MODELO DE AGRUPACIÓN DE PROCESOS DE NEGOCIO BASADO EN UNA REPRESENTACIÓN MULTIMODAL

HUGO ARMANDO ORDOÑEZ ERAZO

Universidad del Cauca
Instituto de Postgrados en Electrónica y Telecomunicaciones
Departamento de Telemática
Doctorado en Ingeniería Telemática
Popayán
2015

HUGO ARMANDO ORDOÑEZ ERAZO

MODELO DE AGRUPACIÓN DE PROCESOS DE NEGOCIO BASADO EN UNA REPRESENTACIÓN MULTIMODAL

Tesis presentada para optar al título de *Philosophiae Doctor* (PhD) en Telemática

Director: Juan Carlos Corrales, PhD
Codirector: Carlos Alberto Cobos Lozada, PhD

Universidad del Cauca
Instituto de Postgrados en Electrónica y Telecomunicaciones
Departamento de Telemática
Doctorado en Ingeniería Telemática
Popayán
2015

Agradecimientos

- A Jehová - Dios fuente de toda sabiduría y conocimiento, por darme la oportunidad de vivir, conocerlo y confiar en Él.
- A mi familia, por su constante apoyo, en especial a mis padres Helena y Guillermo, quienes con amor, cariño y valiosos consejos me animan cada día a seguir adelante con mis sueños, metas y aspiraciones, fuente significativa para poder culminar satisfactoriamente con mi Doctorado
- A la Universidad del Cauca, por brindarme la oportunidad a través del programa de Doctorado en Ingeniería Telemática, de culminar mis estudios de postgrado.
- A mis directores, PhD. Juan Carlos Corrales Muñoz y PhD. Carlos Alberto Cobos Lozada por su tiempo, dedicación, conocimientos y consejos compartidos durante la formación en el Doctorado.
- A los profesores, PhD. Leandro Krug Wives y PhD. Lucineia Heloisa Thom por su colaboración en la estancia de investigación en la Universidad Federal de *Rio Grande do Sul*, Brasil.
- A mis compañeros y docentes por compartir sus conocimientos conmigo en todo momento.
- A la Universidad de San Buenaventura, por el apoyo en la fase final de mi Doctorado.

Resumen Estructurado

Antecedentes

La evolución de la economía y el entorno cambiante de la tecnología, han generado que las técnicas de gestión de procesos empresariales se desarrollen rápidamente tanto en campos académicos como industriales. Para aumentar la flexibilidad y la capacidad de control en la gestión de las organizaciones, son utilizados procesos de negocio (BP). Estos BP permiten describir los servicios que ofrece una organización y los procesos internos que implementan esos servicios. En consecuencia, es común encontrar repositorios con cientos o incluso miles de BP. A medida que los repositorios de BP, aumentan de tamaño, se hace necesario contar con herramientas y técnicas para la gestión de dichos repositorios que permitan buscar o agrupar los BP que cumplen con ciertos criterios de búsqueda.

Con base en lo anterior se han planteado técnicas para la búsqueda o agrupación de BP. Las técnicas para búsqueda de BP, toman un BP como consulta y devuelven una lista ordenada descendentemente de BP en relación con el nivel de similitud entre el BP de consulta y cada uno de los ítems de la lista. Estas propuestas en el proceso de búsqueda se limitan al emparejamiento de entradas y/o salidas, para lo cual toman como base la información textual (por ejemplo nombre o descripción de las actividades, eventos y compuertas lógicas) de los elementos pertenecientes a cada BP. Estas propuestas cumplen parcialmente con las necesidades de los expertos desarrolladores debido a que no tienen en cuenta información relacionada con el flujo de ejecución del BP, comportamiento, estructura, tipo de actividades, tipo de compuertas y tipo de eventos.

Por otra parte, algunas propuestas basadas en agrupación utilizan algoritmos de *clustering* jerárquico para construir una jerarquía de grupos similares con base en la información estructural y de comportamiento de los BP. En estas propuestas los usuarios revisan la jerarquía y seleccionan el grupo de mayor similitud a su búsqueda. Otras propuestas usan algoritmos de *clustering* secuencial, los cuales toman como datos de entrada archivos de ejecución de los BP, conocidos como Log. En estas propuestas, el algoritmo agrupa BP con el mismo tipo de comportamiento en un mismo grupo. A pesar de los aportes ya realizados por estos enfoques, los

resultados pueden ser mejorados teniendo en cuenta más características de información presente en los BP y no centrándose solamente en un solo tipo como la información textual.

Con el fin de alcanzar mayor relevancia en los resultados reportados por un sistema de búsqueda de BP, esta tesis planteó un enfoque de búsqueda multimodal que unifica en un solo espacio de búsqueda información de tipo estructural y lingüística (textual), con el propósito de tener mayor precisión y cobertura en las búsquedas realizadas. Además, esta tesis integra un algoritmo de *clustering* iterativo, para agrupar los resultados recuperados en cada consulta. El algoritmo utiliza una función de similitud basada en lógica difusa, que mide el grado de semejanza entre los tipos de información de los BP recuperados. Los resultados son agrupados, para que el usuario en sus consultas revise de forma organizada un mayor número de BP en menor tiempo.

Objetivos

Proponer un modelo de Agrupación de Procesos de Negocio basado en una representación multimodal que usa *codebooks* (unidades básicas estándar de comportamiento secuencial) y características textuales, buscando obtener resultados más relevantes a las búsquedas realizadas por desarrolladores de BP

Objetivos Específicos

- Definir un diccionario de *codebooks* (unidades básicas estándar de comportamiento secuencial) de BP modelados con la notación BPMN.
- Establecer un modelo de representación multimodal, que incorpore el diccionario previamente definido y las características textuales de los BP.
- Proponer un algoritmo que agrupe BP, basado en el modelo de representación multimodal.
- Medir la calidad de los resultados entregados por el modelo, a través de evaluaciones experimentales con medidas tales como precisión, recuerdo y media F en colecciones cerradas de BP.

Métodos

La metodología de la presente investigación se basó en las fases de construcción del modelo de investigación documental y el modelo para construcción de soluciones. La metodología permitió manejar la complejidad del problema y evaluar adecuadamente las características que el modelo de agrupación y búsqueda multimodal debía cumplir en cada fase. En la fase inicial se realizó una revisión de la literatura y la construcción del estado del arte acerca de la gestión de BP, métodos y algoritmos de búsqueda de BP, modelos de indexación y algoritmos de *clustering*. Luego fue definida la estructura del *codebook* y el algoritmo para su formación, para lograr con esto el cumplimiento del primer objetivo específico. La fase 2 se enfocó en establecer una representación para la indexación y búsqueda multimodal que integra información textual y estructural de los BP, usando el *codebook* previamente definido, la representación en forma de árbol de los BP y la función de ponderación de cada elemento dentro del índice multimodal. Luego, fueron desarrollados los algoritmos de indexación y búsqueda. Lo anterior dio cumplimiento al segundo objetivo específico. Durante la fase 3, se estudiaron algunos de los algoritmos de *clustering* más referenciados en el estado del arte, y fueron adaptados para agrupación de BP, el modelo de búsqueda fue extendido con un algoritmo de *clustering* basado en la teoría de grafos que utiliza un función de lógica difusa para medir el grado de similitud entre los BP a agrupar, además, se adaptó un algoritmo de etiquetado de grupos cumpliendo así con el tercer objetivo específico. En la fase 4 fue ejecutado el proceso de evaluación del modelo de búsqueda y agrupación de BP propuesto. Para esto, se construyó una plataforma que permitió a 56 jueces expertos en gestión de BP, crear un repositorio cerrado de pruebas de BP, el cual contiene un conjunto de BP de consulta y sus respuestas ideales, además la agrupación ideal (formada manualmente por los evaluadores) de los resultados de cada consulta. Finalmente, en esta fase se evaluó la relevancia de los resultados en su presentación como lista ordenada, se comparó con otros modelos de búsqueda del estado del arte, y se evaluó la agrupación con medidas de evaluación interna y externa. Con el proceso de evaluación se dio cumplimiento al cuarto y último objetivo específico.

Resultados

Los resultados alcanzados en el desarrollo de la presente tesis están enfocados en tres aspectos: 1) generación de nuevo conocimiento y desarrollos tecnológicos a través de: la definición de un nuevo modelo de agrupación de BP basado en una representación multimodal y el desarrollo de *MultiSearchBP*; una herramienta *software* para búsqueda y agrupación de BP, la cual implementa el modelo propuesto; construcción de una plataforma Web para la creación colaborativa de repositorios de BP de prueba (repositorios cerrados para evaluación); desarrollo de una plataforma web para la formación colaborativa de grupos de BP; definición de un método para la creación colaborativa de repositorios de prueba de BP y creación de un repositorio de pruebas cerrado de BP. 2) fortalecimiento de la comunidad científica nacional a través de la dirección de tres trabajos de grado a nivel profesional, tutoría de dos trabajos de postgrados a nivel de maestría y orientación de cursos en programas de pregrado y posgrado en las temáticas de recuperación de información, desarrollo de aplicaciones web y gestión de procesos de negocio. 3) apropiación social del conocimiento a través de la publicación de cinco artículos y la presentación de las ponencias en eventos internacionales en temáticas directamente relacionadas con la tesis, además de los contactos realizados para el desarrollo de futuras investigaciones. Finalmente, se cuenta con seis artículos en revistas internacionales, de los cuales a la fecha, tres se encuentran publicados y los otros tres en proceso de evaluación.

Conclusiones

En la búsqueda sobre grandes repositorios de BP es posible contar con técnicas de búsqueda o agrupación que permitan agilizar la gestión de los repositorios. Las técnicas de búsqueda de BP se han desarrollado desde diferentes puntos de vista y varían respecto al tipo de información utilizada para determinar el grado de similitud que se aplicará en la técnica para recuperar los BP. Algunas técnicas combinan grafos, otras, información lingüística (textual) de los BP adicionando ontologías de dominio específico, algunos trabajos utilizan algoritmos genéticos complementados con heurísticas, otros utilizan reglas de asociación sobre eventos ya ejecutados,

entre otros. La propuesta presentada en esta tesis, planteó una estrategia de búsqueda multimodal, capaz de soportar varios tipos de consultas, a saber: textual, estructural, y multimodal (la combinación de las dos anteriores). Además permite tener una representación más amplia del objeto de estudio (proceso de negocio), debido a que almacena información textual y estructural de cada uno de los BP existentes. La evaluación de relevancia de la lista ordenada de resultados demuestra que la combinación de información lingüística (textual) y *codebooks* (estructural) como elementos de búsqueda, permite obtener una mejora sustancial en el nivel de precisión con un menor tiempo de ejecución. La organización automática de los resultados considerados como relevantes en una consulta, a través del algoritmo de agrupación, permite al usuario revisar grupos de BP con algún nivel de similitud en su funcionalidad, en vez de BP por separado, haciendo así que el usuario en sus consultas gaste menos tiempo y revise de forma organizada un mayor número de resultados. Por otra parte, los resultados de la evaluación demostraron que los grupos formados son relevantes y coinciden en alto grado con los grupos formados manualmente por expertos, a través de una estrategia colaborativa.

Palabras Clave: Procesos de negocio, búsqueda multimodal, indexación, *clustering*, etiquetado, evaluación colaborativa.

Structured Abstract

Background

Global economy trends and changing technology environment have leveraged widespread development of business process (BP) techniques both in academic and industrial fields. BP increase flexibility and capacity to manage within organizations. BP allow to describe services offered by organizations as well as internal processes that implement such services. As a result, it is common to find repositories of hundreds and thousands of BP. As BP repositories size increase, it becomes necessary to find tools and techniques for management of such repositories, these tools should search and cluster BP according to search criteria.

Based on the aforementioned, some techniques for BP searching and *clustering* have been proposed. BP Search techniques start from a BP as request and return a BP list arranged in descending order based on the similarity level between the input BP and each one of the items of the list. These approaches for searching are limited to the input/output matching using textual information (activity name or description, events and logic gates) of BP elements. These previous approaches accomplish only partially with the needs of expert developers; the latter is due to the fact that these works do not consider information related with: workflow, behavior, structure, activity type, gate type and event type.

Moreover, there are some approaches based on *clustering* that use hierarchic *clustering* algorithms for building a hierarchy of similar groups based on structural information and behavior of BP. In these approaches, users review the created hierarchy and select the group with higher similarity with their search. Other approaches use sequential *clustering* algorithms which use log files of BP executions as input. In these approaches the algorithm groups BP with similar behavior in the same group. In spite of the contributions made by the previous works, results might be improved if more information features of BP are considered instead of to consider only one type of information such as textual information.

In order to reach higher relevance in results of BP searching systems. This thesis propound a multimodal searching approach that unifies structural and linguistic (textual) information in one single search space in order to increase precision and

covering in performed searches. In addition, the present approach includes an iterative *clustering* algorithm for grouping retrieved results in each request. The algorithm uses a similarity function based on fuzzy logic which measures similarity degree between information types of retrieved BP. Results are grouped so that, during search process, users review in an organized way a larger number or BP and spend less time in.

General Objective

To propound a *clustering* model of Business process based on a multimodal representation using *codebook*s (standard basic units of sequential behavior) and textual features, seeking to obtain more relevant results to business process developers search.

Specific Goals

- To define a *codebook*s dictionary for BPMN specified BP.
- To establish a multimodal representation model which includes the previously defined dictionary as well as the textual features of BP
- To propound an algorithm which groups BP based on the multimodal representation model.
- To measure the quality of results obtained using the model, by means of experimental evaluations that includes measures such as precision, recall and F-measure in closed BP collections

Methods

The methodology is based on the building phases of the documental research model and the model for solutions building. The methodology allowed to handle the complexity of the problem and also allowed to properly assess the features that the *clustering* and multimodal search model should accomplish in each phase. In the initial phase, it was performed a literature review and the state of the art was created that includes the following topics: BP management, methods and algorithms for BP

searching, indexing models and *clustering* algorithms. Later the structure and the algorithm for *codebook* formation was defined, thus fulfilling specific objective one. Phase 2 was focused on establishing an indexation and multimodal search representation that integrates textual and structural information of BP, to do so, the designed *codebook* was used. It was established the tree formal representation of BP and later the weighing function of each element in the multimodal index was defined. Afterwards, it were developed the algorithms for indexing and search, thus achieving specific objective 2. During Phase 3 most referenced *clustering* algorithms in the state of the art were studied and they were adapted for BP *clustering*. Equally, it was proposed and adapted to the search model, a *clustering* algorithm based on graph theory which uses a fuzzy logic function for measuring BP similarity inside each group. Furthermore, a labeling algorithm of groups was developed, thus achieving specific objective 3. Subsequently in phase 4 it was developed the evaluation process of the proposed search and *clustering* model. To do so, it was built a platform that allow to 56 experts in BP management to create a closed repository for testing which contains a set of consultation BP and ideal answers, besides the ideal *clustering* (created manually by evaluators) formed by results of each request. Finally in this phase it was evaluated the relevance of results and the presentation as an ordered list, it was compared with other search models in the state of the art, and also the *clustering* with internal and external measures was evaluated, thus achieving specific objective 4.

Results

The results achieved in the development of this thesis focused on three aspects: 1) generation of new knowledge and technological developments by means of: The definition of a new model of BP grouping based on a multimodal representation y development of MultiSearchBP, a software tool for BP searching and grouping which implements the proposed model. Construction of a Web platform for the collaborative creation of testing repositories of BP (closed repositories for evaluation). Development of a Web platform for collaborative formation of BP groups. The definition of a method for the collaborative creation of testing repositories of BP and creation of repository. 2) Strengthening national scientific community through tutoring 4 undergraduate works, 2 postgraduate thesis and teaching at postgraduate and

undergraduate courses in topics of: information retrieval, Web development and Business Process Management. 3) social appropriation of knowledge through five (5) articles published in international conferences in subjects directly related to the thesis, the respective presentation, in addition to the contacts made for future research. Finally there are six (6) publications in international journals 3 of which are published and three are under evaluation.

Conclusions

Searching in BP large repositories require of *clustering* or searching techniques that speed up repositories management. BP searching techniques have been developed from different angles and vary regarding the type of information used to determine the similarity degree that will be applied in the technique for BP recovering. Some techniques combine graphs, other linguistic (textual) information of BP adding specific domain ontologies, some studies using genetic algorithms complemented with heuristics, others use association rules on already executed events, among others. The proposal presented in this thesis, proposed a strategy of multimodal search, which is able to support various types of queries, namely: textual, structural, and multimodal (the combination of the two). Also allows a wider representation of the object of study (business process) due to the fact that stores textual and structural information of existing BP. The assessment of relevance of the ordered list of results shows that the combination of linguistic information (textual) and *codebook*s (structural) as search features allows to achieve a significant improvement in the precision level with a lower execution time. Automatic organization of results considered relevant to a query, through *clustering* algorithm, allows to the user to review BP groups with some level of similarity in its functionality, instead of BP separately, thus users can review in an organized way a higher number of results and they also spend less time.Moreover, the evaluation results showed that the groups formed are relevant and agree with groups created by experts manually by means of a collaborative approach.

Keywords: Business process, multimodal search, indexation, *clustering*, labeling, collaborative evaluation.

TABLA DE CONTENIDO

Lista de figuras

Lista de tablas

Lista de anexos

Capítulo 1

Introducción

Para tener una visión global de las actividades comerciales que realiza una organización es pertinente que estás sean representadas o modeladas formalmente por procesos de negocio (BP) [1, 2]. Un BP captura un conjunto de procedimientos o actividades interrelacionadas en una estructura organizacional para desarrollar un objetivo de negocio en común, con el fin que la información de estas actividades se convierta en conocimiento explícito y accesible para todos los miembros de la organización [3, 4].

Dado que los BP son un activo esencial, las organizaciones han documentado cada vez más sus procedimientos a través de BP [5, 6]. En consecuencia, el número de actividades que están siendo modeladas dentro de una organización ha aumentado considerablemente, ocasionando que la reutilización de los BP se convierta en una tarea ardua que requiere de tiempo considerable, debido a la cantidad de BP que pueden existir en la organización [7].

Con base en lo anterior, esta tesis se enfrentó al reto de recuperar y agrupar BP, desde grandes colecciones, con el propósito que estos BP estén disponibles de forma organizada o categorizada para su reutilización.

En este capítulo se describe el planteamiento del problema abordado en esta tesis, los objetivos logrados en el desarrollo de la investigación, la metodología de investigación ejecutada en el desarrollo de la tesis, los resultados logrados, los aportes realizados al culminar la tesis, y finalmente como está organizado el resto del documento.

1. Aspectos generales de la investigación

1.1 Planteamiento del problema

En la ola mundial de la tecnología de la información, el entorno empresarial es cada vez más competitivo, requiere que las actividades comerciales se ajusten más rápido a condiciones cambiantes (mayor agilidad de negocio) [8-10], demandando mayor atención en los BP y en la capacidad de adaptarlos rápidamente para responder a condiciones dinámicas [11] . Estos BP especifican el orden de ejecución eventual de las operaciones de una organización teniendo en cuenta datos compartidos, qué socios participan y cómo participan. Actualmente, las organizaciones definen y usan BP para una gran variedad de tareas entre las cuales se encuentran: fabricación de productos, prestación de servicios, adquisición de bienes, manejo de inventarios, entre otras [12, 13]. La utilización de BP hace que las organizaciones centren su atención en el concepto de mejorar la gestión de los recursos operacionales para hacer procesos más maduros, repetibles, tener operaciones de mayor escalabilidad y mejorar su desempeño en general, con el propósito de ofertar nuevos productos y servicios que les permitan ser competitivas en el mercado [14].

Todos estos BP en las organizaciones son normalmente modelados o creados por expertos utilizando herramientas para el diseño de BP donde plasman las operaciones o tareas que necesita ejecutar la organización. Las organizaciones que pretenden diseñar o modelar un nuevo BP tienen que empezar revisando grandes cantidades de información acerca de los BP ya existentes en los repositorios, tarea que demanda tiempo y esfuerzo considerable [15]. Dentro de esta información están las instrucciones del trabajo a realizarse, quién debe realizarlo y la descripción de las conexiones con otros sistemas. Normalmente esta información es almacenada en forma de registros de transacciones conocidos como Log o trazas de ejecución [16-18]. Posteriormente la información revisada sirve como base en el replanteamiento o la remodelación de un nuevo BP que cumpla con los requerimientos actuales de la organización [19, 20].

Debido a la importancia de los BP en las organizaciones, entender y descubrir la semejanza entre los BP existentes en los repositorios, puede ser de utilidad para la identificación de tareas comunes entre BP, las cuales pueden ser reutilizadas en futuras ampliaciones o implementaciones de los BP que comparten dichas tareas [21].

La comprensión de tareas y actividades comunes entre BP, puede ayudar a la toma decisiones acerca de cómo los procesos de la organización se deben fusionar para aumentar la eficacia, identificar oportunidades, normalizar y consolidar los procesos que se ejecutan dentro de la organización [22].

De acuerdo con lo anterior, es necesario contar con un mecanismo de gestión de información eficiente que permita revisar todos los datos contenidos en los BP. De esta forma, nace la minería de BP como alternativa a la búsqueda o descubrimiento de nuevos BP, la cual incorpora técnicas de minería de datos [23, 24], con el propósito de encontrar los BP que representen con mayor similitud el comportamiento de las tareas ejecutadas dentro de una organización [25-27].

Un tema de gran interés en la búsqueda de BP son los datos usados para realizar el proceso de consulta, para lo cual existen propuestas basadas en Log o trazas de ejecución [16, 17], en la estructura [28, 29], y en el comportamiento [30]; todas las anteriores cumplen de alguna manera con los propósitos de búsqueda, encontrando soluciones parciales a las necesidades de los expertos modeladores de BP. Debido que los resultados presentan información mixta que no tiene relación de similitud entre los BP recuperados y el BP de consulta [31, 32].

Con el propósito de disminuir el problema de baja relevancia en la lista ordenada de resultados, se propone el uso de una estrategia de búsqueda multimodal, la cual permite buscar, explorar y descubrir contenido almacenado en varios modos como por ejemplo texto, imágenes y video [33-36]. Este enfoque de consulta tiene las siguientes ventajas: mayor efectividad en las búsquedas, mayor agilidad, y más flexibilidad para realizar varias clases de consultas [37-39], con el propósito de tener una representación más rica del objeto de estudio (proceso de negocio en este caso) y con ello entregar una respuesta más precisa a las necesidades del usuario.

Es así como en la presente tesis, se planteó adicionar a la información textual de los BP, el uso de libros de códigos (*codebooks*) para generar una estructura de unidades básicas estándar de comportamiento secuencial. Estos *codebooks* se pueden construir con base en las propiedades de similitud de patrones secuenciales frecuentes en el contenido de la información a consultar. Generalmente los *codebooks* han sido empleados en el dominio de recuperación de imágenes utilizados como histogramas de patrones visuales [37], y como vocabularios o diccionarios visuales [40]. Además, son utilizados para analizar y buscar ocurrencias

de palabras en transcripciones de texto [41] y también para buscar objetos que se mueven en escenarios complejos en grabaciones de video [42].

Por otra parte, para solucionar el problema de la lista mixta de resultados, esta tesis propuso la utilización de técnicas de agrupamiento por afinidad o *clustering* [43, 44]. En este enfoque se encuentran los algoritmos de *clustering* jerárquico que construyen una jerarquía de grupos similares con base en las características estructurales y de comportamiento de los BP [45, 46]. En estas propuestas los usuarios revisan la jerarquía y seleccionan el grupo de mayor similitud a su búsqueda [47, 48]. Otro tipo de *clustering* de BP es el secuencial, el cual toma como datos de entrada archivos Log de ejecución de los BP. Estos algoritmos agrupan los BP con el mismo tipo de comportamiento a cada grupo [49]. A pesar de los aportes ya realizados por estos enfoques, los resultados pueden ser ampliados y mejorados abarcando más características de información presente en los BP y no centrándose solamente en un solo tipo.

Con el fin de alcanzar mayor relevancia de los resultados reportados en un sistema de búsqueda de BP, en esta tesis se planteó un modelo de agrupación de BP que unifica en un solo espacio de búsqueda unidades estructurales de comportamiento y características textuales existentes en los BP, en lo que se conoce como una representación de indexación y búsqueda multimodal. Adicionalmente, se integró un algoritmo de *clustering* basado en lógica difusa, que utiliza información textual, estructural y de comportamiento para agrupar los resultados de cada consulta. La agrupación es realizada con base en la similitud de los tipos de información contenida en cada uno de los BP recuperados. La agrupación busca garantizar una presentación visual clara y categorizada de los resultados obtenidos en cada consulta [50], logrando así, que el usuario en sus consultas invierta menos tiempo y revise de forma organizada un mayor número de resultados [47]. Esta presentación le permite al usuario la revisión de grupos de BP en lugar de BP individuales y por separado.

Con base en lo anterior, esta tesis se orientó en responder la siguiente pregunta de investigación: ¿Cómo formar grupos o familias de procesos de negocio con base en un modelo de representación de BP multimodal que permita a los expertos desarrolladores obtener resultados más relevantes?

Para responder a la anterior pregunta de investigación, se definió la siguiente hipótesis. Formar grupos o familias de procesos de negocio con base en un modelo de representación de BP multimodal, permite a los expertos desarrolladores obtener resultados de mayor relevancia. Para la comprobación de la hipótesis se plantearon los siguientes objetivos.

1.2 Objetivos

1.2.1 Objetivo general

Proponer un modelo de Agrupación de Procesos de Negocio basado en una representación multimodal que usa *codebook*s (unidades básicas estándar de comportamiento secuencial) y características textuales, buscando obtener resultados más relevantes a las búsquedas realizadas por desarrolladores de BP.

1.2.2 Objetivos específicos

- Definir un diccionario de *codebook*s (unidades básicas estándar de comportamiento secuencial) de BP modelados con la notación BPMN.
- Establecer un modelo de representación multimodal, que incorpore el diccionario previamente definido y las características textuales de los BP.
- Proponer un algoritmo que agrupe BP, basado en el modelo de representación multimodal.
- Medir la calidad de los resultados entregados por el modelo, a través de evaluaciones experimentales con medidas tales como precisión, recuerdo y media F en colecciones cerradas de BP y en experimentos con usuarios.

1.3 Metodología de investigación abordada para el desarrollo de la tesis

La metodología utilizada para el desarrollo de la presente investigación se basó en una adaptación del “Modelo Integral para un Profesional en Ingeniería” [51]. De acuerdo con este modelo, fueron identificados dos grandes componentes: el Modelo de investigación documental en la generación de la base conceptual, con el propósito de realizar aportes científicos partiendo de un hecho abstracto y el Modelo para construcción de soluciones en el proceso de desarrollo del prototipo experimental

que soporta el modelo de búsqueda definido en esta investigación. A continuación son descritos los componentes, las fases ejecutadas en cada uno de estos, y los resultados obtenidos en cada fase.

1.3.1 El Modelo de Investigación Documental

- **Fase Preparatoria:** en esta fase fueron identificados los conceptos más importantes alrededor de la temática de minería de procesos, a saber: metodologías, elementos, algoritmos, medidas de similitud, enfocados esencialmente en el proceso de búsqueda de BP y agrupación o (*clustering*) de BP.
- **Fase Descriptiva:** en esta fase fue realizada una revisión del estado actual del conocimiento acerca de las diferentes técnicas de búsqueda y agrupación de BP en cada uno de los aspectos identificados en la fase anterior.
- **Fase de construcción teórica global:** en esta fase se realizó un balance del conjunto de resultados del estudio adelantado, además de la identificación de vacíos, limitaciones, dificultades, tendencias y logros esperados durante el desarrollo del proyecto, fueron analizadas las propuestas que pueden ser desarrolladas como trabajos de pregrado o maestría en relación con la temática del proyecto.
- **Fase de extensión y publicación:** en esta fase se realizó la divulgación de los resultados obtenidos en el desarrollo del proyecto mediante la presentación de artículos en eventos y revistas nacionales e internacionales, los cuales son presentados en el siguiente apartado.

1.3.2 El Modelo para Construcción de Soluciones

- **Estudio de Pre-factibilidad:** fase en la cual fue analizado el dominio de los problemas identificados en la minería de procesos enfocados en la parte de la búsqueda de BP, se determinó la viabilidad y alcances del proyecto, que permitieron el desarrollo de los dos primeros objetivos específicos
- **Formulación del Proyecto:** en esta fase fueron estudiados los aspectos principales relacionados con el desarrollo del proyecto y la construcción del prototipo.
- **Ejecución del Proyecto.** La ejecución del proyecto materializó los aspectos descritos en el modelo para la construcción de soluciones, en el proceso de

ejecución fueron desglosadas las tareas a desarrollar en el proyecto, de manera que se pueda programar su ejecución. Las tareas ejecutadas en el proyecto fueron:

- **Validación de la solución:** esta fase dio cumplimiento al cuarto objetivo, por medio de un experimento formal apoyado en métodos empíricos de experimentación en recuperación de información [52] . Para complementar y evaluar el modelo, se implementó una herramienta de validación a partir del prototipado evolutivo que incluye los algoritmos de pre-procesamiento, indexación, búsqueda multimodal y agrupamiento. Finalmente, la validez de la propuesta fue comprobada a través de la evaluación experimental de la herramienta a partir de un banco de pruebas previamente definido.
- **El proceso de evaluación**

- Evaluación de la lista de resultados*:* evaluación de relevancia de los resultados con una lista ordenada. Esta evaluación se llevó a cabo con las medidas de Precisión gradada, Recuerdo gradado, Medida F, ANDCG y GenAveP'.
- *Evaluación de la formación del codebook*, con el fin de fijar cuál es el valor estructural (N-componente), del *codebook* que permite tener mejores resultados de acuerdo con las medidas de relevancia, para con ese valor formar el modelo multimodal. Las evaluaciones fueron realizadas por intervalos de resultados teniendo presente listas de resultados de 8, 10, 15 y 20 BP retornados por el modelo de búsqueda multimodal.
- *Evaluación de relevancia de los resultados del modelo multimodal en cada una de las opciones de búsqueda que este ofrece*. Para esto se tuvo en cuenta el mejor componente *codebook* del modelo encontrado en la etapa anterior. Las opciones de consulta evaluadas fueron: búsqueda lingüística, búsqueda por *codebook*, y búsqueda multimodal (la cual corresponde a la combinación de las dos anteriores). Además el modelo multimodal propuesto fue comparado con dos métodos similares de recuperación de BP existentes en el estado del arte.
- *Evaluación de rendimiento*: se realizó un análisis de rendimiento basado en el tiempo de respuesta dependiendo del número de componentes

(nodos) presentes en los BP en cada una de las consultas realizadas al repositorio.

