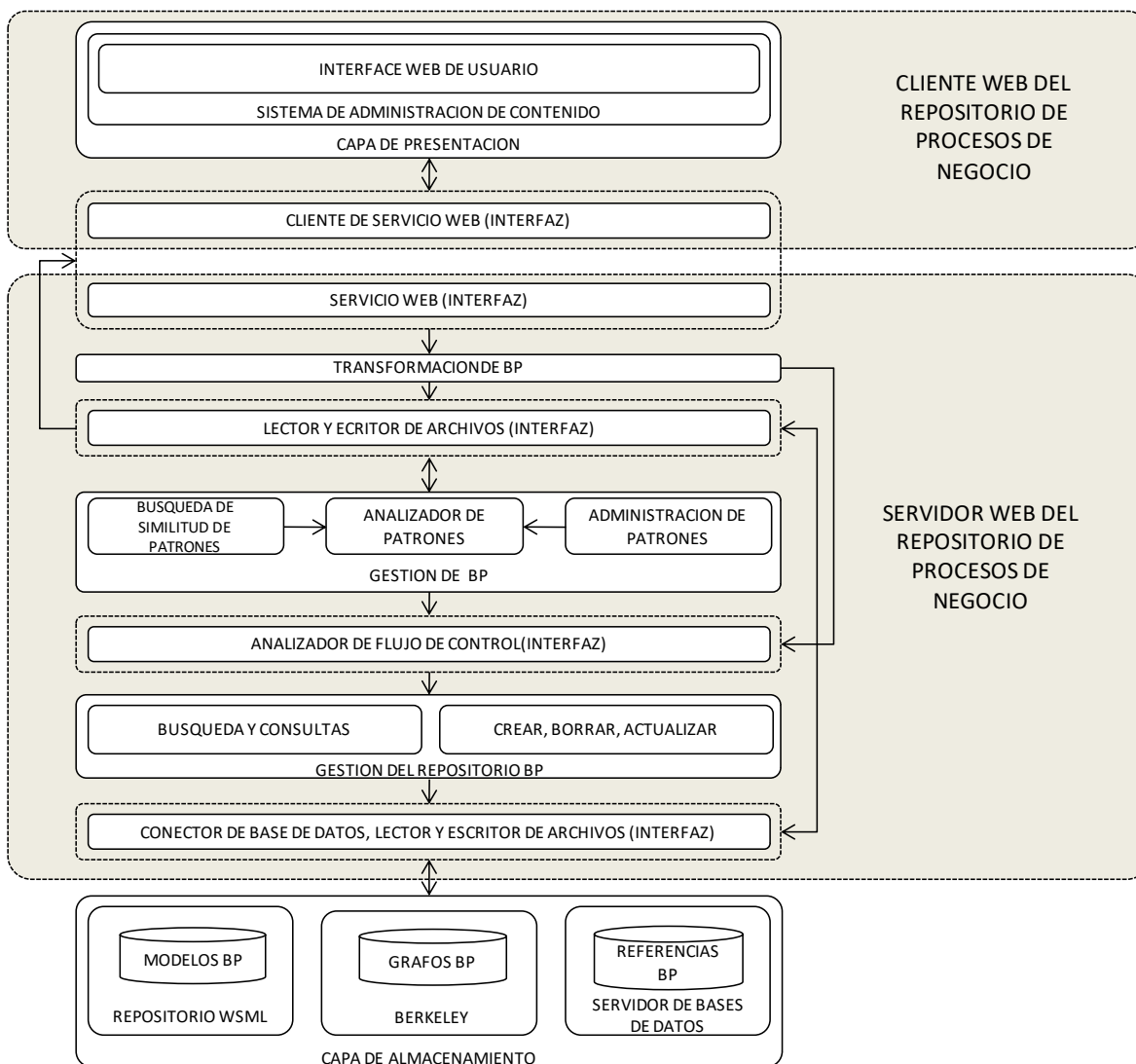


Figura 6. Arquitectura propuesta para el repositorio de procesos de negocio.

3.2.1 Capa de Presentación

Esta capa contiene la lógica de presentación para una aplicación Web, compuesta por un sistema de gestión de contenidos (CMS)³⁷ y provee funcionalidades tales como administración Web y gestión de cuentas de usuarios. A través de formularios, el usuario puede cargar modelos de procesos de negocio al servidor, además de lanzar consultas sobre los procesos almacenados y visualizar los resultados obtenidos, con un buen grado de usabilidad, así como también permite el acceso a los servicios del repositorio publicados en el servidor como servicios Web. En la sección 5.1.4 del Capítulo 5 se presenta una descripción detallada de la capa de presentación correspondiente al prototipo implementado.

3.2.2 Capa de transformación de BP

Esta capa genera una representación formal basada en grafos, a partir de un Modelo BP. Se recibe como entrada un Modelo BP descrito en WSML (Web Service Modeling Language) que contiene un conjunto de tareas, eventos y compuertas. La salida entrega un grafo denominado (Grafo BP) en donde las tareas y eventos se representan como nodos y las compuertas por nodos tipo AND Split,Join; OR Split,Join; XOR Split,Join; Además, los conectores de flujo de control se representan con aristas, esto con el objetivo de generar el Grafo BP a partir del cual se puede hacer la detección de patrones mediante el uso de GraphBlast. La Figura 7 ilustra un ejemplo gráfico de un Modelo BP realizado a través de la versión 1.4 del modelador BPMO. El Anexo No 2 expone una descripción detallada del modelador BPMO, así como también un ejemplo gráfico de un Modelo BP junto con su contenido en formato WSML.

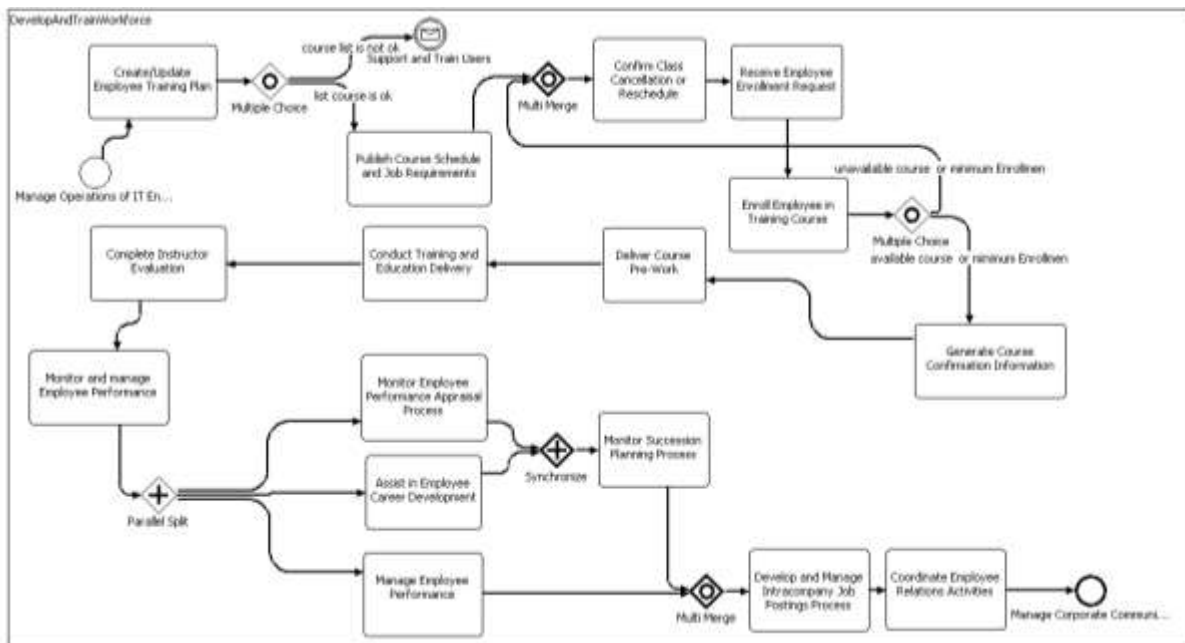


Figura 7. Representación gráfica del Modelo BP de ejemplo "Develop and Train Workforce" descrito en WSML a partir del Modelador BPMO versión 1.4

³⁷ Un CMS, se podría definir como un programa que permite crear una estructura de soporte para la creación y administración de contenidos, principalmente en páginas web, de manera colaborativa por parte de los participantes.

La Tabla 2, presenta la correspondiente representación basada en grafos para cada elemento de BPMO. Es importante destacar que este trabajo está basado en las 11 compuertas definidas en la versión 1.4 del modelador BPMO.








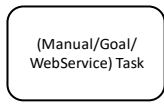
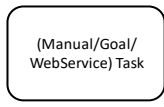
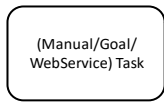


















Elementos BPMO	Etiqueta del Grafo	Representación del Grafo
Eventos Start  End  Timer  Error  Send/Receive Message  	Eventos Los eventos se toman como nodos de tipo Event representado por las etiquetas (E)	 Eventos (Start, End, Timer, Error, SendMessage, ReceiveMessage)
Tareas ManualTask  GoalTask  WebServiceTask 	Tareas Las Tareas se toman como nodos de tipo Task y representados por las etiquetas (T)	 Tareas (Manual, Goal, WebService)
Compuertas DeferredChoice  ExclusiveChoice  InterleavedParallelRouting  MultipleInstantiation 	Compuertas Las Compuertas se toman como nodos de tipo AND, OR, XOR Split;Join. Y son representados por las etiquetas (ANDS, ANDJ, ORS, ORJ, XORS, XORJ)	 Compuertas (DeferredChoice, ExclusiveChoice, InterleavedParallelRouting, Multiple Instantiation)
Discriminator  SimpleMerge 	Los nodos de tipo XOR Split son representados por Etiquetas (XORS)	 Compuertas (Discriminator, SimpleMerge)
ParallelSplit 	Los nodos de tipo AND Split son representados por las etiquetas (ANDS)	 Compuerta ParallelSplit
MultipleMergeSynchronise  Synchronisation 	Los nodos de tipo AND Join son representados por las etiquetas (ANDJ)	 Compuertas (MultipleMergeSynchronise, Synchronisation)
Multimerge 	Los nodos de tipo OR Join son representados por las etiquetas (ORJ)	 Compuerta Multimerge
MultipleChoice 	Los nodos de tipo OR Split son representados por las etiquetas (ORS)	 Compuerta MultipleChoice

Tabla 2. Elementos BPMO y su correspondiente representación en grafos [69].

De manera similar, se presenta un meta modelo utilizando un diagrama de UML, basándose en el trabajo propuesto en [58]. Éste meta modelo muestra una representación abstracta de los conceptos relacionados a las tareas, eventos y compuertas, así como sus interacciones (Figura 7). Un Modelo BP (ProcessModel) contiene un BP Graph (ProcessGraph), el cual es un grafo dirigido compuesto por Nodos (Node) y Aristas (Edge). Los Nodos pueden ser de tipo Function o Connector, mientras que las aristas pueden ser de tipo SequenceFlow o MessageFlow. Los nodos compuerta (Connector) son todos los elementos del Modelo BP, que son utilizados para describir el flujo de control y estos pueden tener muchas aristas de entrada y una de salida o muchas aristas de salida y una de entrada. Los nodos tipo Connector pueden ser OR, XOR, AND (Split, Join) y el nodo Wait, indica una pausa de tiempo definido en el flujo de control. Los elementos Split tienen una arista de entrada y múltiples aristas de salida. Los elementos de tipo Join tienen múltiples aristas de entrada y una arista de salida. Los nodos tipo Function son importantes desde la perspectiva del negocio; estos tienen al menos una arista de entrada y una de salida y son clasificados en nodos de tipo Task y Event. El nodo tipo Task es un elemento de proceso el cual realiza actividades como parte de un proceso, por ejemplo: actividad de configuración de un equipo o la actividad de traducción de un idioma a otro. Los nodos tipo Task también se clasifican como tareas de tipo Manual, WebService o Goal. Finalmente, los Events son utilizados para mostrar el inicio o fin (Start o End de un proceso), así como también presenta el envío y recepción de mensajes entre diferentes Modelos BP.

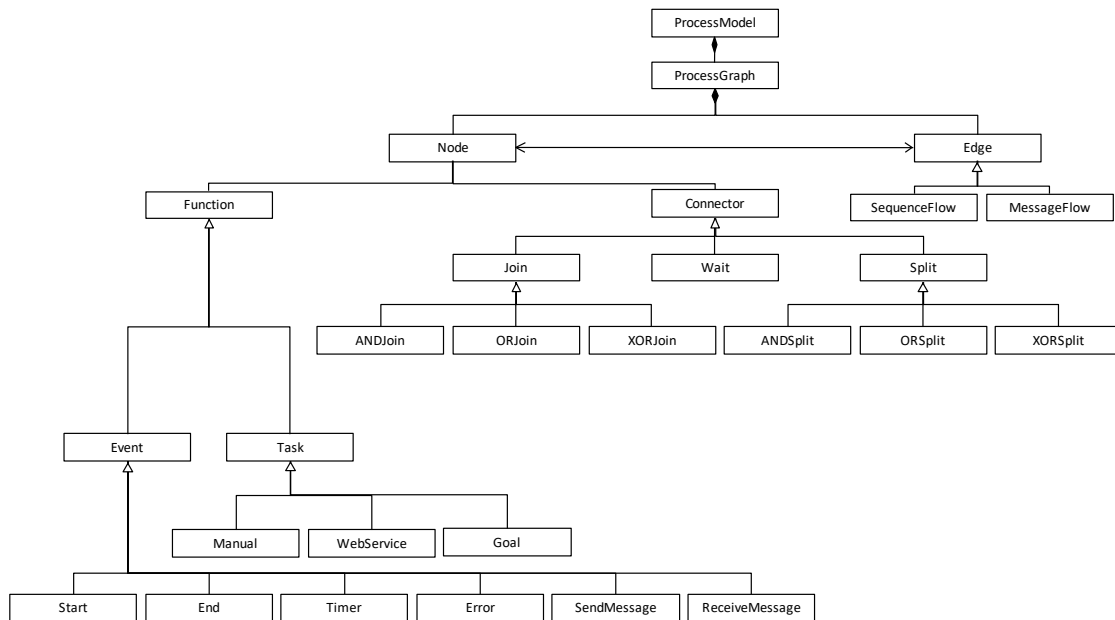


Figura 8. Meta modelo de Referencia.

De esta forma, la clasificación de los nodos compuerta (Connector) en nodos que divergen (SPLIT) el flujo de control y en nodos que convergen (JOIN) el flujo de control, permite generar estructuras que realicen ciclos como lo son las estructuras de tipo WHILE (en la cual primero se evalúa la condición del ciclo y luego se realiza la acción correspondiente a la tarea) o REPEAT (en la cual se realiza primero la acción correspondiente a la tarea y luego se evalúa la condición del ciclo), o la combinación de

estas WHILE-REPEAT (en la cual se realiza primero la acción correspondiente a una tarea, luego se evalúa la condición del ciclo y finalmente se realiza la acción correspondiente a otra tarea). La Figura 9, presenta la representación basada en grafos para una estructura SPLIT-JOIN, y para los ciclos WHILE y REPEAT de un Modelo BP.

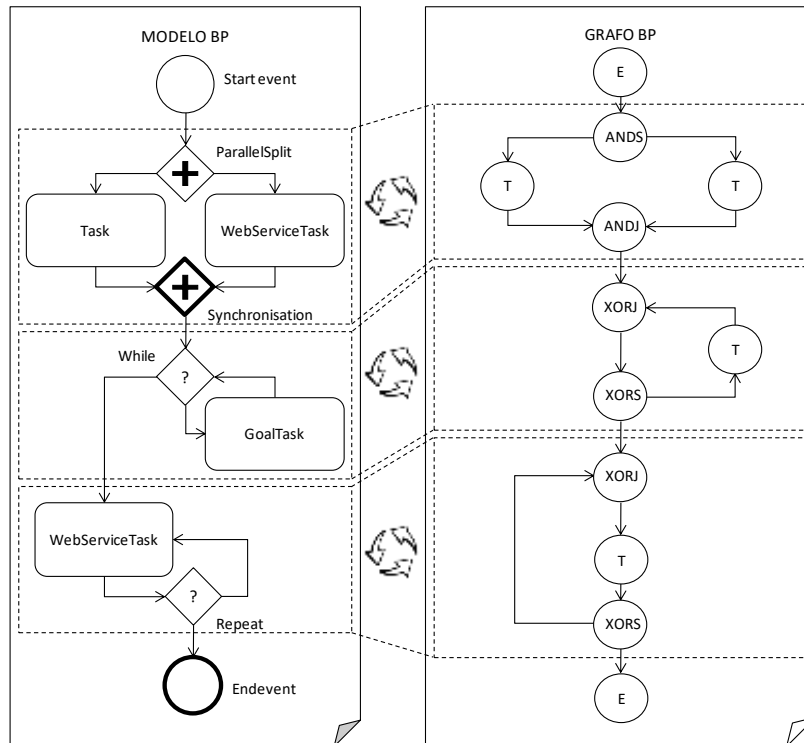


Figura 9. Representación basada en grafos de las estructuras que divergen, convergen y realizan ciclos en el flujo de control de un Modelo BP.

Transformación de un Modelo BP a un Grafo BP

Una vez creado el Modelo BP, la primera fase del proceso de transformación de éste en un Grafo BP, consiste en el uso del API WSMO4J para hacer el proceso de carga del archivo y generar su representación como un objeto Java en memoria (Capas WSMO4J y Carga BPMO). Esta representación del Modelo BP contiene toda la información descrita en el archivo, la cual puede ser accedida a través de métodos provistos por dicha API (Figura 10). A partir de este objeto Java en memoria, se puede realizar un análisis de la estructura de flujo de control, para conseguir la posterior detección de patrones de flujo de control (Capa BPMO Backtracking).

El objetivo principal es generar un Grafo BP a partir del cual se pueda hacer la detección de patrones mediante el uso de GraphBlast (Capas de Transformación BPMO y el API de transformación de BPMO), para lograr este objetivo, se implementó una adaptación del algoritmo de detección de trazas TDA (Algoritmo 1) aplicado en [70], que permite recorrer todo el contenido del objeto Java correspondiente al Modelo BP para generar el Grafo BP (Capa Algoritmo TDA).

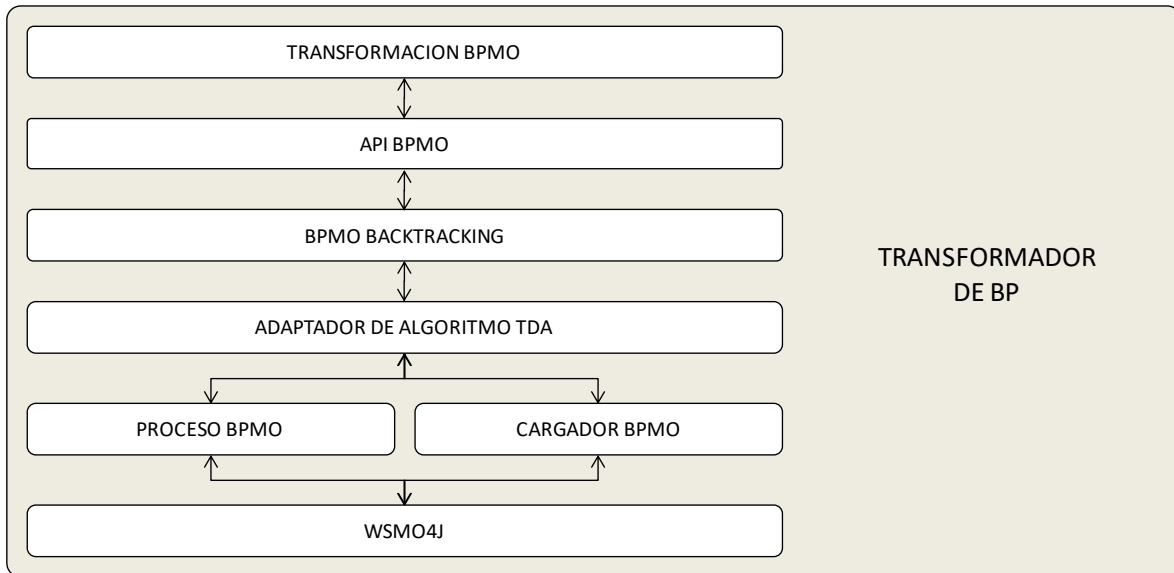


Figura 10. Diagrama modular del proceso de transformación de Modelo BP a Grafo BP.

Algoritmo 1. TDA (Trace Detection Algorithm): Algoritmo de detección de trazas

INPUT: Graph G; **OUTPUT:** Traces Set

1. **GET** Start Node i
 2. **ADD** i to *SetVisitedNodes*
 3. **APPLY** the AdjacencyFunction (i) and **ADD** n to *SetNeighbors*
 4. **IF** *SetNeighbors* > 1 **THEN**
 5. **ADD** *SetVisitedNodes* to *BackTrace*
 6. **IF** *SetNeighbors* = 0 **THEN** **END**
 7. **WHILE** *NodesToVisit* > 0 **THEN**
 8. **GET** (j) the last node **ADDED** to *NodesToVisit*
 9. **IF** (j) is the Last node of G **THEN**
 10. **ADD** (j) to *SetVisitedNodes*
 11. **ADD** *SetVisitedNodes* to *TracesSet*
 12. **IF** *BackTrace* > 0 **THEN**
 13. **GET** Last Set of *BackTrace*
 14. **REPLACE** *SetVisitedNodes* **FOR** *BackTrace*
 15. **ELSE** **RETURN** *TracesSet*
 16. **ELSE**
 17. **ADD** (j) to *SetVisitedNodes*
 18. **APPLY** the AdjacencyFunction (j) **ADD** m to *SetNeighbors*
 19. **IF** m > 1 **THEN**
 20. **ADD** *SetVisitedNodes* to *BackTrace*
 21. **END WHILE**
-

Durante la segunda fase del proceso de transformación, el algoritmo TDA crea varias rutas llamadas "Trazas". Estas trazas son definidas evaluando cada nodo conector, de tal forma que un conector XOR-Split puede crear muchas trazas, así como se evalúen condiciones u ocurran eventos. Por otro lado, un conector AND-Split crea una traza para cada actividad ejecutada. Cuando el algoritmo llega al nodo final, este retrocede al conector que tiene una condición (o condiciones) para evaluar y construir otra traza. El Anexo 3 contiene una descripción del funcionamiento del algoritmo TDA, a través de un ejemplo detallado.

A continuación, en la Figura 11(a) se describe el contenido del Grafo BP generado por la adaptación del algoritmo TDA, el cual es un documento .dat dividido en cinco secciones, las cuales son explicadas según el orden en el cual aparecen al interior del archivo. La primera sección del contenido del Grafo BP corresponde a la primera línea del archivo, consiste en el identificador del archivo del grafo, el cual tiene una única restricción de comenzar con el carácter "#". La segunda sección del Grafo BP corresponde a la segunda línea del archivo y consta del número de nodos del Grafo BP, el cual como mínimo debe ser "2", ya que si es menor no existiría flujo de control alguno.

La tercera sección del Grafo BP, hace referencia a las etiquetas de los nodos, las cuales están asociadas con la posición de los nodos en los grafos y la dirección del flujo de control según el orden en el cual aparecen al interior del archivo. Por ejemplo, la primera etiqueta "E" corresponde al nodo de inicio, el cual tiene "0" como identificador numérico de flujo de control; la segunda etiqueta "T" corresponde al nodo de destino y tiene "1" como identificador numérico de flujo de control y así sucesivamente hasta llegar a la última etiqueta dentro de esta sección. La cuarta sección corresponde al número de aristas del Grafo BP, el cual como mínimo debe ser "1", de acuerdo con la restricción del número de nodos de la segunda sección del Grafo BP, la quinta sección corresponde a las etiquetas de las aristas, las cuales constan del identificador numérico del nodo origen y el nodo destino separados por un "space bar". En cuanto a la representación gráfica del Grafo BP, presentada en la parte (b) de la Figura 11, se evidencia por medio del comportamiento dirigido de sus aristas, la manera en la cual el algoritmo TDA asigna los identificadores numéricos de cada nodo, a medida que recorre cada una de las rutas del Modelo BP, además del etiquetado correspondiente para los elementos de flujo de control del Modelo BP de ejemplo "*Develop and train workforce.wsmf*" descrito en la Figura 7.