- Evaluación de la agrupación: la evaluación de la agrupación contempló dos fases: evaluación interna y externa.
  - ✓ *Evaluación interna**:*** para esto fueron utilizadas medidas de evaluación que permite realizar análisis de la agrupación sin intervención humana. Las medidas utilizadas fueron: Cohesión, Acoplamiento, Silueta, Suma de cuadrados entre grupos (Sum of squares between clusters), Suma de cuadrados dentro del grupo (Sum-of-squares within cluster) y el Índice Davies-Bouldin (DB).
  - ✓ *Evaluación externa:* se creó una colección cerrada de BP con agrupaciones "ideales" para un conjunto específico de consultas, la cual fue creada colaborativamente por 56 usuarios expertos en gestión y modelado de BP. Los resultados del modelo propuesto en la tesis se compararon contra las respuestas ideales y las respuestas entregadas por otros algoritmos identificados en el estado del arte.

La colección y la plataforma para la creación de la colección cerrada de pruebas de BP fueron presentadas por Oroñez H y otros en [53]. Para la comparación de los resultados de agrupación en esta fase se utilizaron las medidas Precisión, Recuerdo y Medida-F.

1.4 Estructura del documento

Además del presente capítulo, donde se presentan los aspectos generales de la investigación, el contenido de la presente tesis, está distribuido en otros cinco capítulos descritos a continuación.

Capítulo 2: en este se definen los conceptos más importantes a tener en cuenta en esta tesis doctoral, presenta una revisión del estado actual del conocimiento y una clasificación de los trabajos relacionados en la temática de la tesis.

Capítulo 3: presenta la formación del *codebook* y el modelo de búsqueda multimodal. Para el *codebook* se describe el algoritmo de pre-procesamiento de los BP, el algoritmo de formación del *codebook* y la equivalencia de cada uno de los elementos del *codebook* en un BP. Para el modelo multimodal se describe el algoritmo de indexación, el algoritmo de búsqueda, los tipos de consultas y

adicionalmente se presenta un ejemplo de búsqueda que integra los elementos mencionados anteriormente.

Capítulo 4: describe el algoritmo de agrupación base, la adaptación de dicho algoritmo y su mejora, adicionalmente, incluye un ejemplo del funcionamiento del algoritmo de agrupación propuesto.

Capítulo 5: presenta la herramienta prototipo que soporta el modelo de búsqueda y agrupación definido previamente, la plataforma de evaluación para la creación de repositorio cerrado de BP, la plataforma para la formación manual de grupos y la experimentación con los resultados obtenidos en cada una de las fases de evaluación.

Capítulo 6: muestra las conclusiones a las que se llegó con el desarrollo de esta tesis, teniendo en cuenta los objetivos planteados al inicio del proyecto, así como también los trabajos futuros que el investigador espera desarrollar.

1.5 Resultados y aportes

El siguiente apartado describe los resultados obtenidos en esta investigación con relación a la generación de nuevos conocimientos y desarrollos tecnológicos (Tabla 1), fortalecimiento de la comunidad científica nacional (Tabla 2), y apropiación social del conocimiento involucrado en el desarrollo de la investigación (Tabla 3). En cada uno de los tipos de resultados obtenidos se definen los resultados, el indicador o los beneficiarios de los mismos.

Tabla 1. Generación de nuevos conocimientos y desarrollos tecnológicos

Resultados	Indicador
Un nuevo modelo de agrupación de procesos de negocio basado en una representación multimodal	Documento de tesis doctoral y trabajos en eventos internacionales y revistas.
MultiSearchBP - Una herramienta software para la búsqueda y agrupación de procesos de negocio, la cual implementa el modelo propuesto	Código fuente de la aplicación, algoritmo de indexación y búsqueda multimodal, algoritmo de *clustering* y método de etiquetado.
Una plataforma Web para la creación colaborativa de repositorios de prueba	Código fuente de la aplicación

de procesos de negocio.	
Una plataforma web para la formación colaborativa de grupos de procesos de negocio	Código fuente de la aplicación
Un método para la creación de repositorios de prueba de procesos de negocio	Método documentado y explicado paso a paso en artículo
Un repositorio de pruebas cerrado de procesos de negocio	Un repositorio con 100 procesos de negocio. El cual cuenta con 6 BP como consultas y las repuestas ideales a cada una de estas consultas. Además los grupos formados manualmente por los 56 evaluadores, con base en los resultados considerados como relevantes en cada consulta. El repositorio pude ser utilizado en futuras investigaciones de búsqueda o agrupación de procesos de negocio

Fuente: elaboración propia

Tabla 2. Fortalecimiento de la comunidad científica.

Resultados	Beneficiarios
Formación de recursos humanos a nivel profesional	Dirección de tres proyectos de trabajo de grado en Ingeniería de sistemas, total de estudiantes participantes (3). Universidad Mariana, San Juan de Pasto
Formación de recursos humanos a nivel de posgrado	Dirección de dos tesis de maestría, en Ingeniería de Software, Universidad de San Buenaventura Cali, con la participación de 2 estudiantes cada una. Los temas en la maestría están enfocados en el modelamiento de procesos de negocio.
Formación de los estudiantes de pregrado y postgrado en la recuperación de información, desarrollo de aplicaciones web y gestión de procesos de negocio	Se orientaron los cursos de Gestión de procesos de negocio, Recuperación de información en el programa de Ingeniera de sistemas de la Universidad Mariana. Además, el autor de esta tesis se desempeñó como Profesor acompañante en la materia de desarrollo de aplicaciones web en el programa de Maestría en Ingeniería telemática de la

	Universidad del Cauca y como Profesor de la materia de Conceptos Avanzados de bases de datos en el programa de especialización en Construcción de software de la Universidad de San Buenaventura Cali.

Fuente: elaboración propia

Tabla 3. Apropiación social del conocimiento involucrado en el desarrollo de la investigación.

Resultados	Indicador
Cinco artículos en eventos internacionales en temáticas directamente relacionadas con la tesis	*1.* H. Ordoñez, M. Obando. W. Mora. "***Modelo para el descubrimiento de procesos de negocio basado en trazas de ejecución generadas con bizagi***". IV Congreso Internacional sobre Aplicación de Tecnologías de la Información y Comunicaciones Avanzadas. Atica 24-26 Octubre 2012, Loja – Ecuador. Artículo publicado como capítulo en el libro: http://www.esvial.org/atica2012/ ISBN: 978-9942-04-296-5, Editorial Universidad Técnica Particular de Loja Loja (UTPL) (Anexo a) 2. H. Ordoñez, C. Figueroa, JC. Corrales, M. Moricio, C. Cobos, L. Krug. "***Business Process Indexing based on Similarity of Execution Cases***". Proceeding EATIS '14 Proceedings of the 7th Euro American Conference on Telematics and Information Systems, Valparaiso, Chile 2 - 4 Abril 2014, Article No. 12, doi 10.1145/2590651.2590664 (Anexo b) 3. H. Ordoñez, J C. Corrales, C. Cobos, L. Krug. "***Collaborative Grouping of Business Process Models***". Proceeding EATIS '14 Proceedings of the 7th Euro American Conference on Telematics and Information Systems. Valparaíso, Chile, 2 - 4 Abril 2014, Article No. 35, doi 10.1145/2590651.2590686 (Anexo c)

	4. H. Ordoñez, J C. Corrales, C. Cobos, L. Krug. L. Thom. "***Collaborative Evaluation to Build Closed Repositories On Business Process Models***". Proceedings of the 16th International Conference on Enterprise Information Systems (ICEIS). 25 – 29 Abril 2014, Lisboa-Portugal, Doi 10.5220/0004881203110318 (Anexo d) *5.* Hugo Ordoñez, Andrés Felipe Escobar, Diana Lorena Velandia, Carlos Cobos, Armando Ordoñez, Juan Carlos Corrales "***Eliciting Requirements in Extreme Programming (XP) Through Business Process Models)***", Proceeding, CONISOFT 2015, Congreso Internacional de Investigación e Innovación en Ingeniería de Software , 27 – 29 Abril 2015, San Luis de Potosí, México (Anexo e)
Tres artículos publicados en revistas internacionales	1. H. Ordoñez, J C. Corales, C. Cobos. "***MultiSearchBP - Entorno Para Búsqueda Y Agrupación De Modelos de Procesos De Negocio***". Polibits journal. Center for Technological Design and Development in Computer Science (CIDETEC) of the National Polytechnic Institute (IPN). Issue 49 (2014, January-June) paginas 29 - 37. Indexada por IB-SciELO, Categoría A1 Publindex-Colciencias. (Anexo f) 2. H. Ordoñez, JC. Corrales, C. Cobos. " ***Business Processes Retrieval based on Multimodal Search and Lingo Clustering Algorithm***". Journal IEEE Latin America Transactions. Indexada por: BBCS-INSPEC, IBGC- JCR, IB-SCI, SCOPUS, SJR. Categoría A2 Publindex-Colciencias. Volumen: 13, Issue: 3, March 2015 (Anexo g) 3. Jose Armando Ordoñez, Hugo Ordoñez, Cristian Figueroa, Juan Carlos Corrales, Carlos Cobos ***"Dynamic reconfiguration of composite convergent services supported by multimodal search***" Lecture Notes in Business Information Processing series by Springer Verlag. . Indexada por:

	SJR. Categoría A2 Publindex-Colciencias. Volumen: 184, 2015 (Anexo h)
Tres artículos en evaluación en revistas internacionales	*1.* H. Ordoñez, C. Figueroa, Juan Carlos Corrales. Corrales, Carlos Cobos, Enrique Herrera-Viedma "***Multimodal Model for Business Process Search***". Information Science Journal. ELSEVIER. Indexada por BBCS-INSPEC, IBGC- JCR, IB-SCI, SCOPUS, CURRENT CONTENTS, ENVIRONMENTAL SCIENCE AND POLLUTION MANAGEMENT, SJR. Categoría A1 Publindex-Colciencias. **(En evaluación)** (Anexo i) *2.* Cristian Figueroa, Hugo Ordoñez, Juan Carlos Corrales, Carlos Cobos, Leandro Krug Wives, Enrique Herrera-Viedma "***Improving Business Process Retrieval Using Categorization and Multimodal Search***". Knowledge - Based Systems Journal. ELSEVIER. Indexada por: BBCS-INSPEC, IBGC-JCR, IB-PsycINFO, IB-SCI, SCOPUS, CURRENT CONTENTS, ENVIRONMENTAL SCIENCE AND POLLUTION MANAGEMENT, PSYCFIRST, PSYCLIT, SJR. Categoría A1 Publindex-Colciencias. **(En evaluación)** (Anexo j) 3. H. Ordoñez, JC. Corrales, C. Cobos, L. Krug. L. Thom "***GROUPING OF BUSINESS PROCESSES MODELS BASED ON INCREMENTAL CLUSTERING ALGORITHM USING FUZZY SIMILARITY AND MULTIMODAL SEARCH***". Journal Knowledge and Information Systems. Springer. Indexada por: BBCS-INSPEC, IB-SCI, SCOPUS, CURRENT CONTENTS, SJR. Categoría A1 Publindex-Colciencias. **(En evaluación)** (Anexo 11)

Fuente: elaboración propia

Capítulo 2

Estado del arte

Este capítulo define los conceptos más representativos que fueron tenidos en cuenta en la presente investigación, entre otros: Procesos de negocio, Búsqueda multimodal, *Clustering* y Lógica difusa. Adicionalmente, presenta una breve descripción y clasificación de los trabajos más destacados relacionados con la temática de esta tesis.

2.1 Introducción

La revisión documental permitió realizar el estudio del conocimiento acumulado en los artículos que se analizaron dentro del área específica de la gestión y modelado de procesos de negocio, búsqueda multimodal, *clustering*, evaluación de recuperación de información y minería de procesos haciendo énfasis en la búsqueda de BP. El análisis realizado sirvió como herramienta para la contextualización del tema de estudio, clasificación de las temáticas e información relacionada y categorización de los trabajos revisados para identificar las brechas existentes. Además, permitió compilar y sistematizar la información recolectada en el desarrollo de la investigación. Por otra parte, los conceptos adquiridos en la revisión documental ofrecieron diferentes posibilidades de comprensión del problema a tratar en la investigación.

2.2 Conceptos teóricos

- **Proceso de negocio (BP):** representa procedimientos o actividades que colectivamente alcanzan un objetivo de negocio en común, definiendo roles y relaciones funcionales. Es decir, organizan la información de las instrucciones del trabajo a realizarse, quién debe realizarlo y describen las conexiones con otros sistemas. Además, define el paso a paso de las tareas a realizar por cada “actor” dentro de una empresa u organización, para garantizar el resultado que se espera lograr. Uno de los propósitos principales de un proceso de negocio es servir como un medio de comunicación que facilita la comprensión de un proceso complejo

entre las diversas partes interesadas o involucradas [54, 55]. Los procesos de negocio son modelados con el fin de determinar los puntos de vista de los usuarios, para comunicar nuevas ideas y desarrollar entendimiento compartido entre las dependencias de una organización [56, 57] .

- **Búsqueda multimodal:** es un tipo de búsqueda que utiliza varias tipos de información como fuentes de entrada para ejecutar las consultas. La combinación de varios tipos de información en la consulta, amplía el espacio de búsqueda en la información a recuperar, haciendo así que el nivel de relevancia en los resultados aumente, y en consecuencia, la búsqueda sea más precisa [37, 38] . En algunos casos como la multimedia, la búsqueda multimodal involucra tipos de la información relacionados con texto, imagen, audio y video, para mejorar la construcción de los índices de búsqueda, los tiempos de acceso a la información, la extracción y ordenamiento de las respuestas, así como también aporta varias clases de consulta dependiendo de la combinación entre los tipos de información utilizada para la consulta [35, 58].
- ***Clustering* o agrupación:** es una técnica de búsqueda de patrones ocultos que puedan existir en algún tipo de información. Se trata de un proceso para agrupar datos inconexos en grupos, de modo que los datos en cada grupo puedan ser similares, sin embargo, diferente a los de otros grupos. Las técnicas de agrupación se utilizan en muchas áreas de aplicación, tales como el análisis de datos, reconocimiento de patrones, procesamiento de imágenes, y la recuperación de información [59, 60]. Para definir la similitud entre objetos, el *clustering* se basa en criterios de distancia o similitud. La cercanía entre objetos se define en términos de una función de distancia, por ejemplo la distancia euclidiana, distancia Manhattan, distancia Mahalanobis, similitud coseno, entre otras [61].
- **Lógica difusa:** es una lógica alternativa a la lógica convencional que pretende introducir un grado de generalidad en las cosas que evalúa. La lógica difusa en comparación con la lógica convencional permite trabajar con información que no es exacta para definir evaluaciones convencionales, contrario con la lógica tradicional que sólo permite trabajar con información definida y precisa [62]. La lógica difusa se puede aplicar en procesos complejos, cuando no existe un modelo de solución simple o un modelo matemático preciso. Es útil también cuando es requerido usar el conocimiento de un experto que utiliza conceptos ambiguos o imprecisos. De la misma manera se puede aplicar cuando ciertas

partes de un sistema a controlar son desconocidas y no pueden medirse de forma precisa y cuando el ajuste de una variable puede producir el desajuste de otras. La lógica difusa ha sido aplicada en muchos campos del conocimiento, tales como sistemas de control, predicción de terremotos, reconocimiento de patrones, visión por computador y también en agrupación o *clustering* de documentos [63, 64].

2.3 Estado actual del conocimiento

La gestión eficaz de los BP como activos de conocimiento requiere poderosos medios (mecanismos de búsqueda) para identificar BP de manera eficiente dentro del repositorio [65]. En el proceso de búsqueda de este tipo de modelos, es necesario definir una representación formal (por ejemplo grafos o máquinas de estado), para la identificación de un conjunto de características estructurales o de comportamiento similares entre una serie de BP a recuperar. Las similitudes detectadas pueden indicar que existen varias versiones de BP similares o incluso duplicados [66], los cuales pueden servir para: unificarse en un solo BP, ampliar un BP con nuevos requerimientos o funcionalidades de la organización, reprogramar o reconfigurar un BP, con el fin de mejorar la agilidad del negocio o propagar los cambios en todos los niveles de la organización.

Las consultas o búsquedas de BP requieren definir una representación que involucre la mayoría de la información existente en estos. Para esto Dijkman y otros [18] han definido las siguientes propiedades:

Definición 1: un BP es un registro (tupla) *BP= {V, E, L, A, C, Peso, Rd, R, A}* en el que:

- *V*, es un conjunto de nodos que pueden ser de tipo: $V \longrightarrow \{Tarea, compuerta, estado\}$
- Todo el conjunto de tareas está representado como *T ⊆ V*, las tareas pueden ser de tipo,
  $T \longrightarrow \{atomica, compuesta\ (\ subproceso\ referenciando\ a\ otro\ modelo\ de\ BP)\}$
- El conjunto de compuertas es representado como *G ⊆ V*, y pueden ser de tipo: $G \longrightarrow \{$*and − split, and –join, xor − split, xor – join}*,* No todos los BP, tienen nodos compuerta.
- Los nodos estado está representado por *S ⊆ V*, que muestra los estados del BP.

- $E \subseteq V \times V$, es un conjunto de relaciones del flujo de ejecución, el cual muestra el orden de ejecución de las diferentes tareas.
- L es el conjunto de etiquetas.
- $l: T \rightarrow L \cup \{\varepsilon\}$ es una función de etiqueta, donde ε significa etiqueta muda, una tarea con la etiqueta como ε es una tarea invisible porque no puede ser observado desde el registro.
- D es un conjunto de datos que se procesa durante la ejecución del BP.
- $Wt: D \rightarrow 2^T$, describe que datos son escritos en cada tarea.
- $Rd: D \rightarrow 2^T$ describe que datos son leídos en cada tarea, R es un conjunto de recursos que pueden ser: una persona, un rol, o un grupo de personas.
- $A: T \rightarrow 2^R$ es una función de asignación, la cual describe que se le asigna a una tarea en el momento de ejecución.
- Perspectiva de flujo de control: {*V, E*} describe las tareas que deben llevarse a cabo y en qué orden.
- Perspectiva de datos: {*T, D, Peso, Rd*} describe los datos que se procesan y cómo son procesados.
- Perspectiva de los recursos: {*T, R, A*} describe quién es el responsable y para qué tareas.

Normalmente el flujo de control de los BP es representado gráficamente, en dicha representación está la lógica temporal lineal de ejecución [67]. Para la representación se pueden utilizar muchas notaciones que permiten capturar el flujo de control de los BP, por ejemplo, redes de Petri, BPMN (*Business Process Modeling Notation*), EPC (cadena de procesos impulsados por eventos), UML AD (*Unified Modeling Language activity diagram*) y BPEL (*Business Process Execution Language*). Cada notación tiene sus propias ventajas y desventajas [56].

Definición 2: propiedades de un BP

- **Lingüística:** es el conjunto de todas las etiquetas de las tareas de los BP, dado que describen la funcionalidad del BP. Ejemplo en la Figura 1, el conjunto de texto que compone las etiquetas de las actividades sería: Ct ={A,B,C,D}
- **Metadatos:** pueden ser cualquier tipo de información, como etiquetas, la información del propietario, nombre del proceso, roles, etc. Por ejemplo, quién

crea el modelo y cuándo; cuál es el objetivo del modelo. Toda la información de metadatos se puede describir utilizando un par clave - valor. Por ejemplo en la Figura 1, un metadato podría ser el nombre del BP,"Estructura secuencial".

- **Estructura Gráfica:** es la estructura topológica de los BP, incluye nodos y los arcos, que representan el flujo de control.
- **Semántica de comportamiento:** describe las tareas que están involucradas y su orden de ejecución. Existen muchas formas para describir el comportamiento de un modelo de BP, tales como la Relación causal, donde se describe que la ejecución de una tarea depende de la ejecución previa de la tarea anterior, es decir, para ejecutar B, primero tiene que haberse ejecutado A (ver Figura 1); la relación de concurrencia, la cual describe que una tarea puede ser ejecutada en paralelo con otra tarea, por ejemplo en la Figura 2, una vez ejecutada la tarea A, las tareas B y C, pueden ejecutarse en paralelo; la relación de conflictos donde se describe que una tarea no se puede ejecutar en la misma instancia con otra tarea, por ejemplo en la Figura 3, la ejecución de las actividades B o C depende de la condición establecida en la compuerta lógica. Es decir, para ejecutar la tarea D, debe haberse ejecutado algunas de las dos tareas anteriores B o C, pero no las dos.
- **Semántica de operación**: describe el paso a paso, y en detalle la operación de los BP utilizando ontologías.

Figura 1. Estructura secuencial.

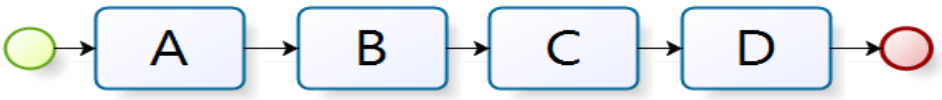


Figura 2. Estructura paralela

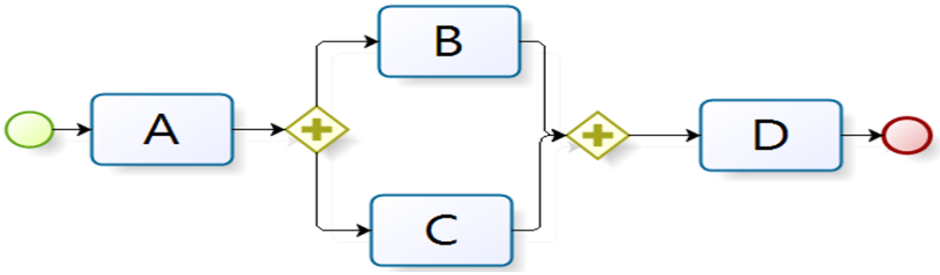
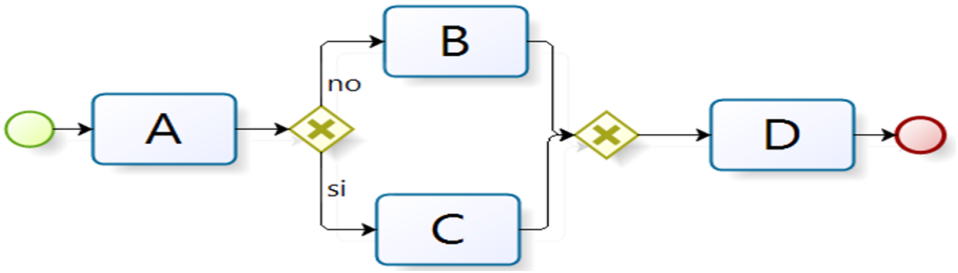


Figura 3. Estructura de selección.



Características de los BP: a pesar de que la representación del flujo de control de los BP en forma de grafos ha sido muy utilizada en sistemas de búsqueda, los BP tienen sus propias características específicas que deben ser consideradas durante la consulta, según [87]. Estas se describen a continuación.

- **Tamaño relativamente pequeño:** esta característica se debe a que el tamaño de los BP no es tan grande. Esto permite mantener una dimensión manejable de las operaciones llevadas a cabo en la organización. En consecuencia, todo el proceso será mucho más fácil de ejecutar con eficiencia y eficacia.
- **Gran número de etiquetas y nombres de tareas incompatibles:** no todos los BP están completamente etiquetados. Las etiquetas de los elementos pueden ser arbitrariamente largas. Algunas tareas idénticas pueden tener diferentes etiquetas. Pueden existir dos tareas diferentes con etiquetas similares.
- **Subgrafos (Sub-estructuras) frecuentes:** en general, no son muy comunes patrones de relaciones entre las diferentes tareas de la empresa.
- **La anidación:** los BP se pueden anidar o componer por medio de colaboración entre roles o dependencias de una o varias organizaciones.
- **Semántica específica:** los BP tienen semántica específica de comportamiento, la cual describe cómo un BP puede ser ejecutado en orden parcial.

El conjunto de características descrito, permite a los usuarios consultar repositorios de BP, basándose en información estructural con el propósito de obtener un conjunto de BP que satisfaga sus necesidades. Estas necesidades pueden ser expresadas a través de consultas del tipo:

- Exacta basada en la estructura gráfica
- Similitud basada en la estructura gráfica
- Exacta basada en la semántica de comportamiento
- Exacta basada en la semántica de operación

A pesar de que las definiciones, propiedades y características descritas anteriormente aportan expresividad y variedad a las consultas ejecutadas a través de los mecanismos de búsqueda de BP, aún los investigadores no han definido cuál de estos tipos de información es el que permite obtener mejores resultados en las búsquedas. Por este motivo, los trabajos más importantes de la literatura, utilizan alguna o algunas de estas propiedades o características, para definir qué tipos de información, podría utilizar un mecanismo de búsqueda de BP.

2.4 Trabajos relacionados

La presente tesis se enfocó en la búsqueda y agrupación (*clustering*) de BP, por lo tanto a continuación se presenta un resumen de los trabajos más destacados en éstas dos temáticas.

2.4.1 Búsqueda BP basada en Lingüística

Reijers y otros [68] proponen un método de compresión lingüística basado en redes de Petri, donde resaltan dos contribuciones realizadas, a saber: 1) un argumento teórico para establecer el grado de compresión de la lingüística, abordando la semiología (estudio de signos) de los gráficos, en donde identifican ocho variables visuales distintas que pueden ser utilizadas para codificar la información de la gráfica del BP, en esta información el color es tomado como una de las variables más eficaces para distinguir los elementos de la notación. 2) la formalización de conceptos en el modelado de *workflows*, para lo cual el método de compresión toma el BP como un grafo dirigido bipartito donde P es un conjunto de nodos llamados lugares, T un conjunto de nodos llamados transiciones y Fp $(P \times T) \cup (T \times P)$ una relación de flujo binario basado en un operador alfa que mapea cada conjunto de nodos T. Para realizar la búsqueda del nuevo modelo, el método de comprensión ejecuta un algoritmo denominado (max-flow-min-cut) que realiza emparejamiento de nodos para encontrar el flujo máximo de coincidencias de los operadores de conexión.

Koschmider, Hornung y Oberweis [69] plantean un sistema de búsqueda de BP que realiza recomendaciones semánticas para extender consultas. Cuenta con un editor de modelos de BP basado en redes de Petri e incorpora un repositorio en el cual todos los BP son etiquetados con un metadato. El método de búsqueda implementado etiqueta actividades basándose en la medida frecuencia de términos

(TF) por BP y la descripción de estados, en la cual se aplican técnicas de recuperación de información. Entre estas técnicas están la creación de índice, la eliminación de palabras vacías y la ponderación de términos presentes en el modelo BP. El sistema cuenta con dos opciones de búsqueda, una básica y otra extendida. La búsqueda básica consulta sobre todos los modelos presentes en el repositorio o sobre un modelo en especial, incorpora la ontología *WordNet* como elemento de generación de sugerencias semánticas en las búsquedas. Por otra parte, la búsqueda extendida considera a cada actividad del BP como un vector de términos agregando una función de costo parcial (FP), con la cual se calcula una función de costo total (FT). El ordenamiento de los resultados de la consulta es realizado con las ft más bajas o de menor peso.

Búsqueda de BP aplicando reglas de asociación. Huang y otros [70] presentan un método de búsqueda basado en descomposición de BP creando un análisis híbrido entre estructura y relevancia. El algoritmo está basado en un análisis iterativo del grafo que representa al BP. La descomposición crea fragmentos de procesos reutilizables (RPF), los cuales cumplen las siguientes características: 1) Un RPF debe ser conectado de manera que todos los nodos puedan llegar desde una entrada de borde o arista, y 2) Cada RPF debe tener sólo una arista de entrada o de salida interconectados con otro fragmento. En este proceso se tiene como meta de búsqueda extraer la frecuencia de ocurrencia más alta en las tareas de los BP representados por los fragmentos generados.

Pérez-castillo, Piattini y Weber [71] presentan un método de comparación de minería de BP dinámica y estática. La solución estática considera el código fuente del artefacto de software y la solución estática, realiza un análisis basado en ingeniería inversa para extraer el conocimiento de los BP. El análisis de la solución estática consiste en la inspección sucesiva de los archivos de código fuente en donde se construye un árbol de sintaxis abstracta de ese código. En el modelo se detectan estructuras específicas o patrones, los cuales indican los elementos que deben ser construidos y cómo estos están interrelacionados en el BP. Este conjunto de patrones se dividen en tres categorías: patrones estructurales, patrones de datos y patrones de eventos. En el mismo sentido, la solución dinámica realiza un pre-análisis donde inyecta declaraciones específicas en ciertas partes del código fuente, en las cuales se marcan en un registro los eventos ejecutados, de esta forma cada evento registrado en el Log especifica la ejecución de una tarea del BP. Los BP

retornados en la búsqueda dependen del cumplimento de reglas de asociación aplicadas sobre las declaraciones específicas del código fuente.

Rosso-Pelayo y otros [72] proponen un método de búsqueda de BP mediante la aplicación de reglas de asociación para información no estructurada. El proceso es llevado a cabo utilizando datos no estructurados en lugar de los registros de las aplicaciones. La ejecución del algoritmo de detección de reglas está dividida en dos: 1) la obtención de la asociación entre documentos y procesos, 2) la construcción de un modelo de lenguaje estadístico para identificación de normas relacionadas con el proceso y las actividades que se presentan en los documentos. La construcción del modelo está dividida en dos actividades principales: el algoritmo analizador, que detecta frases relacionadas con las actividades del proceso por medio de una ontología de dominio, y la identificación de patrones que utiliza una heurística basada en los elementos de la ontología de dominio y las sentencias del documento de búsqueda. En la recuperación de los BP se utiliza la detección de patrones, el cálculo de su frecuencia y las asociaciones de las actividades.