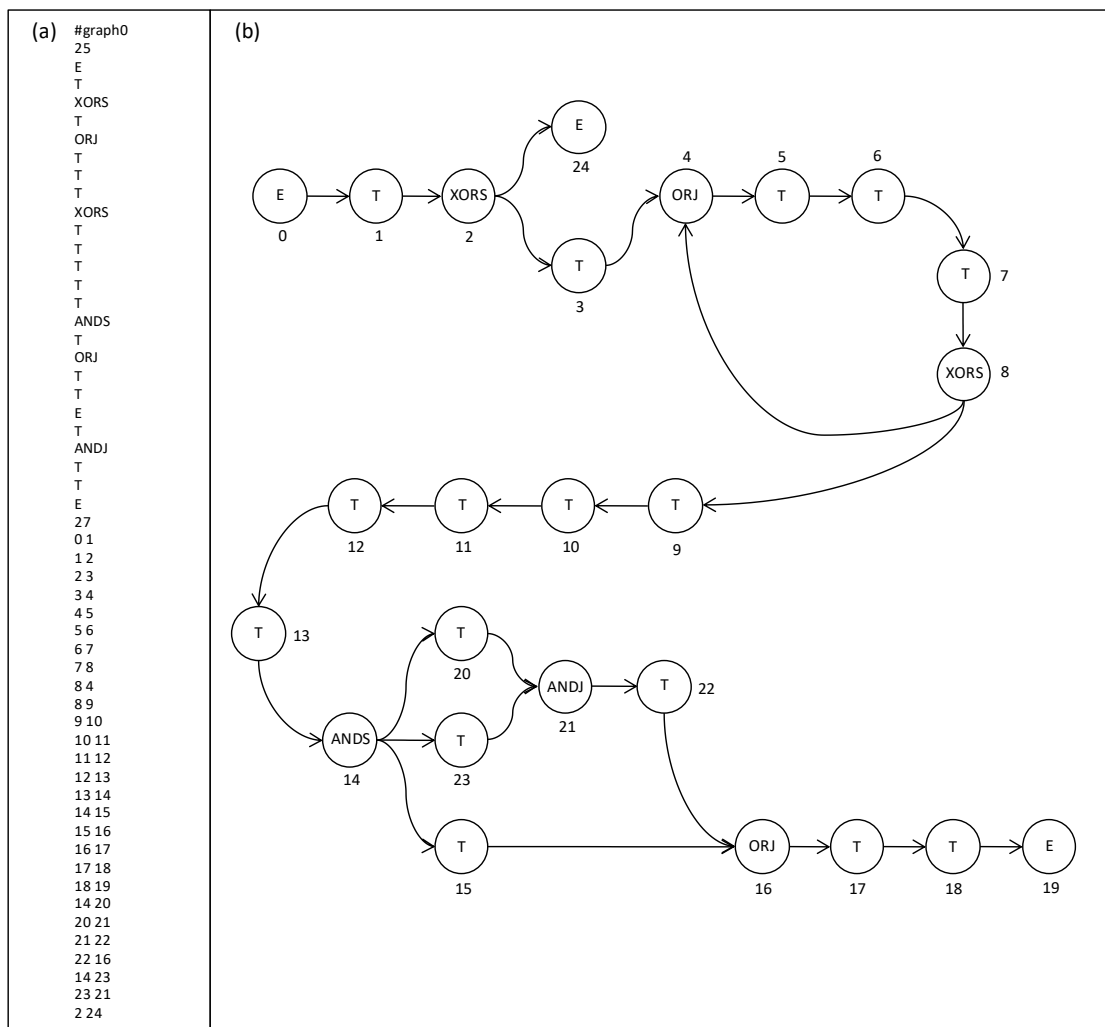


Figura 11. (a) Contenido del archivo .dat del Grafo BP y (b) Su representación gráfica.

3.2.3 Capa de Gestión de BP

Esta capa está compuesta por los siguientes módulos: Gestión de Patrones, Analizador de Patrones y Búsqueda de Similitud de Patrones. A continuación, se explica cada uno de ellos de manera detallada.

- **Gestión de Patrones:**

Para lograr el objetivo de realizar una búsqueda semántica por medio del razonamiento ontológico, sobre las actividades que compone el flujo de control, el presente trabajo propone un mecanismo de búsqueda a partir de la clasificación de los Modelos BP, según los patrones de flujo de control contenidos al interior de estos. Un trabajo relacionado con la clasificación de modelos de procesos de negocio según el flujo de control se presenta en [71], allí se hace una evaluación detallada de los diferentes sistemas de workflows y las notaciones de modelos de procesos de negocio que los soportan, con el propósito de identificar estructuras genéricas y recurrentes (Tabla 3). Por tal motivo, inicialmente se establece un conjunto de 20

patrones desde la perspectiva del flujo de control, los cuales fueron ampliamente adoptados por los teóricos y prácticos, quienes discutieron y compararon las capacidades y la expresividad de estos patrones, de tal manera que fuesen independientes de formalismos y lenguajes específicos.

Pattern	Staffware	WebSphere MQ	FLOWer	COSA	iPlanet	SAP Workflow	FileNet	BPEL	WebSphere BPEL	Oracle BPEL	BPMN	XPDL	UML ADs	EPC
1 (seq)	+	+	+	+	+	+	+	+	+	+	+	+	+	+
2 (par-spl)	+	+	+	+	+	+	+	+	+	+	+	+	+	+
3 (synch)	+	+	+	+	+	+	+	+	+	+	+	+	+	+
4 (ex-ch)	+	+	+	+	+	+	+	+	+	+	+	+	+	+
5 (simple-m)	+	+	+	+	+	+	+	+	+	+	+	+	+	+
6 (m-choice)	-	+	+	+	+	-	+	+	+	+	+	+	+	+
7 (s-sync-m)	-	+	+	-	-	-	+	+	+	+	+	+	-	+
8 (multi-m)	-	-	+/-	+/-	+	-	+	-	-	-	+	+	+	-
9 (s-disc)	-	-	-	-	+	+/-	-	-	-	-	+/-	+/-	+/-	-
10 (arb-c)	+	-	-	+	+	-	+	-	-	-	+	+	+	+
11 (impl-t)	+	+	+	-	-	-	+	+	+	+	+	+	+	+
12 (mi-no-s)	+	-	+	+	+	+/-	+	+	+	+	+	+	+	-
13 (mi-dt)	+	-	+	-	-	+	-	-	-	+	+	+	+	-
14 (mi-rt)	+	-	+	-	-	+	-	-	-	+	+	+	+	-
15 (mi-no)	-	-	+	-	-	-	-	-	-	-	-	-	-	-
16 (def-c)	-	-	+	+	-	-	+/-	+	+	+	+	+	+	-
17 (int-par)	-	-	+/-	+	-	-	-	+/-	+/-	-	-	-	-	-
18 (milest)	-	-	+/-	+	-	-	-	-	-	-	-	-	-	-
19 (can-a)	+	-	+/-	+	+	+	+	+	+	+	+	+	+	-
20 (can-c)	-	-	+/-	-	-	+	+	+	+	+	+	+	+	-

Tabla 3. Resultado de la evaluación – 20 Patrones de flujo de control iniciales [36].

Un estudio más profundo de ellos da origen a la definición de un número adicional de patrones (Tabla 4), el cual genera un total de 43 patrones desde el punto de vista del flujo de control, ésta nueva definición de patrones incluye combinaciones de los 20 patrones iniciales.

Pattern	Staffware	WebSphere	FLOWer	COSA	iPlanet	SAP Workflow	FileNet	BPEL	WebSphere BPEL	Oracle BPEL	BPMN	XPDL	UML ADs	EPC
21 (str-l)	-	+	+	-	+	+	+	+	+	+	+	+	+	-
22 (recur)	+	+	-	+	+	+	-	-	-	-	-	-	-	-
23 (t-trig)	+	-	-	+	-	+	-	-	-	-	-	-	+	-
24 (p-trig)	-	-	+	+	-	+	+	+	+	+	+	+	+	+/-
25 (can-r)	-	-	-	+/-	-	-	-	+/-	+/-	+/-	+/-	+/-	+	-
26 (can-mi)	+	-	-	-	-	+	-	-	-	+	+	+	+	-
27 (comp-mi)	-	-	+/-	-	-	-	-	-	-	-	-	-	-	-
28 (b-disc)	-	-	-	-	-	-	-	-	-	-	+/-	+/-	+/-	-
29 (c-disc)	-	-	-	-	-	+	-	-	-	-	+	+	+	-
30 (s-pjoin)	-	-	-	-	+	+/-	-	-	-	-	+/-	+/-	+/-	-
31 (b-pjoin)	-	-	-	-	-	-	-	-	-	-	+/-	+/-	+/-	-
32 (c-pjoin)	-	-	-	-	-	+	-	-	-	-	+/-	+/-	+	-
33 (g-and-join)	-	-	-	-	-	-	+	-	-	-	+	+	-	+/-
34 (st-pjoin-mi)	-	-	-	-	-	-	-	-	-	-	+/-	+/-	-	-
35 (c-pjoin-mi)	-	-	-	-	-	-	-	-	-	-	+/-	+/-	-	-
36 (dyn-pjoin-mi)	-	-	-	-	-	-	-	-	-	-	-	-	-	-
37 (a-sync-m)	-	+	+	+	-	-	-	+	+	+	-	-	+/-	+
38 (g-sync-m)	-	-	-	-	-	-	+	-	-	-	-	-	-	-
39 (crit-sec)	-	-	+/-	+	-	-	-	+	+	+	-	-	-	-
40 (int-rout)	-	-	+/-	+	-	-	-	+	+	-	+/-	+/-	-	-
41 (tm)	-	-	-	-	-	-	-	+/-	+/-	+/-	+	+	+	-
42 (ts)	-	-	-	-	-	-	-	+/-	+/-	+/-	+	+	+	-
43 (exp-t)	-	-	-	+	+	+	-	-	-	-	+	+	+	-

Tabla 4. Patrones adicionales – Total 43 Patrones de flujo de control [36].

Además, según [71] los patrones de flujo de control pueden ser clasificados en las siguientes subcategorías:

- ✓ **Patrones de flujo de control básicos (Basic Control Flow Patterns)**
Esta clase de patrones captura los aspectos elementales del flujo de control de proceso y son similares a las definiciones de estos conceptos inicialmente propuestos por el WfMC [16].
- ✓ **Patrones de flujo de control de ramificación y sincronización avanzada (Advanced Branching and Synchronization Patterns)**
Es una serie de patrones que se caracteriza por utilizar conceptos más complejos de ramificación y sincronización que surgen a partir de los procesos de negocio.

✓ **Patrones de flujo de control de múltiples instancias (Multiple Instance Patterns)**

Describen las situaciones en donde hay múltiples hilos de ejecución, activos en un modelo de proceso, que se relaciona a una misma actividad (por lo tanto, comparten la misma definición de la implementación). Las múltiples instancias pueden ser declaradas en 3 situaciones:

1. Una actividad puede iniciar múltiples instancias de sí misma, cuando se ha activado (se denota esta clase de actividad como una actividad de múltiples instancias).
2. Una actividad dada se activa en múltiples ocasiones como una consecuencia de recibir activaciones independientes (por ejemplo una parte de un ciclo o una instancia de un proceso, en el cual hay muchos hilos de ejecución concurrentes, como un resultado de un patrón que une el flujo de control).
3. Dos o más actividades en un proceso comparten la misma definición de implementación. Esto podría ser la misma definición de actividad, como en el caso de múltiples instancias de actividades o una definición de un subproceso común, en el caso de un bloque de actividades. Dos (o más) de estas actividades se inician de tal forma que sus ejecuciones se superponen ya sea parcial o totalmente.

Aunque estas 3 situaciones potencialmente involucran múltiples instancias concurrentes de una actividad o un subproceso, la primera de ellas atrae mayor interés, ya que ellas requieren las activaciones y sincronización de múltiples instancias de actividades concurrentes. Este grupo de patrones se enfoca en las diferentes maneras en las cuales los eventos pueden ocurrir.

✓ **Patrones basados en estados (State-based Patterns)**

Estos muestran la situación donde las soluciones son más fácilmente ejecutadas en los lenguajes de procesos que soporten la noción del estado. En este contexto, se consideran los estados de una instancia de un proceso, para incluir una amplia colección de datos asociados con la ejecución actual, incluyendo el estado de varias actividades, así como también la información relevante al proceso, tal como los elementos de las actividades y los elementos de datos.

✓ **Patrones de cancelación y de terminación forzada (Cancellation and Force Completion Patterns)**

Existen patrones que tienen variantes que utilizan el concepto de cancelación de actividades, donde las instancias de actividades son activadas o

desactivadas. Varias formas de excepción manejadas en los procesos son también basadas en conceptos de cancelación.

- ✓ **Patrones de iteración (Iteration Patterns)**
Capturan un comportamiento repetitivo en un workflow.
- ✓ **Patrones de terminación (Termination Patterns)**
Estos abordan las circunstancias sobre las cuales un workflow se considera como completo.
- ✓ **Patrones de activación (Trigger Patterns)**
Estos se encargan de manejar las señales externas que pueden ser requeridas para iniciar ciertas actividades.

Teniendo en cuenta que la evaluación anterior no considera a BPMO entre las notaciones evaluadas, en el presente trabajo se hizo la evaluación de la viabilidad de implementación de los 43 patrones de flujo de control en la notación BPMO, para esto principalmente se tuvo en cuenta la funcionalidad de cada patrón.

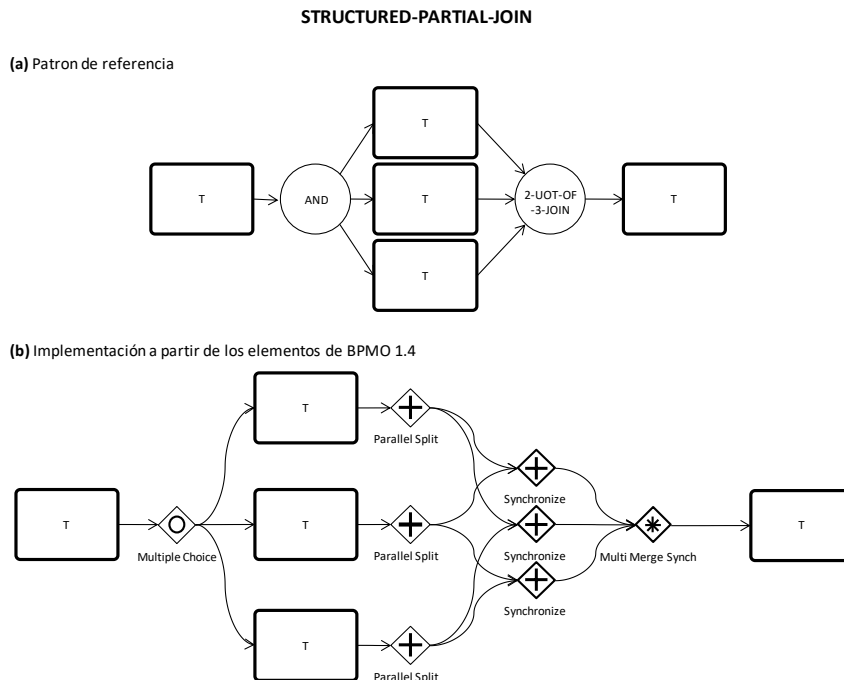


Figura 12. Ejemplo de implementación (STRUCTURED-PARTIAL-JOIN)

Por ejemplo, un patrón descrito en [71] denominado Structured-Partial-Join y cuya funcionalidad consiste en generar un flujo de salida únicamente cuando se cumpla un subgrupo determinado de actividades dentro de un grupo total, es decir, en este caso el elemento “2-out of-3-join” solamente genera un flujo de salida cuando se cumple un subgrupo de 2 actividades de un total de 3 (Figura 12a). El elemento Structured Partial-Join no existe directamente dentro de los elementos de BPMO, pero

analizando su funcionamiento se puede encontrar un equivalente, por medio de una relación de composición a partir de los elementos de BPMO (Figura 12b).

Resultados de la viabilidad de la implementación

Los resultados de la evaluación de la viabilidad de implementación de los patrones en BPMO, según las sub-categorías de control, son (Figura 13):

PATRONES BASICOS DE FLUJO DE CONTROL		PATRONES DE MULTIPLES INSTANCIAS	
1. Sequence	+	12. Multiple Instances without Synchronization	+
2. Parallel Split	+	13. Multiple Instances with a Priori Design-Time Knowledge	+
3. Synchronization	+	14. Multiple Instances with a Priori Run-Time Knowledge	+
4. Exclusive Choice	+	15. Multiple Instances without a Priori Run-Time Knowledge	+/-
5. Simple Merge	+	34. Static Partial Join for Multiple Instances	+/-
		35. Cancelling Partial Join for Multiple Instances	+/-
		36. Dynamic Partial Join for Multiple Instances	+/-
PATRONES DE RAMIFICACION Y SINCRONIZACION AVANZADA		PATRONES DE CANCELACION Y TERMINACION FORZADA	
6. Multi-Choice	+	19. Cancel Task	+
7. Structured Synchronizing Merge	+	20. Cancel Case	+/-
8. Multi-Merge	+	25. Cancel Region	-
9. Structured Discriminator	+	26. Cancel Multiple Instance Activity	-
28. Blocking Discriminator	+/-	27. Complete Multiple Instance Activity	-
29. Cancelling Discriminator	+/-		
30. Structured Partial Join	+	PATRONES DE ITERACION	
31. Blocking Partial Join	+/-	10. Arbitrary Cycles	+
32. Cancelling Partial Join	+/-	21. Structured Loop	+
33. Generalised AND-Join	+	22. Recursion	-
37. Local Synchronizing Merge	+	PATRONES DE TERMINACION	
38. General Synchronizing Merge	+	11. Implicit Termination	-
41. Thread Merge	-	43. Explicit Termination	+/-
42. Thread Split	-	PATRONES DE ACTIVACION	
PATRONES BASADOS EN ESTADOS		23. Transient Trigger	-
16. Deferred Choice	+	24. Persistent Trigger	+/-
17. Interleaved Parallel Routing	+		
18. Milestone	-		
39. Critical Section	-		
40. Interleaved Routing	+/-		
		(+) viabilidad de implementacion ALTA (+/-) viabilidad de implementacion PARCIAL (-) viabilidad de implementacion NULA	
Cantidad patrones de flujo de control	Viabilidad de implementacion en BPMO		
21	ALTA		
12	PARCIAL		
10	NULA		

Figura 13. Resultados del estudio de viabilidad de implementación de los patrones de referencia en la notación BPMO.

Con base en estos resultados, se decidió que el alcance comprende únicamente la sub-categoría de patrones básicos, porque se observó que todos los patrones básicos son implementables en BPMO, así como también los patrones “Multiple Choice”, “Multi Merge” y “Structured Discriminator” de la sub-categoría de Patrones de flujo de control de ramificación y sincronización avanzada, y de igual manera los patrones “Deferred Choice”, “Interleaved Parallel Routing” de la sub-categoría de patrones basados en estados. Además, a partir de las combinaciones entre éstas se generan los 43 patrones restantes. En el Anexo 4, son explicados en forma detallada, los 12 patrones seleccionados y aplicados a este proyecto, de las diferentes sub-categorías explicadas

anteriormente. La Tabla 5, resume la representación basada en grafos de cada uno de estos patrones de BPMO 1.4.

(a) BPMO		(b) GRAPHS
	SEQUENCE	
	PARALLEL SPLIT	
	SYNCHRONIZATION	
	EXCLUSIVE CHOICE	
	SIMPLE MERGE	
	MULTIPLE CHOICE	
	DISCRIMINATOR	
	MULTI MERGE	
	MULTI MERGE SYNCHRONIZATION	
	INTERLEAVED PARALLEL ROUTING	
	MULTIPLE INSTANTIATION	

Tabla 5. Patrones de referencia modelados en (a) BPMO y (b) Representación en grafos.

- **Analizador de Patrones:**

Este módulo es responsable de la detección exacta de los patrones de flujo de control presentados en la Tabla 5, los cuales se encuentran contenidos en un grafo de consulta que representa al Modelo BP, que de ahora en adelante se denominará Grafo de Consulta BP. Para lograr este propósito, se aplica un algoritmo de isomorfismo de grafos llamado VF2 [68] y el método de indexación de grafos es realizado por GraphBlast.

VF2 es un algoritmo de isomorfismo de grafos, el cual es un método de correspondencia determinístico, que permite verificar correspondencias entre grafos. A través de este algoritmo, es posible realizar las comparaciones entre diferentes grafos y determina cuál grafo es igual a otro, en cuanto a su estructura de nodos y aristas (isomorfismo); ó cuál grafo está contenido dentro de otro (subgrafo). El algoritmo es válido mientras que no haya restricciones impuestas a la topología de los grafos.

Algoritmo 2. VF2: Algoritmo de isomorfismo de grafos

PROCEDURE Match (*s*)

INPUT: an intermediate state *s*; the initial state; the initial state s_0 has $M(s_0) = \emptyset$

OUTPUT: the mappings between two graphs

IF $M(s)$ covers all the nodes of G_2 **THEN**

OUTPUT $M(s)$

ELSE

 Compute the set $P(s)$ of the pairs candidate for inclusion in $M(s)$

FOREACH p in $P(s)$

IF the feasibility rules succeed for the inclusion of p in $M(s)$ **THEN**

 Compute the state's obtained by adding p to $M(s)$

CALL $Match(s')$

END IF

END FOREACH

 Restore data structures

END IF

END PROCURE MATCH

Descripción del método de indexación de GraphBlast

GraphBlast utiliza una representación de grafos basada en nodos, que tienen un número de identificación (*node-id*) y una etiqueta de identificación (*node-label*) (Figura 14). Además, se define un identificador de trayectos (*id-path*) de longitud n (esto quiere decir, una lista de $n+1$ *node-ids*) con una arista no etiquetada entre 2 nodos consecutivos y un trayecto de etiquetas (*label-path*) de longitud n como una lista de $n+1$ etiquetas de nodos (*node-labels*). Por ejemplo, un *label-path* es: EANDS, y un *id-path* sería: (0,1).

Para cada grafo y para cada nodo, este encuentra todos los trayectos que empiezan en este nodo y tienen una longitud desde uno hasta un tamaño predefinido, utilizando una variable denominada LP (longitud de trayecto), por defecto $LP=4$, pero este puede variar. La construcción del índice se realiza a través de un conjunto de *id-paths* y *id-labels*, utilizando una tabla de llaves (*hash table*), cuyas llaves son los valores de los *label-paths*. En el presente proyecto se utiliza un *label-path* para definir y describir un patrón de grafo, que se busca dentro de los grafos de procesos almacenados, en otras palabras significa que el *label-path* $h(TTXORJT)$ describe un patrón de grafo predefinido. Entonces, la construcción de los índices se hace por el número de ocurrencias de este *label-path* dentro de cada grafo almacenado (Figura 14b).

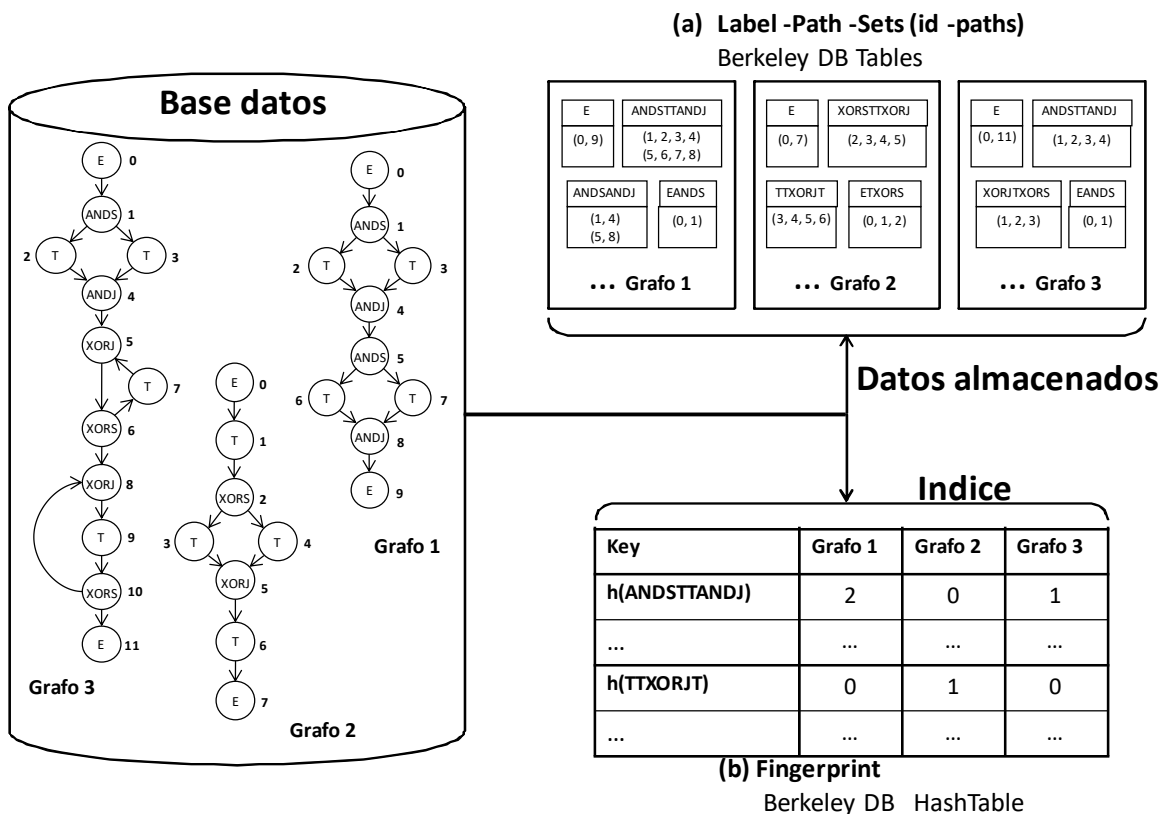


Figura 14. Almacenamiento de los grafos (Data Storing) y la creación del índice (Index) sobre las tablas de la Base de datos de Berkeley

La tabla de llaves (*hash table*) se define como una huella digital (*fingerprint*) de la base de datos y está compuesta por una matriz donde las filas son etiquetadas como *label-paths* y cada columna se asocia a un grafo almacenado en la base de datos. Toda la información relacionada a la construcción del índice se almacena en la base de datos Berkeley [72]. La Figura 14a muestra una colección de *id-paths* de todos los trayectos que representan una secuencia de etiquetas en un conjunto de *label-paths*. Estas colecciones son almacenadas en tablas de la base de datos de Berkeley.

- **Búsqueda de Similitud de Patrones:**

Con la intención de no descartar algunos Modelos BP del repositorio, que son similares al Grafo de Consulta BP, este módulo hace una detección inexacta de subgrafos utilizando un lenguaje llamado GLIDE (Graph Linear DEscriptor) [62]. Las expresiones en GLIDE, denominadas también como expresiones regulares de grafos, realizan la descripción de porciones de grafos y consultas aproximadas, identificando diferentes localizaciones de los patrones de flujo de control dentro del Grafo BP analizado.

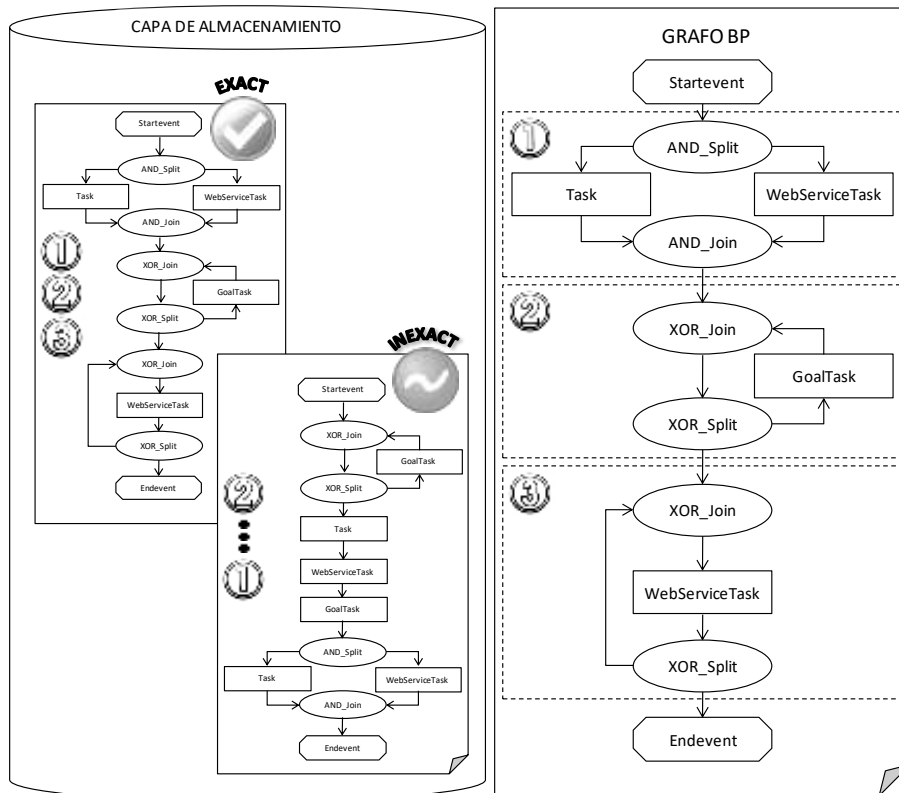


Figura 15. Operación del módulo Búsqueda de Similitud de Patrones.

La Figura 15 presenta un ejemplo de la detección aproximada de subestructuras anidadas entre nodos AND, OR, XOR Split,Join. El Grafo de Consulta BP está compuesto por una secuencia que contiene 3 patrones de flujo de control y contiene las tareas y compuertas que se escriben a continuación:

- (1) AND_Split, Task, WebServiceTask, AND_Join;
- (2) XOR_Join, XOR_Split, GoalTask;
- (3) XOR_Join, WebServiceTask, XOR_Split.

Los 3 patrones se describen en GLIDE así como se muestra en la (Figura 16a). En donde los nodos están expresados solo con sus etiquetas y están unidos por un carácter '/' que representa las aristas. Las ramas se agrupan, usando paréntesis anidados y los ciclos se pueden ver como una arista de corte y un número entero

positivo. Los nodos de las aristas de corte están representados por sus etiquetas, seguido por los caracteres '%', '/' y el número entero. Teniendo en cuenta lo anterior, el usuario puede estructurar una consulta exacta como la de la Figura 15, en el orden de los patrones: (1) (2) (3), utilizando el lenguaje GLIDE de esta manera (Figura 16b):

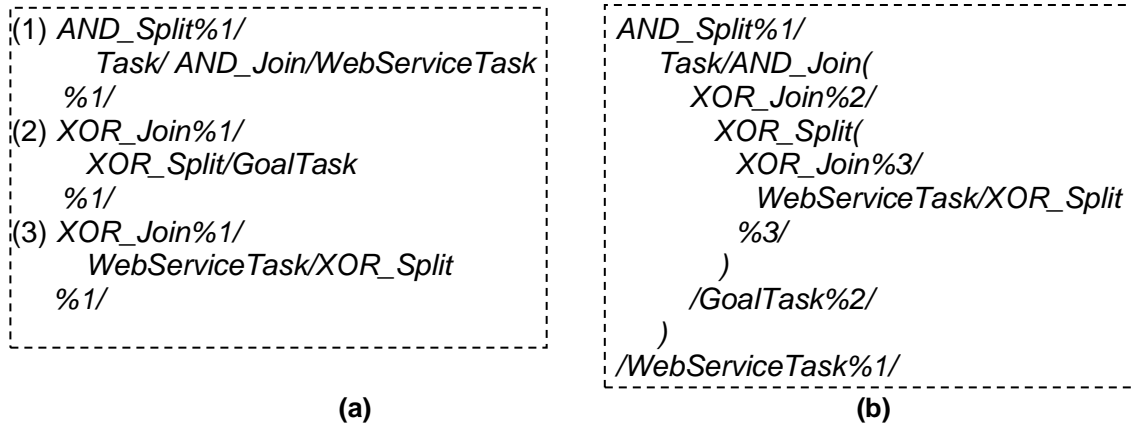


Figura 16. (a) Los Patrones (1) (2) y (3) descritos en Lenguaje GLIDE **(b)** Consulta exacta de los patrones (1) (2) y (3) utilizando el lenguaje GLIDE.

GLIDE también permite realizar consultas inexactas utilizando símbolos (wildcards) tales como: '.' correspondencia entre cualquier nodo individual, '*' correspondencia entre cualquier secuencia de nodos, '?' correspondencia de al menos un nodo, '+' correspondencia de cualquier secuencia de uno o más nodos. En la Figura 17, se presenta una consulta inexacta, donde existe al menos una ruta entre los patrones (2) y (1), la cual se expresa a través del carácter '*'.

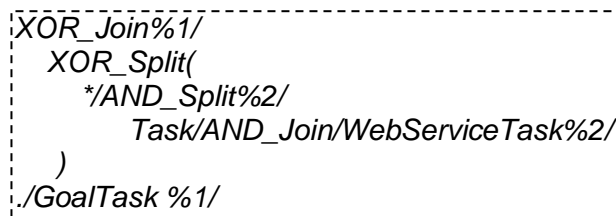


Figura 17. Una consulta inexacta descrita en el lenguaje GLIDE.

Se puede observar que de acuerdo a la secuencia de subestructuras, es posible obtener un resultado de similitud exacta y también un resultado de similitud inexacta, con respecto a los Grafos BP almacenados en el repositorio.

Ranking de Modelos BP

El proceso de ordenar (Ranking) los Modelos BP obtenidos a partir de una consulta realizada por el usuario, está dirigido principalmente por una serie de condiciones que determinan la similitud entre patrones, la similitud entre secuencias de patrones y la similitud semántica entre las funcionalidades de los patrones a partir de las relaciones de composición y especialización entre ellos. En esta sección se abordan las condiciones que determinan los tipos de similitud no semánticos, y en el siguiente Capítulo 4 se abordarán las condiciones para determinar la similitud semántica.

Similitud de Patrones

Tomando como referencia el trabajo definido en [73], en donde se establece una medida de similitud entre nodos pertenecientes a un Grafo BP, en el presente trabajo de grado, se define una medida de similitud para patrones como una adaptación de la medida de similitud para nodos.

Sean Q y T los grafos Consulta (Query) y Objetivo (Target) respectivamente, y P_Q y P_T , la cantidad de patrones detectados en el Query y en el Target, para un determinado tipo de patrón (Parallel Split, Multiple Choice, etc.), la medida de similitud entre la cantidad de patrones de un determinado tipo entre el Query y el Target se define como $Sim_{pattern}(Q, T)$ y está dada por:

$$Sim_{pattern}(Q, T) = \frac{|P_Q \cap P_T|}{|P_Q|}$$

Que representa cuántos de los patrones del Target corresponden con el Query, del total de los patrones del Query. El valor $Sim_{pattern}(Q, T) = 0$, cuando la cantidad de patrones del Target para un determinado tipo de patrón es 0, es decir que el Target no contiene el patrón solicitado en el Query, y $Sim_{pattern}(Q, T) = 1$, cuando la cantidad de patrones del Target para un determinado tipo, es idéntica a la cantidad de dicho patrón solicitada en el Query.

Similitud entre secuencias de Patrones

De manera análoga a la definición $Sim_{pattern}$, considerando una secuencia de patrones como un conjunto de patrones con un orden definido, el cual se puede presentar en varias ocasiones al interior del Query, se define $Sim_{seqPattern}(Q, T)$, como la medida de similitud entre secuencias de patrones por medio de la siguiente expresión:

$$Sim_{seqPattern}(Q, T) = \frac{1}{1 + |P_{seq}(Q)| + |P_{seq}(T)| - 2|P_{seq}(Q) \cap P_{seq}(T)|}$$

En esta función de similitud, el parámetro P_{seq} representa el número de patrones consecutivos ($P_{seq}(Q)$ para el grafo Query y $P_{seq}(T)$ para el grafo Target), y $P_{seq}(Q) \cap P_{seq}(T)$ representa las secuencias mapeadas en un grafo Query con respecto al grafo Target. Por ejemplo, si se selecciona $P_{seq}(Q) = 3$ (secuencia triple), entonces la función considerará todas las secuencias de 3 patrones consecutivos del grafo Query y del grafo Target y sus intersecciones. De esta forma, $tri_{pattern}(Q)$, $tri_{pattern}(T)$ y $|tri_{pattern}(Q) \cap tri_{pattern}(T)|$ se calculan. Además, se puede apreciar que $Sim_{seqPattern}(Q, T)$ nunca toma el valor de 0, pero si valores muy cercanos a 0, cuando las cantidades de secuencias son muy elevadas.

Condiciones del Ranking

Con el propósito de ordenar (Ranking) los Modelos BP, se define $Sim_{Total}(Q, T)$ como una medida de similitud en la que participan las medidas previamente establecidas: $Sim_{Pattern}(Q, T)$ y $Sim_{SeqPattern}(Q, T)$ por medio de la siguiente expresión:

$$Sim_{Total}(Q, T) = \frac{\alpha * Sim_{Pattern}(Q, T) + \beta * Sim_{SeqPattern}(Q, T)}{2}$$

En esta medida, los valores de los coeficientes de Ranking α y β son personalizados, de acuerdo a los criterios de pertinencia del usuario, tomando valores tales que $0 \leq \alpha \leq 1$ y $0 \leq \beta \leq 1$ y $\alpha + \beta = 1$. Con base en esta medida se ordenan los Modelos BP recuperados.

3.2.4 Capa de Gestión del Repositorio

Esta capa contiene los métodos y controladores necesarios para la creación, lectura, actualización y borrado (CRUD), de la información del repositorio por medio de 3 elementos:

BP Reference: Base de datos relacional que almacena la información asociada con los Modelos BP y Patrones de BP (*bpreference*).

BP Model: Sistema de archivos que almacena el contenido de archivos WSML que describen los modelos de procesos de negocio (*wsmfFiles*).

BP Graph: Sistema de archivos que almacena el contenido de archivos DAT, que describen los grafos de los modelos de procesos de negocio divididos en *graphProcessFiles*, *graphPatternFiles*, *BPGraphFocus* y *BPGraphIndex*.

3.2.5 Storage Layer

Esta capa contiene 3 repositorios. El primero, es un sistema de archivos que es el responsable de almacenar los documentos WSML, éste trabaja con funciones de lectura y escritura de archivos del sistema operativo. El segundo, se denomina Berkeley DB y se encarga de almacenar los Grafos BP, a través de una relación directa con GraphBlast. Por último, el tercero utiliza una base de datos relacional con el propósito de entregar resultados ágiles al usuario, esta consideración se tomó de las recomendaciones para la recuperación ágil de información en sistemas de información, descritas en [74]. Además, en la base de datos se almacena toda la información relacionada a la detección de los patrones de un Modelo BP, tal como el tipo de patrón y su cantidad. La Figura 18, muestra una abstracción del diagrama entidad relación correspondiente basado en [48].

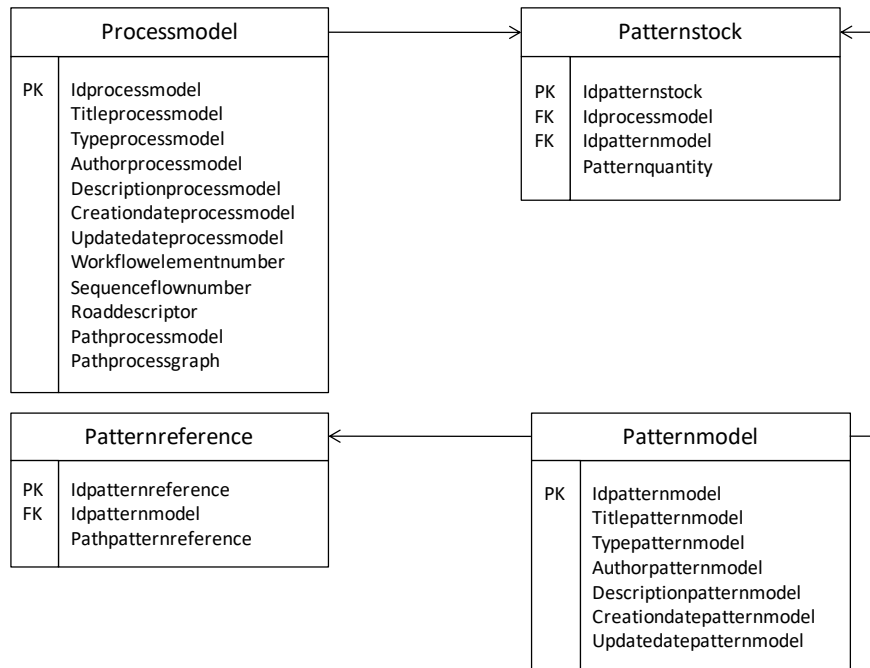


Figura 18. Abstracción del diagrama de entidad relación de la base de datos de referencias.

3.3 Resumen

La teoría de grafos e isomorfismo de grafos aportan conceptos fundamentales que permiten comprender el tratamiento y el proceso de transformación realizado a los procesos de negocio modelados en BPMO, y de esta forma, generar modelos formales representados en grafos. Para el método de indexación de procesos aplicado en el presente trabajo, se tomó como referencia algunos trabajos de recuperación y minería de grafos, seleccionando a GraphBlast [63] como el método de indexación de grafos que permite realizar una búsqueda de patrones exacto e inexacto dentro del repositorio. Además, se demostró cómo la problemática de la recuperación de procesos de negocio se puede convertir en una problemática de detección de isomorfismo de grafos.

A partir de la arquitectura inicial de referencia presentada en [36], se propone y describe la arquitectura del Repositorio de Procesos de Negocio construida en este trabajo, explicando detalladamente las características de funcionamiento de los módulos que la componen. Finalmente, el proceso de ordenamiento (Ranking) de Modelos BP, se realiza a partir de las medidas de similitud de patrones, similitud de secuencias de patrones y similitud total, teniendo en cuenta los patrones de flujo de control que componen los modelos de procesos de negocio. A partir de esta arquitectura, es posible generar una adaptación de búsqueda semántica al proceso de recuperación de Modelos BP, la cual se expone en detalle en el Capítulo 4.

Capítulo IV

4 BÚSQUEDA SEMÁNTICA

En este capítulo se abarca la explicación de la integración de una ontología de patrones de flujo de control, que permite adicionar características de búsqueda semántica a la recuperación de Modelos BP realizada en el presente proyecto (sección 3.2). Para esto, se abordan conceptos relacionados a la búsqueda semántica y al manejo de ontologías que permiten la posterior descripción de la formulación de la ontología de patrones de flujo de control, y la explicación de la definición de una medida de similitud semántica entre patrones aplicada durante el proceso de ordenamiento (Ranking) de Modelos BP recuperados.

4.1 Conceptos fundamentales

4.1.1 Web semántica

Dado que el presente proyecto propone una búsqueda semántica de Modelos BP en un repositorio, el cual es desplegado en un entorno Web, es pertinente conocer la relación entre la semántica y la Web actual, las cuales dan origen a la Web semántica.

El término semántica se refiere a los aspectos del significado, sentido o interpretación de un determinado elemento, símbolo, palabra, expresión o representación formal. La integración de la Web y la semántica producen la Web semántica, la cual propone superar las limitaciones de la Web actual, mediante la introducción de descripciones explícitas del significado en la estructura interna y la estructura global de los contenidos y servicios disponibles en la WWW (World Wide Web) [2]. Según el W3C (World Wide Web Consortium) define la Web Semántica como “una Web extendida, dotada de mayor significado en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida a través de lenguaje natural” [3].

Un servicio de razonamiento desplegado en Internet llamado IRS (Internet Reasoning Service), es la plataforma de servicios Web semánticos del KMI (Knowledge Media Institute) de la Open University, el cual permite a las aplicaciones describir semánticamente servicios Web y ejecutarlos. IRS soporta la provisión de servicios de razonamiento semántico en el contexto de la Web semántica.

4.1.2 Ontologías

De manera concreta, “Una ontología es la especificación formal y explícita de una conceptualización compartida” [75]. En esta definición, conceptualización hace referencia a un modelo abstracto de algún fenómeno del mundo, del que se identifican los conceptos que son relevantes; explícito se refiere a la necesidad de especificar de forma consciente los distintos conceptos que conforman una ontología; formal indica que la especificación debe representarse por medio de un lenguaje de representación formalizado, y compartida refleja que una ontología debe, en el mejor de los casos, dar cuenta de un conocimiento aceptado.

A través de los siguientes elementos, las ontologías pueden representar el conocimiento de un dominio específico:

- **Conceptos:** son las ideas básicas que se intentan formalizar, es decir los elementos básicos de la terminología de dominio que representan clases de objetos del mundo, tanto abstracto como real, y comparten propiedades específicas.
- **Relaciones:** representan la interacción y enlace entre los conceptos del dominio. Suelen formar la taxonomía del dominio.
- **Funciones:** son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología.
- **Instancias:** se utilizan para representar objetos determinados de un concepto.
- **Axiomas:** son reglas de restricción o teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología y permiten inferir conocimiento que no está indicado explícitamente en ella.

Inicialmente es necesario puntualizar que las ontologías cumplen un papel fundamental dentro de la Web Semántica y más aún, en la interoperabilidad semántica entre los sistemas de información, estableciendo correspondencias y relaciones entre los diferentes dominios de aplicación [18]. Con lo cual, una ontología, desde el punto de vista informático, constituye cierta estructura que especifica un vocabulario relativo a un cierto dominio. Este vocabulario define entidades, clases, propiedades, predicados y funciones y las relaciones entre estos componentes [19]. Los principales estándares de ontologías son RDF (Resource Description Framework), DAML (DARPA (Defense Advanced Research Projects Agency) Agent Markup Language), OIL (Ontology Interface Layer) y OWL (Web Ontology Language), WSML (Web Service Modeling Language).

Un análisis más profundo permite evidenciar que la importancia de las ontologías radica en la conversión de la información en conocimiento. Esto es posible a través de metadatos, los cuales son estructuras de conocimiento formalizadas, que referencian los datos, bajo un esquema común normalizado sobre algún dominio del conocimiento. Los

metadatos no sólo especifican el esquema de datos, que debe aparecer en cada instancia, sino que también pueden contener información adicional de cómo hacer deducciones sobre ellos, es decir, cómo establecer axiomas que podrán, a su vez, aplicarse en los diferentes dominios que trate el conocimiento almacenado. De esta forma, en las búsquedas es posible obtener información al compartir los mismos esquemas de anotaciones y, además, realizar inferencias de forma automática buscando información relacionada con los requerimientos de las consultas realizadas por los usuarios.

Los beneficios de utilizar ontologías se pueden resumir de la siguiente forma:

- Proporcionan una forma de representar y compartir el conocimiento utilizando un vocabulario común.
- Permiten usar un formato de intercambio de conocimiento.
- Proporcionan un protocolo específico de comunicación.
- Permiten facilitar la reutilización del conocimiento.

En síntesis, la adición de una ontología de patrones de flujo de control que permita adicionar características semánticas a la recuperación de Modelos BP, constituye un mecanismo que permite recuperar conocimiento, por medio de la capacidad de inferir a partir del razonamiento sobre los axiomas y relaciones entre los elementos pertenecientes al dominio del flujo de control de los procesos de negocio.

4.2 Ontología de Patrones de flujo de Control

4.2.1 Ontologías de procesos de negocio

El primer paso para lograr la definición de una ontología de patrones de flujo de control de procesos de negocio, fue realizar una investigación acerca de las ontologías existentes relacionadas al dominio de los procesos de negocio. Como resultado de dicha investigación, se generó una base de conocimiento de ontologías del dominio de los procesos de negocio, entre las cuales están las ontologías formuladas por el Proyecto SUPER, Sembiz³⁸ y m3pe³⁹. Una vez realizada la investigación, se utilizaron estas ontologías como referencia para la formulación de la ontología de patrones del presente proyecto, las cuales fueron definidas y publicadas entre el 2008 y el 2009.

³⁸ El proyecto Sembiz aspira reducir la brecha entre el nivel de la perspectiva del negocio y la implementación técnica en el nivel de Gestión de Procesos de Negocio (BPM) a través de descripciones semánticas de los procesos de negocio junto con las herramientas de soporte necesarias.

³⁹ El proyecto m3pe, (multi meta model process engineering) fue concebido con el propósito de alcanzar la interoperabilidad de workflows a través de un modelo unificado de modelos de procesos, llamado multi meta model.