Búsqueda de BP aplicando algoritmos genéticos. Turner y Mehnen [73] proponen un algoritmo genético que identifica y extrae patrones de los datos presentes en los BP. El algoritmo consta de dos etapas: 1) Inicia leyendo el registro de eventos (Log), y 2) Realiza un cálculo de relaciones de dependencia entre las actividades exploradas en el BP (sobre la base de un conjunto de heurísticas). La heurística busca determinar la precedencia y orden de las tareas del BP utilizando una medida de dependencia. La medida tiene por objeto determinar el peso de la relación entre tareas, mediante el cálculo de la cantidad de veces que una tarea es directamente precedida por otra. El análisis es llevado a cabo de manera dirigida, en donde todas las tareas de rastreo en el orden de ejecución poseen inicio y final; de esta manera cada tarea es comparada con las aristas de entrada de la tarea que la precede. Así, el algoritmo toma la medida heurística de mayor valor como referencia para recuperar los BP.

El trabajo presentado por Li, Reichert y Wombacher [74], plantea una búsqueda de procesos de negocio que utiliza una colección de variantes de BP empleando un algoritmo genético. Este algoritmo utiliza un modelo original de referencia S como punto de partida con el propósito de encontrar modelos de BP vecinos en la colección dada con una distancia media ponderada. El algoritmo se ejecuta hasta encontrar S', el cual consiste en un nuevo modelo de referencia con una distancia mínima entre S y S', centrándose en encontrar una medida heurística mínima. Dado

que el espacio de búsqueda puede llegar a ser muy grande, el algoritmo toma una decisión rápida sobre qué camino elegir, con el fin de medir la "proximidad" entre el modelo de referencia y un candidato de la colección a ser recuperado.

Búsqueda de procesos sobre repositorios. En el trabajo de Yan, Dijkman and Grefen [75] se presenta un Repositorio para la Gestión Integrada de procesos denominado IPM. Este permite almacenamiento y recuperación de BP. La información de los BP es almacenada mediante conectores en una base de datos XML o una base de datos relacional. IPM está centrado en el almacenamiento de los BP específicos de una compañía. Para almacenar y recuperar los BP o información relacionada, IPM utiliza un lenguaje de consulta propia, denominado IPM *Process Query Language* (IPM-PQL), el cual soporta consultas específicas que comprueban si un BP contiene una determinada actividad o cierta transición entre actividades. IPM también soporta la búsqueda de BP en el repositorio utilizando una serie de clasificaciones tales como tipo de actividad y descripción de actividad. Tanto el lenguaje de modelado como de consulta es propietario y está diseñado para modelar los procesos específicos de una compañía.

La Rosa y otros [76] presentan APROMORE, un repositorio avanzado que mantiene, analiza y reutiliza grandes colecciones de modelos de procesos; además, es una plataforma de código abierto, desplegada sobre la arquitectura SOA, para permitir acceso a los usuarios 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 que implementa unas funcionalidades básicas: importar y exportar modelos, búsqueda y clasificación de modelos, así como funcionalidades de comparación (búsqueda de similitud de procesos y funcionalidad de gestión de procesos). Cabe resaltar que esta aproximación no proporciona consultas semánticas.

Búsqueda de BP aplicando *clustering* secuencial. Ferreira [49] plantea un algoritmo de *clustering* secuencial con el propósito de organizar una serie de objetos en un conjunto de grupos, donde cada grupo contiene objetos que son similares por un tipo de medida. Esta medida depende del tipo de objetos o datos presentes en los BP. Cada grupo está asociado con un modelo probabilístico, por lo general una cadena de Markov (al igual que los presentados por Wives y por Ristov y Korenčić [91, 92]). Si para todos los grupos se conocen las cadenas de Markov, entonces cada secuencia de entrada es asignada a la agrupación que mejor pueda producir tal

secuencia. El algoritmo desarrolla los pasos siguientes: 1) Inicializa los modelos de cluster (es decir, la cadena de Markov para cada grupo) al azar. 2) Asigna a cada secuencia de entrada el grupo que es capaz de producir la mayor probabilidad. 3) Estima de cada modelo del clúster de la serie de secuencias que pertenecen a ese grupo. Finalmente, se repiten los pasos 2 y 3 hasta encontrar los modelos del *cluster*.

En el trabajo presentado por Ferreira y otros [50], se propone un enfoque de *clustering* que agrupa secuencias similares e identifica tópicos temáticos presentes en los BP sin la necesidad de proporcionar información de entrada. La agrupación es realizada con el propósito de encontrar información valiosa sobre el tipo de secuencias que se están ejecutando en los BP. El procedimiento de agrupación incluye: un algoritmo alfa, el cual es capaz de volver a crear el BP de una red de Petri con base en las relaciones encontradas en el registro de ejecución de los BP. Métodos de inferencia que consideran el registro de ejecución como una secuencia simple de símbolos, inspirada en el modelo de Markov (al igual que el presentado por Wives [91]) y que genera un modelo gráfico que considera cadenas de Markov de orden creciente con grafos no cíclicos dirigidos.; un algoritmo de *Clustering* jerárquico que tiene en cuenta un amplio conjunto de trazas de ejecución de un mismo proceso, que separa las trazas en grupos y encuentra el gráfico de dependencias por separado para cada grupo un algoritmo genético donde las soluciones candidatas son evaluadas por una función de aptitud y cada solución es representada mediante una matriz causal, es decir, un mapa de las entradas y dependencias de salida para cada actividad.

Búsqueda de procesos aplicando *clustering* jerárquico. Caicedo y otros [77], presentan un esquema de agrupación de BP (tal como lo hace Ekanayake y también Koehler [65, 67]) para recuperación de esquemas gráficos en grupos similares de (sub) procesos y sus relaciones. Se parte de un macro proceso para llegar hasta las actividades más específicas, para lo cual se toma un conjunto de grafos dirigidos $G_i = <N_i, A_i>$ donde N_i es el conjunto de nodos y $A_i \subseteq N_i * N_i$ es el conjunto de arcos posiblemente etiquetados, generando un esqueleto de agrupación típica de subestructuras. Los grafos son iterativamente analizados para descubrir en cada paso un grupo de sub-estructuras isomorfas. El *clustering* se utiliza para comprimir los grafos sustituyendo a cada ocurrencia de la subestructura con un nodo; este proceso se repite hasta que no haya más compresión posible.

Aiolli, Burattin y Sperduti [45], presentan un algoritmo alfa para *clustering* (tal como lo hace Koehler [67]) que trasforma un BP en dos conjuntos de relaciones entre sus actividades. El algoritmo selecciona las perspectivas que deben ser consideradas como relevantes para la comparación. El objetivo es convertir un determinado BP en dos conjuntos: un conjunto de relaciones entre las actividades que tienen que ocurrir y otro conjunto de relaciones que pueden ocurrir o no. El algoritmo de clusterirg vincula una medida de similitud entre dos grupos, la cual se define como la similitud de todos los pares de actividades pertenecientes a los dos grupos. El objetivo es comenzar con cada uno de los elementos de un grupo único y en cada iteración del algoritmo dos o más grupos se fusionan en uno solo. El algoritmo se ejecuta hasta que todos los grupos formados se fusionan en un solo grupo que contiene todos los elementos con mayor similitud, una vez formado el grupo final, este es graficado a través de una estructura jerárquica en forma de árbol, denominada *dendograma* (comúnmente usada en minería de datos).

2.4.2 Brechas identificadas

En la Tabla 4, se describen las brechas del conocimiento e investigación de cada una de las temáticas identificadas y clasificadas en esta sección, con base en el estudio de los anteriores trabajos**.**

Tabla 4. Brechas identificadas en las propuestas para búsqueda de BP.

Propuestas	Retorno de resultados	Brechas
Basadas en Lingüística • [Koschmider et al, 2011] • [Reijers et al, 2011]	Tf/idf, modelo vectorial, resultados ordenados con base en número de términos relevantes	• Dejan de lado el flujo de ejecución o comportamiento. • No tienen en cuenta similitud en patrones frecuentes.
Reglas de asociación • [Zichen Huan et al, 2010] • [Pérez-castillo et al, 2011]	Detectan estructuras específicas, los resultados son retornados con base al cumplimiento de las reglas asociación.	• Log deben pertenecer a un proceso bien estructurado • Información más relevante tiene que ser proporcionada de forma manual
Algoritmos genéticos • [Chris J et al, 2010] • [Chen Li et al, 2011]	Los resultados son retornados con base a heurísticas, mediante el cálculo de la cantidad de veces que una tarea es directamente precedida por otra	• El retorno de los procesos recuperados depende del cálculo de muchas variables. • La importancia de cada tarea solo depende del número de veces que aparece en el registro log

***Clustering* secuencial** • [Ferreira et al., 2009] • [Diamantini et al, 2011] • [Montani et al, 2014]	Esquemas gráficos dependiendo de sus relaciones Tópicos temáticos presentes en los BP	• Toma un BP azar como punto de partida para formar y estimar los grupos • En el proceso de agrupación se eliminan secuencias de tareas que ocurren una sola vez.
***Clustering* Jerárquico** • [Aiolli et al, 2012] • [Jae-Yoon et al, 2011] • [Sarno et al, 2013]	Utilizan datos tales como: nombre de las actividades, tiempo de duración de cada actividad, número de errores.	• No tienen en cuenta secuencias que pueden compartir información de tipo estructural o de comportamiento.

Fuente: elaboración propia con base en el estado de la cuestión.

En esta tesis se planteó un enfoque de búsqueda multimodal de BP, que a diferencia de las anteriores propuestas, se caracteriza porque:

- La búsqueda es ejecutada utilizando Información lingüística (nombres y descripción de tareas, eventos y compuertas) unida con *codebook*s (tarea-tarea, tarea-compuerta, tarea-evento–compuerta, entre otros tipos de comportamiento secuencial).
- La información de los BP, es almacenada en un esquema de indexación y búsqueda multimodal, con el propósito de tener un espacio más representativo para realizar las consultas y generar respuestas más precisas.
- El proceso de búsqueda se realiza en un solo proceso unificado, sobre el esquema multimodal obteniendo un menor tiempo de ejecución de las consultas.
- Agrupa BP basado en representación multimodal, teniendo en cuenta: información lingüística (descripción de las actividades) y estructural (tipo de tareas y tipo de compuertas).
- Agrupa los resultados relevantes de una consulta, no todo el repositorio.
- En los grupos formados, los elementos comparten información lingüística y estructural, logrando con ello grupos más cohesivos.
- Los grupos son etiquetados, para permitir la identificación de su contenido con facilidad.

2.5 Conclusiones

La visión general sobre los conceptos involucrados en esta tesis, permitió identificar qué tipo de información utilizar en el momento de consultar grandes colecciones de BP almacenados en repositorios. Con base en las características de los BP hay tres tipos de información específica que puede ser identificada a partir de éstos, información de estructura gráfica, de semántica de comportamiento y de semántica de operación. Para acceder a este tipo de información se han definido los tipos de consulta: exacta basada en la estructura gráfica, de similitud basada en la estructura gráfica, exacta basada en la semántica de comportamiento, exacta basada en la semántica de operación.

Con base en los trabajos relacionados, las propuestas basadas en búsqueda de BP se limitan al emparejamiento de entradas y/o salidas, para lo cual toman como base la información textual de los elementos pertenecientes a cada BP. En el proceso de búsqueda estas propuestas no tienen en cuenta información del flujo de ejecución del BP, así como del comportamiento, estructura, tipo de actividades, tipo de compuertas y tipo de eventos.

Por otro lado en las propuestas basadas en el proceso de agrupación utilizan los datos textuales de cada BP. Entre estos datos están: nombre de las actividades, tiempo de duración de cada actividad y número de errores. Además la agrupación elimina secuencias de actividades que ocurren una sola vez, sin tener en cuenta que estas secuencias pueden compartir información de tipo estructural o textual que puede ser relevante en el momento de la selección de los BP que forman cada grupo.

A pesar de los aportes ya realizados por los enfoques basados en agrupación anteriormente nombrados, los resultados pueden ser ampliados o mejorados abarcando un número mayor de características de información como: descripción de las actividades, tipo de tareas, tipo de compuertas, estructura, comportamiento, entre otras presentes en los BP. Centrarse en un solo tópico, por ejemplo el textual, solo permite realizar agrupación de BP mediante la comparación de información correspondiente a los nombres y la descripción de cada uno de los elementos pertenecientes a cada BP. En los grupos formados sobre este tópico se dejan de lado BP que pueden tener similitud en su estructura, tipo de tareas o comportamiento.

A partir del análisis anterior, en esta tesis se planteó un modelo que unifica en un solo espacio de búsqueda unidades estructurales de comportamiento y características textuales existentes en los BP, en lo que se conoce como una representación de búsqueda multimodal. Adicionalmente, integra un algoritmo de *clustering* que utiliza varios tipos de información, tales como textual, estructural y de comportamiento, para agrupar los resultados de la búsqueda. La agrupación es realizada con base en la similitud de los tipos de información contenida en cada uno de los BP recuperados, para lograr así una forma más efectiva en el despliegue de los resultados.

Capítulo 3

Modelo de búsqueda multimodal

El éxito de un mecanismo de búsqueda de BP está dado por el grado de relevancia en los BP recuperados a partir de una consulta. Según Pretschner [78], solo la mitad de los resultados de una consulta sobre los actuales mecanismos de búsqueda son relevantes para el usuario. Este hecho se atribuye a que las consultas son cortas e incompletas en relación con la intención individual de quien usa el mecanismo de búsqueda. En consecuencia investigaciones comerciales y académicas, han planteado el desarrollo de nuevos enfoques para la búsqueda de BP con el propósito de mejorar la precisión y proporcionar los resultados de búsqueda deseados en poco tiempo [18].

Con base en lo anterior, se han planteado una serie de técnicas para la búsqueda de BP, las cuales utilizan representaciones heterogéneas de la información contenida en este tipo de modelos. Entre estas representaciones están las Redes de Petri y los grafos. Las propuestas basadas en redes de Petri modelan y analizan las variantes estructurales de los BP de un repositorio. El fundamento matemático de las Redes de Petri permite comprobar si existen semejanzas entre los BP, para así establecer un nivel de similitud entre los BP a retornar [79, 80]. Las propuestas basadas en grafos eliminan las etiquetas de las actividades (nodos del grafo), con el propósito de establecer un contexto estructural en términos de nodos predecesores y sucesores de los grafos que representan los BP. La evaluación de la similitud es realizada iterativamente hasta encontrar un valor máximo de similitud estructural o alineación óptima para un par de BP comparados [81, 82].

Con el propósito de aumentar la relevancia de los resultados en un mecanismo de búsqueda de BP, en esta tesis se propuso un nuevo modelo que unifica en un solo espacio de búsqueda, unidades estructurales de comportamiento y características textuales existentes en los BP, en lo que se conoce como una representación de búsqueda multimodal. Esta técnica es muy usada en investigaciones relacionadas con recuperación de información multimedia [77].

A continuación, son descritos el modelo de búsqueda multimodal y cada una de sus capas. Capa de pre-procesamiento, la cual contiene los algoritmos de formación de

componente lingüístico y formación del componente del *codebook*. La capa de indexación que ejecuta el algoritmo de ponderación y creación del índice de búsqueda, también presenta una descripción del repositorio. La capa de consulta, en la cual son explicados los tipos de consulta soportados por el modelo, describe el método de puntuación conceptual. Finalmente se presenta un ejemplo de ejecución del modelo de indexación y búsqueda multimodal.

3.1 Modelo de búsqueda multimodal

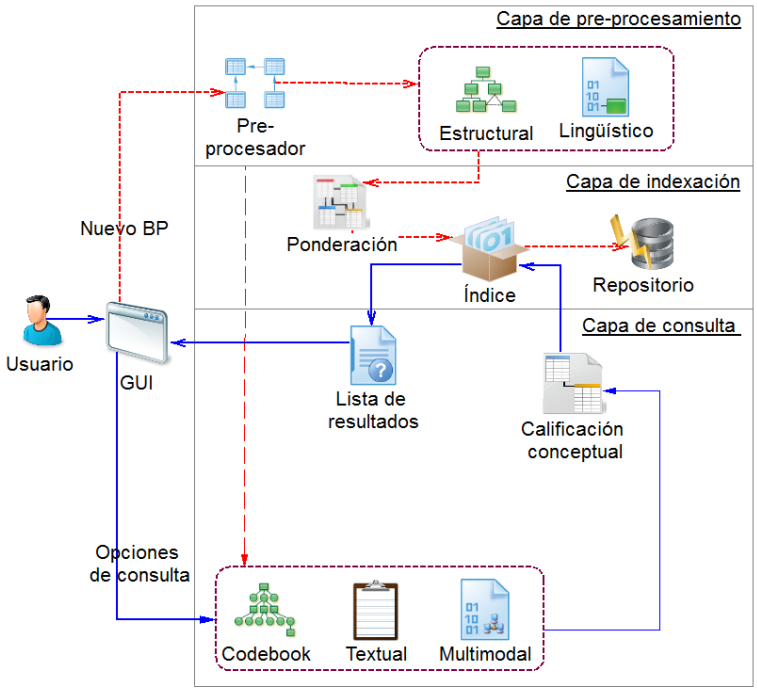
Hasta ahora los modelos de búsqueda multimodal han sido aplicados únicamente en sistemas de recuperación de información multimedia, por lo tanto el modelo de búsqueda multimodal de BP presentado en esta tesis, representa una propuesta novedosa que aprovecha la eficiencia de la búsqueda multimodal, aplicada en el campo de búsqueda de BP. El modelo multimodal permite almacenar y buscar BP. El almacenamiento es organizado en un repositorio que contiene los archivos físicos de los BP modelados con BPMN. En la búsqueda se aplica una estrategia que integra información lingüística y estructural contenida en los BP, permitiendo de esta manera incrementar la efectividad y relevancia de los resultados de las búsquedas a partir de consultas definidas por los usuarios. La arquitectura del modelo multimodal está compuesta por tres capas (ver Figura 4): Pre-procesamiento, Indexación, y Consulta, descritas a continuación.

3.1.1 Capa de Pre-procesamiento

Esta capa está compuesta por un pre-procesador que transforma los BP desde su formato original BPMN, a una representación matricial en la cual cada BP es considerado un vector de términos compuesta por un elemento lingüístico y uno de *codebook*.

- **Pre-procesador**
  El pre-procesador contiene un algoritmo que toma un BP en notación BPMN y construye el *codebook* de comportamiento mínimo (el cual puede estar formado por dos o más componentes secuenciales, presentes en cada BP) y el componente lingüístico. El algoritmo es descrito a continuación.

Figura 4. Arquitectura de modelo de búsqueda multimodal.

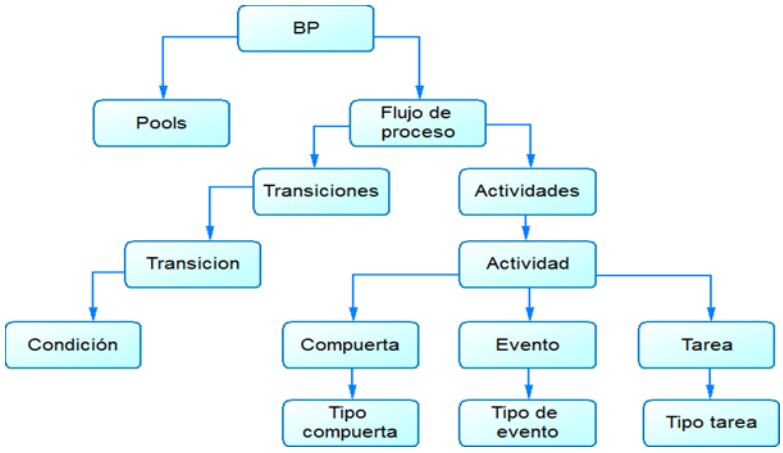


- **Formación del Componente Lingüístico**

  El algoritmo del pre-procesador, toma cada modelo de BP (BP_i) del repositorio y lo representa en forma de árbol (A_i) teniendo en cuenta el flujo de ejecución. A continuación, por cada árbol extrae las características textuales (nombre de actividad, tipo actividad y descripción); cada una de las características textuales extraídas se transforma a minúsculas, se eliminan caracteres especiales y palabras vacías, luego se aplica el algoritmo de porter [79], para realizar *stemming*, con el propósito de convertir cada uno de los componentes textuales a su raíz léxica (por ejemplo "*Running*" y "*Runner"*, en "*Run*"). Adicionalmente, con las características textuales procesadas se forma un vector $Ct_i = \{Ct_{i,1}, Ct_{i,2}, \ldots, Ct_{i,M}\}$, donde M es el número total de características textuales del repositorio. Dicho vector es luego utilizado para crear la matriz "componente de características textuales" *MC*. En esta matriz cada componente $Ct_{i,j}$ representa el peso de la característica textual *j* en el BP_i.

- **Formación del componente de *codebook***
  En este paso el algoritmo pre-procesador incorpora una estrategia de formación (*codebooks*) para generar unidades estructurales básicas secuenciales de los BP. Estos *codebooks* son construidos con base en los patrones secuenciales frecuentes en la estructura de cada uno de los BP existentes en el repositorio. El pre-procesador estructural crea los *codebooks* tomando cada uno de los BP del repositorio y recorriendo de manera secuencial la estructura en árbol de los mismos (Figura 5). Como resultado, el pre-procesador estructural utiliza los *codebooks* para crear la matriz *MCd* de componentes estructurales. Para esto, de cada árbol A_i se extrae un vector de transiciones (*vt*) que contiene las transacciones de nodos que se encuentren secuencialmente interconectados entre sí dentro de la estructura del BP. Formalmente, sea *P* el total de transiciones posibles en el repositorio, para cada BP_i, se crea un vector $Cd_i = \{Cd_{i,1}, Cd_{i,2}, \ldots, .Cd_{i,P}\}$, en este vector cada *codebook* está formado por la unión de dos o más transiciones.

Figura 5. Estructura del árbol que representa un modelo de BP.



Dicho vector representa una fila *i* de la matriz *MCd* de componentes *codebook*, en la cual *N* es el número total de BP del repositorio, *i* es un BP específico, *j* es el índice para un *codebook* específico del repositorio y *P* es el total de *codebooks* del repositorio.

Por ejemplo, la muestra un fragmento de un BP con sus actividades. Cada actividad es representada con una cadena de texto que define el tipo de nodo (StartEvent,

TaskUser, TaskService). El tipo de nodo se refiere a la funcionalidad de cada actividad dentro del BP. La Tabla 5 muestra la correspondencia entre las actividades del BP de la Figura 6 y sus tipos de nodos.

Tabla 5. Ejemplo de las actividades del BP en la figura 6 y la representación de sus tipos de nodos

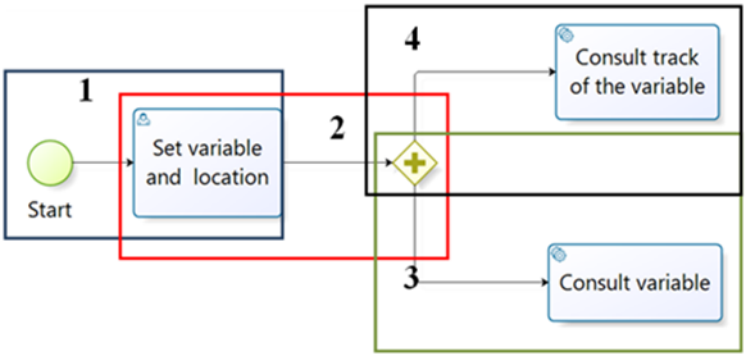
Actividad	**Tipo de representación**
Start	StartEvent
Set variable and location	TaskUser
Route	RouteParallel
Analyze variable tracking	TaskService
Analyze Variable	TaskService

Fuente: elaboración propia.

En el ejemplo de la Figura 6, pares de secuencias de nodos o componentes se toman (tipos de nodos) para representar el conjunto de *codebooks*. Estos pares de nodos son representados como cadenas de texto secuenciales (es decir, la adición de la cadena de texto del nodo anterior con el nodo actual): Cd_i= {StartEvent_TaskUser$_1$, TaskUser_ParallelRoute$_2$, ParallelRoute_TaskService$_3$, ParallelRoute_TaskService$_4$}. Donde *StartEvent_TaskUser$_1$* corresponde a la concatenación del evento *Start* con la actividad "*Set variable and location"* de la Figura 6. Y así de manera similar para el resto de los componentes de vector Cd_i.

Los *codebooks* se forman simulando la técnica de n-gramas (secuencia de N caracteres formando un grama en cadenas de texto). A diferencia de los n-gramas, que son secuencias lineales simples de caracteres, los *codebook* se forman al unirse componentes n-estructurales utilizando la secuencia y la semántica del flujo de control de los BP representado por los tipos de nodos. De acuerdo con Chang y. Wang [83], la semántica de comportamiento de los BP se describe a través de las actividades que están involucradas en el orden de ejecución. El orden de ejecución en esta tesis está representado por los *codebooks* o secuencias de componentes interconectados. Es importante señalar que los *codebooks* respetan la secuencialidad y la semántica del flujo de control de los BP.

Figura 6. Estructura de cada codebook.



3.1.2 Capa de Indexación

En esta capa se crea el índice de búsqueda multimodal, el cual está compuesto por dos espacios de búsqueda: 1) Indexación textual de las funciones de negocio de cada BP y 2) indexación estructural (caracterización entre tipos de tareas, tipos de eventos y tipos de conexiones). Estos dos espacios de búsqueda se unifican en una estructura multimodal para crear un índice más amplio que permita tener una representación mayor del objeto de estudio (el repositorio de BP). Finalmente, el índice de búsqueda almacena una estructura conceptual denominada matriz índice (*MI*) de términos a través del BP, $MI_{i,j} = \{MCd_{i,j} \cup MC_{i,j}\}$ (similar al modelo vectorial de recuperación de información [79]). Esta matriz contiene en cada celda un peso (w_{ij}) que refleja la importancia de cada componente textual/estructural en su raíz léxica o *codebook* para cada BP (ver figura 7).

- **Ponderación**

La matriz *MI* está creada con base en la ecuación (1) propuesta por Salton [79], donde $F_{i,j}$ es la frecuencia observada del componente textual o del *codebook j* en el BP_i. Max(F_i) es la mayor frecuencia observada en el BP_i, *N* es el número de *BP* en la colección (repositorio), y n_j es el número de *BP* en los que aparece el componente textual o *codebook j*. La Figura 7, muestra gráficamente la matriz índice constituida por los dos espacios o componentes de la *MI*: el primero (recuadro verde a la izquierda) representa el peso de los elementos textuales en cada BP, y el segundo (recuadro azul a la derecha), el peso de los elementos de cada *codebook* en cada BP

$$w_{i,j} = \frac{F_{i,j}}{\max(F_i)} \times \log\left(\frac{m}{m_j + 1}\right) \quad (1)$$

Figura 7. Matriz índice (MI)

	MC					MCd				
BP1	**Ct1**	**Ct2**	**Ctl**		**CtL**	**Cd1**	**Cd2**	**Cdk**		**CdK**
BP2	***W 1,1***					***W 1,1***				
BP..		***W 2,2***					***W 2,2***			
BPn			**..**					**..**		
BP..				***W n,l***					***W n,k***	
BPN					***W N,L***					***W N,K***

- **Índice**

Este componente está conformado por un archivo almacenado dentro del sistema de archivos del sistema operativo. Este archivo almacena el índice de búsqueda multimodal, el cual contiene indexado cada uno de los BP del repositorio, además de la referencia al archivo BPMN que representa a cada BP indexado.

- **Repositorio**

Uno de los principales problemas con respecto a la búsqueda de BP es la falta de un repositorio cerrado de acceso libre para la ejecución de las pruebas. Esto dificulta la comparación del modelo de búsqueda definido en esta tesis con propuestas para la búsqueda de BP desarrolladas previamente por la comunidad científica [81]. Por esta razón, para la validación de esta tesis, se desarrolló una colección de pruebas cerrada, compuesta de: un repositorio de 100 BP descritos en lenguaje BPMN, en el marco de las telecomunicaciones y la geo-referenciación; una lista predefinida de 6 BP como consultas; y una lista de resultados ideales para cada consulta (conjunto de BP relevantes).

Para generar el conjunto de BP relevantes, la evaluación se basó en una estrategia colaborativa donde 56 revisores expertos de diferentes instituciones ejecutaron comparaciones entre pares de BP (un BP de consulta y cada BP del repositorio). La descripción de la estrategia y los resultados la construcción del repositorio están presentados en [84] (ver anexo d).

Por otra parte, el repositorio actúa como la unidad central de almacenamiento de BP, es similar a una base de datos que comparte información acerca de los artefactos de ingeniería producidos o utilizados por una organización [85, 86]. Está encargado de almacenar y representar todos los atributos de información presentes en los BP (roles, descripción de actividades, temporizadores, mensajes, llamadas a servicios) [35].

3.1.3 Capa de Consulta

Esta capa es la encargada de permitir a los usuarios realizar búsquedas de BP a partir de tres opciones de consulta: textual, *codebook*, y multimodal. Cada consulta es representada a través de un vector de términos $q= \{t_1, t_2, t_3,\ldots t_n\}$. A este vector se le aplica el mismo mecanismo de pre-procesamiento aplicado en la capa de indexación, con lo cual se obtienen los términos del vector consulta reducidos a su raíz léxica. Las opciones de consulta soportadas por el modelo multimodal se describen a continuación.

- **Consulta textual:** en esta opción el usuario puede digitar una o varias palabras clave representadas en lenguaje natural, las cuales forman un vector de consulta $qpc=\{pc_1, pc_2, . pc_k\}$. El sistema pre-procesa las palabras clave, genera un vector de consulta con los términos y luego compara esta consulta con el componente textual del índice (*MI*) con el fin de entregar aquellos BP con mayor similitud respecto a la consulta planteada por el usuario.
- **Consulta estructural:** en esta opción el usuario tiene la posibilidad de elegir uno o varios *codebook* de una lista de componentes estructurales formados a partir de la colección de BP existentes en el repositorio, con ello se crea un vector de consulta $qcd=\{cd_1,cd_2,\ldots cd_n\}$. Los elementos utilizados en la consulta son comparados con el componente del índice (*MI*) que contiene los elementos estructurales con el propósito de retornar los BP con mayor similitud.
- **Consulta multimodal:** esta opción integra las dos opciones de consulta anteriores. En esta, el usuario escribe palabras de consulta y selecciona *codebooks*, en este sentido el sistema crea un vector de consulta $qmg = \{qpc \cup qcd\}$ que es comparado con cada uno de los modelos representados en la matriz *MI* teniendo en cuenta los dos espacios de búsqueda (información textual e información estructural).

Finalmente, en el proceso de comparación entre vector de consulta y los BP registrados en el índice de búsqueda, se ejecuta un mecanismo de correspondencia y refinamiento de resultados, es decir la comparación de los datos de la consulta con los BP almacenados en el repositorio. Para esto, los datos introducidos en estas opciones de consulta son representadas en un vector de términos q= {t_1, t_2, t_3,…tn}. Una vez se obtiene la cadena de consulta procesada, se ejecuta la búsqueda en el espacio elegido por el usuario a través de la ecuación (2).

- **Puntuación conceptual**

Cuando el usuario hace una consulta utilizando los BP procesados en el índice de búsqueda, este módulo genera una lista de resultados teniendo en cuenta las tres opciones de consulta nombradas anteriormente (textual, estructural y multimodal). En este proceso se ordenan y filtran los BP retornados por las opciones de consulta haciendo uso de la ecuación (2) adaptada de Apache Lucene [87].

$$Puntuacion = (q,d) = coord(q,d) * \sum_{t \in q} (tf(t \in d) + idf(t)^2 * norm(t,d)) \quad (2)$$

La clasificación de la lista de resultados es realizada en forma decreciente (mayor a menor), con base en el nivel de puntuación alcanzada por los BP al efectuar la ejecución de la consulta. Para esto, en la ecuación (2), se tienen las siguientes consideraciones:

- *t* es un término (textual / *codebook*) de la consulta *q*.
- *d* es el modelo de BP consultado.
- *tf (t ∈ d)* es la frecuencia de aparición (número de veces) del término *t* en el BP *d* específico.
- *idf(t)* define la importancia relativa del término *t* en el repositorio de *BP* como un todo, expresada como la frecuencia de aparición del término *t* en la colección de BP (expresa numéricamente cuán relevante es el término *t* para los BP de la colección).
- *Coord (q, d)* es un factor de puntuación que está determinado por el número de términos pertenecientes a la consulta y que además existen en el modelo de BP consultado.

Una vez realizados los cálculos de ponderación, los resultados se ordenan, se filtran y se listan en orden descendente de acuerdo con la similitud que presentan respecto a la consulta ingresada por el usuario.

- **Lista de resultados**

Una vez los resultados son ordenados y filtrados se listan de acuerdo al nivel de similitud que presentan con respecto al BP en la opción de consulta seleccionada por el usuario.

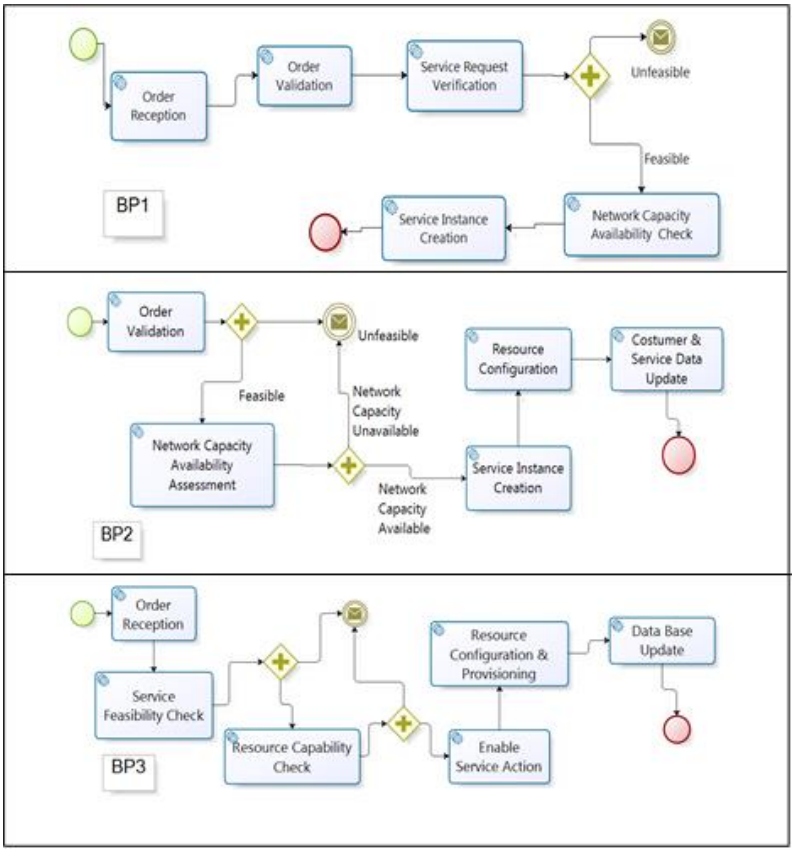
3.2 Ejemplo de ejecución

A continuación se presenta un ejemplo del modelo de búsqueda multimodal en el que se describe cada uno de los pasos realizados: Pre-procesamiento, Indexación y Consulta. Para este ejemplo, se examina un repositorio con tres BP (BP1, BP2, y BP3) (Figura 8).

Supóngase que previo a la consulta del usuario, el sistema ha procesado los BP del repositorio aplicando los algoritmos para generar los componentes lingüístico y estructural. Estos algoritmos ponderan cada uno de los elementos que pertenecen a los BP para construir el índice multimodal. Una vez que cada BP se analiza y pondera, se almacena físicamente en el repositorio.

En la fase de búsqueda, el usuario puede utilizar un BP del repositorio o un BP externo (que no existe en el repositorio) para ejecutar la consulta. Cuando el BP no está en el repositorio se procesa aplicando todos los algoritmos mencionados anteriormente, y entonces se añade al repositorio. En este punto, el usuario puede elegir entre las opciones de consulta (textual, *codebook*, y multimodal). De esta manera, una vez que el usuario elija una opción, los resultados se muestran de acuerdo con la similitud entre el BP de consulta y los BP almacenados en el repositorio. A continuación, se describen en detalle los pasos ejecutados en el modelo:

Figura 8. Repositorio de ejemplo (con tres BP).



Pre-procesador: inicia tomando cada uno de los BP del repositorio representado en la Figura 8 (BP1, BP2, BP3). A continuación, crea el componente de *codebook*, en el cual el conjunto de *codebooks* formados por cada BP representa una fila de la matriz (MCd), de igual forma todos los elementos lingüísticos que conforma cada BP representan una fila de la matriz (MC). La formación de cada una de estas matrices puede verse a continuación en la Figura 9.

Figura 9. Ejemplo de construcción de los componentes de codebook y lingüístico.

	MC								MCd			
BP_1	Order	Reception	Verification	Network	Capacity	Availability	Assessment	...	StartEvent_TaskService	TaskService_TaskService	TaskService_TaskService	...
BP_2	$ct_{j,1}$	$ct_{j,2}$	$ct_{j,3}$	$ct_{j,4}$	..	..	$ct_{j,p}$	..	$cd_{i,1}$	$cd_{i,2}$	$cd_{i,3}$	$cd_{i,p}$
BP_3	$ct_{j+1,1}$	$ct_{j+1,2}$	$ct_{j+1,3}$	$ct_{j+1,4}$	..	..	$ct_{j+1,p}$	..	$cd_{i+1,1}$	$cd_{i+1,2}$	$cd_{i+1,3}$	$cd_{i+1,p}$

Indexación: en esta fase es tomado cada uno de elementos existentes en las matrices formadas en la etapa anterior (Figura 9), para crear la matriz multimodal $MI_i = \{\mathrm{MCd_{i,j}} \cup \mathrm{MC_{i,j}}\}$. Una vez la matriz MI es construida, se realiza el cálculo de la ponderación de cada uno de sus elementos $w_{i,j}$ cuyos valores son calculados con la ecuación 1, como se muestra en la Tabla 6. El índice que forma el espacio de consulta es creado en el sistema de archivos, con el propósito de guardar referencia a la ruta especifica de los archivos que representan los BP, para garantizar su recuperación. Por ejemplo, el cálculo del peso para el elemento $w_{0,0}$ en el BP1, sería $\mathrm{w_{0,0}} = \frac{1}{3} * \mathrm{Log}\left(\frac{3}{1+1}\right) = 0{,}138.$

Tabla 6. Ejemplo de cálculos de ponderación wi,j.

BP1	0,138	0,301	0,301	0,602	0,602	0,301	0,000	0138	0,301	0,301	0,602	0,201	0,602	0,602	0,201
BP2	0,138	0,398	0,398	0,602	0,398	0,398	0,000	0,138	0,301	0,265	0,602	0,398	0,602	0,301	0,602
BP3	0,138	0,301	0,398	0,398	0,301	0,398	0,301	0,201	0,301	0,602	0,265	0,602	0,602	0,265	0,602

Fuente: elaboración propia.

Consulta: en esta fase, el usuario puede introducir nuevos BP que se pre-procesan, ponderan y anexan al índice multimodal y, además se almacenan en el repositorio (con el fin de ser utilizados en futuras consultas), o selecciona un BP de consulta existente en el repositorio (Figura 8). Este BP de consulta es pre-procesado para formar un vector que incluye los componentes lingüísticos y de *codebook* (Tabla 7). Más tarde, se toma el vector de consulta y el peso de cada uno de sus componentes es calculado utilizando la ecuación (1). Finalmente, el vector se compara con los pesos en la matriz índice multimodal, y de esta forma los BP que más se asemejan al BP de consulta son recuperados de manera ordenada, utilizando la ecuación (2).

Figura 10. Ejemplo BP consulta**.**

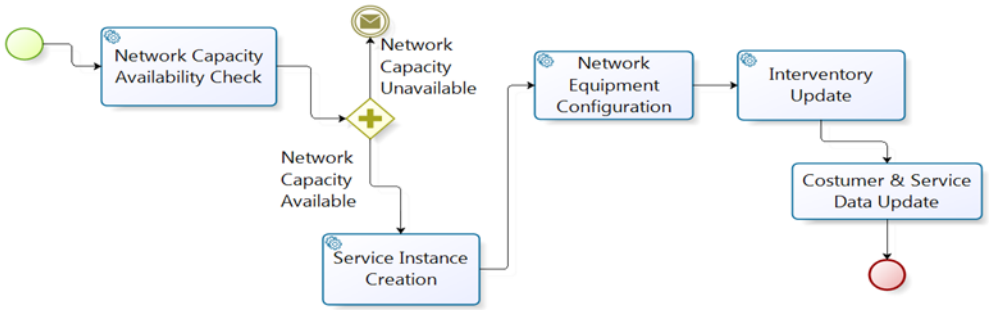


Tabla 7. Ejemplo de cálculo de pesos para BPq.

BPq	0,138	0,201	0,301	0,602	0,602	0,301	0,602	0,201	0,201	0,301	0,602	0,201	0,602	0,602	0,201

Fuente: elaboración propia.

Por ejemplo, en la consulta realizada con el BPq de la Figura 10, se obtiene la clasificación presentada en la Tabla 8.

Tabla 8. Resultados de las opciones de consulta.

Opciones de consulta	**BP1**		**BP2**		**BP3**	
	Similitud	**% Similitud porcentual**	**Similitud**	**% % Similitud porcentual**	**Similitud**	**% % Similitud porcentual**
Consulta por *Codebook*	0.9506	95.06	0.9036	90.36	0.8926	89.26
Consulta textual	0.5304	53.04	0.7094	70.94	0.1735	17.35
Consulta Multimodal	0.9537	95,37	**0.9693**	**96.93**	0,9233	92,33

Fuente: elaboración propia.

Teniendo en cuenta que idealmente, el BP con la mayor similitud al BP de consulta es BP2, se puede observar que los mejores resultados fueron obtenidos con la opción multimodal, seguida de los resultados generados por el *codebook* y, finalmente, la textual. Esto muestra que los resultados de la opción multimodal tienen mayor nivel de relevancia, ya que la lista de resultados se organiza de una manera similar a la clasificación generada por los jueces humanos (ver capítulo 5).

3.3 Conclusiones

El uso de información heterogénea en mecanismos de búsqueda de BP, ha permitido representar formalmente los BP a través de grafos y redes de Petri, con el propósito de aumentar el nivel de relevancia en los resultados reportados y disminuir el tiempo de consulta con este tipo de mecanismos.

Por otra parte, la representación de la información estructural de comportamiento por medio de *codebooks* en forma de cadenas de texto, hace que disminuya el tiempo requerido para la ejecución de la indexación, ya que no utiliza algoritmos complejos y exhaustivos de isomorfismo de grafos como en el caso de algunos trabajos

mencionados anteriormente [80-82]. El *codebook* formado por secuencias de dos componentes (N=2), permite obtener mayor nivel de precisión en las consultas ejecutadas sobre el índice multimodal.

La aplicación de una estrategia de búsqueda multimodal, permitió tener una representación ampliada de la información contenida en cada uno de los BP del repositorio. La representación de la información en el índice de búsqueda multimodal en forma de cadenas de texto, aporta mayor efectividad en las búsquedas, mayor agilidad y flexibilidad para realizar varias clases u opciones de consulta.

En el ejemplo de ejecución explicado arriba, se puede evidenciar la sencillez en el proceso de indexación y creación del índice de búsqueda. Por otra parte, el índice de búsqueda por su robustez hace posible ejecutar varios tipos de consulta que pueden ser: textual, estructural o multimodal. Los resultados obtenidos en el ejemplo y en la experimentación que son presentados en un capitulo posterior demuestran que la unión de la información lingüística y estructural de los BP en un solo espacio de búsqueda, genera resultados más relevantes en comparación con otras técnicas identificadas en el estado del arte.

Capítulo 4

Algoritmo de *clustering* propuesto para extender el modelo de búsqueda multimodal

En la actualidad las organizaciones definen BP que representan líneas de productos o servicios basados en conjuntos de características que permanecen constantes en función de una familia de productos o servicios dada [88]. En consecuencia, identificar manualmente un grupo de BP que represente una familia de productos, puede considerarse como una tarea dispendiosa que demanda tiempo considerable. Como alternativa, se han planteado mecanismos para la detección automática de BP que puedan explicar el comportamiento de las líneas de productos trabajadas en la empresa [89]. Estos mecanismos han incorporado algoritmos de agrupación o *clustering*, con el propósito de reunir en un mismo grupo un conjunto coherente de BP que representen una línea o familia de productos, con base en características comunes que pueden ser de flujo, de control, de finalidad, de estructura, de función del proceso o producto que representan, de manera que cada grupo podría ser utilizado posteriormente para generar un modelo de proceso de mayor comprensión [90]. En este sentido, los ingenieros (modeladores de BP) pueden explorar organizadamente los resultados agrupados para plantear posibles sugerencias sobre cómo rediseñar los BP, a fin de incorporar los cambios más frecuentes y significativos de una vez para todos los elementos de cada grupo.

Los resultados de la agrupación proporcionan valores de pertenencia que representan el nivel de correspondencia entre un BP y el grupo que lo contiene, esto puede considerarse como el grado de similitud entre las características de los BP pertenecientes a un mismo grupo.

A continuación se describe el algoritmo que sirve como base para extender el modelo de búsqueda multimodal para hacer agrupaciones de BP. Posteriormente son descritas las mejoras realizadas a este algoritmo y su adaptación al modelo de búsqueda multimodal, haciendo énfasis en el cálculo de similitud entre BP, la forma de agrupación de los BP y el método de etiquetado de grupos, finalmente se presenta un ejemplo de agrupación de BP.

4.1 Algoritmo base

El algoritmo propuesto en esta tesis, está basado en la teoría de grafos y es una ampliación del algoritmo denominado BestStar desarrollado por Leandro Krug Wives en 1999 [91], para agrupar documentos. Originalmente el algoritmo tenía como propósito eliminar las desventajas que presentaba el algoritmo ***Star*** en el proceso de agrupación. Entre las desventajas del algoritmo ***Star*** están**:** un umbral mínimo de similitud entre los documentos pertenecientes a cada grupo, asignación de los documentos al primer grupo que cumpla con el valor del umbral y baja afinidad entre los documentos pertenecientes a cada grupo. Las mejoras planteadas en BestStar se basán, en que el usuario no tiene que establecer previamente un umbral de similitud, además realiza un análisis de los elementos ya asignados cada vez que se selecciona un nuevo centroide de grupo, para reasignar los documentos al grupo (estrella) de mayor similitud, según sea necesario, incluso si ya están agrupados; por lo tanto, la afinidad entre los documentos en cada grupo aumenta.

4.2 Adaptación del algoritmo base para agrupar bp

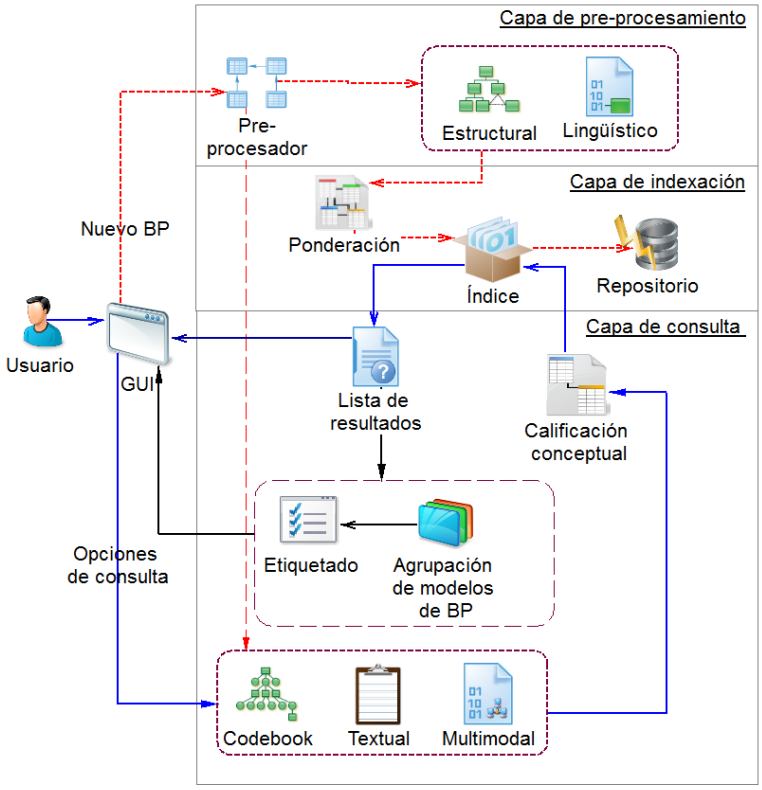
Debido a que BestStar fue inicialmente construido para agrupar documentos, en esta tesis se realizaron algunas mejoras al algoritmo, con la intención de agrupar BP. Entre las mejoras realizadas se encuentran:

- Adaptación del algoritmo para trabajar con BP. Utilizando información lingüística y estructural de los BP a agrupar.
- La agrupación es realizada bajo demanda, o sea sobre los resultados de una consulta específica.
- Los grupos son formados en tiempo de ejecución y solo representan una parte parcial del repositorio de BP.
- Implementación de un método de etiquetado de los grupos formados, el cual facilita la identificación del contenido de cada grupo.

La Figura 11, presenta la arquitectura del modelo de búsqueda multimodal extendido con el algoritmo de *clustering* propuesto (BestStarBP).

A continuación se describen los métodos ejecutados para el proceso de agrupación de BP, a saber: cálculo de similitud entre los BP de la lista de resultados, agrupación de BP y etiquetado de grupos.

Figura 11. Expansión del modelo de búsqueda multimodal para agrupación de BP.



4.2.1 Cálculo de similitud entre BP recuperados

Los BP retornados como relevantes para cada consulta, son utilizados para formar grupos con base en el nivel de similitud entre los términos (lingüísticos y estructurales) de cada uno de estos BP. Los niveles de similitud son calculados por medio de una función de lógica difusa (ecuación 3), y son almacenados en una matriz de similitud por BP (Mterc) representada en la Figura 12. Esta matriz en cada celda almacena al grado de similitud entre dos BP, es decir, qué tan similares son. Los valores almacenados en esta matriz se encuentran entre 0 y 1, teniendo las siguientes consideraciones, 0 sin similitud y 1 totalmente similar. Es importante tener presente que un BP es totalmente similar a sí mismo, por esta razón la diagonal principal de esta matriz contiene valores de 1, además la similitud de un BP_i con un BP_j es igual que la similitud de BP_j con BP_i, por esto la matriz es simétrica y la mitad inferior puede ser ignorada. En consecuencia, solo los elementos de la diagonal superior (ya que es un matriz triangular) son los utilizados para formar los grupos.

Figura 12. Matriz (Mterc) de similitud entre BP.

	BP1	BP2	BP3	BP4	BP5	BP6
BP1	***1***	**0.4**	**0.2**	**0.7**	**0.8**	**0.9**
BP2	0.4	***1***	**0.5**	**0.3**	**0.6**	**0.7**
BP3	0.2	0.5	***1***	**0.3**	**0.4**	**0.3**
BP4	0.7	0.3	0.3	***1***	**0.7**	**0.6**
BP5	0.8	0.6	0.4	0.7	***1***	**0.5**
BP6	0.9	0.7	0.3	0.6	0.5	***1***

En el cálculo de similitud, se toman cada uno de los BP que formarán la matriz Mterc, para determinar el grado de similitud (gs) entre los BP que pueden existir en un grupo; utilizando la ecuación (3), la cual está basada en la teoría de la lógica difusa.

$$\boldsymbol{gs}(\boldsymbol{X},\boldsymbol{Y}) = \left(\sum_{\boldsymbol{h}=\boldsymbol{1}}^{\boldsymbol{k}} \boldsymbol{gi}(\boldsymbol{a},\boldsymbol{b})\right)/\boldsymbol{n} \quad (3)$$

Donde *X* y *Y*, son vectores que representan cada uno de los BP, *k* es el número de elementos comunes entre ellos, *n* es el número de elementos presentes en *X* y *Y*, *gi* define el grado de igualdad (*gi*) entre los pesos del elemento h^{th} (*a* en *X* and *b* en *Y*). En la ecuación (4) se expone el (*gi*) representado en lógica difusa.

$$\boldsymbol{gi}(\boldsymbol{a},\boldsymbol{b}) = \mathbf{1/2}\ \left[(\boldsymbol{a} \rightarrow \boldsymbol{b}) \wedge (\boldsymbol{b} \rightarrow \boldsymbol{a}) + (\overline{\boldsymbol{a}} \rightarrow \overline{\boldsymbol{b}}) \wedge (\overline{\boldsymbol{b}} \rightarrow \overline{\boldsymbol{a}})\right] \quad (4)$$

En la ecuación anterior se tiene, $\mathbf{1} - \boldsymbol{x}; \boldsymbol{a} \rightarrow \boldsymbol{b} = \mathbf{max}\{\boldsymbol{c} \in [\mathbf{0},\mathbf{1}] \mid \boldsymbol{a} * \boldsymbol{c} \leq \boldsymbol{b}\}; \boldsymbol{and} \wedge = \boldsymbol{min}$. En esta ecuación un atributo puede tener diferentes grados de importancia en distintos vectores (BP). En este sentido, en lugar de calcular el promedio o el producto entre dos BP, la función determina el grado de igualdad entre ellos. Por lo tanto, cuando se ejecuta el cálculo de la similitud entre todos los BP, los valores de similitud calculados son almacenados en la matriz *Mterc,* como se muestra en la Figura 12.

4.2.2 Agrupación de modelos de BP

Una vez formada la matriz *Mterc*, se ejecuta el algoritmo de *clustering* (BestStartBP) con el propósito de formar los grupos que serán visualizados por los usuarios. El algoritmo 1 describe el proceso ejecutado para la formación de los grupos.

Para formar los grupos de BP el *algoritmo 1*, recibe como parámetro de entrada una matriz (*Mterc*) que contiene similitud de la información textual y estructural de los BP retornados como relevantes en una consulta. De esta matriz se toma cada BP y se

verifica si no está asignado a un grupo, con el fin de crear un nuevo grupo (líneas 1-3). Una vez creado un grupo con el *BP* actual asignado como centroide, es seleccionado otro BP y se comprueba el nivel de similitud, si el BP comprobado no está asignado aún, se agrega al grupo creado, de lo contrario se busca el grupo con mayor similitud y se adiciona (líneas 6-10). De esta forma son creados los grupos, utilizando como centroide de grupo un BP que aún no ha sido agrupado. Cuando un *BP* ya está asignado se obtiene la similitud del grupo original y se compara con el *BP* actual, con el propósito de crear un nuevo grupo (líneas 14 - 17). Si el nivel de similitud entre el BP actual y el BP asignado es mayor al del grupo original, entonces el BP ya asignado se elimina del grupo original y es agregado al nuevo grupo (Líneas 21-24). Finalmente los grupos que quedan con un solo BP son eliminados, y la formación de los grupos de *BP* es retornada para que el usuario la visualice (Líneas 29-30).

Pseudocódigo algoritmo de agrupación.

Algorithm 1. Algorithm for BP Group

Require: A Matrix (*Mterc*)

Ensure: A group create (G_{BP}) for the query *BP-Input*

```
1   for all BP_i ϵ Mtrec_L do
2   if BP_i not assigned then
3   star = CreatNnewCluter(BP_i)
4     for all BP_i ϵ Mtrec_J do
5       BP_j =getCurrentBP(Mtrec_J )
6       Similarity = getSimilarity(BP_i, BP_j)
7       if BP_j not in star then
8        addToSatr(BP_j)
9       Else
10       ClusterGreaterSimilarityFound(Similrity)
         AddToMostSimilarCluster(BP_j)
11      Endif
12    Endfor
```

```
13      AddG_BP(star)
14    Else
15    originalCluster= G_BP(i)
16     BPprevious = getBpOriginalCluster(0)
17     star = CreatNnewCluter()
18     similarityOriginal= getSimilarityOriginalCluster(BPprevious)
19     for all BP_i ∈ Mtrec_J do
20      BP_j =getCurrentBP(Mtrec_J )
21      Similarity = getSimilarity(BPprevious, BP_j)
22      if Similarity > similarityOriginal then
23       addToSatr(BP_j)
24       originalClusterRemove(BPprevious)
25      endIf
26     Endfor
27    Endif
28    Endfor
29    removeClusterWithOneBP()
30    return G_BP
```

4.2.3 Etiquetado

En los algoritmos de *clustering* basados en teoría de grafos, los grupos formados no están identificados con una etiqueta que defina su contenido. En consecuencia en esta tesis se adaptó un método de etiquetado basado en Suffix Arrays [92, 93], que identifica el contenido de cada grupo de BP formado, a fin de mejorar la interacción con el usuario. De este modo el usuario puede ver de qué trata el contenido del grupo o grupos a revisar (finalidad o funcionalidad de los BP).

El proceso de etiquetado inicia creando un resumen o snnipet (*S*) con los nombres de las tareas o actividades, las cuales definen la funcionalidad de cada uno de los BP del grupo a etiquetar. Posteriormente pre-procesa la cadena *S*, convirtiéndola a minúsculas, eliminado caracteres especiales, palabras vacías y, finalmente crea un

arreglo de sufijos *As*, ordenados lexicográficamente, con el propósito de encontrar la frase con mayor frecuencia en *S*, que identifique el contenido del grupo.

En el algoritmo, *S* es procesado como un conjunto de caracteres $S = \{s_1, s_2, s_3, s_n\}$, y se forma *S´ [i,j]* un arreglo de subcadenas de *S,* que van desde el índice *i* hasta el índice *j*. Posteriormente es creado un arreglo *As* de enteros*,* que contiene las posiciones iniciales de los sufijos de *S* en orden lexicográfico. Entonces As*[i]* almacena la posición inicial del *i*-esimo sufijo más pequeño perteneciente a *S.* Seguidamente se recorre el arreglo de subcadenas *S´* a través de una búsqueda binaria, para encontrar el sufijo más común y de longitud mayor, que comienza con ($) carácter de separación de términos, posteriormente el sufijo encontrado es retornado para etiquetar un grupo de BP.

4.3 Ejemplo del algoritmo

En el ejemplo, es utilizado el BP denominado *Activate service*, para ejecutar una consulta sobre el repositorio representado en la Figura 13, el modelo de búsqueda multimodal recupera los BP con mayor similitud al BP de consulta y genera una lista de resultados, tal como muestra la Figura 14. Posteriormente, los resultados son agrupados por el algoritmo BestStarBP, a partir de los niveles de similitud almacenados en la matriz (*Mterc*), los grupos de BP son formados a través de las reglas de relación implementadas en el algoritmo.

Ejecutando el algoritmo BestStarBP sobre la matriz de similitud representada en la Figura 12, los grupos formados serían:

- **Grupo 1**: BP1, BP4, BP5, BP6
- **Grupo 2:** BP2, BP3

La Figura 15, hace una representación gráfica de los grupos formados y los BP pertenecientes a cada grupo.

Figura 13. Ejemplo de repositorio para ejemplo agrupación.