4.2.2 Clasificación de patrones

Tomando como referencia el estudio realizado en [73], en el cual se hace una definición general de los patrones de flujo de control para procesos de negocio, en el presente proyecto se realizó una clasificación de los patrones básicos definidos para BPMO, según sus funcionalidad y comportamiento.

Clasificación de patrones según su funcionalidad

Las categorías definidas para la clasificación de patrones según su funcionalidad son: AND, XOR, OR y SEQ. La primera de ellas denominada AND, hace referencia a un funcionamiento síncrono del flujo de control, es decir que un patrón que sincronice el flujo de control puede ser clasificado como tipo AND. La segunda categoría es denominada XOR y hace referencia a un funcionamiento asíncrono del flujo de control, es decir que un patrón que permita generar asincronía en el flujo de control, puede ser clasificado como tipo XOR. La tercera categoría es denominada OR y su funcionamiento consiste en que de acuerdo a una serie de condiciones establecidas, puede permitir la sincronía o asincronía del flujo de control. Finalmente, la cuarta categoría es denominada SEQ, y hace referencia a un funcionamiento secuencial del flujo de control, en el cual no se presenta sincronía ni asincronía.

Clasificación de patrones según su comportamiento

Por otra parte, las categorías definidas para la clasificación de patrones según su comportamiento son SPLIT, JOIN y KEEP. La primera de estas tres categorías denominada SPLIT hace referencia a los patrones que presentan un comportamiento de divergencia en el flujo de control en 2 o más ramas, en contraste con la categoría ya mencionada, la segunda categoría denominada JOIN, hace referencia a los patrones que presentan un comportamiento de convergencia en el flujo de control de 2 o más ramas, Finalmente, la tercera categoría denominada KEEP, hace referencia a los patrones que no realizan divergencia ni convergencia del flujo de control, es decir, aquellos que mantienen la coherencia del flujo de control.

En la Tabla 6, se muestra a manera de resumen la clasificación de los patrones básicos definidos para BPMO, según su funcionalidad y comportamiento. En esta tabla el patrón "Sequence" tiene una funcionalidad SEQ y un comportamiento KEEP lo que significa que en este patrón el flujo de control es secuencial y coherente. Seguidamente el patrón "Parallel Split", el cual tiene una funcionalidad AND y un comportamiento SPLIT, lo que significa que sincroniza y diverge el flujo de control. Luego, el patrón "Synchronise" presenta una funcionalidad AND y un comportamiento JOIN, lo que significa que sincroniza y converge el flujo de control, así como el patrón "Multi Merge Synch". A continuación, se aborda el patrón "Simple Merge" que posee una funcionalidad XOR y un comportamiento JOIN, lo que significa que el flujo de control es asíncrono y convergente así como en el patrón "Discriminator". Seguidamente, es presentado el patrón denominado "Multi Merge", cuya funcionalidad es OR y su comportamiento JOIN, esto

significa que el flujo de control es convergente y puede ser síncrono o asíncrono, dependiendo del conjunto de condicionales que posea. A continuación, el patrón “Deferred Choice” presenta una funcionalidad XOR y un comportamiento SPLIT, lo que significa que el flujo de control es asíncrono y diverge, así como los patrones “Multiple Instantiation”, “Interleaved Routing” y “Exclusive Choice”. Finalmente, se aborda el patrón “Multiple Choice”, el cual tiene una funcionalidad OR y un comportamiento SPLIT, esto significa que el flujo de control es divergente y puede ser síncrono o asíncrono dependiendo del conjunto de condicionales que posea.

Nombre del Patron	Funcionalidad				Comportamiento		
	AND	OR	XOR	SEQ	SPLIT	JOIN	KEEP
Sequence				+			+
Parallel Split	+				+		
Synchronise	+					+	
Simple Merge			+			+	
Multi Merge Synch	+					+	
Discriminator			+			+	
Multi Merge		+				+	
Deferred Choice			+		+		
Multiple Instantiation			+		+		
Interleaved Routing			+		+		
Multiple Choice		+			+		
Exclusive Choice			+		+		

Tabla 6. Clasificación según la funcionalidad y comportamiento de los patrones de BPMO.

4.2.3 Relaciones entre patrones

Tomando como referencia el estudio realizado en [73], en el cual se hace una definición general de las relaciones entre los patrones de flujo de control de los procesos de negocio, en el presente proyecto se aplican dichas relaciones entre los patrones básicos definidos para BPMO, estas relaciones son denominadas relaciones de especialización y composición entre patrones, y son descritas a continuación.

Relación de especialización entre patrones

Una relación de **Especialización** entre patrones, existe donde uno es una forma más restringida de otro. Como ejemplo de esta relación, el patrón “Multiple Choice” es una forma más especializada del patrón “Parallel Split”, el cual es más general. Así, El patrón “Parallel Split” presenta una funcionalidad más general correspondiente a un AND y un comportamiento SPLIT lo cual corresponde a que el flujo de control diverja de manera síncrona. En cambio, el patrón “Multiple Choice” además de tener la funcionalidad de AND y el comportamiento SPLIT, también tiene la funcionalidad de XOR, En consecuencia, un patrón que posee funcionalidad OR constituye una forma más especializada de un patrón con funcionalidad AND, siempre y cuando su comportamiento sea semejante. Para lo cual, una mayor especialización, se refiere a que funcionalidad se acerca más a la EXCLUSIÓN. Entonces, así como existe una relación de especialización entre el patrón “Parallel Split” y el patrón “Multiple Choice”, también existe el mismo tipo de relación entre

el patrón “Multiple Choice” y el patrón “Exclusive Choice”, en donde el patrón “Exclusive Choice” se acerca más a la EXCLUSIÓN.

La Figura 19, muestra una secuencia de imágenes que describen la funcionalidad AND y el comportamiento SPLIT del patrón “Parallel Split”, en la cual se puede apreciar que el flujo de control diverge de manera síncrona a través de cada una de sus ramas. En especial se puede apreciar que la imagen 4 de la secuencia de imágenes (remarcada con un cuadro azul), corresponde con la funcionalidad síncrona del patrón “Multiple Choice” (recuadro rojo), Pero este a su vez, debido a que presenta una funcionalidad de tipo OR, puede generar un flujo de control síncrono o asíncrono (recuadros verdes) dependiendo de las condiciones del flujo de control de entrada. Por consiguiente, la funcionalidad síncrona o asíncrona del patrón “Multiple Choice”, lo acerca más a la exclusión, definiendo así su relación de especialización respecto al patrón “Parallel Split”.

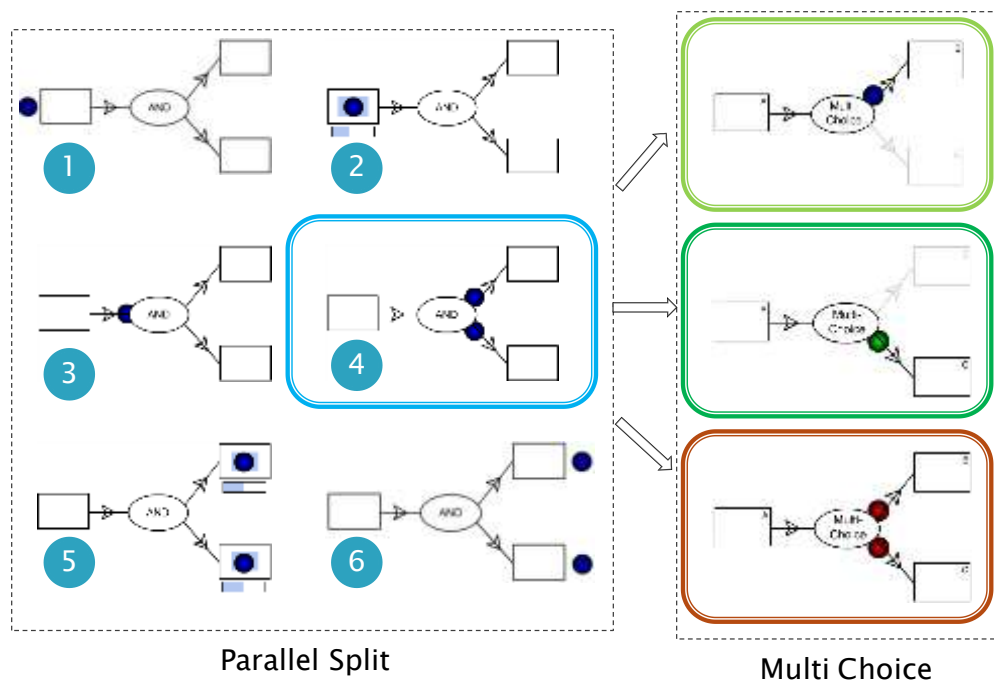


Figura 19. Relación de especialización entre los patrones Parallel Split y Multiple Choice.

La Tabla 7, presenta todas las relaciones de especialización entre los patrones de flujo de control de referencia definidos en [63], para esto, los patrones se clasifican en dos grupos. El primero de ellos (parte izquierda de la tabla), es denominado patrones especializados, y su principal característica, consiste en que su funcionalidad es excluyente como se menciona en la sección anterior. El segundo grupo (parte derecha de la tabla), es denominado patrones generales, y a diferencia del primer grupo presentado, los patrones que componen este grupo se caracterizan por su funcionalidad generalizada.

Nr	Specialized Pattern Name	General Pattern
3	Synchronization	31. Blocking Partial Join
		32. Cancelling Partial Join (only where N=M)
		33. Generalized AND-Join
		38. General Synchronizing Merge
4	Exclusive Choice	6. Multi-Choice
5	Simple Merge	8. Multi-Merge
6	Multi-choice	2. Parallel Split
7	Structured Synchronizing Merge	37. Acyclic Synchronizing Merge
9	Structured Discriminator	28. Blocking Discriminator
		30. Structured Partial Join
13	MIs with a priori D/T Knowledge	14. MIs with a priori R/T Knowledge
14	MIs with a priori R/T Knowledge	15. MIs without a priori R/T Knowledge
19	Cancel Activity	25. Cancel Region
20	Cancel Case	25. Cancel Region
26	Cancel Multiple Instance Activity	19. Cancel Activity
27	Complete MI Activity	26. Cancel MI Activity
28	Blocking Discriminator	31. Blocking Partial Join
29	Cancelling Discriminator	32. Cancelling Partial Join
30	Structured Partial Join	31. Blocking Partial Join
34	Static Partial Join for MIs	36. Dynamic Partial Join for MIs
37	Acyclic Synchronizing Merge	38. General Synchronizing Merge
40	Interleaved Routing	17. Interleaved Parallel Routing

Tabla 7. Relaciones de especialización entre patrones de flujo de control [63].

Relación de composición entre patrones

En algunos casos, es posible que un patrón pueda ser expresado a través de una combinación de dos (o más) patrones. Por ejemplo, el patrón “Multiple Choice” que tiene una funcionalidad OR y un comportamiento SPLIT (flujo de control síncrono o asíncrono y divergente), se puede expresar mediante una combinación de funcionalidades del patrón “Parallel Split” (flujo de control síncrono y divergente) y el patrón “Exclusive Choice” (flujo de control asíncrono y divergente). En consecuencia, este tipo de relación es denominada relación de **Composición** entre patrones. Ya que la funcionalidad del flujo de control de un patrón, es el resultado de la composición (superposición de las funcionalidades) de dos o más patrones.

En la Figura 20, se presenta una secuencia de imágenes que describe de manera detallada, el ejemplo mencionado anteriormente para el patrón “Multiple Choice”. En la parte inicial de esta secuencia (recuadro de contorno azul al lado izquierdo de la figura), se observa una representación de la funcionalidad divergente y síncrona del flujo de control del patrón “Parallel Split”, que combinada con la funcionalidad divergente y asíncrona del patrón “Exclusive Choice” (recuadros de contorno azul en la parte central de la figura), permiten expresar la funcionalidad síncrona o asíncrona y divergente correspondiente al patrón “Multiple Choice”(recuadros al lado derecho de la figura).

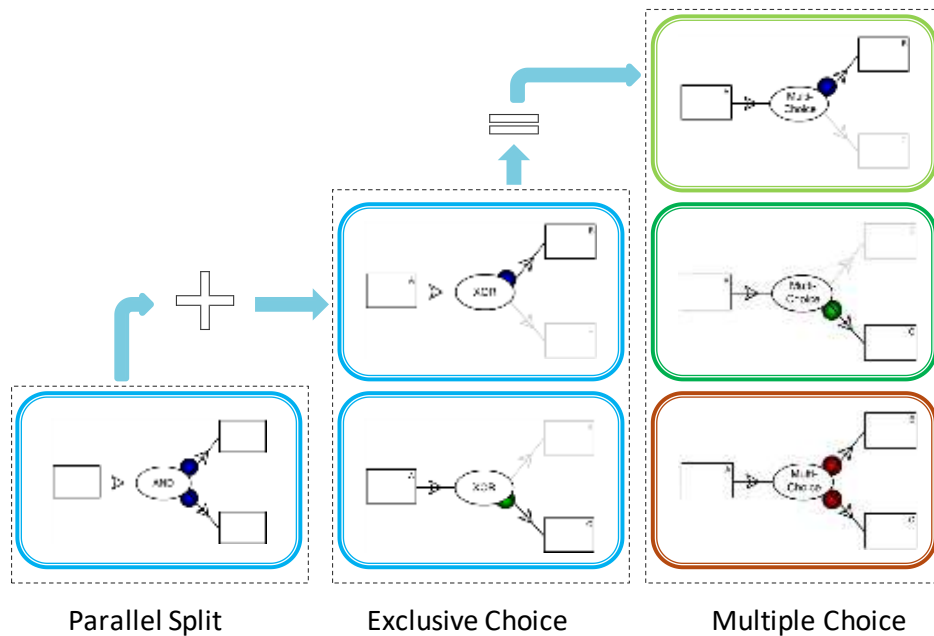


Figura 20. Relación de composición entre los patrones Parallel Split, Exclusive Choice y Multiple Choice.

La Tabla 8, presenta todas las relaciones de composición entre los patrones de flujo de control de referencia definidos en [63], para esto, los patrones se clasifican en dos grupos. El primero de ellos (parte izquierda de la tabla) es denominado patrones elementales, y su principal característica, consiste en que su funcionalidad no se puede expresar como la combinación de las funcionalidades de los demás patrones. El segundo grupo (parte derecha de la tabla), es denominado patrones compuestos, y a diferencia del primer grupo presentado, los patrones, que integran este grupo, se caracterizan por tener una funcionalidad que puede ser expresada como una combinación de las funcionalidades de los patrones elementales.

Nr	Pattern Name	Composed From
6	Multi-Choice	2. Parallel Split in conjunction with 4. Exclusive Choice
10	Arbitrary Cycles ⁹	4. Exclusive Choice together with 5. Simple Merge
12	MI without Synchronization ⁴	6. Multi-Choice together with 8. Multi-Merge
21	Structured Loop ⁴	4. Exclusive Choice together with 8. Multi-Merge
29	Cancelling Discriminator	25. Cancel Region together with 28. Blocking Discriminator
32	Cancelling Partial Join	25. Cancel Region together with 31. Blocking Partial Join
35	Cancelling Partial Join for MIs	34. Static Partial Join for MIs together with 26. Cancel MI Activity

Tabla 8. Relaciones de composición entre patrones de flujo de control

La Figura 21, presenta por medio de líneas continuas, las relaciones de especialización y por medio líneas punteadas, las relaciones de composición existentes entre los 43 patrones de referencia descritos en [71]. De manera general, se observa que solo existen relaciones de especialización o composición entre 32 de los 43 patrones de referencia. Además, se pueden evidenciar las relaciones anteriormente explicadas.



Figura 21. Relaciones de especialización y composición entre patrones de flujo de control

A partir de las relaciones entre patrones descritas anteriormente, se realizó una abstracción, en la cual se tienen en cuenta únicamente las relaciones aplicables entre los 12 patrones básicos de flujo de control, implementados en BPMO. La Figura 22, muestra la abstracción de relaciones mencionadas a partir de un diagrama, que contiene 5 recuadros que representan los patrones básicos implementados en BPMO, que presentan relaciones de especialización y composición entre sí. Inicialmente, se evidencia la relación de composición entre los patrones “Exclusive Choice” y “Parallel Split” (denotada por las flechas punteadas), los cuales componen el patrón “Multiple Choice”. Por otra parte, las relaciones de especialización presentes (denotadas por flechas continuas), corresponden a la especialización del patrón “Parallel Split”, que respectivamente es el patrón “Multiple Choice”, la especialización del patrón “Multiple Choice”, que respectivamente es el patrón “Exclusive Choice”, y finalmente la especialización del patrón “Multi Merge”, que respectivamente es el patrón “Simple Merge”.

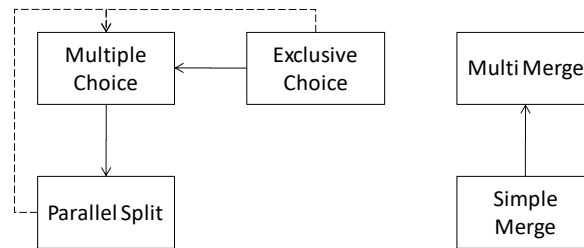


Figura 22. Relaciones de especialización y composición para los patrones básicos de BPMN.

4.2.4 Definición de la ontología de patrones

A partir de la abstracción de las relaciones entre patrones básicos realizada y la clasificación de patrones básicos, según funcionalidad y comportamiento, se generó la ontología para patrones básicos de flujo de control, descritos en BPMN. Inicialmente se identificaron los principales conceptos relacionados al dominio del flujo de control de los procesos de negocio. Esto con el fin de establecer los súper conceptos, de los cuales heredarían atributos, los conceptos pertenecientes a la ontología de patrones de flujo de control. Así, el primer concepto padre para la ontología de patrones de flujo de control, fue denominado “controlflowpattern”, y hace referencia a cualquier tipo de patrón de flujo de control, ya que lo que diferencia las categorías de patrones entre sí, son sus atributos. Entre las formas de categorizar patrones, a manera de síntesis se nombran las siguientes categorías, las cuales ya fueron descritas en las secciones 3.2.3 y 4.2.3:

- Patrones básicos.
- Patrones de ramificación y sincronización avanzados.
- Patrones de múltiples instancias.
- Patrones basados en estados.
- Patrones de cancelación y de terminación forzada.
- Patrones de iteración.
- Patrones de terminación.
- Patrones de activación.
- Patrones elementales.
- Patrones compuestos.
- Patrones especializados.
- Patrones generales.

Cada uno de los 12 patrones básicos de flujo de control, es representado por medio de una instancia del concepto padre “controlflowpattern”, la cual contiene los siguientes atributos:

- **hasFunction:** corresponde a una instancia del concepto “function”, la cual puede tomar los valores AND, XOR, OR o SEQ.
- **hasBehaviour:** corresponde a una instancia del concepto “behaviour”, la cual puede tomar los valores SPLIT, JOIN o KEEP.

- ***isSpecializedOf***: corresponde a una instancia del concepto “controlflowpattern”, la cual puede tomar los valores “Sequence”, “Parallel Split”, “Synchronise”, “Simple Merge”, “Multi Merge Synch”, “Discriminator”, “ Multi Merge”, “ Deferred Choice”, “ Multiple Instantiation”, “ Interleaved Routing”, “ Multiple Choice” y “ Exclusive Choice”.
- ***composeTo***: corresponde a una instancia del concepto “controlflowpattern”, la cual puede tomar los valores “Sequence”, “Parallel Split”, “Synchronise”, “Simple Merge”, “Multi Merge Synch”, “Discriminator”, “ Multi Merge”, “ Deferred Choice”, “ Multiple Instantiation”, “ Interleaved Routing”, “ Multiple Choice” y “ Exclusive Choice”.

Por otra parte, el segundo concepto padre de la ontología de patrones de flujo de control, fue denominado “function”, que hace referencia al tipo de funcionalidad de las descritas previamente en la sección 4.2.2, que puede tener cada patrón (AND, XOR, OR, SEQ). De esta forma, cada tipo de funcionalidad, es a su vez representada por una instancia del concepto padre “function”.

Finalmente, el tercer concepto padre de la ontología de patrones de flujo de control, fue denominado “behaviour”, que hace referencia al tipo de comportamiento de los descritos previamente en la sección 4.2.2, que puede tener cada patrón (SPLIT, JOIN, KEEP). Cada tipo de comportamiento es a su vez representado por una instancia del concepto padre “behaviour”.

La Figura 23, muestra la representación gráfica de la ontología de patrones básicos de flujo de control para BPMO, donde los nodos representan conceptos o instancias y las aristas, las relaciones de súper conceptos y subconceptos existentes entre los conceptos de la ontología. Adicionalmente, una descripción formal en formato WSML de la ontología está contenida en el Anexo 5.

Para la implementación de la ontología de patrones de flujo de control, se utilizó un EMF (Eclipse Meta Framework) denominado WSMT (Web Service Modeling Toolkit), el cual provee funcionalidades de edición para la creación de ontologías en formato WSML, un visualizador para ver una representación gráfica de la ontología y una interfaz para realizar consultas por medio del razonador IRIS.

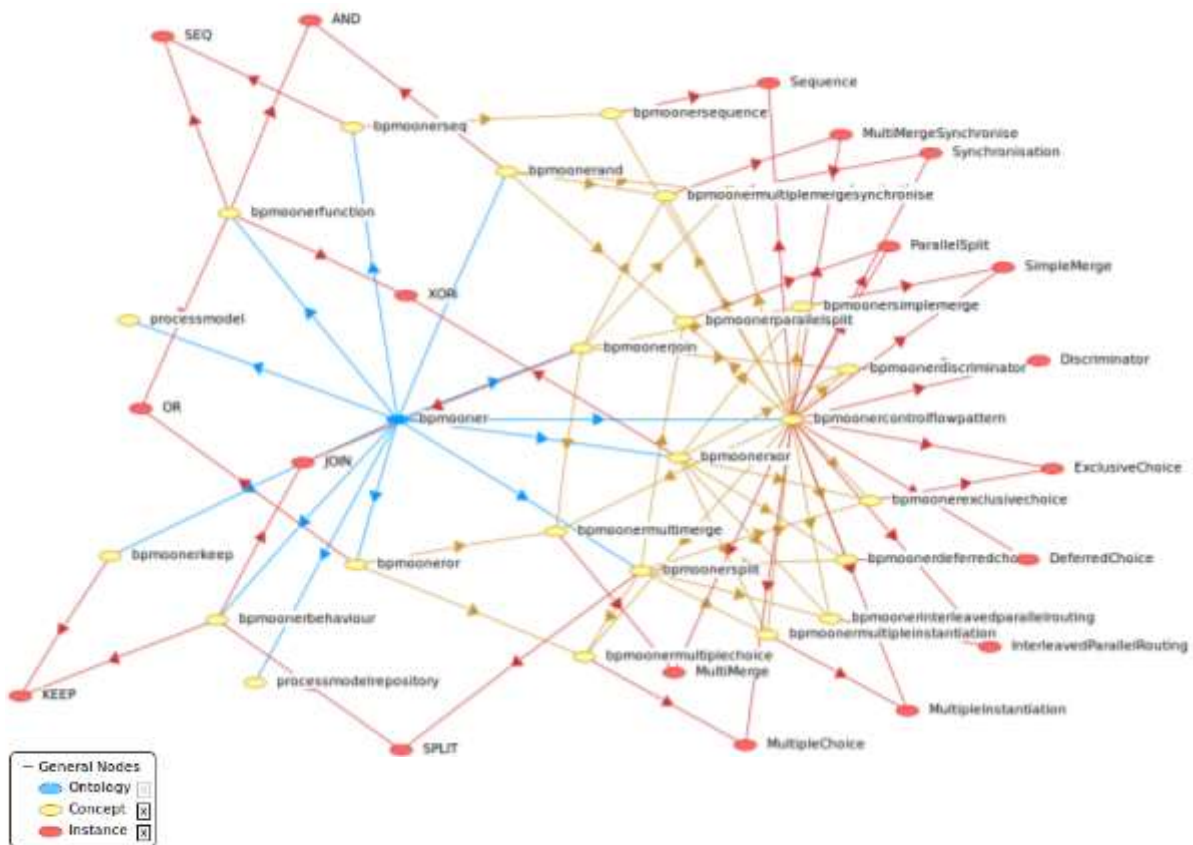


Figura 23. Ontología de patrones de flujo de control básicos.

4.3 Integración de la Ontología de Patrones con la Arquitectura

En esta sección, se explica la manera en que se integra la ontología de patrones básicos, con la arquitectura del repositorio propuesta en la sección 3.2, por medio de la adaptación de un Razonador semántico, al interior de modulo de Recuperación de Modelos BP. A partir de este razonador, es posible realizar el posterior cálculo de la medida de similitud, entre los flujos de control de los Modelos BP, a partir de la cual, finalmente se realiza el ordenamiento (ranking) de los Modelos BP recuperados según la medida de similitud de su flujo de control respecto al flujo de control, del Modelo BP de consulta.

4.3.1 Razonador Semántico

Para lograr la integración de la ontología de patrones básicos de flujo de control, con la arquitectura propuesta en la sección 3.2, se adiciona un módulo de razonamiento semántico denominado IRIS (Integrated Rule Inference System) [76]. Este razonador es utilizado para inferir que instancias, de los conceptos de la ontología, cumplen con atributos semánticamente similares, durante la recuperación de Modelos BP según sus patrones de flujo de control.

IRIS, es un razonador Datalog⁴⁰, que utiliza estrategias de evaluación de tipo bottom-up (se evalúan partes individuales con detalle y luego se enlazan para formar un componente de evaluación más grande, que a su vez se enlaza con otros componentes hasta que se evalúa el sistema completo). Adicionalmente, el soporte para las variantes basadas en las reglas WSML requiere varias extensiones las cuales son denominadas:

- WSML-Core, el cual está basado en planes (funciones libres y negación por defecto), proporcionados por un Datalog con tipos primitivos de XML Schema.
- WSML-Flight, requiere un Datalog extendido con desigualdades y localmente ordenado en niveles, con un tipo determinado de negación por defecto.
- WSML-Rule, requiere el irrestricto uso de funciones simbólicas, la negación bien definida por defecto y no requiere las condiciones de reglas de seguridad.

WSML, está basado en diferentes lógicas formales denominadas: Lógica de Descripción (Description Logics), Lógica de Primer Orden (First-Order Logic) y Programación Lógica (Logic Programming), las cuales se utilizan para el modelado de servicios web semánticos. Con base en estas lógicas formales se han definido cinco variantes, cada una con distinto nivel de expresividad lógica. De estas cinco variantes, la más adecuada para describir ontologías es WSML-Flight, ya que es la variante más expresiva y combina tanto la lógica de descripción, como la de primer orden con una complejidad computacional manejable [77] (Figura 24).

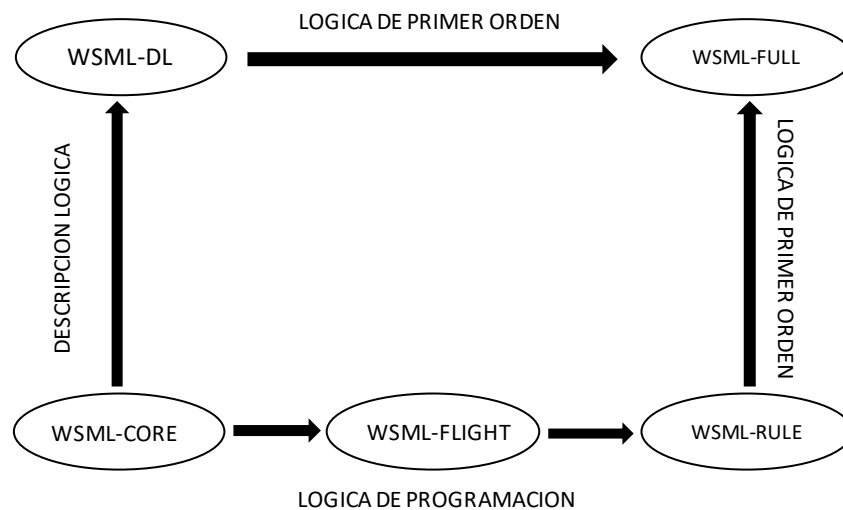


Figura 24. Variantes del lenguaje WSML (Web Service Modeling Language)

4.3.2 Similitud semántica de patrones

Tomando como referencia el trabajo definido en [71], donde se establece las relaciones de especialización y composición de patrones de flujo de control, y a partir de la ontología

⁴⁰ Datalog, es un lenguaje de reglas y consultas para bases de datos deductivas. Las consultas están basadas en lógica de primer orden.

presentada anteriormente, es definida una medida de similitud entre patrones $Sim_{SemPattern}(Q, T)$, por medio de los atributos definidos para cada instancia del concepto padre "controlflowpattern", estos atributos son: "hasFunction", "hasBehaviour", "isSpecializedOf" y "composeTo", los cuales ya han sido mencionados con anterioridad en la sección 4.2.4. De esta manera, cuando el Modelo BP de consulta contenga un determinado patrón, tomando en cuenta los atributos de éste, no solo serán recuperados aquellos Modelos BP que contengan este patrón de manera exacta, sino que también todos los Modelos BP que tengan alguna relación semántica con dicho patrón.

Sean Q y T los patrones Consulta (*Query*) y Objetivo (*Target*) respectivamente, y A_Q y A_T arreglos de atributos del *Query* y del *Target* para un determinado tipo de patrón (Parallel Split, Multiple Choice, etc.), la medida de similitud semántica entre los patrones *Query* y el *Target* se define como $Sim_{SemPattern}(Q, T)$ y está dada por:

$$Sim_{SemPattern}(Q, T) = \frac{|A_Q \cap A_T|}{|A_Q \cup A_T|}$$

Donde cada uno de los atributos de los arreglos A_Q y A_T , están asociados a un coeficiente de tal manera que la suma de cualquier subconjunto de coeficientes del total, sea diferente, es decir que no se repitan. Por ejemplo, si los valores de estos coeficientes son: $A_Q = \{1, 2, 5, 9\}$, caso en el cual los cuatro atributos están definidos para el Query, un ejemplo del cálculo de esta medida de similitud con $A_T = \{1, 9\}$ es $A_Q \cap A_T = \{1, 2, 5, 9\} \cap \{1, 9\} = \{1, 9\}$, y cuya norma $|A_Q \cap A_T| = 10$. Los posibles valores de $|A_Q \cap A_T|$ son: 1, 2, 3, 5, 7, 8, 9, 10, 11, 12, 14, 15, 17. Además, para este caso $A_Q \cup A_T = \{1, 2, 5, 9\} \cup \{1, 9\} = \{1, 2, 5, 9\}$ y su norma $|A_Q \cup A_T| = 17$, así $Sim_{SemPattern}(Q, T) = 10/17$.

El valor $Sim_{SemPattern}(Q, T)$ es nulo, cuando ninguno de los atributos del Query corresponde con los del Target, y $Sim_{SemPattern}(Q, T) = 1$, cuando todos los atributos del Query corresponden con los Target.

4.4 Ranking de modelos de procesos recuperados

4.4.1 Condiciones del ranking semántico

A partir de la definición de la medida de similitud semántica a utilizar, en esta sección se define $Sim_{Total}(Q, T)$ de manera que contemple el valor de $Sim_{SemPattern}(Q, T)$, con el propósito de ordenar (ranking) los Modelos BP, teniendo en cuenta la semántica del flujo de control. $Sim_{Total}(Q, T)$ es una medida de similitud en la que participan las medidas de similitud previamente establecidas en el capítulo anterior en la sección 3.2.3.

Así, los Modelos BP son ordenados teniendo en cuenta las medidas $Sim_{Pattern}(Q, T)$, $Sim_{SeqPattern}(Q, T)$ y $Sim_{SemPattern}(Q, T)$ por medio de la siguiente expresión:

$$Sim_{Total}(Q,T) = \frac{\alpha * Sim_{Pattern}(Q,T) + \beta * Sim_{SeqPattern}(Q,T) + \delta * Sim_{SemPattern}(Q,T)}{3}$$

En esta medida, los valores de los coeficientes de ordenamiento α , β y δ son personalizados de acuerdo a los criterios de relevancia o pertinencia del usuario, tomando valores tales que $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$, $0 \leq \delta \leq 1$ y $\alpha + \beta + \delta = 1$.

El diagrama modular que presenta los componentes adicionales del módulo Búsqueda de Similitud Semántica para búsqueda semántica, es ilustrado en la Figura 25.

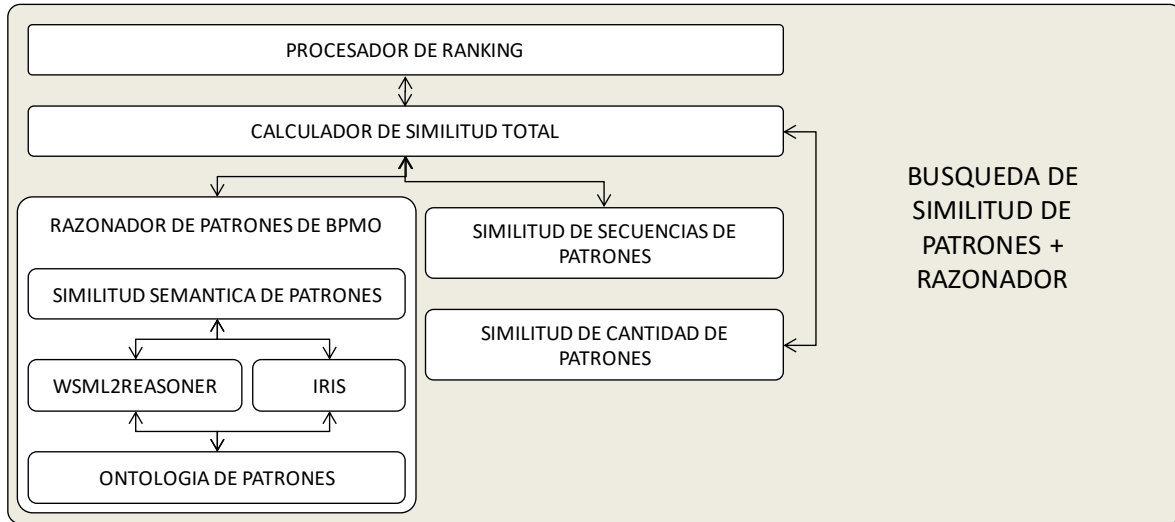


Figura 25. Diagrama modular del módulo Búsqueda de Similitud de Patrones con la adaptación de búsqueda semántica.

4.5 Resumen

En este capítulo se describió la integración del proceso de recuperación de Modelos BP, junto con una ontología de patrones de flujo de control, que permitió adicionar características de búsqueda semántica al repositorio propuesto (sección 3.2). Fueron abordados los conceptos de semántica, Web semántica y ontologías, explicando cómo estos inciden en la transformación de la Web actual. Posteriormente, fue descrita una clasificación de patrones según su funcionalidad y comportamiento, especificando relaciones de especialización y composición entre patrones y, a partir de estas relaciones, se generó una abstracción de acuerdo a los 12 patrones básicos de flujo de control implementados en BPMO que permitió crear la ontología de patrones básicos de flujo de control. Finalmente, se integró la ontología de patrones con la arquitectura del repositorio, mediante un razonador semántico y se definieron las medidas de similitud semántica para el ordenamiento de modelos de procesos recuperados, contemplando las medidas de similitud de patrones, similitud de secuencias de patrones y similitud total, propuestas en el presente capítulo.

Capítulo V

5 PROTOTIPO Y EXPERIMENTACIÓN

En este capítulo se aborda la descripción detallada del prototipo que valida la arquitectura propuesta en los Capítulos 3 y 4. Además, se especifica el plan de pruebas realizado sobre el mismo, junto con los resultados obtenidos, a partir de la ejecución de dichas pruebas.

5.1 BPMOONER (BPMO – Ontological Enterprise Repository)

El prototipo construido para evaluar el mecanismo propuesto consiste en 2 aplicaciones Web Java, implementadas a partir de la adopción del Modelo de Construcción de Soluciones [78], como referencia metodológica para soportar el proceso de desarrollo. La primera de ellas es una aplicación Web servidora llamada “BPMOONER-server” que contiene toda la lógica de negocio, la cual contiene una interfaz para ser accedida por medio de servicios Web, y la segunda es una aplicación Web cliente llamada “BPMOONER-client”, que permite visualizar la información acerca de los Modelos BP recuperados.

5.1.1 Modelo de Casos de Uso del Sistema

La definición de los casos de uso del sistema parte de la identificación de los requisitos funcionales del mismo. A continuación se detallan los requisitos funcionales abstraídos para el sistema implementado:

- i. **Detectar Patrones de flujo de control al interior de un modelo de proceso de negocio descrito en BPMO.** Teniendo en cuenta el conjunto de los 12 patrones básicos definidos de BPMO, el sistema debe realizar una detección mediante un algoritmo de isomorfismo de grafos.
- ii. **Determinar una medida de similitud de patrones.** El sistema debe calcular, de manera cuantitativa, una medida de similitud, que representa la cantidad de patrones que comparten el Modelo BP Consulta y el Modelo BP Objetivo.
- iii. **Determinar una medida similitud entre secuencias de patrones.** El sistema debe calcular, de manera cuantitativa, la medida de similitud entre secuencias de patrones, al interior de los modelos de procesos de negocio descritos en BPMO.
- iv. **Determinar una medida de similitud semántica de patrones.** El sistema debe calcular, de manera cuantitativa, una medida de similitud semántica entre los 12 patrones básicos de BPMO a partir su funcionalidad, comportamiento y relaciones de especialización y composición.
- v. **Determinar una medida de similitud total.** El sistema debe calcular, de manera cuantitativa, una medida de similitud total entre el flujo de control de los modelos de procesos de negocio descritos en BPMO, a partir de las medidas de similitud entre patrones, similitud entre secuencias de patrones y similitud semántica de patrones.

- vi. **Configurar parámetros de comparación.** El sistema debe permitir ajustar los coeficientes que determinan el valor de importancia asignado a cada medida de similitud presente en el proceso del cálculo de la medida de similitud total.
- vii. **Ordenar los Modelos BP recuperados según su medida de similitud total con respecto al BP Model de consulta.** A partir de la comparación entre un Modelo BP de Consulta y un conjunto de Modelos BP disponibles en el repositorio, el sistema debe ordenar (Ranking) dicho conjunto de acuerdo a la medida de similitud total existente entre sus flujos de control.
- viii. **Visualizar contenido de los Modelos BP recuperados.** El sistema debe permitir visualizar el contenido de los Modelos BP recuperados, mediante una interfaz gráfica de usuario.

A partir del análisis de los requisitos funcionales del sistema se definieron tres casos de uso, los cuales se describen a continuación:

Caso de Uso No. 1: *Publicar Modelos BP.*

Iniciador: Usuario (Agente Externo: Persona, Aplicación)

Propósito: Detectar patrones de flujo de control al interior de los Modelos BP.

Resumen: Este caso de uso satisface el requisito funcional (i), a partir de la detección de los 12 patrones básicos de BPMO mediante un algoritmo de isomorfismo de grafos.

Caso de Uso No. 2: *Recuperar Modelos BP*

Iniciador: Usuario (Agente Externo: Persona, Aplicación)

Propósito: Entregar un conjunto de Modelos BP ordenados según la medida de similitud total de su flujo de control, con respecto a un Modelo BP de consulta.

Resumen: Este caso de uso obedece al requisito funcional (i, ii, iii, iv, v, vi y vii). Una vez ejecutadas las comparaciones entre una tarea de consulta y un conjunto de tareas publicadas, se determina cuales de estas últimas resultan más relevantes para satisfacer la petición definida por la primera.

Caso de Uso No. 3: *Visualizar contenido de los Modelos BP recuperados.*

Iniciador: Usuario (Agente Externo: Persona, Aplicación)

Propósito: Mostrar al usuario el contenido de los Modelos BP recuperados para la posterior reutilización de éstos en el proceso de modelado de nuevos Modelos BP.

Resumen: Este caso de uso satisface el requisito funcional (viii), a partir de la visualización del contenido de los Modelos BP recuperados mediante una interfaz gráfica de usuario.

La Figura 26, presenta el diagrama de casos de uso del sistema desarrollado, donde se pueden observar los 3 casos de uso del sistema representados por elipses. El primero de ellos corresponde al caso de uso Publicar Modelos BP, el segundo Recuperar Modelo BP y el tercero Visualizar contenido de los Modelos BP recuperados.

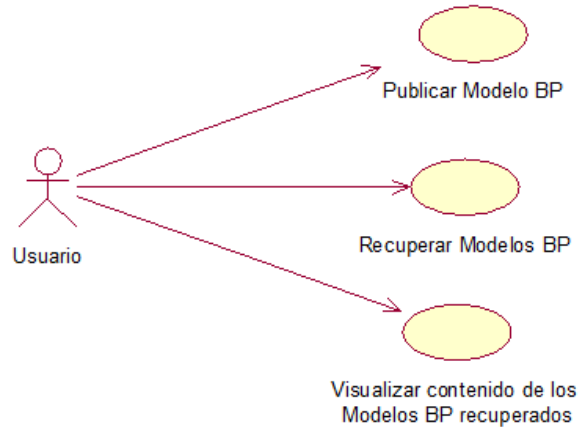


Figura 26. Diagrama de Casos de Uso del Sistema

La definición adecuada de los casos de uso del sistema y su continua verificación a lo largo de su periodo de implementación, permiten orientar las actividades del proceso de desarrollo hacia la satisfacción de los requerimientos fundamentales de la solución.

5.1.2 Software utilizado para la implementación

Dado que la versión de la implementación del algoritmo de isomorfismo de grafos VF2 obtenida a través del contacto establecido con la Dra. Rosalba Giugno de la Universidad de Catania (Italia) está compilada en g++ (Linux), se escogió el sistema operativo Linux (Ubuntu 9.10), el cual permite la ejecución de este algoritmo. Respecto a la selección de Eclipse Galileo como el IDE para desarrollo del prototipo se tuvo en consideración las siguientes razones:

1. Es un IDE multiplataforma implementado y para desarrollo en lenguaje Java, que facilitó la utilización de la versión conseguida de GraphBlast, la cual estaba implementada en lenguaje Java (JGraphBlast).
2. Los EMFs WSMO Studio y WSMT para el modelado de los procesos de negocio en BPMO y las ontologías en formato WSML, están disponibles como Eclipse plugins.
3. Eclipse Galileo JEE está provisto de plugins que permiten el control mediante su interfaz de servidores y de bases de datos.

La versión de JDK que soporta la implementación es 1.6.0.20, el servidor Web seleccionado fue Apache Tomcat 6.0.18 y el entorno de ejecución de servicios Web fue Apache Axis2 1.4.1. El servidor de bases de datos relacional empleado fue PostgreSQL 8.4 y para la base de datos de grafos fue Berkeley DB. Para el manejo de los Modelos BP y la ontología fueron utilizadas las APIs wsmo, wsmo4j, wsmlgrammar, upo, wsml2reasoner e iris. Para el diseño de la interfaz gráfica se eligió Adobe Illustrator y Adobe Dreamweaver. En cuanto a la visualización de las referencias de los Modelos BP recuperados, se utilizó una grilla implementada en javascript llamada flexiGrid. La gestión

de contenido fue hecha a través de un CMS Java llamado Liferay, y para la subida de los archivos al servidor, se utilizó el API Mime.

Roles del Sistema

- **Usuario anónimo:** puede acceder a todas las funcionalidades de búsqueda del sistema excepto la de publicar Modelos BP.
- **Usuario registrado:** posee las funcionalidades del usuario anónimo más la de publicar Modelos BP.
- **Administrador:** Tiene control total sobre el sistema (Administración de usuarios, patrones de flujo de control y Modelos BP).

5.1.3 Diagramas de Paquetes, subsistemas y capas del Sistema

La arquitectura lógica de las siguientes Figuras organizan el software utilizado en paquetes, subsistemas y capas. Este es implementado en 3 capas diferentes (Aplicación, Mediación, Fundamentación), en donde son presentadas las interacciones entre los paquetes más relevantes que componen cada capa (Figuras 27 y 28).

BPMOONER-client:

Capa de Aplicación: Gestiona los paquetes que implementan las funcionalidades del prototipo, está compuesta por los siguientes paquetes:

- **com:** Este es el paquete general que contiene los llamados a los demás paquetes.
- **bpmoonerwebserviceclient:** Este paquete provee las funcionalidades que permiten invocar los servicios Web de la App Web BPMOONER-server.
- **uploadmanager:** Este paquete contiene los servlets que permiten cargar los archivos desde un navegador y la invocación a los servicios Web de la App Web BPMOONER-server.

Capa de Mediación: Contiene todas las interfaces de aplicación de programa (APIs) utilizadas por el prototipo, está compuesta por los paquetes:

- **apacheaxis2:** Es una plataforma, open source de servicios Web basada en XML. Consiste en una implementación de un servidor SOAP escrito en Java y contiene varias utilidades y APIs para generar y desplegar aplicaciones de servicios Web.
- **flexigrid:** Es una grilla implementada en Java Script, que proporciona una completa funcionalidad para la visualización de datos, mediante la comunicación por medio de JSON o XML.

- **mime:** Es un API implementada en Java, que permite cargar archivos desde formularios Web.
- **apachetomcat:** Implementa los servlets y las Java Server Pages (JSP), especificados por Oracle Corporation, proveyendo un entorno para ejecutar código Java en cooperación con un servidor Web. Este adiciona herramientas para gestión y configuración pero también puede ser configurado a través de archivos de configuración en formato XML.
- **jquery:** Es un API implementada en Java Script, que proporciona una gran variedad de funcionalidades, a nivel de interfaz gráfica de usuario.
- **webbrowser:** Un navegador web es un programa que permite ver la información que contiene una página web (ya se encuentre ésta alojada en un servidor dentro de la World Wide Web o en un servidor local).

Capa de Fundamentación: Incluye el software básico, que permite el rendimiento del prototipo, está compuesta por los siguientes paquetes:

- **Linux (Ubuntu 9.10):** Es el sistema operativo que soporta el prototipo.
- **soap:** Es el protocolo para el intercambio de mensajes basados en XML sobre redes de computadores, normalmente utiliza HTTP/HTTPS. SOAP provee la una plataforma básica de mensajería sobre la cual más capas abstractas pueden ser construidas.