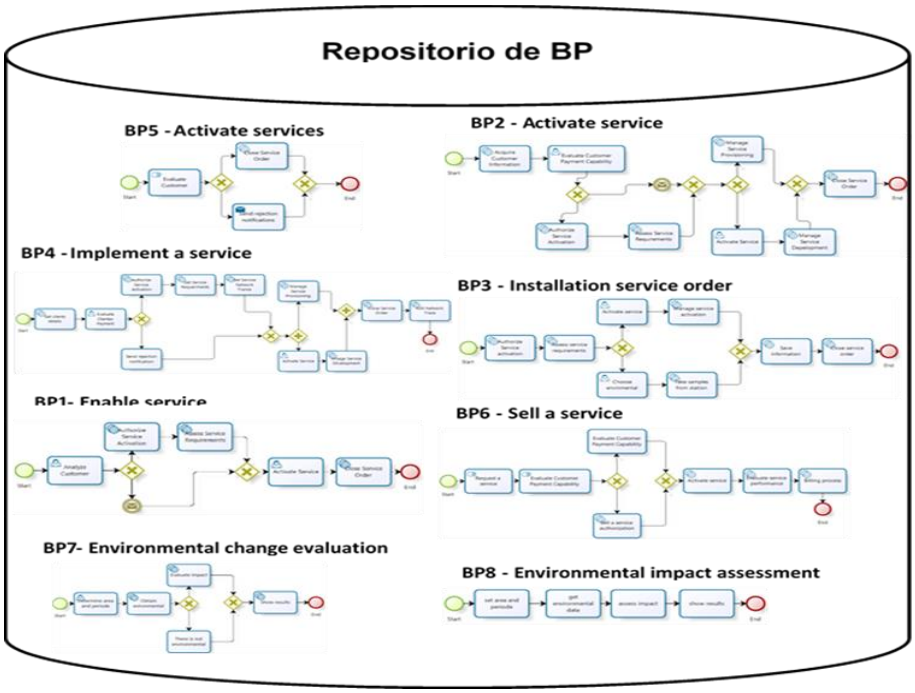


Figura 14. Lista de resultados generada para el ejemplo de agrupación.

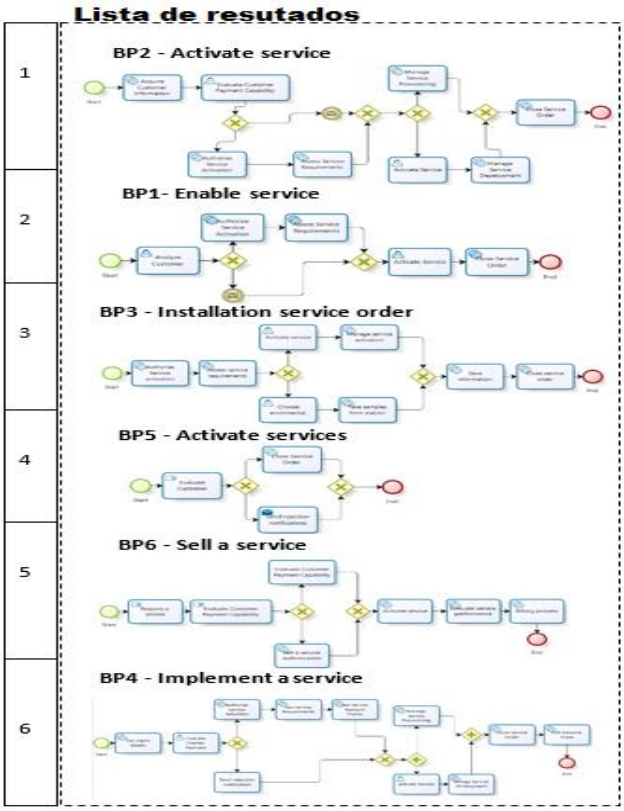
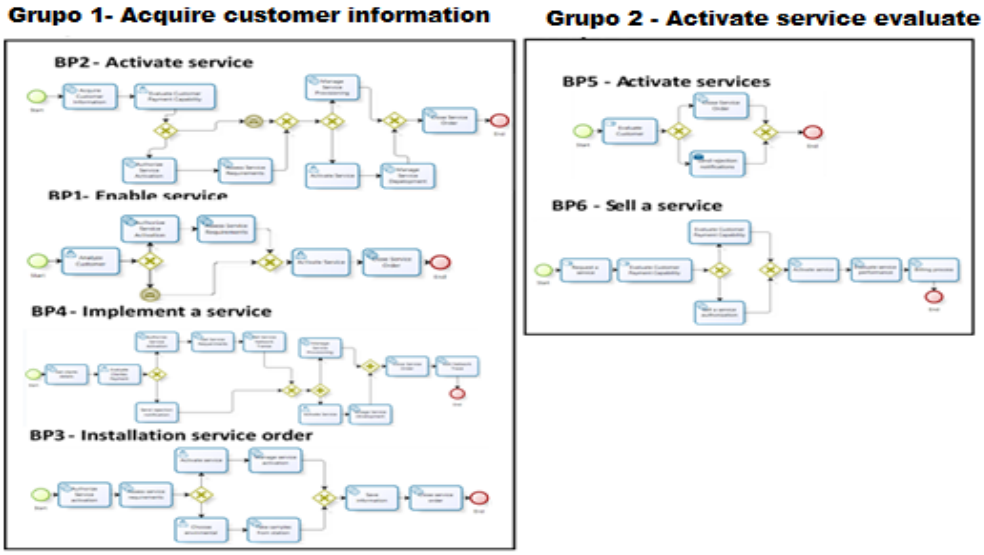
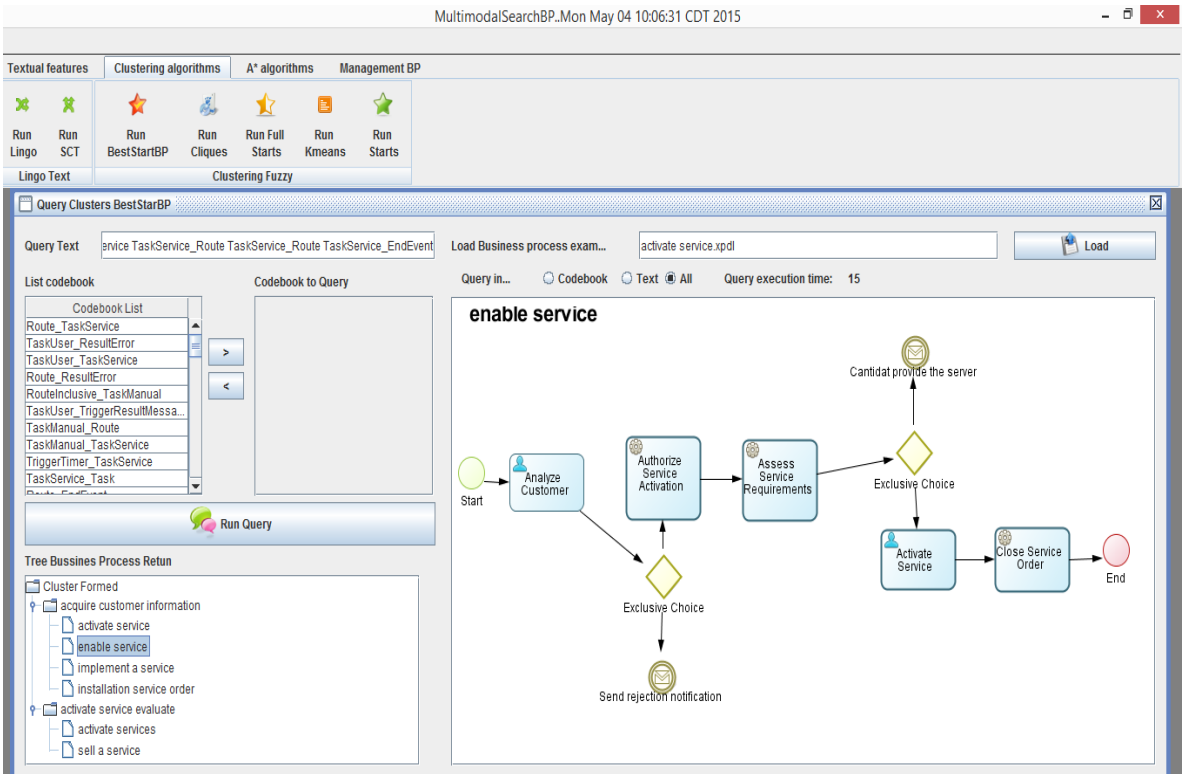


Figura 15. Grupos formado en el ejempló de agrupación.



Después de la formación de grupos inicia la fase de etiquetado, en esta se genera un resumen o Snippet con las etiquetas de los BP de cada grupo y el método de etiquetado encuentra la frase más representativa para el grupo. Finalmente, el usuario utiliza las etiquetas de los grupos para acceder a los BP de cada grupo, como se muestra en la Figura 16.

Figura 16. Interfaz de ejecución del algoritmo BestStarBP en la herramienta MultiSearchBP.



4.4 Conclusiones

La adaptación del algoritmo de *clustering* al modelo de búsqueda propuesto, permite organizar en forma de grupos los resultados entregados en cada una de las opciones de consulta. La función de similitud, basada en lógica difusa, permite definir el grado de similitud entre los BP pertenecientes a un mismo grupo por medio del número de elementos comunes entre ellos, los cuales pueden ser estructurales, textuales o ambas al mismo tiempo. El método de etiquetado le permite al usuario identificar con mayor claridad la información o funcionalidad contenida en cada uno de los BP

existentes en cada grupo, haciendo así que la revisión de cada grupo sea más coherente.

El algoritmo de *clustering* permite agrupar BP que representan diferentes funcionalidades para una misma línea de productos, esto hace posible evidenciar los cambios realizados a estos (BP), además, reduce el tiempo del análisis por parte de los ingenieros de procesos (usuarios del sistema de búsqueda de BP). Por otra parte, la agrupación permite a los ingenieros de procesos hacer algunas sugerencias sobre cómo rediseñar BP, con el fin de incorporar los cambios más frecuentes y significativos de una vez para todos los BP de un mismo grupo.

Capítulo 5
Prototipo y experimentación

Este capítulo presenta el prototipo que soporta el modelo de búsqueda y agrupación definido en los capítulos 3 y 4, la plataforma de evaluación para la creación del repositorio cerrado de BP, la plataforma para la formación manual de grupos y los resultados obtenidos en cada una de las fases de evaluación.

5.1 Introducción

La evaluación de un sistema de búsqueda y agrupación resulta de máxima importancia para garantizar su adecuado funcionamiento, es decir, la recuperación de información pertinente a una consulta realizada y una correcta adaptación a las necesidades de los usuarios. Durante el proceso de evaluación se realizaron actividades que incluyeron la evaluación de la estrategia de búsqueda planteada, la evaluación y aplicación de diferentes algoritmos de recuperación, la evaluación de ordenamiento y agrupación de los resultados según su relevancia o utilidad para el usuario. En este sentido, es de suma importancia el uso de herramientas (prototipos) que permiten hacer pruebas a través de la interacción con usuarios.

Para realizar de manera eficaz un proceso de evaluación de un sistema de recuperación de información es estrictamente necesario contar con:

- Una colección (repositorio) de ítems de información (fuente) de los que se obtuvieron las preguntas que se le plantearían al sistema.
- Una serie de juicios de relevancia de los ítem de información recuperados y expresados en varios niveles (muy relevante, relevante, algo relevante, poco relevante e irrelevante), según el tipo de información almacenada en la colección.
- Se usaron las medidas de exhaustividad y precisión para analizar los resultados.

Una vez ejecutada la tarea de evaluación, fueron obtenidos los valores de similitud entre la consulta y los resultados retornados por el sistema de búsqueda. Estos valores determinaron la calidad de los resultados alcanzados en la aplicación de las medidas de evaluación.

5.2 Plataforma de búsqueda

La plataforma construida para soportar el modelo de búsqueda definido en esta tesis, se denomina ***MultiSearchBP***, está implementada sobre la tecnología Java y soportada por una arquitectura organizada en 3 capas. 1) un nivel de presentación desde el cual el usuario puede gestionar los BP (adicionar, eliminar, modificar y buscar BP) almacenados en el repositorio y el índice, además de hacer las consultas de BP. 2) un nivel de lógica de negocio que se encarga de gestionar los BP, extraer las características estructurales y los componentes textuales de los BP e indexarlos, también responde a las opciones de búsqueda con dos tipos de respuesta: lista lineal ordenada de BP o grupos temáticos de BP que se relacionan con la consulta del usuario (diseñador) y finalmente, 3) un nivel de almacenamiento que se encarga de dar persistencia a los procesos de negocio y al índice de búsqueda. Esta herramienta fue presentada por Ordoñez, Corrales y Cobos [94]. Además en el Anexo 6, se hace una descripción de esta herramienta.

5.3 Plataforma para construcción colaborativa de una colección cerrada de prueba de bp

La plataforma se denominada ***CollaborativeGroupBP,*** fue presentada por Ordoñez y otros [53]. Es una plataforma web, desarrollada bajo la tecnología java y PostgreSQL (RDBMS) para el almacenamiento de la información gestionada en la construcción del repositorio cerrado y la formación de los grupos de BP. La funcionalidad en la plataforma está distribuida en dos módulos, definición de resultados relevantes y formación colaborativa de grupos. El anexo 3, presenta la descripción de la plataforma.

5.4 Evaluación

Para determinar la calidad del modelo propuesto fue necesario someterlo a un proceso de evaluación experimental, con el objetivo de verificar la eficiencia en el proceso de búsqueda y agrupación de BP con base en el patrón de similitud definido para las opciones de consulta que permite el modelo. El proceso de evaluación fue desarrollado en dos fases, que a su vez se subdividieron en varias actividades:

Fase 1: Evaluación de la lista ordenada de resultados, contempló las siguientes actividades: Afinamiento del *Codebook*, Evaluación de la mejor opción de consulta

(textual, estructural o multimodal) y Comparación con otros mecanismos de búsqueda identificados en el estado de arte.

Fase 2: Evaluación de la agrupación de BP, contempló la evaluación y comparación del algoritmo propuesto con otros algoritmos identificados en el estado del arte, usando medidas internas y medidas externas.

5.4.1 Objetivos de la evaluación

El proceso de evaluación planteado en esta tesis trazó como objetivo general: permitir al modelador de BP aumentar la capacidad de gestión para el re-modelamiento de los BP contenidos un repositorio, a través de un modelo que utiliza información textual y estructural para la búsqueda y agrupación de BP. Para lograrlo se plantearon los siguientes objetivos específicos:

- Establecer qué longitud de *codebook* (número de componentes del BP interconectados secuencialmente), permite obtener resultados más relevantes en las consultas ejecutadas sobre el índice multimodal.
- Definir la opción de búsqueda que obtenga mejores resultados en consultas ejecutadas sobre el índice multimodal.
- Debido a que no existe de libre acceso, en esta investigación se decidió crear una colección cerrada de pruebas (repositorio), construido colaborativamente, que contenga una lista predefinida de consultas y sus respectivas respuestas ideales a cada una de las consultas definidas. Además, una formación ideal de grupos realizada a partir de los resultados considerados como relevantes en cada consulta. Teniendo presente que un sistema de búsqueda o recuperación de información no puede evaluarse sobre la base de la eficacia de apreciaciones individuales.
- Comparar el modelo de búsqueda multimodal de BP propuesto con otros métodos de búsqueda de BP identificados en el estado del arte.
- Realizar la comparación sobre la colección de prueba (repositorio) creada, para determinar la calidad de los resultados de búsqueda con base en la exhaustividad y la precisión. En el uso de las medidas se asumió que los resultados ofrecidos están ordenados según su relevancia respecto a la consulta planteada.

- Evaluar el rendimiento en el proceso de búsqueda de BP, con base en el tiempo empleado por el modelo multimodal para recuperar BP, con diferente número de elementos (nodos).
- Evaluar la agrupación del modelo de búsqueda propuesto, en relación a la similitud de los grupos formados manualmente por los expertos evaluadores frente a los grupos formados automáticamente por el modelo de búsqueda propuesto.

Los objetivos de la evaluación se plantearon con el propósito de definir un proceso eficaz de representación del conocimiento, el cual pueda conseguir en alto nivel de efectividad en las funciones de búsqueda y agrupación automática de BP, especificadas en el modelo de búsqueda propuesto en esta tesis.

La descripción del proceso de evaluación está organizada de la siguiente manera: la sección 5.4.2 describe la fase de evaluación de la lista de resultados, la sección 5.4.2.1 presenta las medidas de relevancia utilizadas para en el proceso de evaluación de la lista de resultados, la 5.4.2.2 presenta la definición del mejor N componente estructural de *Codebook*, 5.4.2.3 presenta la definición de la mejor opción de búsqueda en el modelo multimodal, la sección 5.4.2.4 describe la evaluación comparativa con otros modelos de búsqueda, la 5.4.2.5 presenta la evaluación del rendimiento. La evaluación de agrupación es presentada en la sección 5.4.3, en la sección 5.4.3.1 son descritos los algoritmos analizados para evaluación de resultados, la sección 5.4.3.2 describe la evaluación interna de la agrupación, en la sección 5.4.3.3 son presentados los resultados de la evaluación interna, la sección 5.4.3.4 describe la evaluación externa, en la cual son comparados los grupos formados manualmente frente a los grupos formados por el modelo propuesto, y finalmente en la sección 5.4.3.5 son presentados los resultados de la evaluación externa.

5.4.2 Evaluación de lista de resultados

La evaluación de la relevancia de los resultados contempla tres fases con un objetivo bien definido. La primera, se ejecuta para calcular el mejor componente estructural del *codebook* (valor de N componentes) que permita alcanzar mayor relevancia y calidad en la lista de resultados de búsqueda; La segunda, busca determinar cuál es la mejor opción de búsqueda entre (lingüística, *codebook*, y

multimodal); y la tercera compara los resultados generados por la mejor opción de búsqueda de la etapa anterior con los resultados generados con otras herramientas de búsqueda de BP identificadas en el estado del arte.

La evaluación de la lista se basó en la estimación de la relevancia de los resultados generados por el modelo multimodal. La experimentación fue realizada con base en el modelo de pertinencia presentado en Ordoñez y otros [84], el cual permite a un grupo de expertos en el campo de gestión de BP emitir juicios de similitud entre un conjunto de modelos de BP de entornos reales y un subconjunto de 6 BP como consultas. Los juicios de similitud obtenidos en esa plataforma fueron considerados como BP relevantes y utilizados para estimar la relevancia de los resultados del modelo multimodal.

Dentro de la evaluación experimental, se utilizó el modelo multimodal para generar listas de resultados, en las cuales se consideraron listas de 6, 8, 10, 15, y 20 primeros BP del repositorio de acuerdo a su similitud con el BP de consulta. Con el propósito de evaluar la relevancia de los resultados obtenidos en la ejecución de cada búsqueda, a partir de la aplicación de medidas ampliamente empleadas en la evaluación de sistemas de recuperación de información [95].

5.4.2.1 Medidas para la evaluación de la relevancia

Las medidas para la evaluación calculan el nivel de la relevancia de los resultados de una herramienta de búsqueda de BP en forma decreciente, gradual, y continúa. Miden la ganancia gradual de relevancia de un ítem de la lista de resultados basada en la posición del ítem dentro de la lista, teniendo presente que los BP de mayor relevancia son útiles si aparecen en las primeras posiciones de la lista de resultados [96].

Para lo anterior fueron aplicadas las medidas de relevancia gradadas (*Pg y Rg*) [96], que proporcionan una clasificación (T_i) de los BP del repositorio retornados los cuales son considerados similares a un BP de consulta (Q) de acuerdo con diferentes niveles de relevancia. Pg y Rg tienen en cuenta la suma total de grados de relevancia entre los BP de la lista de resultados. Por otra parte, para medir la calidad de la lista de resultados generados por el modelo multimodal se utilizaron las medidas ANDCG (*Average Normalized Discounted Cumulated Gain*) y GenAveP' (*Generalized Average Precision*) las cuales fueron presentadas y mejoradas en el

trabajo de Küster and König-Ries [96]. Estas medidas cuantifican la calidad de la lista de resultados producidos por las herramientas de recuperación de servicios Web, pero son plenamente aplicables al campo de búsqueda de BP. Estas medidas son descritas a continuación.

- **Precisión gradada (Pg):** estima la capacidad del sistema para recuperar solo elementos relevantes (es decir, aquellos elementos considerados como similares a una consulta según los evaluadores expertos), evitando los elementos no relevantes (es decir, falsos positivos), relacionando los valores mínimos entre la herramienta automática y los resultados de los evaluadores expertos con la sumatoria del total de los resultados. Ecuación (5)

$$\mathbf{P_g} = \frac{\sum \mathbf{T_i} \in \mathbf{T}\ \mathbf{min}\{\mathbf{f_r}(\mathbf{Q},\mathbf{T_i}),\mathbf{f_e}(\mathbf{Q},\mathbf{T_i})\}}{\sum \mathbf{T_i} \in \mathbf{T} \mathbf{f_e}(\mathbf{Q},\mathbf{T_i})} \quad (5)$$

- **Recuerdo gradada (Rg):** evalúa la capacidad del sistema de recuperar la mayor cantidad de elementos relevantes, con relación a aquellos elementos considerados como relevantes por los expertos. Para esto, realiza la sumatoria de todos los valores mínimos de relevancia encontrados en la lista de resultados generados por una herramienta automática (f_e) y los resultados de los expertos (f_r). Ecuación (6)

$$\mathbf{R_g} = \frac{\sum \mathbf{T_i} \in \mathbf{T}\ \mathbf{min}\{\mathbf{f_r}(\mathbf{Q},\mathbf{T_i}),\mathbf{f_e}(\mathbf{Q},\mathbf{T_i})\}}{\sum \mathbf{T_i} \in \mathbf{T} \mathbf{f_r}(\mathbf{Q},\mathbf{T_i})} \quad (6)$$

- **Medida F gradada:** evalúa la combinación armónica entre Pg y Rg, el cual determina el porcentaje de los documentos verdaderamente relevantes recuperados. Ecuación (7)

$$\mathbf{Mf_g} = \frac{\mathbf{2} * \mathbf{R_g} * \mathbf{P_g}}{\mathbf{R_g} + \mathbf{P_g}} \quad (7)$$

- **ANDCG y GenAvep':** Las medidas ANDCG y GenAvep' a diferencia de la precisión y recuerdo que miden la calidad de los resultados en términos del número de elementos relevantes obtenidos, tienen en cuenta la calidad de la lista de resultados generados por la herramienta, en este caso si una herramienta entrega más elementos relevantes en la parte superior de la lista de resultados entonces será mejor calificada.

Las ecuaciones (8) y (9), presentan las medidas ANDCG y $\mathrm{GenAveP}'$ dónde $CG(i) = \sum_{j=1}^{i} g(r_j)$ es la ganancia acumulada en la lista de resultados *i*, es decir, la ganancia (g) que una herramienta automática asigna a los primeros *i* ítems en la lista. La medida $\mathrm{ICG(i)}$ evalúa la ganancia que un usuario asigna a los primeros i ítems en la lista (en este caso corresponden a los elementos considerados como relevantes). El $DCG(i) = \sum_{j=1}^{i} \frac{g(i)}{disc(i)}$ es similar al CG pero utiliza un factor de descuento $disc(i)$ el cual asigna mayor valor a los primeros elementos y reduce el valor a los últimos elementos de la lista de resultados. En esta tesis el factor de descuento que se utilizó es: $disc(i) = \max(1, log_b i)$.

$$\mathbf{ANDCG} = \frac{1}{|\mathrm{R}|}\sum_{\mathrm{i=1}}^{|\mathrm{L}|}\frac{DCG(i)}{IDCG(i)} \quad (8)$$

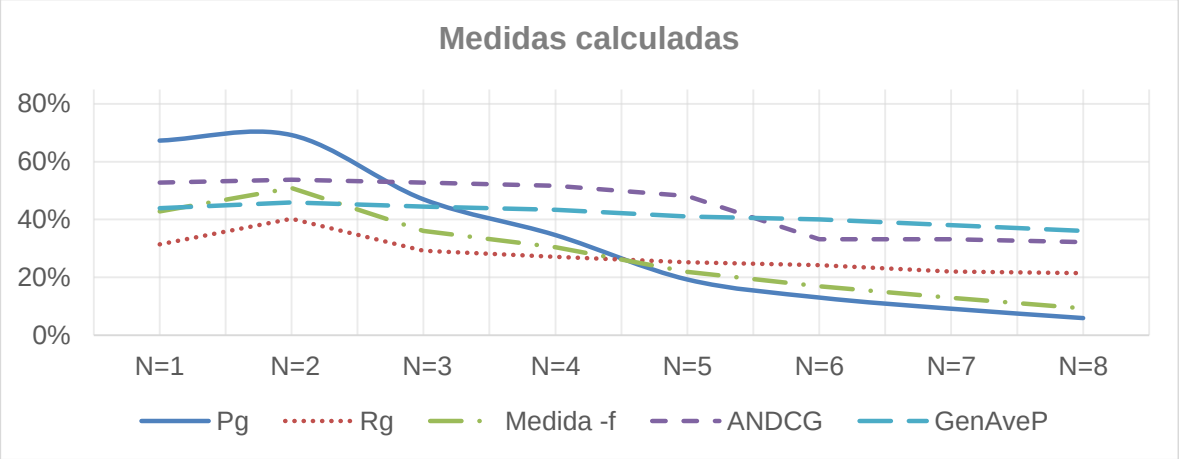
$$\mathbf{GenAveP'} = \frac{\sum_{\mathrm{i=1}}^{|\mathrm{L}|}\frac{CG(i)}{i}}{\sum_{\mathrm{i=1}}^{|\mathrm{R}|}\frac{ICG(i)}{i}} \quad (9)$$

5.4.2.2 Cálculo del mejor N componente estructural del codebook

En este proceso, se evaluó la formación de *codebook* para determinar cuál es el valor estructural del *codebook* (N-componente) que permita tener mejores resultados de acuerdo con las medidas de relevancia a aplicar en la evaluación del modelo de búsqueda propuesto para con el valor de N formar el modelo multimodal. Las evaluaciones fueron realizadas por intervalos teniendo presente listas de resultados de 8, 10, 15 y 20 BP retornados por la herramienta desarrollada para soportar el modelo de búsqueda multimodal.

Para la evaluación de la búsqueda por *codebook*, se realizaron varias consultas desplegando listas de resultados con los números de ítems mencionados anteriormente, además los *codebook* fueron construidos por secuencias de componentes estructurales con valores de N = 1 hasta N= 8. La Figura 17 muestra los promedios porcentuales de cada valor de N, obtenidos en las diferentes listas de resultados de esta evaluación.

Figura 17. Valores obtenidos en el refinamiento del Codebook.



La Figura 17 permite observar que a medida que el número de componentes (*N*) del *codebook* aumenta, se reducen considerablemente los valores de *Pg* y *Rg*, y en consecuencia la Medida F; alcanzado un ligero equilibro entre los *codebook* de *N*=5 a *N*=8. En este caso los valores más altos en estas medidas (69%, 40%, y 51% respectivamente) se reportan en el componente *N*=2.

En cuanto a las medidas ANDCG y GenAveP', se puede observar que mantienen mayor estabilidad que las medidas anteriores, con una pequeña caída al 38 de ANDCG en n=6. Lo cual permite concluir que la calidad de la lista de resultados presenta cierta independencia del valor de *N* definido para los *codebooks* manteniendo una calidad promedio de 55%.

En relación con el refinamiento del *codebook*, se concluyó que las secuencias de dos componentes (N=2) representan de mejor manera la semántica de comportamiento de los BP para la ejecución de búsquedas, en este sentido el *codebook* formado por secuencias de dos componentes (nodos) obtuvo resultados con los mejores valores en cada una de las medidas evaluadas. Cabe señalar que los BP recuperados en las consultas realizadas para el refinamiento se asemejan en la forma que operan o se ejecutan, aspecto importante para el usuario que realiza la consulta.

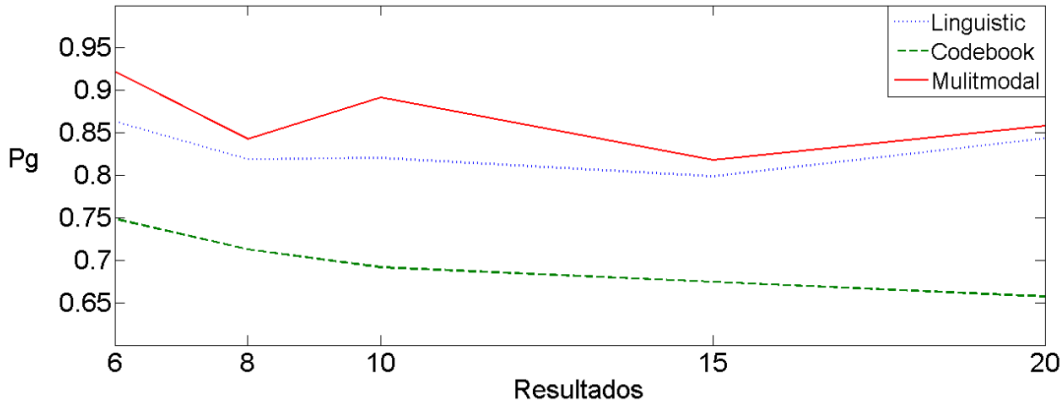
5.4.2.3 Definición de la mejor opción de búsqueda

Esta sección presenta la evaluación de relevancia de los resultados obtenidos por el modelo multimodal en cada una de las opciones de búsqueda que este ofrece. Para

esto se tiene en cuenta que el mejor componente *codebook* del modelo es N=2, conforme lo presentado en la sección anterior. Las opciones de consulta evaluadas fueron: búsqueda lingüística, búsqueda por *codebook*, y búsqueda multimodal (la cual corresponde a la combinación de las dos anteriores). A continuación se presentan los resultados para cada medida de relevancia.