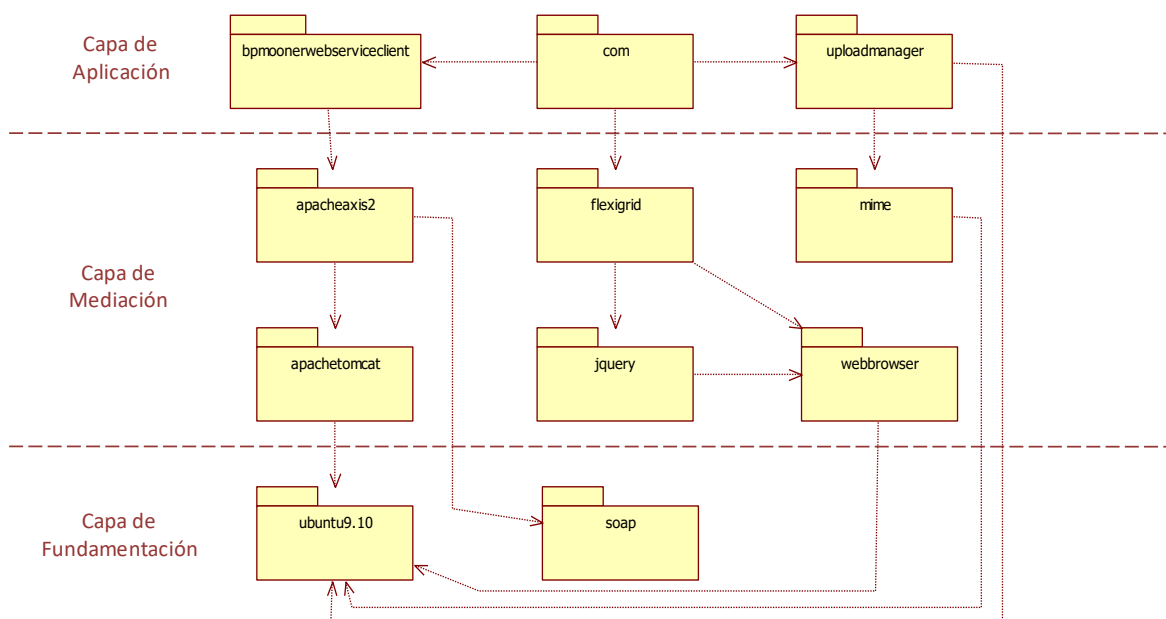


Figura 27. Diagrama de paquetes BPMOONER-client

BPMOONER-server:

Capa de Aplicación: Gestiona los paquetes que implementan las funcionalidades del prototipo, está compuesta por los siguientes paquetes:

- **com:** Este es el paquete general que contiene los llamados a los demás paquetes.
- **bpmomodel:** contiene la representación de los objetos que componen un BPMO Model.
- **parser:** provee las funcionalidades para realizar la transformación de un BPMO Model a un BP Graph
- **bpmo2graph:** proporciona la adaptación del algoritmo TDA, específica para BPMO.
- **basegraphmodel:** contiene la representación de los objetos que componen un BP Graph.
- **webservice:** implementa los servicios Web, que provee la App Web BPMOONER-server.
- **bpmoweb service:** implementa las funciones específicas para BPMO, invocadas por el paquete webservice.
- **servlet:** contiene los servlets encargados de la interfaz de administración de la App Web BPMOONER-server.
- **bpmanager:** este es el paquete general, que contiene los llamados a los paquetes patternmanager, similaritypatternsearch y patternanalyzer.
- **patternmanager:** provee las funciones para realizar la administración de los patrones de flujo de control.
- **similaritypatternsearch:** implementa el cálculo de las medidas de similitud y el ordenamiento de los Modelos BP.
- **patternanalyzer:** implementa la detección de patrones de flujo control al interior de los Modelos BPMO.
- **jgraphblast:** provee las funciones necesarias para invocar el algoritmo de isomorfismo de grafos.
- **gmlprocessgraphwriter:** permite generar un archivo gml, que representa el BP Graph.
- **gmlprocessgraphmodel:** contiene los objetos, que representan un grafo en formato gml.
- **repositorymanager:** este es el paquete general, que contiene los llamados a los paquetes bpmomodelaccess, bpreferenceaccess, bpgraphaccess.
- **bpmomodelaccess:** implementa las funciones de CRUD para los archivos WSMML que representan los Modelos BP.
- **bpreferenceaccess:** implementa las funciones de CRUD para una base de datos relacional de referencias de los Modelos BP.
- **bpgraphaccess:** implementa las funciones de CRUD para los archivos .DAT, que representan los Grafos BP.

Capa de Mediación: Contiene todas las interfaces de aplicación de programa (APIs) utilizadas por el prototipo, está compuesta por los paquetes:

- **apachetomcat:** Implementa los servlets y las Java Server Pages (JSP) especificados por Oracle Corporation, proveyendo un entorno para ejecutar código Java, en cooperación con un servidor Web. Este adiciona herramientas para gestión y configuración pero también puede ser configurado a través de archivos de configuración, en formato XML
- **apacheaxis2:** Es una plataforma, open source de servicios Web basada en XML. Consiste en una implementación de un servidor SOAP, escrito en Java y contiene varias utilidades y APIs para generar y desplegar aplicaciones de servicios Web.
- **wsmo4j:** es un API y una implementación de referencia para la creación de Servicios Web Semánticos y aplicaciones de procesos de negocio semánticas basadas en la WSMO y BPMO.
- **wsmo:** estandariza las definiciones de interfaz de componentes de infraestructura de bajo nivel (Por ejemplo, analizadores, serializadores, validadores, almacenes de datos, fábricas, etc.)
- **wsm1grammar:** puede validar y determinar la variante de WSM1.
- **iris:** es un motor de registro de datos de código abierto, ampliado con los tipos de datos XML Schema, que incorpora en los predicados, símbolos de función y la negación fundada por defecto.
- **wsm12reasoner:** traduce ontologías WSM1 en la sintaxis apropiada, para un motor de razonamiento subyacente especificado. Dependiendo de la variante de la ontología WSM1, y también permite que el usuario escoja el motor para realizar el razonamiento.
- **postgreSQL:** permite la conexión y la ejecución de sentencias SQL con el motor de bases de datos desde la App web.
- **upo:** API de referencia de la cual depende wsmo4j.
- **Berkeley:** motor de bases de datos de grafos.

Capa de Fundamentación: Incluye el software básico, que permite el rendimiento del prototipo, está compuesta por los siguientes paquetes:

- **Linux (Ubuntu 9.10):** Es el sistema operativo que soporta el prototipo.
- **soap:** Es el protocolo para el intercambio de mensajes basados en XML sobre redes de computadores, normalmente utiliza HTTP/HTTPS. SOAP provee la una plataforma básica de mensajería sobre la cual más capas abstractas pueden ser construidas.

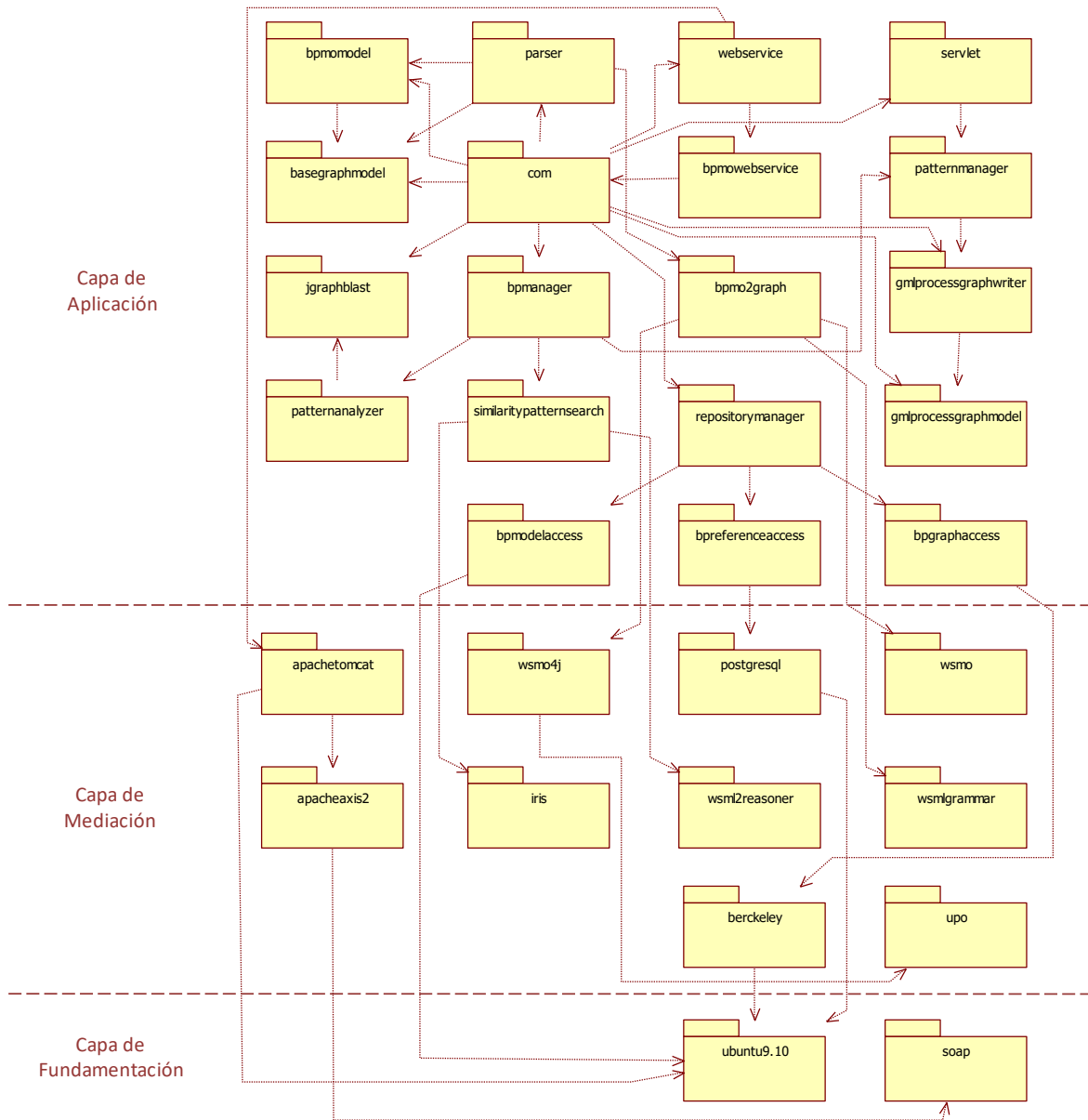


Figura 28. Diagrama de paquetes BPMOONER-server.

5.1.4 Interfaces gráficas de usuario del Sistema

En esta sección son presentadas las interfaces gráficas (GUIs) BPMOONER-client, las cuales permiten publicar Modelos BP, si el usuario es registrado y recuperar Modelos BP, tanto para usuarios registrados como no registrados. También se presentan las interfaces gráficas de BPMOONER-server, las cuales permiten realizar las funciones de administración del repositorio.

Interfaces BPMOONER-client: La Figura 29, muestra los componentes de la interfaz para el usuario anónimo, entre los cuales están una barra de ingreso para los usuarios registrados, la cual cuenta con un campo para el nombre de usuario y otro para

contraseña. Para los usuarios anónimos, se provee una barra de búsqueda rápida que permite buscar por medio de substring coincidencias con cada uno de los campos (identificador, título, autor, tipo, descripción, etc.) de cada una de las referencias de los Modelos BP. También es provista una barra de búsqueda por patrones de flujo de control, a la cual se le puede adicionar o no la característica de semántica del flujo de control. Finalmente, consta de una barra de funcionalidades adicionales entre las que están: visualizar modelos de patrones de flujo de control, visualizar contenido WSML de los Modelos BP y visualizar la WSDL del servicio Web del servidor.

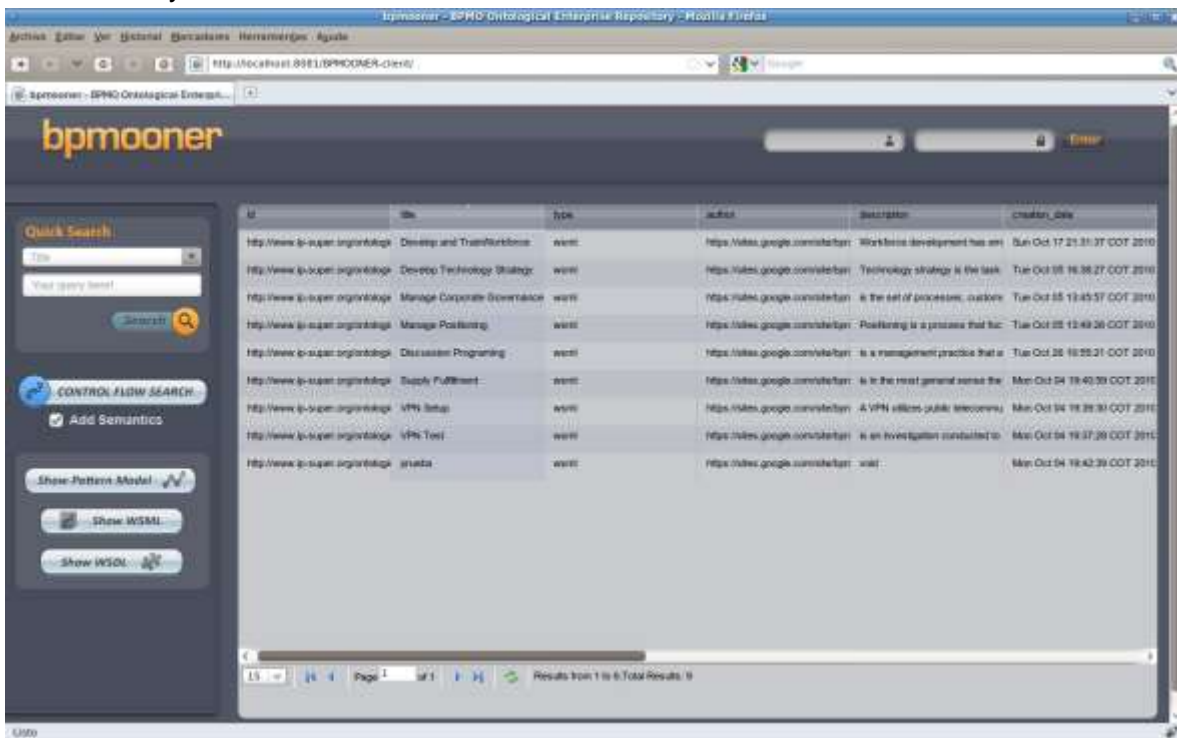


Figura 29. Interfaz para el usuario anónimo.

La Figura 30, muestra los componentes de la interfaz para el usuario registrado, entre estos están una barra de salida del sistema para cerrar sesión, se provee una barra de búsqueda rápida que permite buscar por medio de substring coincidencias con cada uno de los campos (identificador, título, autor, tipo, descripción, etc.) de cada una de las referencias de los Modelos BP. También es provista una barra de búsqueda por patrones de flujo de control, a la cual se le puede adicionar o no la característica de semántica del flujo de control. Adicionalmente, consta de una barra de funcionalidades adicionales entre las que están: visualizar modelos de patrones de flujo de control, visualizar contenido WSML de los Modelos BP y visualizar la WSDL del servicio Web del servidor. A diferencia de la interfaz para el usuario anónimo, ésta consta con una barra para realizar la publicación de un Modelo BP. Tanto para el usuario anónimo como para el usuario registrado la interfaz del BPMOONER-client, consta de una grilla para la visualización de la lista de los Modelos BP obtenidos a partir de una consulta, esta grilla cuenta con funcionalidades como paginado de resultados, ocultamiento de columnas y reorganización de columnas.

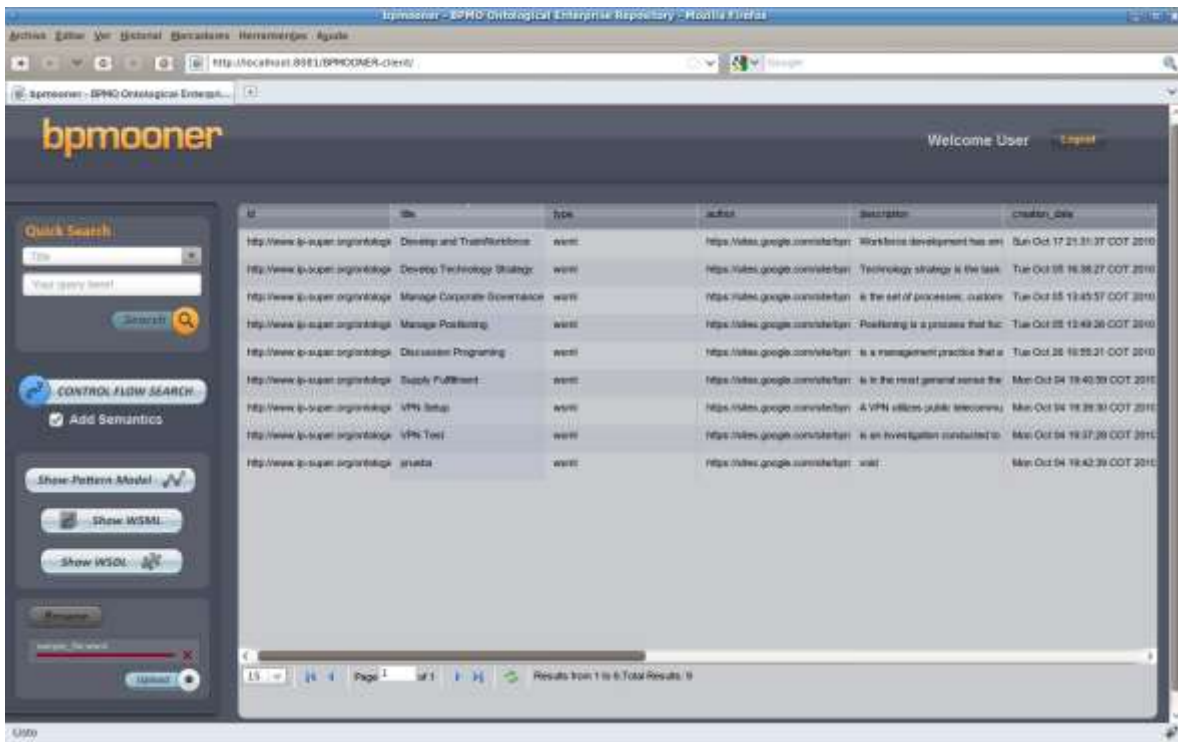


Figura 30. Interfaz para el usuario registrado.

Interfaces BPMOONER-server:

La Figura 31, muestra la interfaz de bienvenida de BPMOONER-server, la cual consta de un módulo para iniciar sesión.

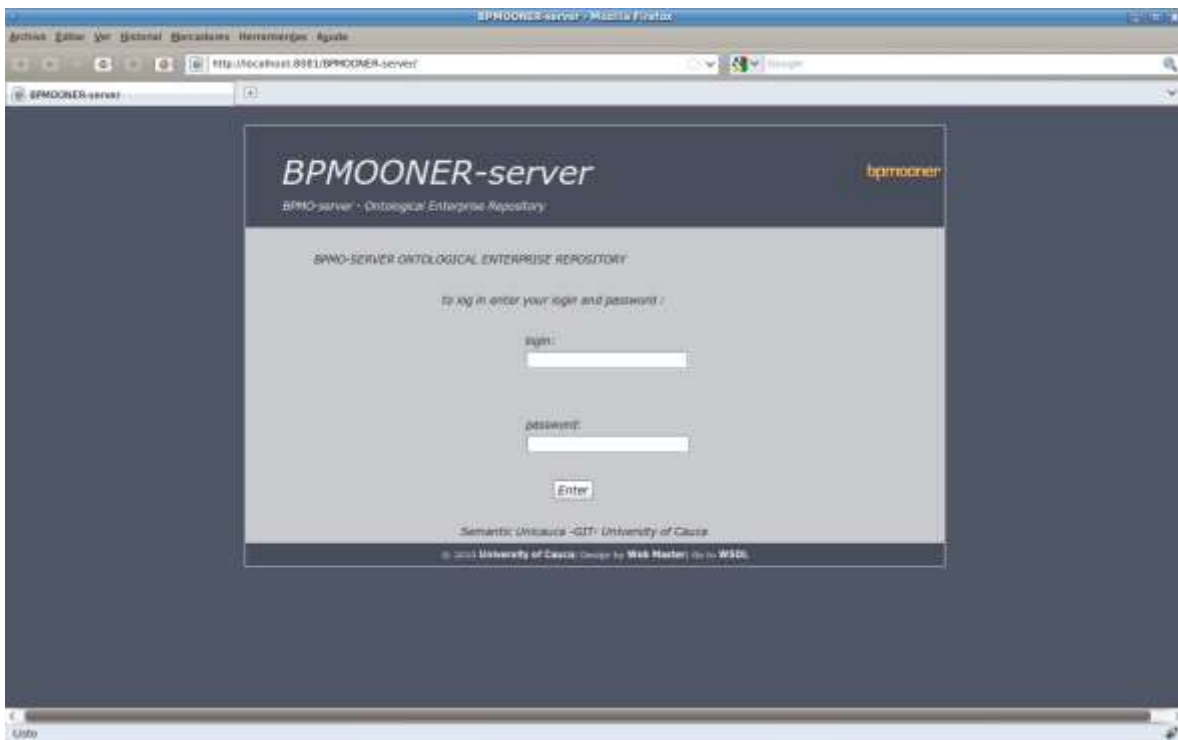


Figura 31. Interfaz de bienvenida del BPMOONER-server.

Una vez se inicia sesión y el nombre de usuario y contraseña son validados, en la Figura 32 se presenta la interfaz del BPMOONER-server que provee la funcionalidad para cerrar sesión, los controles para hacer lectura, creación, actualizado, borrado de las referencias de los Modelos BP almacenados en la base de datos del sistema. Además, por medio de esta interfaz se generan inicialmente los modelos de los patrones mediante un script SQL, así como los archivos para el control de cuentas de usuario.

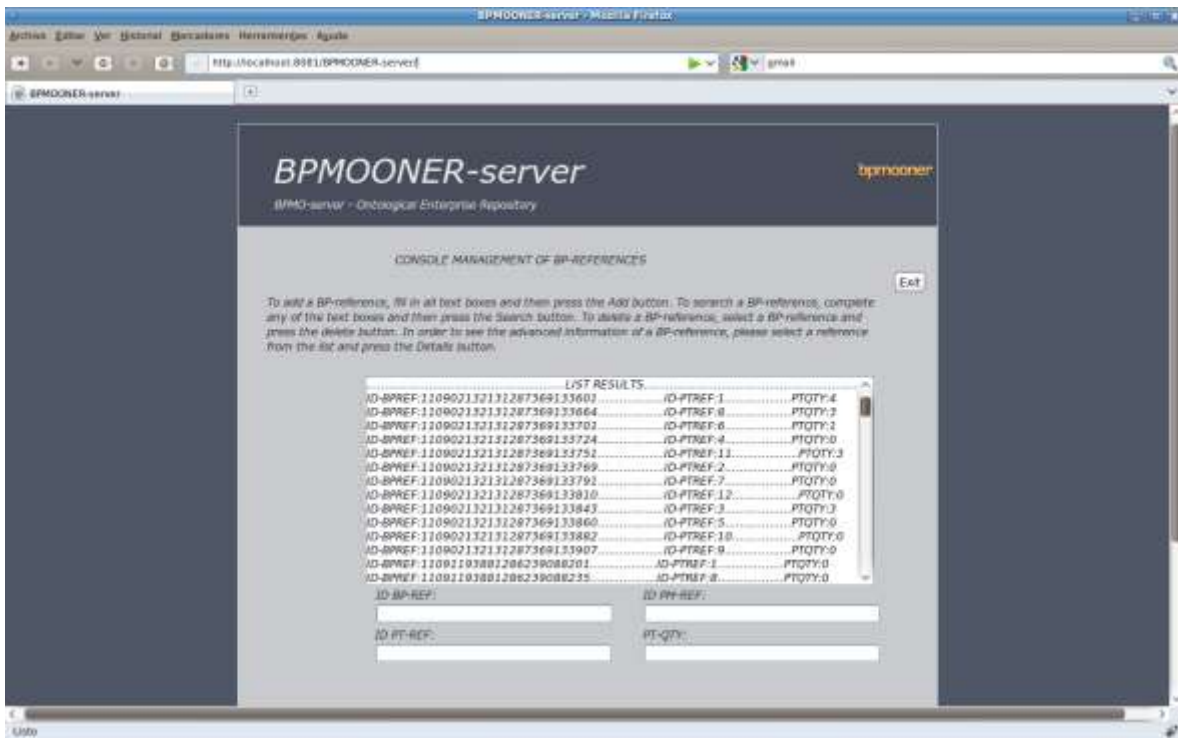


Figura 32. Interfaz de administración del BPMOONER-server.

5.2 Experimentación y Pruebas

Una vez concluida la implementación del prototipo operacional, es necesario someterlo a un proceso de evaluación y puesta a punto que permita identificar y corregir deficiencias en la operación del sistema y garantizar su óptimo desempeño.

Con el fin de llevar a cabo actividades de evaluación sobre el prototipo desarrollado, se requirió inicialmente el modelado de los Modelos BP. Posteriormente se procedió a realizar su publicación en el BPMOONER-server, permitiendo almacenar las referencias definidas para los Modelos BP o lo que se ha denominado *benchmark del prototipo*.

Benchmark, puede definirse como un punto de referencia a partir del cual es posible evaluar comparativamente el rendimiento de un sistema respecto de otro, el cual por lo general representa las mejores prácticas de su dominio. Para el caso de interés, el benchmark está integrado por los resultados obtenidos de la ejecución del sistema sobre un conjunto de Modelos BP predefinidos. De esta manera, para evaluar el rendimiento del prototipo, particularmente, para determinar que tan adecuado es el cálculo de la medida

de similitud obtenida por el mecanismo de comparación, se requiere precisamente conocer cuáles son los resultados de similitud apropiados, para cada una de las comparaciones ejecutadas por el prototipo, entendiendo estos resultados apropiados como aquellos que se ajustan al criterio de un experto del dominio.

5.2.1 Descripción del Benchmark de Referencia

El benchmark de referencia consiste en un conjunto de resultados de similitud, obtenidos a partir de las comparaciones manuales entre los flujos de control de los Modelos BP publicados, realizadas por un grupo de 8 evaluadores, con amplio conocimiento del dominio de los procesos de negocio.

Para generar este benchmark fue necesaria la utilización de una herramienta que permitiera llevar a cabo la comparación manual entre Modelos BP, a través de una interfaz gráfica de usuario, que soportara el acceso simultáneo de varios evaluadores y que almacenara los resultados de las comparaciones en una base de datos. Con base en estos requisitos se empleó el prototipo implementado en el marco del trabajo de grado: **“Plataforma para la Evaluación de Sistemas de Recuperación de Servicios Basados en Comportamiento”** que se llevó a cabo en la Universidad del Cauca.

En esta herramienta se realizaron en total 700 comparaciones manuales, entre parejas de procesos de negocio, garantizando que cada una de las 700 posibles parejas (7 procesos de consulta x 100 procesos publicados) fuera comparada por 8 evaluadores.

Para determinar el nivel de rendimiento del sistema, se compararon los resultados registrados en el benchmark del prototipo con aquellos consignados en el benchmark de referencia, a partir de un análisis estadístico basado en la aplicación de un conjunto de medidas empleadas en la evaluación del desempeño de sistemas de recuperación de información. Dichas medidas se describen en la siguiente sección.

5.2.2 Medidas de Desempeño

Una funcionalidad básica del sistema desarrollado en el presente proyecto, consiste en generar un ordenamiento (Ranking) de los Modelos BP (dispuestos por orden de similitud) para satisfacer una petición definida, por medio de un Modelo BP de Consulta. En este sentido, es posible evaluar la calidad de los resultados obtenidos en la ejecución de esta operación del sistema, a partir de la aplicación de medidas estadísticas ampliamente empleadas en la caracterización del desempeño de sistemas de recuperación de información (p.ej. motores de búsqueda en internet). Estas medidas se denominan: *precisión (p)*, *exhaustividad (r)*, *promedio de la efectividad(o)*, por sus siglas en inglés: *precision, recall, overall*.

Estas operan sobre la cardinalidad de tres conjuntos conformados por los Modelos BP recuperados por el prototipo, en respuesta a un Modelo BP de Consulta específico: el conjunto de *verdaderos positivos (TP)* (Modelos BP publicados y recuperados

correctamente de acuerdo con los resultados registrados en el benchmark de referencia), el conjunto de *falsos positivos (FP)* (Modelos BP publicados e incorrectamente recuperados) y el conjunto de *falsos negativos (FN)* (Modelos BP publicados que no se recuperan, a pesar de ser relevantes, de acuerdo con el benchmark de referencia). De esta manera, las medidas *precisión*, *exhaustividad* y *promedio de la efectividad* se definen como:

Precisión (p): se refiere a la proporción de Modelos clasificados como *verdaderos positivos* respecto al número total de Modelos BP publicados y recuperados por el prototipo, esto es:

$$p = \frac{|TP|}{|TP| + |FP|} \quad \text{Ecuación 1}$$

Exhaustividad (r): identifica la proporción de Modelos BP clasificados como *verdaderos positivos* respecto al número total de Modelos BP considerados como relevantes (es decir, todos aquellos Modelos BP que deben ser recuperados de acuerdo con el benchmark de referencia):

$$r = \frac{|TP|}{|TP| + |FN|} \quad \text{Ecuación 2}$$

Promedio de la efectividad (o): relaciona las medidas de *precisión* y *exhaustividad*, y se emplea para determinar la calidad del proceso de comparación y recuperación de Modelos BP. Matemáticamente se calcula:

$$o = r * \left(2 - \frac{1}{p} \right) \quad \text{Ecuación 3}$$

Con el fin de evaluar el rendimiento general del sistema, en cuanto a la calidad del método para estimar la similitud y a su capacidad para recuperar Modelos BP realmente relevantes, se llevó a cabo la aplicación de las medidas de desempeño descritas, sobre los resultados de similitud registrados en el benchmark del prototipo para cada una de los Modelos BP de consulta establecidos.

La siguiente sección detalla el plan de pruebas dispuesto para el sistema implementado junto con los resultados obtenidos a partir de su ejecución.

5.2.3 Plan de Pruebas y Resultados Obtenidos

A continuación se describen las pruebas de calidad y eficiencia ejecutadas sobre el prototipo desarrollado (Tabla 9):

Plan de Pruebas.

Publicación de un Modelo BP
Prueba de Rendimiento: permite medir el tiempo que le toma al sistema llevar a cabo la carga del archivo del Modelo BP a la App Web BPMOONER-client, la invocación del servicio Web para publicar el Modelo BP en la App Web BPMOONER-server, la validación de la versión del Modelo BP (versión 1.4), la transformación del Modelo BP a Grafo BP, la detección de los patrones de flujo de control y el almacenamiento de las referencias en una base de datos relacional.
Recuperación de Modelos BP según su flujo de control
Prueba de Rendimiento: permite determinar el tiempo que le toma al sistema generar una lista ordenada (Ranking) de Modelos BP según la medida de similitud de su flujo de control con respecto a los Modelos BP publicados para satisfacer un Modelo BP de consulta determinado.
Prueba de Calidad: permite evaluar la calidad de los resultados del ordenamiento generado (Ranking) generado por el sistema en términos de <i>precisión</i> , <i>exhaustividad</i> y <i>promedio de la efectividad</i> .

Tabla 9. Plan de Pruebas.

Los resultados de las pruebas de rendimiento están condicionados por las prestaciones del equipo donde se ejecuta el software del prototipo (Servidor DELL PowerEdge T100). En este caso, el equipo empleado para desplegar el plan de pruebas descrito, cuenta con las siguientes especificaciones técnicas (Tabla 10):

Procesador	Intel XEON Serie 3000 4 núcleos de 3GHz
Caché del microprocesador	16MB de caché de nivel 2
Memoria	8 GB (4 DIMM)
Disco Duro	1TB (5400 RPM)
Sistema Operativo	Ubuntu 9.10

Tabla 10. Especificaciones Técnicas del Equipo empleado para las Pruebas del Prototipo.

Resultados

A continuación se presenta una representación gráfica, que resume la información obtenida a partir de la ejecución del plan de pruebas descrito

- **Publicación de un Modelo BP**

Prueba de Rendimiento:

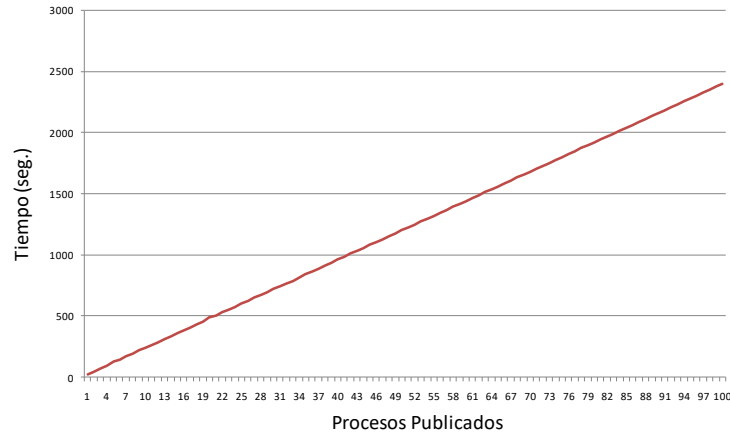


Figura 33. Gráfica de rendimiento del sistema durante la Publicación de Modelos BP.

De acuerdo con el comportamiento de la curva que se ilustra en la Figura 33, se observa que existe una relación de linealidad entre el número de Procesos de Negocio Publicados y el tiempo que toma el sistema en la ejecución de esta actividad. Por otra parte, se considera que el tiempo empleado en esta operación es aceptable (20 segundos para la publicación promedio de un Modelo BP), teniendo en cuenta la complejidad del algoritmo del parser y la detección de patrones y la gran cantidad de invocaciones realizadas a los métodos de las APIs.

- **Recuperación de BP Models según su flujo de control**

Prueba de Rendimiento:

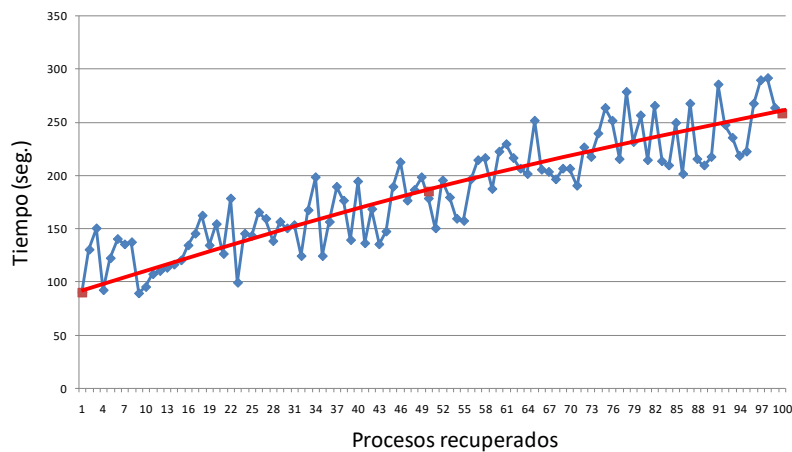


Figura 34. Gráfica de rendimiento del sistema durante la Recuperación de Modelos BP.

De acuerdo con el comportamiento de la curva que se ilustra en la Figura 34, se observa que existe cierta linealidad a partir de la normalización de los datos, entre el número de Procesos de Negocio Recuperados y el tiempo que toma el sistema en la ejecución de esta actividad. Por otra parte, se considera que el tiempo empleado en esta operación es aceptable (solo un incremento de aproximadamente 100 milisegundos por cada 50 modelos adicionales recuperados), teniendo en cuenta la complejidad del proceso de lectura y mapeo de la ontología y el ordenamiento (Ranking) de Modelos BP.

Prueba de Calidad:

Para las medida de calidad de *exhaustividad* (r) se utilizó como benchmark de referencia los resultados del numeral I del Anexo C del trabajo de grado “**PLATAFORMA PARA LA EVALUACIÓN DE SISTEMAS DE RECUPERACIÓN DE SERVICIOS BASADOS EN COMPORTAMIENTO**”, llevado a cabo dentro del Grupo de Ingeniería Telemática en la Universidad del Cauca, los cuales son presentados a continuación en la Tabla 11:

Resultados medidas de calidad							
TJ	Q	Rg	Pg	R	P	Og	O
Expertos	Q1	0.15	0.97	0.58	0.89	0.15	0.51
	Q2	0.14	0.91	0.54	0.72	0.12	0.33
	Q3	0.11	0.89	0.37	0.63	0.10	0.15
	Q4	0.06	0.71	1.00	0.59	0.04	0.31
	Q5	0.13	0.90	0.53	0.63	0.12	0.22
	Q6	0.12	0.75	1.00	0.53	0.08	0.11
	Q7	0.21	0.94	0.52	0.71	0.20	0.31
Intermedio	Q1	0.41	0.52	0.67	0.04	0.04	-14.33
	Q2	0.89	0.54	0.67	0.04	0.12	-14.00
	Q3	0.83	0.55	0.67	0.07	0.15	-8.67
	Q4	0.14	0.34	1.00	0.07	-0.14	-11.50
	Q5	0.94	0.61	1.00	0.10	0.35	-7.80
	Q6	0.12	0.14	1.00	0.04	-0.59	-23.50
	Q7	0.79	0.81	0.75	0.07	0.60	-9.75
Novato	Q1	0.59	0.52	1.00	0.04	0.05	-21.50
	Q2	0.44	0.54	0.67	0.04	0.06	-14.00
	Q3	0.38	0.55	0.67	0.07	0.07	-8.67
	Q4	0.06	0.34	1.00	0.07	-0.06	-11.50
	Q5	0.43	0.64	1.00	0.10	0.19	-7.80
	Q6	0.08	0.14	1.00	0.04	-0.38	-23.50
	Q7	0.34	0.81	0.75	0.07	0.26	-9.75

Tabla 11. Benchmark de referencia para la medida de calidad de *exhaustividad*.

Para realizar las pruebas de calidad del prototipo en la recuperación de Modelos BP (Figuras 35 - 40), se definieron 6 escenarios de pruebas dependiendo de la configuración de los coeficientes α : coeficiente de similitud de patrones, β : coeficiente de similitud de secuencias de patrones, δ : coeficiente de similitud semántica de patrones, de la función que calcula la medida de similitud entre los flujos de control de los Modelos BP (sección 4.4.1), estos escenarios son:

- **Escenario 1:** Función de similitud de flujo de control con coeficientes $\alpha > \beta > \delta$, los resultados obtenidos en este escenario, en su mayoría son útiles pero pocos corresponden con los resultados que se esperaba que el prototipo recuperara. La Tabla 12 muestra los resultados de las medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida para cada una de las 7 consultas en este escenario. Se observa que los valores de precisión en su totalidad superan el 50% para las 7 consultas, mientras que la exhaustividad presenta valores por

debajo del 30%, lo que significa que los coeficientes de este escenario provocan un comportamiento lejano al benchmark de referencia, aunque sus resultados son en promedio útiles. Por otra parte, el promedio de efectividad corresponde a valores por debajo del 22%.

	PRECISION	RECALL	OVERALL
Q1	0,55	0,12	0,02181818
Q2	0,81	0,16	0,12246914
Q3	0,71	0,22	0,13014085
Q4	0,95	0,13	0,12315789
Q5	0,68	0,15	0,07941176
Q6	0,82	0,24	0,18731707
Q7	0,93	0,23	0,21268817

Tabla 12. Medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida en el Escenario 1.

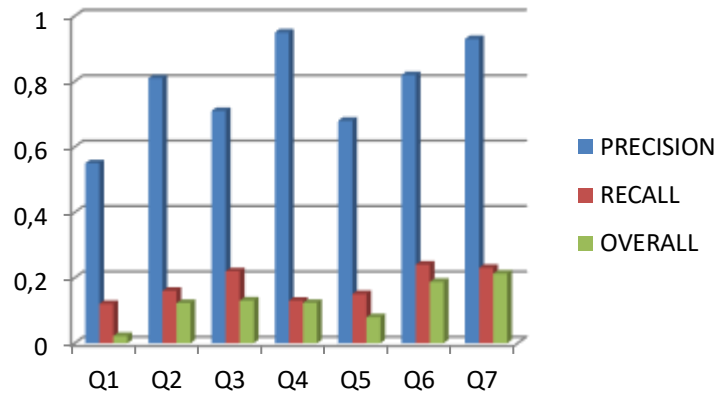


Figura 35. Escenario de Pruebas No 1.

- Escenario 2:** Función de similitud de flujo de control con coeficientes $\alpha > \delta > \beta$, los resultados obtenidos en este escenario en su mayoría son útiles, no tanto como el primer escenario, pero es mayor la cantidad de los resultados que corresponden con los resultados que se esperaba que el prototipo recuperara. La Tabla 13, muestra los resultados de las medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida para cada una de las 7 consultas en este escenario. Se observa que los valores de precisión en su totalidad, al igual que en el Escenario 1, superan el 50% para las 7 Consultas, mientras que la exhaustividad presenta valores por debajo del 50%, lo que significa que los coeficientes de este escenario provocan un comportamiento más cercano al benchmark de referencia, comparado con el Escenario 1. Por otra parte, el promedio de efectividad corresponde a valores por debajo del 32%.

	PRECISION	RECALL	OVERALL
Q1	0,76	0,47	0,32157895
Q2	0,59	0,36	0,10983051
Q3	0,65	0,48	0,22153846
Q4	0,72	0,31	0,18944444
Q5	0,52	0,46	0,03538462
Q6	0,55	0,41	0,07454545
Q7	0,61	0,49	0,17672131

Tabla 13. Medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida en el Escenario 2.

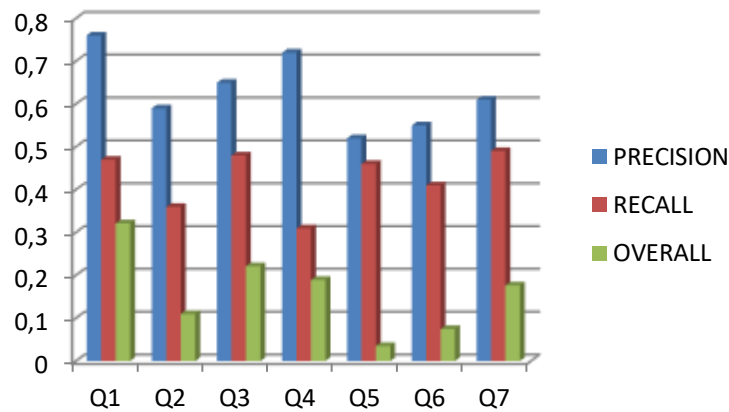


Figura 36. Escenario de Pruebas No 2.

- Escenario 3:** Función de similitud de flujo de control con coeficientes $\beta > \delta > \alpha$, los resultados obtenidos en este escenario, son medianamente útiles pero la cantidad de los resultados que corresponden con los resultados que se esperaba que el prototipo recuperara, es mucho mayor que los escenarios anteriores. La Tabla 14, muestra los resultados de las medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida para cada una de las 7 consultas en este escenario. Se observa que los valores de precisión en su totalidad, al igual que en el Escenario 1, están por debajo del 68% para las 7 Consultas, mientras que la exhaustividad presenta valores por debajo del 90%, lo que significa que los coeficientes de este escenario provocan un comportamiento muy cercano al benchmark de referencia, lo que quiere decir que los resultados recuperados corresponden, en gran medida, con los esperados pero de la totalidad de los recuperados hay pocos que son útiles. Por otra parte, el promedio de efectividad corresponde en más de la mitad de los casos a valores negativos, debido al bajo nivel de precisión respecto a la exhaustividad en estos casos.

	PRECISION	RECALL	OVERALL
Q1	0,65	0,92	0,42461538
Q2	0,39	0,59	-0,33282051
Q3	0,38	0,47	-0,29684211
Q4	0,68	0,65	0,34411765
Q5	0,45	0,78	-0,17333333
Q6	0,65	0,95	0,43846154
Q7	0,45	0,75	-0,16666667

Tabla 14. Medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida en el Escenario 3.

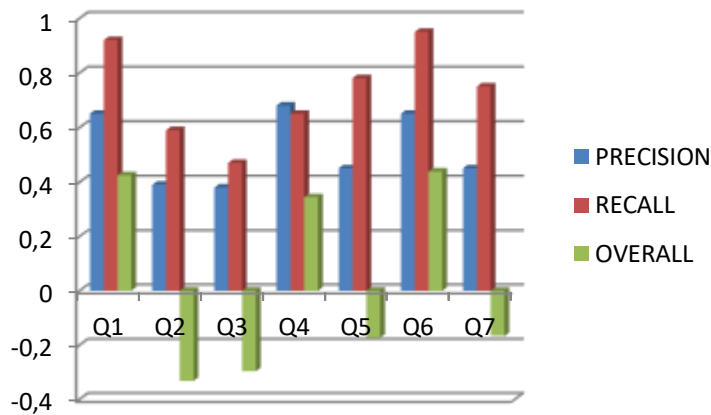


Figura 37. Escenario de Pruebas No 3.

- Escenario 4:** Función de similitud de flujo de control con coeficientes $\beta > \alpha > \delta$, en promedio este es el mejor escenario, ya que la mayoría de los resultados son útiles y corresponden con los resultados que se esperaba que el prototipo recuperara. La Tabla 15, muestra los resultados de las medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida para cada una de las 7 consultas en este escenario. Se observa que los valores de precisión en su totalidad, al igual que en el Escenario 1, están por encima del 85% para las 7 Consultas, mientras que la exhaustividad presenta valores por encima del 77%, lo que significa que los coeficientes de este escenario, provocan un comportamiento muy cercano al benchmark de referencia, lo que quiere decir que los resultados recuperados corresponden en gran medida con los esperados, de esta forma la gran mayoría de los recuperados son útiles. Por otra parte, el promedio de efectividad está por encima del 63%, que corresponde con el nivel más alto de los 6 escenarios definidos.

	PRECISION	RECALL	OVERALL
Q1	0,85	0,77	0,63411765
Q2	0,92	0,85	0,77608696
Q3	0,93	0,81	0,74903226
Q4	0,87	0,79	0,67195402
Q5	0,97	0,87	0,84309278
Q6	0,91	0,86	0,77494505
Q7	0,96	0,85	0,81458333

Tabla 15. Medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida en el Escenario 4.

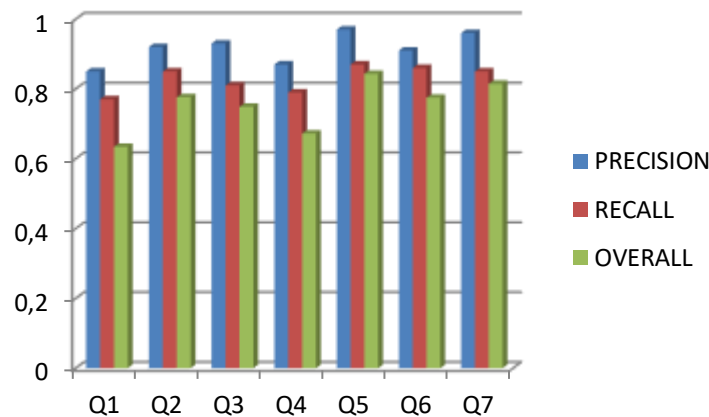


Figura 38. Escenario de Pruebas No 4.

- Escenario 5:** Función de similitud de flujo de control con coeficientes $\delta > \alpha > \beta$, no por mucho es superado por el anterior escenario ya que presenta una buena relación de precisión y recall. La Tabla 16, muestra los resultados de las medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida para cada una de las 7 consultas en este escenario. Se observa que los valores de precisión en su totalidad, al igual que en el Escenario 1, están por encima del 83% para las 7 Consultas, mientras que la exhaustividad presenta valores por encima del 79%, lo que significa que los coeficientes de este escenario, provocan un comportamiento muy cercano al benchmark de referencia, lo que quiere decir, que los resultados recuperados corresponden en gran medida con los esperados, de esta forma la gran mayoría de los recuperados son útiles. Por otra parte, el promedio de efectividad está por encima del 54% que corresponde a un buen nivel de acercamiento al benchmark de referencia, pero no tan bueno comparado con el escenario anterior.

	PRECISION	RECALL	OVERALL
Q1	0,83	0,87	0,69180723
Q2	0,76	0,79	0,54052632
Q3	0,82	0,92	0,71804878
Q4	0,84	0,83	0,67190476
Q5	0,77	0,88	0,61714286
Q6	0,8	0,82	0,615
Q7	0,74	0,89	0,5772973

Tabla 16. Medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida en el Escenario 5.

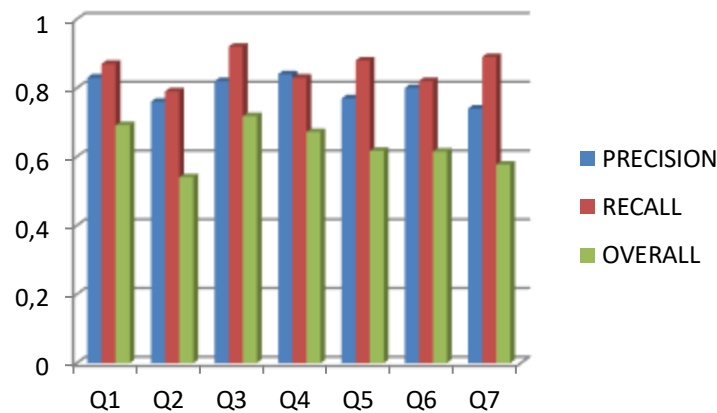


Figura 39. Escenario de Pruebas No 5.

- Escenario 6:** Función de similitud de flujo de control con coeficientes $\delta > \beta > \alpha$, este escenario presenta un buen nivel de precisión y recall, pero no alcanza al overall de los escenarios 4 y 5. La Tabla 17, muestra los resultados de las medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida para cada una de las 7 consultas en este escenario. Se observa que los valores de precisión en su totalidad, están por encima del 72% para las 7 Consultas, mientras que la exhaustividad presenta valores por encima del 59%, lo que significa que los coeficientes de este escenario, provocan un comportamiento muy cercano al benchmark de referencia, lo que quiere decir que los resultados recuperados corresponden en gran medida con los esperados, de esta forma la gran mayoría de los recuperados son útiles. Por otra parte, el promedio de efectividad está por encima del 36% que corresponde a un comportamiento aceptablemente cercano al benchmark de referencia.