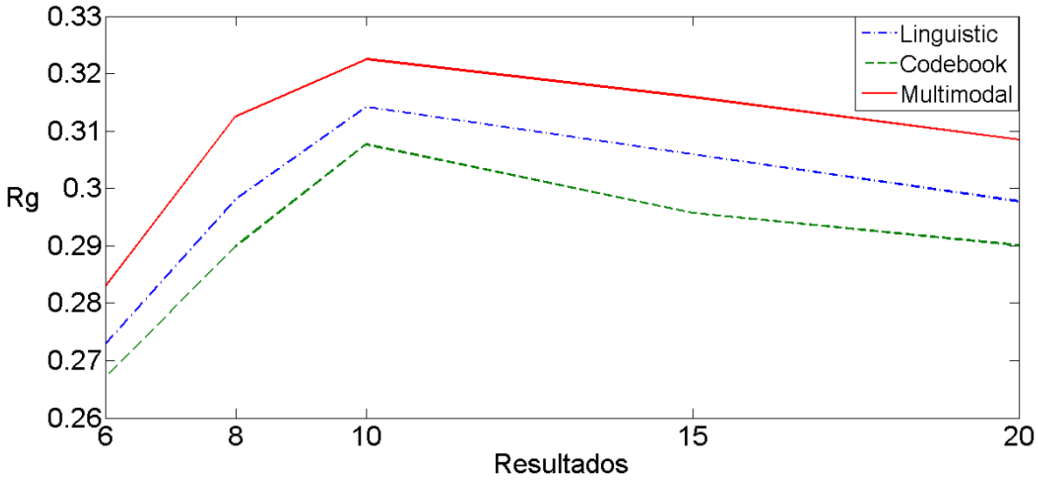
La Figura 18 presenta los resultados obtenidos para listas con 6 a 20 ítems de resultados en cada una de las opciones de consulta del modelo multimodal. En los resultados obtenidos, las listas con 6 resultados presentan los mayores niveles de Pg, en este caso la opción lingüística alcanzó el 86,3%; la de *codebook* el 74,86%; y la multimodal obtuvo 92,24%. Esto demuestra que el modelo multimodal que integra las opciones lingüística y *codebook* incrementa el nivel de Pg a 92,24% reportando tan solo un 8% de falsos positivos (es decir, BP no relevantes recuperados).

Figura 18. Pg. Mejor opción de búsqueda.



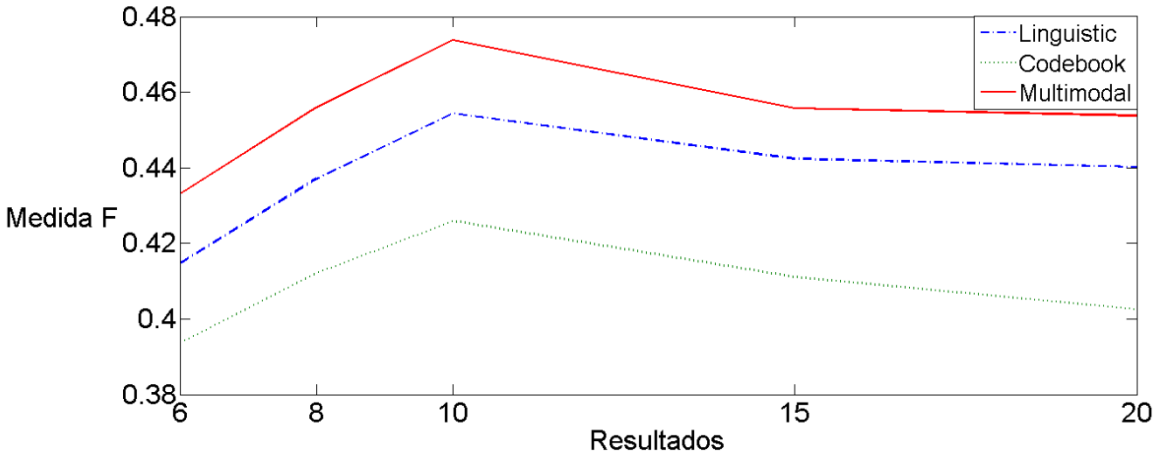
La Figura 19 despliega los niveles de Rg para cada uno de las opciones del modelo multimodal. Se puede observar que para cada una de las opciones de consulta las listas de resultados con 10 items obtienen los niveles más altos de Rg, 31.41% para la opción lingüística, 30,74% para *codebook*, y 32,25% para multimodal. Estos resultados permiten evidenciar que la opción multimodal obtiene el menor número de falsos negativos (67,75%), es decir, BP relevantes que no fueron recuperados por las otras opciones en cada consulta.

Figura 19. Rg. Mejor opción de búsqueda.



La Figura 20, presenta los resultados de la evaluación con la Medida F, en estos se observa que al igual que el Rg la lista de resultados con 10 ítems obtiene mejores niveles de armonía entre Pg y Rg para cada una de las opciones del modelo, un 45,43% para la opción lingüística, 42,59% para *codebook*, y 46,63% para multimodal. De lo cual se puede concluir que la opción de búsqueda que logra mayor armonía es la multimodal, debido a que alcanzó los mejores valores en las dos medidas anteriores (Pg y Rg).

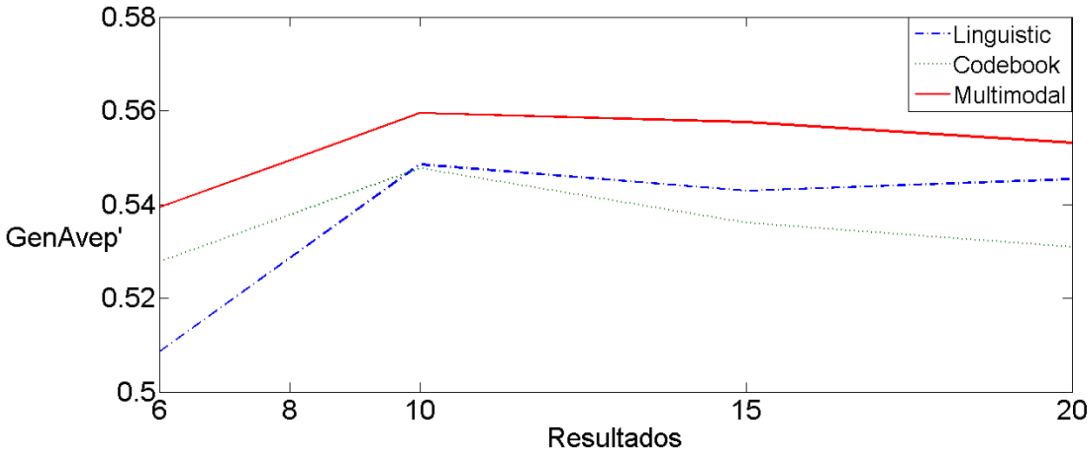
Figura 20. Medida F. Mejor opción de búsqueda.



La Figura 21 presenta los resultados de la medida GenAveP', en la cual se puede observar que la opción multimodal presenta valores ligeramente mayores que las

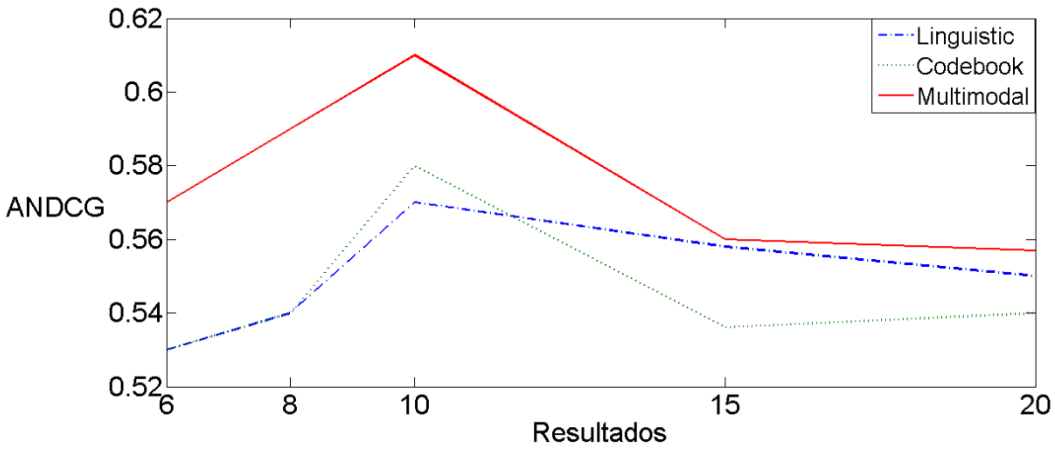
otras dos opciones, indicando que esta opción ofrece una mejor calidad en la lista de resultados, debido que posiciona mayor número de ítems (BP) relevantes en las primeras posiciones de la lista de resultados.

Figura 21. GenAvep'. Mejor opción de búsqueda.



En la Figura 22 se muestran los resultados de la medida ANDCG la cual, como se explicó anteriormente, es muy similar a la medida GenAveP' en el sentido de que mide la calidad de la lista de resultados. Sin embargo, en esta figura se puede observar que los resultados de ANDCG difieren de los resultados de GenAveP', debido que la primera utiliza un factor de descuento que penaliza los resultados obtenidos al final de la lista. De esta manera la Figura 22 permite evidenciar, que la opción de consulta multimodal presenta mejor calidad en la lista de resultados, clasificando mayor cantidad de elementos (BP) relevantes en la parte superior de la lista, a diferencia de las opciones de consulta lingüística y *codebook* por separado.

Figura 22. ANDCG. Mejor opción de búsqueda.



Por otra parte, cabe destacar que al igual que en las gráficas para Rg, Medida-f, ANDCG y GenAveP', el mejor nivel de precisión del modelo de búsqueda multimodal se obtuvo en listas de resultados con 10 ítems, debido que la mayoría de los ítems (BP) recuperados tienen alto grado de relevancia. Por lo tanto, la evaluación comparativa con otros modelos toma este número de ítems para la lista de resultados.

5.4.2.4 Evaluación comparativa con otros modelos de búsqueda

En esta tesis se evalúo el modelo multimodal propuesto comparando la relevancia de sus resultados con dos métodos similares de búsqueda de BP, el primero corresponde a una herramienta denominada "BeMantics" (*Behavioral Semantics Business Process Retrieval*) [97], la cual permite ejecutar consultas a través de características semánticas, estructurales, y de comportamiento; mientras el segundo corresponde a una implementación del algoritmo A* [98], el cual hace la búsqueda teniendo en cuenta el componente estructural.

Para la comparación del modelo multimodal con la herramienta BeMantics y el algoritmo A* se utilizó el mismo conjunto de prueba creado por Ordoñez y otros [84], y las consultas que se ejecutaron permitieron comparar la opción multimodal del modelo propuesto contra las características estructurales de los otros dos métodos, debido a que estos presentaron los mejores valores de relevancia, según los

resultados mostrados en las publicaciones en que fueron presentadas estas herramientas (BeMantics [97], A* [98]).

- *BeMantics:* está compuesto por dos módulos principales: un módulo de indexación basado en semántica del comportamiento, y un módulo de análisis estructural y semántico.

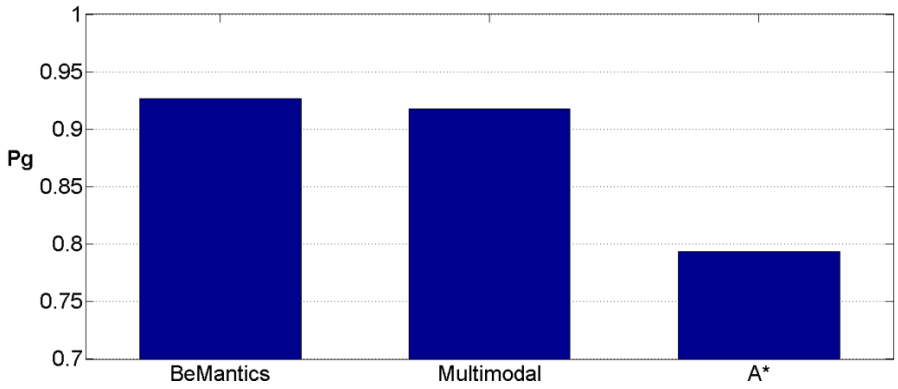
El módulo indexador busca sub-estructuras dentro de un BP de consulta representado en un documento BPMO (Business Process Modeling Ontology). Dicho documento es transformado a un modelo formal basado en grafos y entregado al módulo Control-flow patterns detector, el cual encuentra un conjunto de patrones de control de flujo en el BP de consulta y procede a recuperar una lista de BP del repositorio (BP store) que contengan un conjunto similar de patrones (pre-ranked graphs).

A continuación la lista de grafos pre-ranked es ingresada al analizador estructural y semántico el cual realiza un conjunto de operaciones de edición (substituir o eliminar nodos, eliminar o adicionar aristas) con el fin de hacer que los grafos pre-ranked provenientes del repositorio sean lo más similares posible al grafo de consulta. Los valores de las operaciones de edición son predefinidos por los usuarios, pero en el caso específico de la operación de substitución de nodo, el costo se calcula a través de un analizador lingüístico que calcula un valor de distancia léxica o distancia semántica. La distancia léxica es calculada a través de la similitud entre palabras definida en una base de datos léxica llamada WordNet [99]; la distancia semántica es calculada contando el número de saltos entre conceptos de una ontología de dominio que contiene conceptos del entorno de las telecomunicaciones.

- **A***: El algoritmo está enfocado en hacer comparaciones estructurales entre grafos para encontrar un nivel de similitud entre estos, para este proceso el algoritmo realiza los siguientes pasos:
- Dado un conjunto de grafos (G1….Gn) y un grafo de entrada de consulta Gq. El algoritmo inicia con la descomposición recursiva fuera de línea de cada uno de los grafos en pequeños subgrafos hasta que estos representen un solo vértice. Todos estos subgrafos son almacenados en una estructura de datos compacta, con el propósito de hacer menor el tiempo de ejecución.
- La similitud es calculada tomando cada uno de los subgrafos coincidentes para formar la medida de distancia la cual está dada por el subgrafo máximo común entre un grafo del conjunto y el de consulta.

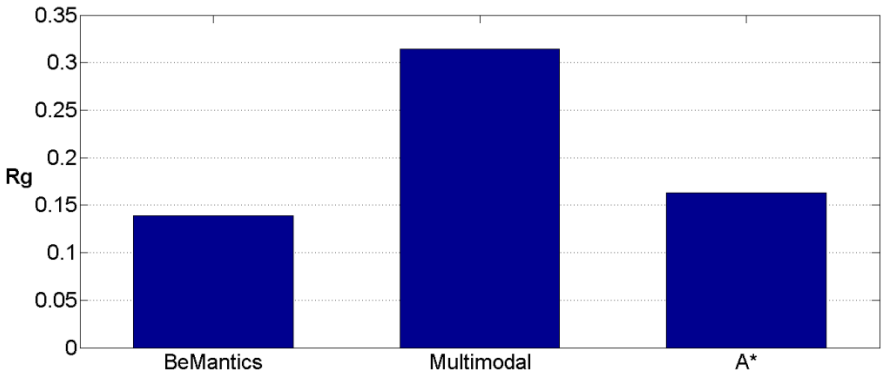
Comparando la precisión gradada (Figura 23), BeMantics obtuvo el valor más alto de precisión promedio (92,85%) lo cual indica que redujo el número de falsos positivos a tan solo un 7,15%. Esto es debido a que BeMantics realiza una comparación entre cada nodo del BP consulta, con cada nodo del repositorio de BP utilizando un algoritmo de isomorfismo de grafos el cual le permite obtener mayores valores de precisión. Sin embargo, el modelo Multimodal obtiene valores similares en los resultados combinando los criterios estructurales y lingüísticos presentes en los BP, los cuales son procesados mediante algoritmos de extracción de texto, siendo capaz de reducir la probabilidad de obtener resultados irrelevantes (falsos positivos) al 8%. Por otra parte, A* obtiene el menor nivel debido que solo utiliza algoritmos de isomorfismo de grafos como función de similitud.

Figura 23. PG. Para BeMantics, A* y Multimodal.



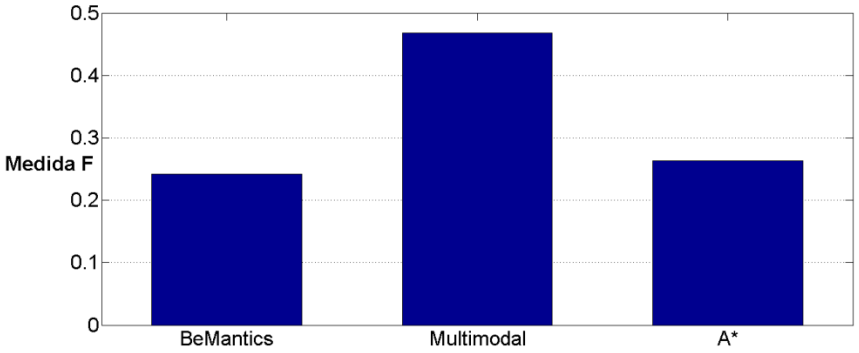
En cuanto a la medida de Recuerdo gradado (Figura 24), las herramientas muestran valores bajos 13,9% para BeMantics, 16,6% para A* y 30,6% para Multimodal. Aunque estos valores son bajos Multimodal obtiene un valor que supera en 250% a BeMantics y 100% a A*. Esto se debe a que las tres herramientas tienen listas de resultados limitadas, con un máximo de diez BP en concordancia a que muchas aplicaciones de IR, especialmente en entornos web, donde los usuarios sólo se centran en la revisión de los primeros diez o quince resultados devueltos en el conjunto de respuestas [100, 101]. Por lo tanto, BeMantics, A* y Multimodal pueden obtener falsos negativos (perder BP relevantes en la lista de resultados), pero al mismo tiempo aumentar la precisión reduciendo el número de falsos positivos.

Figura 24 . Rg. Para BeMantics, A* y Multimodal.



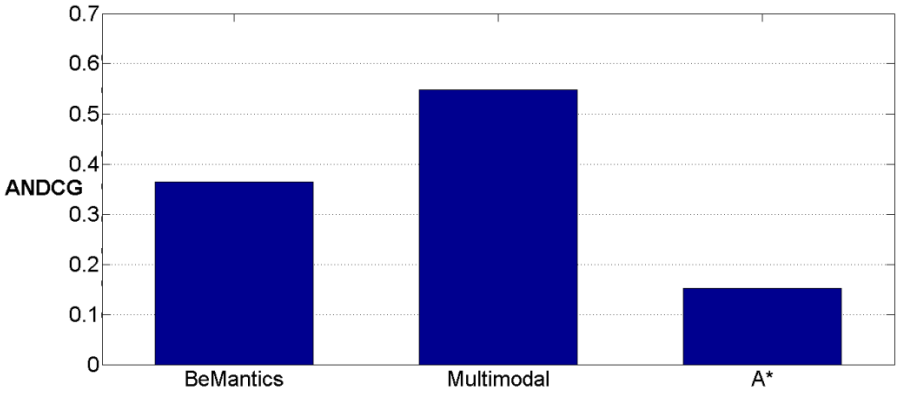
En cuanto a la efectividad de las herramientas, estas son caracterizadas por el rendimiento de la clasificación en las listas de resultados, para esto la Medida F (Figura 25Figura 25) permite observar la armonía de los resultados de Pg y Rg. Las herramientas obtuvieron los siguientes valores: BeMantics alcanzó 24,14% y A* el 26,31%, lo que demuestra que el orden y la calidad de los resultados poseen grados bajos de armonía. En este sentido, la comparación semántica y estructural nodo a nodo entre el BP consulta y los BP existentes en el repositorio que realizan los algoritmos de BeMantics, hace que aumente la precisión, pero disminuya notablemente el recuerdo, haciendo así, que la armonía de los resultados baje considerablemente. En relación a la baja armonía presentada por A*, esta se presenta porque A* realiza únicamente comparación estructural nodo a nodo entre el BP consulta y los BP del repositorio, a través de un algoritmo de isomorfismo de grafos. Por otra parte, el 46,81% que alcanzó Multimodal evidencia que el modelo propuesto obtiene mejor rendimiento en la clasificación de los resultados recuperados en las consultas realizadas, debido que MultiModal pondera elementos lingüísticos y estructurales compartidos entre el BP de consulta y los BP del repositorio, a través de algoritmos de extracción de texto. En promedio Multimodal logra una mejora de 191% con relación a BeMantics y de 176% a A*, en la presentación de los resultados usando una “lista ordenada de resultados”.

Figura 25. Medida F. Para BeMantics, A* y Multimodal.



La Figura 26 muestra la medida ANDCG, en la cual se puede observar que el modelo multimodal posee una mejor calidad en la lista de resultados con respecto a BeMantics y A* los cuales no logran ubicar tantos elementos relevantes al inicio de la lista de resultados como lo hace el modelo multimodal el cual obtiene una diferencia de 150% con BeMantics y 360% con A*.

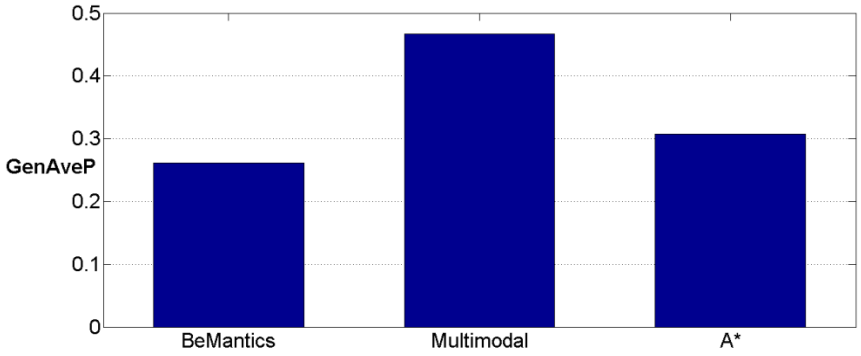
Figura 26. ANDCG. Para BeMantics, A* y Multimodal.



La diferencia en la medida ANDCG y GenAveP' (Figura 27) es que esta última posee un factor de descuento que penaliza con mayor valor a los elementos retornados al final de la lista de resultados, en este caso al igual que el anterior es el modelo multimodal el que obtiene los valores más altos que oscilan entre 260% con relación a BeMantics y 153% a A*. Esto implica que los algoritmos de extracción de texto implementados en MultiModal, ordena y ubican de mejor manera un mayor númerode elementos (BP) relevantes en las primeras posiciones de la lista de resultados, a diferencia de los algoritmos de isomorfismo de grafos implementados

en Bemantics y A*, que ubican un menor número de ítems relevantes en las primeras posiciones, generando listas poco ordenadas con ítems dispersos en ellas.

Figura 27. GenAvep'. Para BeMantics, A* y Multimodal.



5.4.2.5 Rendimiento

Comparando el análisis de rendimiento ejecutado en los modelos para búsqueda de BP, se encontró que Multimodal presentó el mejor tiempo de respuesta para todos los BP (con diferente cantidad de nodos), siendo 588 veces más rápido que BeMantics y 1083 veces más rápido que A* para un BP de consulta con 10 nodos. Con respecto a BP de consulta con 20 a 30 nodos, Multimodal es 2500 veces más rápido que BeMantics y 482 veces más rápido que A*. El número de veces es calculado con base en el tiempo (Tabla 9) empleado en la ejecución de una consulta con un BP con determinado número de nodos.

Tabla 9. MultiModal vs BeMantics y A* – Comparación y análisis de rendimiento.

Herramienta	10 Nodos	20 Nodos	30 Nodos
	Tiempo (ms)	Tiempo (ms)	Tiempo (ms)
BeMantics	10000	100000	1000000
A*	18410	19260	285000
Multimodal	17	40	60

Fuente: elaboración propia.

Lo anterior es debido a que BeMantics ejecuta una comparación estructural exhaustiva y basada en un mecanismo de corrección de errores que busca hacer un BP del repositorio tan similar como sea posible al BP de consulta, A* realiza análisis estructural basado en subgrafos para encontrar los BP que más subgrafos tenga para recuperarlos, mientras que el enfoque Multimodal ejecuta un algoritmo de

extracción de texto simple que toma unos pocos milisegundos para recuperar los resultados (17 - 60 ms) en BP con 10 - 30 nodos.

5.4.3 Evaluación de la agrupación

Medir el rendimiento de una agrupación o *clustering* no es una tarea trivial, debido a que no existe una metodología estándar para esto. En este sentido el proceso de evaluación de *clustering* se basa en la utilización de diversas métricas que permiten hacer evaluación de la calidad interna y externa [102, 103] de los grupos formados.

Con base en lo anterior, para realizar la validación de la presente tesis, fue realizado un proceso de evaluación que contempla dos fases, i) *Evaluación interna,* donde se aplican métricas que miden la proximidad de los grupos formados por el método propuesto (BestStartBP) y los algoritmos que se describen en la siguiente sección y ii) *Evaluación externa,* donde se compararon los grupos formados por el método propuesto contra los grupos formados colaborativamente por un conjunto de jueces evaluadores, la cual es considerada como una agrupación “ideal”. Esta agrupación fue presentada por Ordoñez y otros [53].

5.4.3.1 Algoritmos utilizados en el proceso de evaluación de agrupación

En esta sección son descritos los algoritmos de agrupamiento que se analizaron para evaluar el algoritmo de *clustering* BestStartBP, propuesto en esta tesis, el cual fue implementado en la herramienta MultiSearchBP para el proceso de evaluación. Entre los algoritmos se encuentran: Lingo [104] y STC (Sufix Tree *Clustering*) [105], algoritmos para agrupación de documentos web, K-means [106], algoritmo para la agrupación de datos y Stars, Cliques, FullStart [91], algoritmos basados en teoría de grafos. Adicionalmente estos algoritmos fueron adaptados para realizar agrupamiento de BP con base en los resultados entregados por el modelo de búsqueda multimodal previamente explicado.

- **STC**

Toma cada BP y extrae el texto de todos sus componentes, y crea una secuencia sintáctica ordenada de términos textuales para generar el agrupamiento de los BP basado en la información extraída. Este algoritmo consta de dos pasos:

  - **Paso 1:** identificar grupos base. En este paso el algoritmo crea un árbol de sufijos a partir del vector que contiene todos los componentes textuales de

los BP. A partir de este vector detecta una raíz de tal manera que se garantiza que cada nodo contiene al menos dos hijos internos (un par). Luego, las aristas entre nodos se etiquetan con una parte del texto resumen con el propósito de formar la etiqueta de dicho nodo.

- **Paso 2:** combinar grupos base. En este paso el algoritmo asigna una clasificación a cada grupo base, teniendo en cuenta el número de BP que el grupo contiene y que están relacionados con una serie de elementos textuales. Para esto se usa una función de grupo base (s(B)), en la cual está contemplado un grupo B con elementos P así: s(B) = |B| * f(|P|) , donde |B| es el número de BP en el grupo base B; |P| es el número de elementos en P que no tienen calificación 0 (es decir que estén conectados a, al menos, algún nodo del árbol de sufijos); f es una función que penaliza los P de un solo elemento.

- **LINGO**

Este algoritmo, construye un resumen (Snippet) con los términos textuales contenidos en cada componente de los BP retornados por una consulta. El algoritmo consta de cuatro pasos principales.

- **Paso 1:** la extracción de características. Aquí se identifican frases o términos que pueden ser candidatos para etiquetas de grupo. Esto se realiza calculando el número de veces que los términos o frases identificadas aparecen en los BP contenidos en la lista de resultados generados en una consulta.
- **Paso 2:** inducción de etiquetas de grupo. Este paso forma descripciones significativas de los grupos tomando la información de la matriz de términos por cada BP.
- **Paso 3:** identificación de los BP pertenecientes a cada grupo. Compara fragmentos de texto con todas y cada una de las etiquetas de grupo. Para esto se forma una matriz Q en la que cada etiqueta de grupo se representa como un vector columna. De forma que $C=Q^{T}A$, donde A es el termino original de la matriz de término BP. De esta manera, el elemento $c_{i,j}$ de la matriz C indica el peso de adhesión del BP j en el grupo i.
- **Paso 4:** formación final de grupos de BP. En este paso se calcula la ponderación de la etiqueta dependiendo del número de veces que los

términos de la etiqueta aparecen en cada uno de los BP asignados al grupo identificado por dicha etiqueta.

- **K-Means**

En este algoritmo se debe especificar de antemano el número de grupos de BP (k-clusters) a formar. Para ello, se seleccionan k BP aleatoriamente, que representarán el centroide o media de cada grupo de BP. Posteriormente cada uno de los BP existentes en la lista de resultados, es asignado al centroide del grupo más cercano de acuerdo con una función de distancia (la más utilizada es la euclidiana). Para cada uno de los grupos formados se calcula el centroide de todas sus BP. Estos centroides son tomados como los nuevos centros de sus respectivos grupos. A continuación se describe el algoritmo K-Means:

  - **Paso 1:** elegir k BP que actúan como centroides (k determina el número de grupos de BP a formar).
  - **Paso 2:** cada BP, es añadido al grupo con mayor similitud o cercano.
  - **Paso 3:** calcular el centroide de cada grupo de BP, con el propósito de convertirse en los nuevos centroides.
  - **Paso 4:** si no se llega a un criterio de convergencia (por ejemplo, dos iteraciones no cambian las clasificaciones de los grupos de BP), se vuelve al paso 2).

- **Stars**

El algoritmo es ejecutado iterativamente, en cada iteración selecciona un BP de la lista de resultados para formar un nuevo grupo, seguidamente evalúa todos los BP restantes para encontrar los más similares (el algoritmo considera que un BP es similar a otro cuando su nivel de similitud es mayor o igual a un umbral mínimo de aceptación previamente definido) para adicionarlos al grupo creado. De esta forma, en cada iteración ejecutada el grupo formado es representado como un grafo, en el cual el BP inicial referencia el nodo principal (central) y los BP que conforman el grupo, están conectados al nodo principal por medio de aristas, formando una figura muy similar a una estrella (de ahí el nombre de estrellas). El algoritmo ejecuta los siguientes pasos:

  - **Paso 1:** seleccionar un BP que no es parte de ningún grupo y crear un nuevo Grupo.

- **Paso 2:** Introducir en el grupo creado todos los BP con mayor similitud posible.
- **Paso 3:** Si todavía existen BP sin asignar a un grupo, repite los pasos 1 y 2.

- **Clique**

Este algoritmo es ejecutado de forma iterativa, en cada iteración todos los BP recuperados se comparan entre sí, con el propósito de asignarlos a un mismo grupo dependiendo de la similitud entre ellos. Un BP es similar a otro, si el grado de similitud es mayor que o igual al factor de aceptación mínimo definido (umbral) previamente. Los pasos del algoritmo son:

- **Paso 1:** selecciona un BP que no es parte de ningún grupo y crea un nuevo grupo.
- **Paso 2:** selecciona un nuevo BP y lo compara con el BP del paso anterior.
- **Paso 3:** si el BP seleccionado es mayor o similar a todos los BP del grupo actual, se añade a ese grupo.
- **Paso 4:** si el BP seleccionado, ya está asignado en algún grupo, vuelve al paso 2.
- **Paso 5:** mientras existan BP sin asignar a un grupo, vuelve al paso 1

- **FullStart**

En la agrupación, este algoritmo, considera los BP que ya están agrupados, y asigna los BP a todos los grupos donde la similitud es mayor o igual con el umbral establecido, sin sacarlos de los grupos donde ya hacen parte. Este algoritmo ejecuta los siguientes pasos.

- **Paso 1:** selecciona un BP que no es parte de ningún grupo y lo añade a un nuevo grupo.
- **Paso 2:** selecciona un nuevo BP y lo agrega a todos los grupos con los cuales es mayor o similar.
- **Paso 3:** realiza los pasos 1 y 2 hasta que se asignan todos los BP en al menos un grupo.

5.4.3.2 Evaluación interna

La evaluación interna determina cuáles grupos son mejores que otros, haciendo énfasis en la separación entre grupos y la cercanía o similitud de los elementos

pertenecientes a cada grupo. Además, la evaluación interna mide densidad, distancias entre los objetos en los mismos grupos (si son más pequeñas, los grupos son más compactos) y separación entre los grupos (distancias más grandes indican mayor separación, que se considera mejor) [107].

Como punto importante en la validación de la tesis, se realizó un estudio de la calidad de la solución, aplicando algunas métricas de análisis interno para agrupaciones que no requieren intervención humana. Las métricas utilizadas son descritas a continuación.

- **Cohesión:** expresa el promedio de similitud entre los elementos de un grupo, cuanto mayor sea el grado de similitud el grupo será más cohesionado. (Ecuación 10).

$$\mathbf{cohesion}(\mathbf{c}) = \frac{\sum_{\mathbf{i>j}} \mathbf{sim}(\mathbf{c_i},\mathbf{c_j})}{\frac{\mathbf{m(m-1)}}{\mathbf{2}}} \quad (10)$$

Donde sim (c_i,c_j) es el grado de similitud entre los elementos c_i y c_j existentes en el grupo *C*, y *m* es el número de elementos existentes en el grupo.

- **Acoplamiento:** se utiliza para expresar la similitud media entre todos los pares de elementos, donde un elemento pertenece al grupo *C* y el otro no. Idealmente, el acoplamiento debe ser bajo, ya que este valor comprueba qué tan cercanos están los elementos dentro de cada grupo. (Ecuación 11).

$$\boldsymbol{coupling}(\boldsymbol{c}) = \frac{\sum_{i,j} sim(c_i,q_j)}{m*n} \quad (11)$$

Donde *sim* (c_i, q_j) es la similitud entre el elemento c_i del grupo *C* y el elemento q_j de otro grupo; *m* es el número de elementos en *C* y *n* es el número de elementos externos a *C*.

- ***Silueta***: se deriva de la cohesión y el acoplamiento, muestra cuáles elementos están bien ubicados dentro del grupo y cuáles tienen una posición intermedia entre el grupo al que fueron asignados y los otros grupos, cuanto mayor sea el grado de silueta mejor será la distribución de los grupos, es decir, los grupos tendrán mayor separación entre sí. (Ecuación 12).

$$\boldsymbol{silhouette}(\boldsymbol{i}) = \frac{b_i - a_i}{\max(a_i,b_i)} \quad (12)$$

Donde a_i es la disimilitud promedio (es decir, la distancia) entre el i^{th} elemento del grupo y los otros objetos del mismo grupo, y b_i es la disimilitud promedio mínimo entre el i^{th} elemento de cualquier grupo que no contiene el elemento.

- ***Suma de cuadrados entre grupos "Sum of squares between clusters" (SSB):*** permite medir la separación entre grupos, donde *k* es el número de grupos, n_j es el número de elementos en el grupo *j*, c_j es el centroide del grupo *j* y *x* es la media del repositorio (data set). (Ecuación 13).

$$\boldsymbol{SSB} = \sum_{i=1}^{k} n_j \, dist^2(c_j - \bar{x}) \quad (13)$$

- ***Suma de cuadrados dentro de grupo "Sum-of-squares within cluster" (SSW)***: mide internamente la cohesión de los grupos formados, con base en el nivel de similitud entre los elementos existentes en cada grupo. (Ecuación 14).

$$\mathbf{SSW} = \sum_{\mathbf{i=1}}^{\mathbf{k}} \sum_{\mathbf{x} \in \mathbf{c_i}} \mathbf{dist^2}(\mathbf{m_i}, \mathbf{x}) \quad (14)$$

Donde *k* es el número de grupos, *x* un punto del grupo c_i, y m_i es el centroide del grupo c_i.

- ***Índice Davies-Bouldin (DB)****:* Mide la dispersión de los elementos dentro de los grupos formados y la separación entre ellos. Valores pequeños para el índice DB indican grupos compactos, cuyos centros están bien separados los unos de los otros. Consecuentemente, la agrupación con menor valor de índice DB se toma como la óptima. (Ecuación 15).

$$\boldsymbol{DB} = \frac{1}{k} \sum_{i=1, i \neq j}^{k} max\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right) \quad (15)$$

En la ecuación anterior, *k* es el número de grupos, σ_i es la distancia promedio entre cada elemento del grupo *i* con el centroide del grupo *i*, σ_j es la distancia promedio entre cada elemento del grupo *j* con el centroide del grupo *j*, y $(c_{i,j})$ es la distancia entre los centroides de los dos grupos.

5.4.3.3 Resultados de evaluación interna

A continuación, la Tabla 10 presenta los resultados promedio obtenidos por cada algoritmo en cada una de las medidas evaluadas.

Tabla 10. Resultados evaluación interna de agrupación.
(Ch para Cohesión, Ac para Acoplamiento, Si para Silueta, DB para índice Davies-Bouldin)

Algoritmo	Ch	Ac	Si	SSB	SSW	DB
MultiSearchBP	**1,382**	**0,167**	**0,80**	0,**330**	**0,004**	**0,167**
Cliques	1,088	0,283	0,56	0,133	0,069	0,913
FullStars	1,035	0,726	0,25	0,138	0,087	0,725
Stars	0,948	0,274	-0,008	0,282	0,063	0,920
K-means	0,620	0,246	0,097	0,051	0,048	0,583
Lingo	0,147	0,340	-0,043	0,129	0,070	0,527
STC	0,164	0,676	-0,036	0,148	0,079	0,817

Fuente: elaboración propia.

En relación a la ***Cohesión*** MultiSearchBP obtiene el valor más alto con diferencia a los demás algoritmos. MultiSearchBP mejora en promedio 51,73 % en relación con los algoritmos evaluados. La mejora se debe a que MultiSearchBP no permite solapamiento, es decir, elementos que existan en varios grupos al mismo tiempo, expresando así que los grupos formados por MultiSearchBP contienen elementos con más cercanía o similitud entre ellos, además MultiSearchBP tiene la ventaja, de re-agrupar los BP al grupo con mayor peso de similitud, aun cuando ya han sido agrupados, en consecuencia, el nivel de cercanía de los elementos (BP) dentro de los grupos formados aumenta.

En cuanto al ***Acoplamiento***, MultiSearchBP obtiene el menor valor, alcanzando una mejora promedio de 45%, en relación a los algoritmos evaluados. Esto se debe a que los elementos considerados como similares entre los grupos formados comparten relativamente un subconjunto de características comunes, las cuales pueden ser textuales o estructurales, características no tenidas en cuenta por los demás algoritmos, debido a que estos se enfocan en un solo tipo de información (textual) para realizar la agrupación.

El 0,8 alcanzado por MultiSearchBP en el coeficiente ***silueta*** expresa que la agrupación es buena, además permite corroborar los promedios de mejora en las medidas anteriores (Cohesión y Acoplamiento). Esto demuestra que los grupos formados por MultiSearchBP contienen elementos bien ubicados, generando así grupos mejor distribuidos. Por lo tanto, el valor menor del coeficiente silueta para el resto de algoritmos, significa que los grupos formados contienen elementos

intermedios, es decir elementos que pertenecen a varios grupos al mismo tiempo, esto hace que los elementos al interior de los grupos estén dispersos o tengan baja similitud.

Para el ***SSB***, el promedio mayor (0,330), es alcanzado por MultiSearchBP, la separación de los grupos formados es buena, dado que los elementos (BP) son asignados al grupo con mayor similitud. Por esta razón, es baja la presencia de elementos (BP) intermedios al interior de los grupos formados. Lo anterior, se debe también a que el número de grupos formados y la cantidad de elementos por grupo son proporcionales al número de ítems recuperados en la lista de resultados.

Respecto a ***SSW***, la variación de los elementos entre los grupos formados por MultiSearchBP es baja, por lo tanto, los elementos existentes en cada grupo deben compartir información textual y estructural en el grupo de mayor similitud a ellos, generando así grupos con mayor cohesión. Por otra parte, el índice ***DB***, expresa que la agrupación generada por MultiSearchBP, contiene elementos que están bien ubicados dentro de cada grupo, dicho de otra manera no están dispersos con base en la información compartida entre ellos, haciendo así que la agrupación pueda considerarse óptima.

Para aumentar la eficacia en la agrupación MultiSearchBP, cuenta con la ventaja de la eliminación del solapamiento (elementos que puedan existir en varios grupos al mismo tiempo) y la reasignación de los elementos a un grupo con mayor similitud, características determinantes en la calidad de los grupos formados. Como se observa en los resultados de la Tabla 10, la calidad de los grupos formados por los algoritmos que permiten solapamiento es baja, debido a que producen demasiados grupos con duplicación excesiva de los elementos pertenecientes en cada grupo. Por consiguiente, los grupos formados por estos algoritmos son poco cohesionados, es decir, contienen elementos dispersos con bajo nivel de similitud entre ellos, por esta razón los grupos son poco distantes o estrechamente semejantes. Motivo por el cual la agrupación es poco eficaz.

5.4.3.4 Evaluación externa

La evaluación externa es utilizada cuando se cuenta *a priori* con una formación "ideal" de grupos de datos. Es decir, se conocen las clases (o categorías) de los objetos de datos, que van a ser comparados con los grupos creados por el algoritmo que se pretende evaluar. En consecuencia, la validación externa es más precisa que

la interna. Este es un tipo de validación especialmente importante, cuando se trata de encontrar el mejor método de agrupación para una tarea específica y por lo general implica la comparación de una variedad de algoritmos sobre un repositorio (dataset) específico [108].

Una medida externa evalúa la calidad de una agrupación mediante la comparación de los grupos producidos por una técnica de agrupación automática, contra los grupos generados anteriormente en una etapa de formación realizada por usuarios expertos en la temática inmersa en los datos. Como métricas de evaluación externa en la fase de comparación con usuarios, fueron utilizadas las medidas Precisión ponderada, Recuerdo ponderado y Medida F ponderada (mide la armonía entre precisión y recuerdo), tomadas desde el campo de recuperación de información [109, 110].

Para evaluar precisión ponderada, recuerdo ponderado y medida –F ponderada, tomamos la formación de grupos {C_1, C_2…..C_k}, generada automáticamente por MultiSearchBP y la comparamos con la colección ideal de grupos $\{C_1^i, C_2^i, \ldots . C_h^i\}$ generada colaborativamente por 56 usuarios expertos, presentada por Ordoñez y Otros. [53]. En la evaluación se ejecutaron los siguientes pasos: (a) encontrar para cada grupo ideal C_n^i, el grupo distinto C_m que más se aproxime en la colección que se está evaluado, y evaluar $P(C, C^i)$, definida en la ecuación (16), $R(C, C^i)$, determinada por la ecuación (17), $F(C, C^i)$, especificada en la ecuación (18). (b) Calcular la precisión ponderada, recuerdo ponderado y medida F ponderada usando la ecuación (19).

$$P(C, C^i) = \frac{|C \cap C^i|}{|C|} \quad (16)$$

$$R(C, C^i) = \frac{|C \cap C^i|}{|C^i|} \quad (17)$$

$$F(C, C^i) = \frac{2P(C,C^i)R(C,C^i)}{P(C,C^i)+R(C,C^i)} \quad (18)$$

$$P = \frac{1}{T} \sum_{j=1}^{h} |C_j^i| \, P(C_m, C_j^i); \quad (19)$$

$$R = \frac{1}{T} \sum_{j=1}^{h} |C_j^i| \, R(C_m, C_j^i);$$

$$F = \frac{2PR}{P+R} ;$$

$T = \sum_{j=1}^{h} |C_j^i|$.

En la ecuación (19), *C* es un grupo de BP, C^i es un grupo ideal de BP.

- Número de grupos formados (**Ng**): mide la cantidad de grupos formados, con base en el número de resultados retornados como relevantes en cada consulta realizada por un usuario.
- Número de elementos por grupo (**Ne**): mide el número de elementos al interior de cada grupo formado

5.4.3.5 Resultados evaluación externa

En esta fase, los grupos formados por MultiSearchBP y los algoritmos evaluados fueron comparados con los grupos formados manualmente por un grupo de expertos. En el proceso de formación manual de grupos se ejecutaron las mismas 6 consultas de la fase de evaluación previa. Con los resultados considerados como relevantes por los evaluadores en cada consulta se formaron los grupos y asignaron sus etiquetas. A continuación, la Tabla 11 presenta en forma general los resultados de los grupos formados por los expertos en cada consulta.

Tabla 11. Elementos relevantes y grupos formados por consulta.

Consulta	Elementos relevantes	Grupos formados	Elementos por grupo
Q1	10	3	3,33
Q2	12	4	3
Q3	13	3	4,33
Q4	11	3	3,66
Q5	14	4	3,5
Q6	14	4	3,5

Fuente: elaboración propia.

La Tabla 12, expone los promedios alcanzados de Precisión, Recuerdo y Medida F, en la evaluación de los grupos formados por MultiSearchBP y los algoritmos evaluados, en comparación a los grupos formados manualmente por los expertos (evaluación externa). En relación con la precisión, los mejores valores son reportados por MultiSearchBP (0,801), seguido, en segundo lugar, por Lingo (0,502) y STC (0,401), en tercer lugar, el resto de algoritmos mantiene un promedio entre (0,34 y 0,38). En general, MultiSearchBP aumenta la precisión en 30% con relación a Lingo

y 40% con STC y 42% en relación al resto de algoritmos. El promedio de precisión permite definir que los grupos generados por MultiSearchBP tienen estrecha similitud con los grupos formados manualmente (“ideales”), debido al número elevado de elementos compartidos. Por otra parte, la combinación de información estructural y textual utilizada para realizar la agrupación por parte de MultiSearchBP, permite crear grupos con mayor similitud a los grupos formados por los expertos, quienes tuvieron en cuenta varios tipos de información presente en los BP para realizar la agrupación. En este contexto, Lingo y STC obtienen valores elevados debido a la cantidad de grupos formados y el número de elementos por grupo, esto hace que los ítems considerados como relevantes existan de alguna forma en uno o varios grupos.

En relación con el recuerdo, los mejores valores son alcanzados por MultiSearchBP logra (0,638), STC que reporta el segundo mejor promedio (0,581) y K-means este algoritmo obtiene el tercer mejor valor con (0,424), el resto de algoritmos mantienen un promedio de recuerdo que oscila entre (0,134 y 0,30). En consecuencia, MultiSearchBP tiene un aumento de 6% con relación a STC, 19% de aumento en relación con K-Means y un promedio de 59% en relación al resto de algoritmos. El valor del recuerdo alcanzado demuestra que algunos de los elementos en los grupos formados por MultiSearchBP, se encuentran dispersos en los grupos de la formación manual. Además, la eliminación de factores como el solapamiento, un valor de umbral y el número de grupos a formar permiten que MultiSearchBP reduzca el valor de falsos negativos (FN) es decir aquellos elementos del grupo j que fueron ubicados en un grupo diferente al que indicaba su etiqueta. La cantidad de grupos formados por STC y el número de elementos por grupo, hacen que este algoritmo aumente el valor de verdaderos positivos (VP), es decir, aquellos elementos que fueron ubicados por el algoritmo en el mismo grupo que indicaba la agrupación manual realizada por los expertos, sin embargo también aumenta el valor de los falsos negativos (FN), como resultado, el valor de recuerdo disminuye un poco. Por otra parte, el valor de los grupos a formar y el número de iteraciones a realizar es un factor determinante para K-means, en consecuencia, la precisión disminuye, debido que el número de falsos positivos aumenta (FP), es decir, aquellos elementos que fueron ubicados por el algoritmo en el grupo j, pero en la formación manual realizada por los expertos pertenecen a otro grupo.

Los valores de Medida F más altos, al igual que en la medida anterior, son obtenidos por MultiSearchBP que logra un 23% más que STC y un 33% más que K-means. El

(0,7065) de Medida-f de MultiSearchBP determina el rendimiento de la agrupación realizada por el método propuesto. Esto permite definir que los grupos creados son relevantes y coinciden en alto grado con los grupos formados manualmente por los expertos en un ambiente colaborativo.

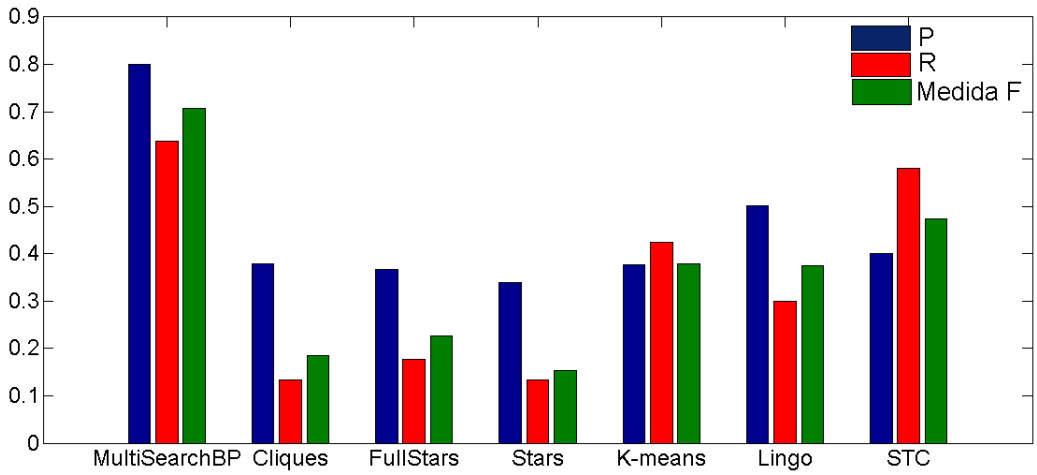
Tabla 12. Promedios de Precisión, Recuerdo y Medida-F en la evaluación externa.

Algoritmo	Medida	Q1	Q2	Q3	Q4	Q5	Q6	PROMEDIO	Nc	Ne
MultiSearchBP	P	**0,857**	0,786	**0,764**	**0,730**	**0,844**	**0,823**	**0,801**	6,3	**2,9**
	R	**0,714**	0,614	**0,543**	**0,763**	**0,610**	**0,581**	**0,638**		
	Medida-F	**0,779**	0,690	**0,635**	**0,746**	**0,708**	**0,681**	**0,707**		
Cliques	P	0,684	0,385	0,632	0,018	0,373	0,183	0,379	2,3	6,9
	R	0,158	0,121	0,174	0,036	0,132	0,175	0,133		
	Medida-F	0,257	0,184	0,272	0,024	0,195	0,179	0,185		
FullStars	P	0,449	0,385	0,621	0,030	0,441	0,267	0,366	19,6	17,9
	R	0,241	0,158	0,174	0,071	0,173	0,250	0,178		
	Medida-F	0,314	0,224	0,271	0,042	0,248	0,258	0,226		
Stars	P	0,449	0,385	0,632	0,018	0,373	0,183	0,340	**4,0**	8,4
	R	0,168	0,121	0,174	0,036	0,132	0,175	0,134		
	Medida-F	0,245	0,184	0,272	0,024	0,195	0,179	0,153		
K-means	P	0,759	0,450	0,292	0,037	0,421	0,303	0,377	**3,0**	6,3
	R	0,462	0,400	0,338	0,083	0,583	0,675	**0,424**		
	Medida-F	0,575	0,424	0,313	0,051	0,489	0,418	0,**378**		
Lingo	P	0,583	0,483	0,411	0,487	0,502	0,545	0,**502**	12,3	2,6
	R	0,279	0,300	0,235	0,346	0,331	0,309	0,300		
	Medida-F	0,378	0,370	0,299	0,405	0,399	0,395	0,374		
STC	P	0,405	0,341	0,338	0,432	0,441	0,446	0,401	12,0	7,3
	R	0,530	0,565	0,533	0,708	0,523	0,628	0,**581**		
	Medida-F	0,459	0,426	0,413	0,537	0,478	0,522	0,**473**		

Fuente: elaboración propia.

La Figura 28, presenta los resultados de la evaluación externa según el tipo de algoritmo. En estos resultados, el algoritmo basado en teoría de grafos que mejores resultados género fue FullStars, el cual logra un rendimiento (Medida F) de 0,2263 en relación con la formación manual de los expertos, esto es debido a que FullStar permite solapamiento y genera un considerable número de grupos, los cuales contienen elementos comunes entre ellos, una de las desventajas presentes en estos algoritmos es el valor de umbral para determinar la similitud de los elementos para asignar a cada grupo, esto hace que en la agrupación se dejen de lado algunos elementos que pueden ser relevantes para cada grupo (FN). Por otra parte, el algoritmo para agrupación de documentos web que mejores resultados reportó fue STC con 0,4725 de rendimiento, ya que al igual que FullStar permite solapamiento y genera un número considerable de grupos con elementos comunes entre sí. Aunque este algoritmo no requiere de un umbral o del número de grupos a formar, se basa en un árbol de sufijos construido a partir de la información textual presente en los BP, lo que hace que agrupe elementos que comparten solo este tipo de información. En relación a K-means algoritmo para agrupación de datos, el rendimiento alcanzado sobre la formación manual fue 0,3783, aunque K-means no permite solapamiento, el valor de (K) grupos a formar hace que el número de VP disminuya, debido a este parámetro.

Figura 28. Promedios alcanzados en la evaluación manual.



Finalmente, los valores de Ng y Ne obtenidos por MultiSearchBP, demuestran que los grupos y el número de elementos pertenecientes a cada grupo, son

proporcionales con el número de resultados retornados como relevantes en cada una consulta.

5.5 Certeza de la investigación

Para evaluar la significación estadística de los resultados mostrados en la Tabla 13, se aplicaron los test no paramétricos de Friedman (Average rankings) y de Wilcoxon (Signed Ranks) sobre los resultados obtenidos por los algoritmos en cada una de las consultas. El test de Friedman generó una clasificación ordenada ascendentemente según el valor promedio de rendimiento, considerando una distribución chi-cuadrado con 6 grados de libertad. La Tabla 13, muestra los resultados obtenidos para la Precisión, el recuerdo y la medida F, e incluye el valor del estadístico de Friedman y el valor P para cada prueba. La clasificación de los algoritmos según el test de Friedman sobre los valores de Recuerdo y Medida F, demuestra que la mejor agrupación de los resultados la obtienen los algoritmos MultiSearchBP, en primer lugar, STC en segundo lugar y K-means en tercer lugar. Este orden permite corroborar los resultados obtenidos en la evaluación externa, donde los mejores resultados fueron alcanzados por estos algoritmos y en el mismo orden de clasificación. En relación con la Precisión, los mejores resultados los entrega MultiSearchBP seguido de Lingo y K-means.

Tabla 13. Calidad de clasificación de los algoritmos según el test de Friedman*.*

Algoritmo	**Precisión**		**Recuerdo**		**Medida F**	
	Test	**Orden**	**Test**	**Orden**	**Test**	**Orden**
MultiSearchBP	**1**	**1**	**1.3333**	**1**	**1**	**1**
STC	4.9167	5	2.1667	2	2.3333	2
K-means	4.1667	3	2.6667	3	2.8333	3
Lingo	2.8333	2	3.8333	4	3.8333	4
FullStars	4.6667	4	2.1667	5	5.3333	5
Stars	5.4167	6	6.3333	6	6.4167	6
Cliques	5.0	7	6.5	7	6.25	7
Valor P de la Prueba	0.004		1.2676 E-5		9.4605 E-6	
Estadístico de Friedman	18.8750		32.5714		33.2321	

Fuente: elaboración propia.

El test Wilcoxon fue utilizado para comparar los promedios de los resultados en Precisión, Recuerdo y Medida F de los algoritmos, con el propósito de determinar si los resultados de un algoritmo dominan estadísticamente a los de otro.

La Figura 29. Muestra en resumen la aplicación del test de Wilcoxon sobre los resultados de la Tabla 13. Los puntos negros en las fila significan el dominio del algoritmo nombrado en la fila sobre el algoritmo referenciado en la columna, los puntos sobre (encima) la diagonal principal tienen un nivel de significancia del 90% y los puntos debajo de la diagonal tienen un nivel significancia del 95%.

El test refleja que en relación con la Medida F, MultiSerachBP tiene un dominio con un nivel de significancia de 95% sobre el resto de algoritmos, seguido por STC el cual tiene un dominio con 95% de significancia a los algoritmos basados en teoría de grafos (Cliques, FullStarts y Starts) y Lingo. Por otra parte, los algoritmos Lingo y K-means también logran un dominio significativo del 95% sobre los algoritmos basados en teoría de grafos. Los resultados del test de Wilcoxon, al igual que los resultados del test de Friedman permiten demostrar que MultiSearchBP, STC, K-means y Lingo obtienen los mejores resultados en Medida F con relación a la evaluación manual realizada por los expertos.

En relación con los resultados de Precisión solamente se puede afirmar que MultiSearchBP obtiene resultados que dominan significativamente (con un 97%) a los resultados de los otros algoritmos y que los resultados de Lingo dominan a los de STC.

Figura 29. Resultados test de Wilcoxon.

Precisión

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
MultiSearchBP (1)	-	●	●	●	●	●	●
Cliques (2)	○	-					
FullStars (3)	○		-				
Stars (4)	○			-			
K-means (5)	○				-		
Lingo (6)	○					-	●
STC (7)	○					○	-

Recuerdo

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
MultiSearchBP (1)	-	●	●	●	●	●	●
Cliques (2)	○	-	○		○	○	○
FullStars (3)	○		-	●	○	○	○
Stars (4)	○			-	○	○	○
K-means (5)		●	●	●	-		
Lingo (6)	○	●	●	●		-	○
STC (7)		●	●	●		●	-

Medida F

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
MultiSearchBP (1)	-	●	●	●	●	●	●
Cliques (2)	○	-	○		○	○	○
FullStars (3)	○		-	●	○	○	○
Stars (4)	○			-	○	○	○
K-means (5)	○	●	●	●	-		
Lingo (6)	○	●	●	●		-	○
STC (7)	○	●	●	●		●	-

Los resultados del Recuerdo ubican a MultiSearchBP de primero, el cual tiene un dominio con un nivel de significancia de 95% sobre el resto de algoritmos excepto sobre STC y K-means, pero domina a todos con un nivel de significancia del 90%. En segundo lugar esta STC, el cual tiene un dominio con un 95% de significancia sobre los algoritmos basados en teoría de grafos (Cliques, FullStarts y Starts) y Lingo. Por otra parte, los algoritmos Lingo y K-means también logran un dominio significativo del 95% sobre los algoritmos basados en teoría de grafos.

Según Gaeta y otros[111], la certeza es una medida de la confianza que determina si un mecanismo o un proceso puede generar información de confianza, basado en esta afirmación y en los valores de los resultados obtenidos en la evaluación del modelo propuesto, podemos afirmar con certeza plena que el conocimiento descrito en esta tesis es verdadero y válido. Esta afirmación se basa además en la evidencia del conocimiento comunicable y reconocible hecho a través de las publicaciones y participaciones en eventos científicos.

Adicionalmente, el conocimiento adquirido en el desarrollo de las tesis da como resultado un juicio categórico. Y las deducciones a partir de los juicios categóricos producen un razonamiento o argumento concluyente [112]. Es decir, expresan verdades no condicionadas, sino que responden a la realidad como tal; la cual para este caso, está plasmada en la comprobación de la hipótesis de investigación.

5.6 Conclusiones

El prototipo para búsqueda (descubrimiento) y agrupación de BP, permite realizar varios tipos de consulta para ampliar el proceso de búsqueda. Las opciones de consulta aportan flexibilidad al usuario ya que es posible replantear las búsquedas para aprovechar de mejor manera el espacio de consulta y de esta forma aumentar la relevancia y pertinencia en los resultados retornados.

El espacio de trabajo conjunto provisto por la plataforma de evaluación permite a los jueces identificar y contextualizar de mejor manera los grupos a crear. La plataforma colaborativa, permite a los jueces evaluadores adicionar elementos que verdaderamente pertenecen al contexto del grupo creado.

La evaluación colaborativa permite a los jueces tener la visión general de los juicios de relevancia emitidos por cada uno de jueces sobre los ítems retornados en la lista de resultados. Por tal razón, un juez puede comparar la concordancia o discordancia en el juicio de relevancia emitido para un ítem evaluado, y así corroborar o cambiar su apreciación.

La aplicación de la escala de relevancia entre un BP de consulta y un BP en la lista de resultados, permite identificar la apreciación que cada uno de los jueces tiene con respecto a cada resultado.

Los resultados obtenidos en la evaluación demuestran la eficiencia y relevancia en el proceso de búsqueda de BP, ya que éstos presentan similitud con la evaluación hecha por los expertos humanos. En este sentido, se lograron niveles de Precisión gradada que se encuentran entre el 86% como punto mínimo y 92,24% como punto máximo. Los resultados obtenidos en la medida de Recuerdo gradada son bajos debido a que en el proceso de descubrimiento solo se están evaluando los primeros 10 resultados y no toda la lista de resultados relevantes, por ende no son tenidos en cuenta los BP clasificados como falsos positivos.