	PRECISION	RECALL	OVERALL
Q1	0,78	0,71	0,50974359
Q2	0,72	0,63	0,385
Q3	0,73	0,74	0,46630137
Q4	0,79	0,67	0,49189873
Q5	0,72	0,59	0,36055556
Q6	0,74	0,69	0,44756757
Q7	0,76	0,72	0,49263158

Tabla 17. Medidas de calidad precisión, exhaustividad y promedio de efectividad obtenida en el Escenario 6.

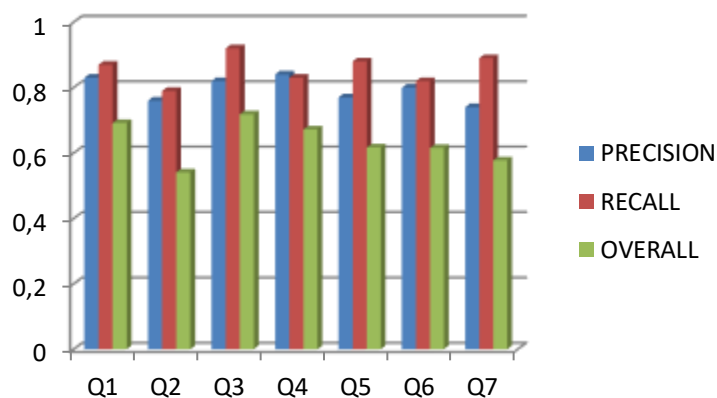


Figura 40. Escenario de Pruebas No 6.

Durante este proceso se realizaron 4200 comparaciones automáticas, 100 para cada uno de los 7 modelos de consulta, en cada uno de los 6 escenarios definidos. La Figura 40 resume el comportamiento promedio de las medidas de calidad para los 7 Modelos BP de consulta, en cada uno de los escenarios. Esta, permite evidenciar que el mejor comportamiento según las medidas de calidad utilizadas, corresponde al escenario 4 en el cual se alcanzó el mayor nivel de efectividad promedio (Overall).

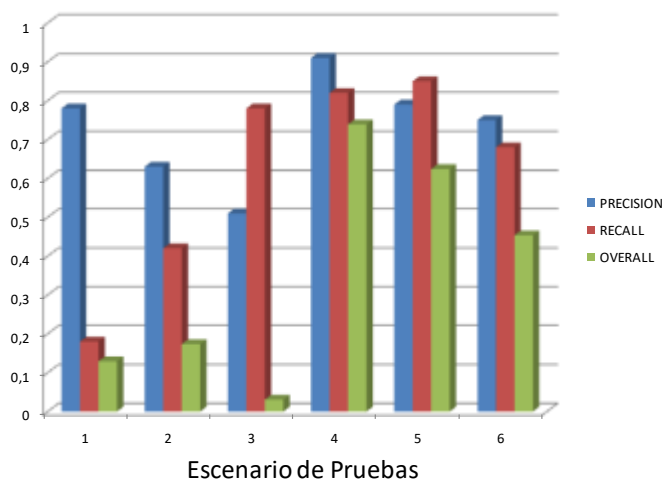


Figura 41. Medidas de Calidad: Precisión, Exhaustividad y Promedio de Efectividad.

5.3 Resumen

En este capítulo fueron abordadas las siguientes temáticas: el prototipo que valida la arquitectura propuesta en los Capítulos 3 y 4, el diagrama de Casos de Uso del Sistema, el Software utilizado para la implementación, los Diagramas de Paquetes, subsistemas, capas del Sistema, las Interfaces gráficas de usuario, la especificación del plan pruebas realizadas, la descripción del Benchmark de Referencia, junto con los resultados obtenidos a partir de la ejecución de dichas pruebas.

Capítulo VI

6 CONCLUSIONES Y TRABAJOS FUTUROS

Este capítulo describe inicialmente las principales contribuciones del trabajo realizado, posteriormente presenta las conclusiones a las que se llegó durante su desarrollo, y finalmente propone los trabajos futuros.

6.1 Contribuciones

Entre las principales contribuciones de éste proyecto de grado se destacan las siguientes:

- La generación de una ontología que describe las relaciones basadas en funcionalidad, comportamiento, especialización y composición, entre los 12 patrones básicos definidos en BPMO. Esta ontología fue formulada tomando como referencia las ontologías del proyecto SUPER, Sembiz, m3pe y permite dotar de semántica al mecanismo de recuperación de Modelos BP.
- La definición de un mecanismo de comparación semántica de patrones de flujo de control entre procesos de negocio descritos en BPMO, el cual está basado en el cálculo de la similitud semántica entre los parámetros que describen dichos patrones. Este mecanismo se compone de una serie de algoritmos que comprenden isomorfismo de grafos, detección de trazas en un grafo, cálculo de la distancia y similitud entre atributos de instancias de conceptos pertenecientes a la ontología, los cuales fueron propuestos a partir del estudio de algoritmos aplicados en el descubrimiento e indexación de procesos de negocio.
- La aplicación Web *BPMOONER (BPMO Ontological Enterprise Repository) ó Repositorio Empresarial Ontológico de BPMO*, fue definida a partir del mecanismo de comparación antes mencionado y constituye una herramienta automática para la recuperación de Modelos BP. Esta aplicación permite buscar y comparar un Modelo BP con todos los Modelos BP almacenados en el repositorio, partiendo del cálculo de similitud semántica entre los procesos comparados, para luego generar un Ranking de Modelos BP de acuerdo a los valores de similitud encontrados.
- Los resultados de comparación de ésta herramienta pueden contribuir a la disminución del tiempo de despliegue de nuevas soluciones empresariales, en la medida en que disminuyen el tiempo que requiere la búsqueda de Modelos BP, independizando esta actividad de la interpretación subjetiva de personal técnico que realiza las búsquedas en forma manual.
- El modelado en BPMO de un banco de 20 procesos de negocio. Estos procesos sirvieron de base para las comparaciones realizadas por la aplicación *BPMOONER*.

Finalmente, el aporte de la aplicación *BPMOONER (BPMO Ontological Enterprise Repository)* ó *Repositorio Empresarial Ontológico de BPMO*, al trabajo de grado de maestría “Descubrimiento Automático de Procesos de Negocio Basado en Semántica del Comportamiento” del Ing. Cristhian Figueroa Martínez desarrollado en la Universidad del Cauca. Esta herramienta sirve de soporte para la fase de Pre-Matching, encargada de generar el Ranking de Modelos BP, que facilita la comparación estructural del módulo Analizador Semántico del Comportamiento definido en su arquitectura. De esta manera se contribuye a la investigación en torno a la temática del descubrimiento de procesos de negocio soportado en la aplicación de ontologías independientes del dominio de aplicación, con el objetivo de incentivar futuros trabajos dentro del grupo de Ingeniería Telemática (GIT) de la Universidad del Cauca, integrando las líneas de investigación relacionadas con aplicaciones y servicios en internet y servicios avanzados de telecomunicaciones.

6.2 Conclusiones

Dentro del proyecto realizado se abordó la problemática de recuperación de modelos de procesos de negocio descritos en BPMO (Modelos BP), la cual fue resuelta a partir de la detección de patrones de flujo de control por medio de un algoritmo de isomorfismo de grafos y el razonamiento semántico, sobre una ontología de patrones de flujo de control que describe las funcionalidades de los patrones y sus relaciones de especialización y composición.

Al finalizar este proyecto se entregó la aplicación Web *BPMOONER*, la cual se basa en el mecanismo antes mencionado, permitiendo recuperar una lista ordenada de modelos de procesos de negocio (Ranking de Modelos BP), según la medida de similitud de su flujo de control con respecto a un modelo de consulta. Esta aplicación admite configurar los coeficientes que determinan el cálculo de la medida de similitud para personalizar la manera en que será generado el Ranking.

Con lo anterior se puede concluir que el trabajo realizado facilita la recuperación de modelos de procesos de negocio almacenados en un repositorio, a partir de un método de búsqueda semántico, basado en el razonamiento ontológico sobre las actividades que componen el flujo de control de dichos procesos.

A continuación son descritas las principales conclusiones que surgieron durante la ejecución del presente proyecto:

- Teniendo en cuenta las principales notaciones para modelar procesos de negocio disponibles en el entorno académico y empresarial, el proceso de investigación realizado, permitió seleccionar a BPMO como la notación con las mejores características para el modelado de procesos de negocio semánticos. Aunque ésta notación, propuesta en el marco del proyecto SUPER, reúne las mejores características de cada una de las notaciones evaluadas, no cuenta con una

documentación adecuada, lo cual dificulta la utilización de algunos plugins de desarrollo como el modelador BPMO 1.4.

- A través del estudio realizado, en torno a los diferentes repositorios de modelos de procesos que se encuentran en el entorno académico y empresarial, fue posible generar un estado del arte que sirvió como base de conocimiento para formular la arquitectura de referencia propuesta. El repositorio de modelos generado a partir de esta arquitectura tiene las características de ser un repositorio basado en capas, extensible y modular que proporciona una fácil adaptación para incluir notaciones de modelos de procesos adicionales a BPMO y así realizar búsquedas de procesos descritos en diferentes notaciones.
- Transformar el problema de recuperación de procesos de negocio en un problema de recuperación de grafos, es posible a través del uso de una técnica de indexación de grafos, la cual permite realizar búsquedas rápidas sobre los índices que identifican las representaciones formales basadas en grafos de los procesos de negocio.
- La utilización de una ontología de patrones de flujo de control, permite recuperar modelos de procesos de negocio, que aunque no posean un flujo de control sintácticamente similar al modelo de consulta, si posee similitud semántica en cuando a las funcionalidades de sus patrones con respecto al modelo de consulta.
- Las pruebas de rendimiento realizadas sobre la aplicación Web *BPMOONER*, permiten concluir que el tiempo que toma realizar la publicación y recuperación de los modelos de procesos, es considerable debido a que dicho procedimiento involucra un gran número de operaciones complejas y dispendiosas, relacionadas principalmente con el razonamiento sobre las ontologías de soporte y el manejo de arreglos de objetos. Sin embargo, en contraste al tiempo que puede tomar realizar dicho procedimiento de forma manual, los resultados de estas pruebas son muy pequeños, lo cual puede considerarse como una mejora en la disminución del tiempo de despliegue de nuevas soluciones en las empresas.

6.3 Trabajos Futuros

El presente trabajo puede considerarse como una propuesta inicial de un mecanismo de búsqueda semántica de modelos de procesos de negocio, que pretende aportar soluciones a la problemática de recuperación de modelos de procesos de negocio. Considerando lo anterior, en el campo de investigación del presente trabajo de grado, se proponen los siguientes trabajos futuros:

Implementación de una ontología que contenga las relaciones existentes entre todos los patrones de flujo de control de referencia viables en BPMO

La ontología implementada (*bpmooner-1.0.0.wsml*) contempla solamente 12 patrones básicos de BPMO. Este trabajo de grado se podría extender por medio de la

implementación en el lenguaje WSML, de una ontología que soporte la totalidad de los 21 patrones de flujo de control de referencia viables en BPMO.

Adicionar a la capa de BP Parser el soporte para lenguajes de modelado adicionales

A partir de la arquitectura propuesta se propone la implementación de parsers que permitan generar a partir de diferentes notaciones de modelado de procesos de negocio, tales como BPMN, YAWL, EPC, etc., una representación formal unificada basada en grafos.

Búsqueda sintáctica y semántica unificada de procesos de negocio

A través de las características de modularidad, extensibilidad y reusabilidad de la arquitectura propuesta, es posible realizar búsquedas sintácticas y semánticas de varios modelos de procesos de negocio, creados a partir de diferentes lenguajes de modelado, utilizando una representación formal unificada basada en grafos. Es posible adicionar el análisis sintáctico (a través de una base de datos léxica como WordNet) al mecanismo de recuperación de modelos de procesos, propuesto con el fin de acentuar la precisión en los resultados obtenidos.

Plataforma para la búsqueda y composición de procesos de negocio semánticos

Es posible crear una plataforma que gestione de manera automática los modelos de procesos y que además de realizar búsqueda de modelos de procesos de negocio, permita un entorno de composición de procesos con los resultados generados a partir del Ranking. El usuario podría utilizar la plataforma para visualizar los modelos de procesos y a partir de los resultados del Ranking, generar nuevos modelos de procesos.

Experimentación de la aplicación *BPMOONER (BPMO Ontological Enterprise Repository)* en un entorno empresarial real

Se propone realizar la experimentación de la aplicación Web *BPMOONER* en un entorno empresarial real, para determinar su desempeño y validar su utilidad en actividades de gestión y reingeniería de procesos de una empresa.

REFERENCIAS

1. EHRIG, M.K., Agnes; OBERWEIS, Andreas, *Measuring Similarity between Semantic Business Process Models*. 2006.
2. Castells, P., *La Web Semántica*. 2005, Universidad Autónoma de Madrid.
3. W3C, *Guía Breve de Web Semántica*. 2005, World Wide Web Consortium Oficina Española.
4. W3C, *Guía Breve de Servicios Web*. 2005, World Wide Web Consortium Oficina Española.
5. OLAYA, A.P., Nelson, *Los Servicios Web Semánticos, una solución a los problemas de interoperabilidad Modelos Digitales de Terreno*. 2006.
6. LOZANO, T., *Ontologías en la Web semántica*. I Jornadas de ingeniería Web' 01, 2001.
7. SAMPER ZAPATER, J.J., *Ontologías para servicios Web semánticos de información de tráfico: Descripción y herramientas de explotación*. 2005.
8. Jung, J., Choi, I., and Song, M. , *An integration architecture for knowledge management systems and business process management systems*. . Comput. Ind 2007. 58(1): p. 21-34s.
9. SILVER, B., *The 2006 BPMS Report: Understanding and Evaluating BPM Suites*. 2005.
10. ANAYA, V.O., Angel. , *BMPS: Soportando Procesos Extendidos*. 2006.
11. Pérez C, N., et al. (2008) *Gestión de Procesos de Negocio Semánticos*.
12. Vanhatalo, J., J. Koehler, and F. Leymann, *Repository for Business Processes and Arbitrary Associated Metadata*, in *Proc of the BPM Demo Session at the Fourth International Conference on Business Process Management*. 2006.
13. IBM, *alphaWorks: BPEL Repository: Overview*. 2007, IBM, SOA and Web Services.
14. MA Zhilei, *BPL Business Process Library*. SUPER Project, 2008.
15. ZAREMBA, M.W., Brian; BROWNE, Jimmie,, *ebXML Registry/Repository Implementation in the Agri-food industry*. 2003.
16. Specification, T.W.M.C., *Workflow Management Coalition, Terminology & Glossary*. 1999.
17. Anghi M., M.R., Spoorti P. , *Implement Business Process Management to realize Cost Savings and High Return on Investments*. 2009, TATA Consultancy Services.
18. White, S.A., *Introduction to BPMN*. 2006, IBM Software Group.
19. LIST, B., KORHERR, Birgit, *An Evaluation of Conceptual Business Process Modelling Languages*. 2005: Vienna University of Technology.
20. Curtis, B., Kellner, M. and Over, J., *Process Modeling*. Communication of the ACM, 1992. 9.
21. G. Keller, M.N., and A.W. Scheer, *Semantische Prozessmodellierung auf der Grundlage Ereignisgesteuerter Prozessketten (EPK)*. 1992, University of Saarland: Saarbrücken.
22. Aalst, W.M.P.v.d., *Formalization and Verification of Event-driven Process Chains*. 2000, Department of Mathematics and Computing Science: Eindhoven University of Technology.
23. Aalst, W.M.P.v.d., *Projecting multi-dimensional Petri nets*. 2000: Eindhoven University of Technology.
24. Züllighoven, D.R.a.H., *Understanding and Using Patterns in Software Development*. Theory and Practice of Object Systems, 1996.
25. N. Russell, A.H.M.t.H., W.M.P. van der Aalst, *new YAWL: Specifying a Workflow Reference Language using Coloured Petri Nets*. 2007.
26. S. Heymans, e.a., *Deliverable D1.1. Process Modelling Ontology and Mapping to WSMO*. . 2007, SUPER Consortium.
27. MELERO, R., *Acceso abierto a las publicaciones científicas: definición, recursos, copyright e impacto [En línea]*. El profesional de la información, 2005. 14(4): p. 255-266.
28. Marta Sabou, J.P., *Towards semantically enhanced Web service repositories*. 2007.
29. Maleshkova, M., *Acquisition and Management of Semantic WebService Descriptions*. 2008, .NET XML Web Services Repertory.

30. Uwe Keller, M.S., Dieter Fensel, *WOOGLÉ meets Semantic Web Fred*. 2004.
31. Mahmoud, E.A.-M.a.Q.H., *Investigating Web Services on the World Wide Web*. www 2008, 2008.
32. Karl Czajkowski, e.a., *The WS-Resource Framework*. 2004.
33. Julie Allinson, S.F.a.S.L., *SWORD: Simple Web-service Offering Repository Deposit*. 2009.
34. Daniel Bachlechner, H.L., Katharina Siorpaes, *Web Service Discovery – A Reality Check*. 2006.
35. *Web Services Search Engine*. .
36. Yan, Z., R. Dijkman, and P. Grefen, *Business Process Model Repositories - Framework and Survey*. 2009.
37. KACZMAREK, M.K.M., *Deliverable 3.4. Business Process Library Final Prototype*. 2008.
38. KACZMAREK, M.K., Mihail, *Deliverable 3.1 Business Process Library Design and first Prototype*. 2007.
39. Barry, B., *Deliverable 6.9 Process Ontology and Query Reasoner - Final Implementation*. 2008.
40. *An overview of ebXML Registry and Repository standard*.
41. *What you always wanted to know about ebBP in five simple questions*.
42. VANHATALO, J., *Building and Querying a Repository of BPEL Process Specifications. Research Report*. 2004, Université Nice Sophia Antipolis.
43. VANHATALO, J., *Building and Querying a Repository of BPEL Process Specifications*. 2004, IBM Research: Zurich Laboratory.
44. I. Choi, K.K., and M. Jang., *An xml-based process repository and process query language for integrated process management. Knowledge and Process Management*, 2007: p. 303-316.
45. I. Choi, H.J., M. Song, and Y. Eyu., *Ipm-epdl: an xml-based executable process definition language. Computers in Industry*, 2005. 56: p. 85-104.
46. M. Song, J.A.M., and I.B. Arpinar., *RepoX: An XML Repository for Workflow Designs and Specifications*. 2001, University of Georgia, USA.
47. T. Theling, J.Z., P. Loos, and D. Vanderhaeghen, *An architecture for collaborative scenarios applying a common bpmn-repository*. In *Proceedings of DAIS 2005*, 2005: p. 169-180.
48. G. Decker, H.O., and M. Weske., *Oryx C An Open Modeling Platform for the BPM Community*. In *Proceedings of BPM 2008*, 2008: p. 382-385.
49. C. Beerli, A.E., T. Milo, and A. Pilberg., *Bp-mon: Query-based monitoring of bpel business processes*. . *SIGMOD Record* 2008. 1(37): p. 21-24.
50. T.W. Malone, K.C., and G.A. Herman, *Organizing Business Knowledge: The MIT Process Handbook*. MIT Press, 2003.
51. S. Fiorini, J.L., and C. Lucena., *Process reuse architecture.*, in *In Proceedings of CAiSE 2001*. 2001: Interlaken, Switzerland. p. 284-298.
52. Yang., G., *Towards a library for process programming*, in *In Proceedings of BPM 2003*. 2003: Eindhoven, The Netherlands. p. 120-135.
53. C. Liu, X.L., X. Zhou, and M. Orłowska. *Building a repository for workflow systems*. in *In Proceedings of Technology of Object-Oriented Languages and Systems*. 1999. Nanjing, China.
54. A. Wasser, M.L., and R. Karni. *Processgene query- a tool for querying the content layer of business process models*. in *Proceedings of BPM 2006*,. 2006. Vienna, Austria,.
55. R. Lu, S.S., and G. Governatori., *On managing business processes variants*. *Data & Knowledge Engineering*, 2009. 68(7): p. 642-664.
56. I. Markovic, A.P., and N. Stojanovic., *A framework for queying in business process modeling*. *Multikonferenz Wirtschaftsinformatik*, 2008.

57. M. Elhadad, a.M.B., *Effective business process outsourcing: The prosero approach. International Journal of Interoperability in Business Information Systems*, 2008. 3(1).
58. La Rosa, M.a.R., Hajo A. and Aalst, Wil M.P. van der and Dijkman, Remco M. and Mendling, Jan and Dumas, Marlon and Garcia-Banuelos, Luciano, *APROMORE : An Advanced Process Model Repository*, Q.e.T.R. #27448, Editor. 2009, Queensland University of Technology.
59. Han, X.Y.a.J., *gSpan: Graph-Based Substructure Pattern Mining. ICDM'02*, 2002.
60. Han, X.Y.a.J., *CloseGraph: Mining Closed Frequent Graph Patterns. KDD'03*, 2003.
61. F. Zhu, X.Y., J. Han, and P. S. Yu, *gPrune: A Constraint Pushing Framework for Graph Pattern Mining. PAKDD'07*, 2007.
62. Giugno, R. and D. Shasha, *GraphGrep: A Fast and Universal Method for Querying Graphs.*, in *Proc. of ICPR*. 2002. p. 112-115.
63. A. Ferro, R.G., M. Mongiovi', A. Pulvirenti, D. Skripin, D. Shasha . *Graphblast: Multi-Feature Graphs Database Searching*, in *Workshop On Network Tools And Applications In Biology - NETTAB 2007*. 2007.
64. S. Srinivasa, M.H.S., *GRACE: A Graph Database System. COMAD 2005b*, 2005.
65. X. Yan, J.H., P. S. Yu, *Graph Indexing: A Frequent Structurebased Approach. ACM SIGMOD 2004*, 2004.
66. T. Jin, J.W., N. Wu, M. La Rosa and A.H.M. ter Hofstede, *Efficient and Accurate Retrieval of Business Process Models through Indexing.* , Q.e.T.R. #31996, Editor. 2010: Queensland University of Technology
67. Ullmann, J.R., *An Algorithm for Subgraph Isomorphism. Journal of the ACM*, 1976. 23(1): p. 31-42.
68. Luigi P. Cordella, P.F., Carlo Sansone and Mario Vento., *A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs. IEEE Transactions on Pattern Analysis and Machine intelligence*, 2004. 26(10).
69. Daniel Felipe Rivas, D.S.C., Cristihan Figueroa, Juan Carlos Corrales, Rosalba Giugno, *Business Process Model Retrieval based on Graph Indexing Method rBPM 2010*, 2010.
70. Corrales, J.C., *Behavioral matchmaking for service retrieval.*, in *Computer Science*. 2008, University of Versailles Saint-Quentin-en-Yvelines: Versailles.
71. Russell, N., et al., *Workflow Control-Flow Patterns: A Revised View.* , in *BPM Center Report BPM-06-22* BPMcenter.org, Editor. 2006, BPM Center.
72. Edwards, J., *Oracle Berkeley DB Java Edition*. 2006.
73. Daniela Grigori, J.C.C., Mokrane Bouzeghoub, Ahmed Gater *Ranking BPEL Processes for Service Discovery. IEEE Transactions on Services Computing 2010*(IEEE computer Society Digital Library).
74. R. Dijkman, M.D., B. van Dongen, R. Käärik, and J. Mendling. , *Similarity of business process models: Metrics and evaluation in BETA Research School* W.P. 269, Editor. 2009: Eindhoven, The Netherlands
75. Gruber, T., *A translation Approach to portable ontology specifications*. 1993, Knowledge Acquisition. p. 199-220.
76. Bishop, B. and F. Fischer, *IRIS - Integrated Rule Inference System. Advancing Reasoning on the Web: Scalability and Commonsense - ARea2008*, 2008. Vol-350(2).
77. de Bruijn, J., y otros, *The Web Service Modeling Language WSML*. 2005: WSML Deliverable D16.1v0.2.
78. SERRANO, C., *Un Modelo Integral para un Profesional en Ingeniería.*. 2003: Universidad del Cauca.