En el proceso de agrupación, los grupos son formados mediante correlación y similitud directa entre características textuales, estructurales o ambas. La estructura de árbol formada permite al usuario revisar las categorías y seleccionar el grupo de mayor similitud a su consulta.

Capítulo 6

Conclusiones y trabajos futuros

En este capítulo se definen las conclusiones a las que se llegó con el desarrollo de esta tesis, así como los trabajos futuros, teniendo en cuenta los objetivos planteados al inicio del proyecto.

6.1 Conclusiones

6.1.1 Búsqueda de BP

- En relación con la búsqueda de BP, se puede concluir que la búsqueda sobre grandes repositorios puede llegar a ser computacionalmente muy costosa, como se ha destacado en la literatura (que se refiere a las aplicaciones de gestión de BP). El proceso de búsqueda de BP, se ha desarrollado desde diferentes puntos de vista, enfocándose principalmente en los tipos de información contenida en los BP. Entre estos tipos de información se encuentran: información de estructura gráfica, de semántica de comportamiento, de semántica de operación y lingüística o textual. Los tipos de información son utilizados para determinar la estrategia de búsqueda a aplicar, la cual debe estar basada en una función de similitud que permita definir el grado de correspondencia entre un BP de consulta y los BP a recuperar del repositorio. En algunos trabajos se combinan grafos con métricas de similitud para editar distancias entre nodos con el propósito de reducir operaciones de comparación. Otros trabajos utilizan la lingüística (información textual) de los BP adicionando ontologías de dominio, algunos trabajos utilizan algoritmos genéticos complementados con heurísticas, otros utilizan reglas de asociación sobre eventos ya ejecutados. Los tipos de información contenida en los BP, permitió definir en esta tesis un modelo de búsqueda multimodal, que explota una técnica de recuperación de información basada en el modelo vectorial, que unifica en un solo espacio de búsqueda información textual y estructural, permitiendo así cubrir mayor cantidad de información en la ejecución de las consultas, a través de varios tipos de búsqueda.

6.1.2 Codebook

- A partir de la revisión bibliográfica que permitió construir el estado del arte, se puede concluir que esta tesis es la primera en usar *codebooks* para representar en forma de cadenas de texto la secuencia de comportamiento de los BP.
- La representación a través de cadenas de texto de la información estructural de los BP en forma de *codebook*, permite aumentar el rendimiento en el proceso de indexación y búsqueda. En consecuencia, esto permite una disminución de tiempo considerable en este proceso, ya que es 2500 veces más rápido que los algoritmos basados en grafos [97, 98] . El aumento en el rendimiento se debe a que la formación del *codebook* es realizada por medio de algoritmos de extracción de texto simples, los cuales se ejecutan en periodos más cortos de tiempo.
- Los *codebook* son construidos a través de la semántica de comportamiento (información estructural), la cual representa la secuencia lógica en que los BP pueden ejecutarse. Esta característica permite a los ingenieros modeladores plantear consultas para recuperar BP con una determinada semántica de comportamiento, estas consultas pueden ser expresadas por un conjunto de *codebook*s que representen la semántica de comportamiento en los BP a recuperar.

6.1.3 Esquema multimodal

- En relación con el esquema de búsqueda multimodal, partir de la revisión bibliográfica que permitió construir el estado del arte, se puede concluir que esta es la primera tesis en utilizar un esquema de indexación y búsqueda multimodal de BP.
- El esquema de búsqueda multimodal permite a los ingeniero modeladores plantear varios tipos de consultas, a saber: textual, estructural, y multimodal (la combinación de las dos anteriores). Además, permite tener una representación más amplia del objeto de estudio (Repositorio), debido a que puede almacenar varios tipos de información de los BP existentes.
- La sencillez del esquema multimodal, a diferencia de otras propuestas identificadas en el estado del arte, permite fácilmente la ampliación de la información indexada y de las opciones de consulta. Con base en lo anterior, en futuros trabajos es posible agregar otro tipo de información como: tareas ya

ejecutadas en archivos log, información del usuario que ejecuta las tareas y fechas de ejecución entre otras.

- El esquema de búsqueda multimodal demuestra que la representación en forma textual de los componentes lingüísticos y estructurales de los BP, permite obtener mejores resultados que las propuestas que representan los BP en forma de grafos (por ejemplo BeMantics y A*). De esta forma, el esquema multimodal permite resultados más rápidos, superando a BeMantics en 2500 veces y A* en 482 veces. En el caso de las herramientas BeMantics y A*, lo anterior es propiciado al requerir de algoritmos complejos y exhaustivos de isomorfismo de grafos.
- La relevancia del esquema multimodal fue evaluada con las medidas de precisión gradada (Pg), recuerdo gradada (Rg), Medida-F, ANDC y GenAvep'. Además, se comparó con otras herramientas de búsqueda de BP, BeMantics y una implementación del algoritmo A*, las cuales están basadas en características estructurales, semánticas y de comportamiento. El análisis de la relevancia del esquema multimodal demuestra que la combinación de información lingüística y *codebook*s (estructural) como elementos de búsqueda permite obtener un nivel alto de efectividad caracterizado por el rendimiento de la clasificación en la lista de resultados. Basado en la Medida F el esquema multimodal alcanzó 46,81%, evidenciando mayor rendimiento en la clasificación de la lista de resultados generados para cada consulta, logrando en promedio una mejora de 191% con relación a BeMantics y de 176% a A*, en clasificación de la lista de resultados. Por otra parte, en las medidas ANDCG y GenAveP', el esquema multimodal recupera BP con mejor posición, a través del ordenamiento de la lista de resultados, obteniendo así una diferencia de 150% sobre BeMantics y de 360% sobre A*.

6.1.4 Agrupación

- En relación con la agrupación de BP, esta tarea fue realizada usando un algoritmo de *clustering* iterativo e incremental que aplica una función de similitud basada en lógica difusa, trabajando sobre el conjunto de resultados retornados en una consulta. Se puede concluir que este algoritmo permite la reasignación de los elementos (BP) al grupo con mayor similitud, aun cuando ya han sido asignados inicialmente, lo cual permite tener grupos más homogéneos.

- La organización automática de los resultados considerados como relevantes en una consulta, a través del algoritmo de agrupación, puede servir como un punto de partida para el análisis de conjuntos de BP que comparten similitud estructural, lingüística o funcional.
- La función de similitud que utiliza el algoritmo de agrupación, facilitó la adaptación del algoritmo al esquema de búsqueda multimodal. Debido a que el cálculo de similitud es realizado con base en la información contenida en la lista de resultados. La información de la lista de resultados está internamente pre-procesada y organizada en una matriz que permite recuperar los elementos lingüísticos o estructurales de cada BP, de manera más directa y eficiente para realizar el cálculo de similitud entre los BP a agrupar.
- El método de etiquetado, implementado en el algoritmo de agrupación, se basa en las descripciones textuales de todos los elementos que existen en los BP de cada grupo. Estas descripciones definen la finalidad o funcionalidad de los BP. En consecuencia, las etiquetas retornadas por el algoritmo hacen referencia a las funcionalidades de todos los BP que existen en cada grupo, lo que permite al usuario identificar con mayor facilidad la funcionalidad de cada uno de los BP existentes en un grupo a revisar o analizar.
- En relación con el análisis de la agrupación interna en los algoritmos comparados, MultiSearchBP obtiene mejores grupos, con relación a las medidas empleadas, a saber: cohesión, acoplamiento, silueta, Suma de cuadrados (*Sum-of-squares*), Suma de cuadrados entre clusters (*Sum of squares between clusters*) e índice DB. Las medidas aplicadas permitieron evidenciar que utilizar información textual y estructural en la formación de grupos de BP, permite que la agrupación generada sea más compacta, y, en consecuencia, los elementos compartan un mayor subconjunto de características comunes, basadas en los tipos de información utilizada para la agrupación. La eliminación del solapamiento, es decir, de elementos (BP) que puedan existir en varios grupos a la vez, hace posible tener elementos más similares dentro de cada grupo, además de permitir mayor separación entre los grupos formados.
- En la evaluación externa de los grupos de BP, MultiSearchBP alcanza un nivel de 70% en el rendimiento en la agrupación determinado por la medida F, lo que implica que los grupos formados son relevantes y coinciden en alto grado con los grupos formados manualmente y en forma colaborativa por los expertos. Los niveles de precisión alcanzados en cada consulta, evidencian que MultiSearchBP

crea grupos relevantes a las consultas de los usuarios de forma automática y sin intervención humana. Facilitando así al usuario la revisión de las categorías de BP, con el propósito de seleccionar el grupo de mayor similitud a una consulta realizada por él.

6.1.5 Plataforma de evaluación colaborativa

- En la presente tesis se modeló y construyó una plataforma de evaluación colaborativa que posibilita la creación de repositorios de BP cerrados (de prueba). La plataforma permite definir un conjunto de consultas preestablecidas y las respuestas ideales a cada consulta, característica primordial en un repositorio de pruebas. Además la plataforma hace posible la formación manual de grupos con base en los resultados relevantes de cada consulta. Los grupos creados manualmente pueden ser considerados como la agrupación ideal, en el proceso de evaluación de algún mecanismo de agrupación de BP. La plataforma fue construida debido a que en la construcción del estado del arte no se encontró una plataforma que permitiera construir colaborativamente repositorios de BP de prueba.
- La plataforma cuenta con una infraestructura que integra a un grupo de jueces evaluadores en un entorno colaborativo, con el propósito de crear, almacenar, gestionar y recuperar grupos de modelos de BP, definidos manualmente en consenso por ellos.
- La plataforma proporciona un marco para evaluar la relevancia de los resultados proporcionados por un mecanismo de búsqueda de BP. Una vez obtenidos los resultados considerados relevantes en cada consulta, la plataforma integra un conjunto de elementos que permiten la comunicación y el intercambio de ideas y sugerencias entre los jueces que realizan la evaluación, con el propósito de refinar la consistencia de los resultados.
- La plataforma posibilita la creación manual de grupos de BP a los jueces evaluadores, a través de una visión consensuada en un ambiente colaborativo. La agrupación manual es realizada a partir de los resultados considerados como verdaderos relevantes por los jueces evaluadores en cada consulta. Para este proceso la plataforma provee una pizarra colaborativa que hace referencia a un espacio de trabajo conjunto para la formación de los grupos. Este espacio provee a cada juez la opción de crear uno o varios grupos teniendo en cuenta la

información compartida entre el BP de consulta y los BP en la lista de resultados. Esta información puede ser de tipo estructural, tipo textual o multimodal. Para cada grupo creado se define un nombre y los elementos pertenecientes. Además, los jueces pueden realizar cambios en cuanto al nombre o etiqueta de cada grupo. Cada juez puede seleccionar uno o varios elementos de la lista de resultados y arrástralos para adicionarlos al grupo, de la misma forma un juez puede eliminar uno o varios elementos del grupo. Además, la plataforma cuenta con una sala de chat donde los jueces evaluadores pueden compartir puntos de vista para definir la pertinencia de un nombre de un grupo y de los elementos pertenecientes a él.

6.1.6 Método para la construcción de repositorios de prueba

En relación con el método propuesto en esta tesis para la construcción colaborativa de repositorios de evaluación de sistemas de recuperación de BP, se puede concluir que:

- Está compuesto por tres etapas, 1) Evaluación individual, 2) Búsqueda de consensos en evaluaciones discordantes y 3) Refinamiento de resultados. El método fue planteado como instrumento de consolidación que permite a un conjunto de jueces emitir juicios con relación a los resultados relevantes que se deben entregar frente a una consulta de BP en una colección (lista) de BP previamente almacenados. En efecto, los resultados considerados como relevantes por el conjunto de jueces, serán los que representaran las repuestas ideales para cada consulta en el repositorio cerrado.
- El método sirve como base para la generación de repositorios de BP de evaluación, estables, sostenibles y reutilizables. Al mismo tiempo de libre acceso, y que pueda ser compartido y utilizado en futuras investigaciones por cualquier actor interesado en la temática de gestión de BP.
- En el proceso de evaluación colaborativa es conveniente no definir jornadas de trabajo demasiado largas (que no sobrepasen las 3 horas), ya que esto genera cansancio en los evaluadores. El cansancio puede llevar a que los evaluadores realicen sus evaluaciones de forma sesgada y los resultados no sean idóneos o adecuados.

6.2 Trabajos futuros

El estudio presentado en esta tesis sugiere las siguientes posibilidades de trabajo futuro:

- Complementar el modelo multimodal con ontologías de dominio específico que permitan agregar semántica a las búsquedas para tener más precisión en los resultados.
- Definir un modelo de indexación que incluya información adicional como: flujos de datos, flujos de control, flujos de ejecución, información de usuarios y fechas, entre otras. Donde los pesos de similitud sean definidos por una función que pondere cada tipo de información según sea el interés del usuario que realiza las consultas.
- Diseñar un modelo de categorización semántica, automático o semiautomático, que realice estructuración de un repositorio de BP en categorías funcionales, a través de patrones estructurales, de comportamiento, metadatos o la documentación de los BP.
- Incorporar en la herramienta de evaluación un módulo de evaluación automática que genere gráficas de relevancia y ampliar la evaluación aplicando nuevas medidas para la búsqueda de BP.
- Mejorar el algoritmo de agrupamiento de manera que sea capaz de decidir de forma automática qué métrica de similitud (lógica difusa, similitud por cosenos, distancia euclidiana, entre otras), proporciona los mejores resultados.
- Incorporar en MultiSearchBP un método de agrupación jerárquica con el fin de crear categorías y subcategorías de los BP presentes en el repositorio, a partir de una consulta.
- Evaluar el uso de métodos algebraicos de descomposición de matrices con el objetivo de analizar la semántica implícita en la descripción de los BP y su uso en el esquema multimodal de búsqueda propuesto.

Referencias bibliográficas

[1] C. U. Pyon, J. Y. Woo, and S. C. Park, "Service improvement by business process management using customer complaints in financial service industry," *Expert Systems with Applications,* vol. 38, pp. 3267-3279, 2011.
[2] V. B. Vukšić, M. P. Bach, and A. Popovič, "Supporting performance management with business process management and business intelligence: A case analysis of integration and orchestration," *International Journal of Information Management,* vol. 33, pp. 613-619, 2013.
[3] T. Schlegel, K. Vidačković, S. Dusch, and R. Seiger, "Management of interactive business processes in decentralized service infrastructures through event processing," *Journal of King Saud University - Computer and Information Sciences,* vol. 24, pp. 137-144, 2012.
[4] M.-J. Son and T.-w. Kim, "Business process management-based job assignment in ship hull production design," *Ocean Engineering,* vol. 88, pp. 12-26, 2014.
[5] J. Lee, D. Muthig, and M. Naab, "A feature-oriented approach for developing reusable product line assets of service-based systems," *Journal of Systems and Software,* vol. 83, pp. 1123-1136, 2010.
[6] R. d. Santos Rocha and M. Fantinato, "The use of software product lines for business process management: A systematic literature review," *Information and Software Technology,* vol. 55, pp. 1355-1373, 2013.
[7] M. Kunze and M. Weske, "An Open Process Model Library," *Business Process Management Workshops, BPM 2011 International Workshops Clermont-Ferrand, France, August 29, 2011 Revised Selected Papers, Part II,* pp. 26-38, 2012.
[8] T. Raghu and a. Vinze, "A business process context for Knowledge Management," *Decision Support Systems,* vol. 43, pp. 1062-1079, 2007.
[9] Y. Alotaibi and F. Liu, "A novel secure business process modeling approach and its impact on business performance," *Information Sciences,* vol. 277, pp. 375-395, 2014.
[10] I. Khodyrev and S. Popova, "Discrete Modeling and Simulation of Business Processes Using Event Logs," *Procedia Computer Science,* vol. 29, pp. 322-331, 2014.
[11] P. K. Stefan Appel n, Sebastian Frischbier,Tobias Freudenreich, Alejandro Buchmann, "Modeling and execution of events tream processing in business processes," *Information Systems,* 2014.
[12] H. Bae, S. Lee, and I. Moon, "Planning of business process execution in Business Process Management environments," *Information Sciences,* vol. 268, pp. 357-369, 2014.
[13] C.-C. Chen, H.-S. Shih, H.-J. Shyur, and K.-S. Wu, "A business strategy selection of green supply chain management via an analytic network process," *Computers & Mathematics with Applications,* vol. 64, pp. 2544-2557, 2012.

[14] K. Iizuka, Y. Iizuka, and C. Suematsu, "E-Business Process Modeling Issues: From the Viewpoint of Inter-organizational Process Efficiency and Information Sharing," *Procedia Computer Science,* vol. 22, pp. 820-827, 2013.
[15] A. Jiménez-Ramírez, B. Weber, I. Barba, and C. del Valle, "Generating Optimized Configurable Business Process Models in Scenarios Subject to Uncertainty," *Information and Software Technology,* 2014.
[16] D. Greenwood and R. Ghizzioli, "Goal-Oriented Autonomic Business Process Modelling and Execution," *Systems Technology,* vol. 1, p. 18, 2009.
[17] H. H. Chang and I. C. Wang, "Enterprise Information Portals in support of business process, design teams and collaborative commerce performance," *International Journal of Information Management,* vol. 31, pp. 171-182, 2011.
[18] R. Dijkman, M. Dumas, B. van Dongen, R. Käärik, and J. Mendling, "Similarity of business process models: Metrics and evaluation," *Information Systems,* vol. 36, pp. 498-516, 2011.
[19] R. Alas, M. Zernand-Vilson, and M. Vadi, "Management Techniques in Estonian Organizations: Learning Organization and Business Process Reengineering," *Procedia - Social and Behavioral Sciences,* vol. 62, pp. 494-498, 2012.
[20] Z. Huang, X. Lu, and H. Duan, "Resource behavior measure and application in business process management," *Expert Systems with Applications,* vol. 39, pp. 6458-6468, 2012.
[21] J. Tang, L. G. Pee, and J. Iijima, "Investigating the effects of business process orientation on organizational innovation performance," *Information & Management,* vol. 50, pp. 650-660, 2013.
[22] H. a. Reijers, R. S. Mans, and R. a. van der Toorn, "Improved model management with aggregated business process models," *Data & Knowledge Engineering,* vol. 68, pp. 221-243, 2009.
[23] W. M. P. V. D. Aalst, "Process-Aware Information Systems : Lessons to be Learned from Process Mining," *Technology*.
[24] W. M. P. V. D. Aalst and B. F. V. Dongen, "ProM : The Process Mining Toolkit," *Industrial Engineering*.
[25] W. M. P. Aalst, V. Rubin, H. M. W. Verbeek, B. F. Dongen, E. Kindler, and C. W. Günther, "Process mining: a two-step approach to balance between underfitting and overfitting," *Software & Systems Modeling,* vol. 9, pp. 87-111, 2008.
[26] W. M. P. Aalst, Van Der, M. Pesic, and M. Song, "Beyond Process Mining: From the Past to Present and Future," *Technology Management,* vol. 8, pp. 1-15, 2009.
[27] R. P. J. C. Bose and W. M. P. V. D. Aalst, "Abstractions in Process Mining : A Taxonomy of Patterns," *Computer and Electrical Engineering*.
[28] M. Ehrig, "Measuring Similarity between Semantic Business Process Models," *Reproduction,* vol. 5, p. 10, 2008.
[29] R. A. T.-R. Dafne A. Rosso-Pelayo, Miguel Gonzales-Mendoza, Neil Hernandez-Gress, "Business Process MIning and Rules Detection for Unstructured Information," *Ninth Mexican International Conference on Artificial Intelligence,* 2010.

[30] D. Grigori, Corrales, Juan Carlos, Bouzeghoub, Mokrane, Gater, Ahmed, "Ranking BPEL Processes for Service Discovery," vol. 3, pp. 178-192, 2010.
[31] W. Tan, W. Xu, F. Yang, S. Li, and Y. Du, "A Framework for Business Process Simulation Based on Multi-Agent Cooperation," *Computer,* vol. 14, p. 12, 2009.
[32] N. This, "Managing and Measuring Business Processes," *Management,* vol. 5, pp. 295-313, 2007.
[33] D. He, G. Ritchie, and J. Lee, "References to graphical objects in interactive multimodal queries," *Knowledge-Based Systems,* vol. 21, pp. 617-628, 2008.
[34] C. Lauer, "Contending with Terms: “Multimodal” and “Multimedia” in the Academic and Public Spheres," *Computers and Composition,* vol. 26, pp. 225-239, 2009.
[35] A. Reimerink, M. García de Quesada, and S. Montero-Martínez, "Contextual information in terminological knowledge bases: A multimodal approach," *Journal of Pragmatics,* vol. 42, pp. 1928-1950, 2010.
[36] Y.-T. Juang, S.-L. Tung, and H.-C. Chiu, "Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions," *Information Sciences,* vol. 181, pp. 4539-4549, 2011.
[37] J. C. Caicedo, J. BenAbdallah, F. A. González, and O. Nasraoui, "Multimodal representation, indexing, automated annotation and retrieval of image collections via non-negative matrix factorization," *Neurocomputing,* vol. 76, pp. 50-60, 2012.
[38] B. L. Kennedy, S.-f. Chang, and A. Natsev, "Query-Adaptive Fusion for Multimodal Search," *Proceedings of the IEEE,* vol. 96, p. 22, 2008.
[39] H.-L. Luo, H. Wei, and F.-X. Hu, "Improvements in image categorization using *codebook* ensembles," *Image and Vision Computing,* vol. 29, pp. 759-773, 2011.
[40] Y.-C. Hu, B.-H. Su, and C.-C. Tsou, "Fast VQ *codebook* search algorithm for grayscale image coding," *Image and Vision Computing,* vol. 26, pp. 657-666, 2008.
[41] M. E. Fonteyn, M. Vettese, D. R. Lancaster, and S. Bauer-Wu, "Developing a *codebook* to guide content analysis of expressive writing transcripts," *Appl Nurs Res,* vol. 21, pp. 165-8, Aug 2008.
[42] Q. Zhong, Z. Qingqing, and G. Tengfei, "Moving Object Tracking Based on *Codebook* and Particle Filter," *Procedia Engineering,* vol. 29, pp. 174-178, 2012.
[43] K. Rizman, "An efficient k 0 -means *clustering* algorithm," *Pattern Recognition Letters,* vol. 29, pp. 1385-1391, 2008.
[44] I. Martínez, "Restricted Conceptual *Clustering* Algorithms based on Seeds," vol. 11, pp. 174-187, 2007.
[45] F. Aiolli, A. Burattin, and A. Sperduti, "Metric for *Clustering* Business Processes Based on Alpha Algorithm Relations," *Business,* p. 17, 2011.
[46] I. S. Engineering and U. States, "Hierarchical *clustering* of business process models," *Computer,* vol. 5, pp. 613-616, 2009.
[47] C. Diamantini, D. Potena, and E. Storti, "*Clustering* of Process Schemas by Graph Mining Techniques (Extended Abstract)," *Methodology,* vol. 4, p. 7, 2011.
[48] J. Melcher, D. Seese, and I. Aifb, "Visualization and *Clustering* of Business Process Collections Based on Process Metric Values," *Measurement,* vol. 8, p. 9, 2008.
[49] D. R. Ferreira, "Applied Sequence *Clustering* Techniques for Process Mining," *Science,* pp. 492-513, 2009.

[50] D. Ferreira, M. Zacarias, M. Malheiros, and P. Ferreira, "Approaching Process Mining with Sequence *Clustering*: Experiments and Findings," *Engineering,* vol. 7, pp. 1-15, 2008.
[51] C. Serrano, "Un Modelo Integral para un Profesional en Ingeniería," *Universidad del Cauca,* 2003.
[52] C. Wohlin, "Experimentation in software engineering: an introduction," p. 390, 2005.
[53] H. Ordonez, J. C. Corrales, C. Cobos, and L. K. Wives, "Collaborative grouping of business process models," *EATIS 2014, Valparaiso -Chile,* pp. 1-2, 2014.
[54] H. A. Reijers, R. S. Mans, and R. A. van der Toorn, "Improved model management with aggregated business process models," *Data & Knowledge Engineering,* vol. 68, pp. 221-243, 2009.
[55] R. Rajnoha, A. Sujová, and J. Dobrovič, "Management and Economics of Business Processes Added Value," *Procedia - Social and Behavioral Sciences,* vol. 62, pp. 1292-1296, 2012.
[56] M. Indulska, "Modeling languages for business processes and business rules : A representational analysis," *Information Systems,* vol. 35, pp. 379-390, 2010.
[57] X. Zhao and C. Liu, "Version management for business process schema evolution," *Information Systems,* vol. 38, pp. 1046-1069, 2013.
[58] N. Sarter, "Multimodal information presentation: Design guidance and research challenges," *International Journal of Industrial Ergonomics,* vol. 36, pp. 439-445, 2006.
[59] E. Fersini, E. Messina, and F. Archetti, "A probabilistic relational approach for web document *clustering*," *Information Processing & Management,* vol. 46, pp. 117-130, 2010.
[60] P. N. Karamolegkos, C. Z. Patrikakis, N. D. Doulamis, P. T. Vlacheas, and I. G. Nikolakopoulos, "An evaluation study of *clustering* algorithms in the scope of user communities assessment," *Computers & Mathematics with Applications,* vol. 58, pp. 1498-1519, 2009.
[61] C. C. Aggarwal and C. Zhai, "A Survey of Text *Clustering* Algorithms," pp. 77-128, 2012.
[62] E. I. Papageorgiou, J. D. Roo, C. Huszka, and D. Colaert, "Formalization of treatment guidelines using Fuzzy Cognitive Maps and semantic web tools," *J Biomed Inform,* vol. 45, pp. 45-60, Feb 2012.
[63] I. G. I. Publishing, "Data Mining and Knowledge Discovery Technologies."
[64] L. Han and G. Chen, "A fuzzy *clustering* method of construction of ontology-based user profiles," *Advances in Engineering Software,* vol. 40, pp. 535-540, 2009.
[65] C. C. Ekanayake, M. Dumas, L. García-Bañuelos, M. Rosa, and A. H. M. Hofstede, "Approximate Clone Detection in Repositories of Business Process Models," vol. 7481, pp. 302-318, 2012.
[66] M. K. Markus Guentert, and Mathias Weske, "Evaluation Measures for Similarity Search Results in Process Model Repositories," *Springer-Verlag Berlin Heidelberg,* 2012.
[67] J. Koehler, "The Process-Rule Continuum — How can the BPMN and SBVR Standards interplay ?," *Continuum,* 2010.

[68] H. A. Reijers, T. Freytag, J. Mendling, and A. Eckleder, "Syntax highlighting in business process models," *Decision Support Systems,* vol. 51, pp. 339-349, 2011.
[69] A. Koschmider, T. Hornung, and A. Oberweis, "Recommendation-based editor for business process modeling," *Data & Knowledge Engineering,* vol. 70, pp. 483-503, 2011.
[70] Z. Huang, J. Huai, X. Liu, and J. Zhu, "Business Process Decomposition based on Service Relevance Mining," *Science,* 2010.
[71] R. Pérez-castillo, M. Piattini, and B. Weber, "An Empirical Comparison of Static and Dynamic Business Process Mining," *Society,* vol. 11, pp. 272-279, 2011.
[72] D. a. Rosso-Pelayo, R. a. Trejo-Ramirez, M. Gonzalez-Mendoza, and N. Hernandez-Gress, "Business Process Mining and Rules Detection for Unstructured Information," *2010 Ninth Mexican International Conference on Artificial Intelligence,* pp. 81-85, 2010.
[73] C. J. Turner and J. Ö. Mehnen, "A Genetic Programming Approach to Business Process Mining," *Applied Sciences,* pp. 1307-1314, 2008.
[74] C. Li, M. Reichert, and A. Wombacher, "Mining business process variants: Challenges, scenarios, algorithms," *Data & Knowledge Engineering,* vol. 70, pp. 409-434, 2011.
[75] Z. Yan, R. Dijkman, and P. Grefen, "Business process model repositories – Framework and survey," *Information and Software Technology,* vol. 54, pp. 380-395, 2012.
[76] M. La Rosa, H. A. Reijers, W. M. P. van der Aalst, R. M. Dijkman, J. Mendling, M. Dumas, and L. García-Bañuelos, "APROMORE: An advanced process model repository," *Expert Systems with Applications,* vol. 38, pp. 7029-7040, 2011.
[77] J. C. Caicedo, J. BenAbdallah, F. a. González, and O. Nasraoui, "Multimodal representation, indexing, automated annotation and retrieval of image collections via non-negative matrix factorization," *Neurocomputing,* vol. 76, pp. 50-60, 2012.
[78] S. G. Alexander Pretschner, "Ontology Based Personalized Search," *11th IEEE Intl. Conf. on Tools with Artificial Intelligence,* 1999.
[79] R. Christopher D. Manning, Prabhakar, Schütze, Hinrich, "An Introduction to Retrieval Information," p. 428, 2008.
[80] S. Smirnov, M. Weidlich, J. Mendling, and M. Weske, "Action patterns in business process model repositories," *Computers in Industry,* vol. 63, pp. 98-111, 2012.
[81] R. Dijkman, B. Gfeller, J. Küster, and H. Völzer, "Identifying refactoring opportunities in process model repositories," *Information and Software Technology,* vol. 53, pp. 937-948, 2011.
[82] Y. Yang, W.-p. Yang, and Y. Liang, "Application of Weighted Fuzzy *Clustering* Method to Supplier Selection Under E-Business," pp. 1293-1299, 2013.
[83] H. H. Chang and I. C. Wang, "Enterprise Information Portals in support of business process, design teams and collaborative commerce performance," *International Journal of Information Management,* vol. 31, pp. 171-182, 2011.
[84] O. Hugo, J. C. Corrales, C. Cobos, L. Krug Wives, and L. Thom, "COLLABORATIVE EVALUATION TO BUILD CLOSED REPOSITORIES ON BUSINESS PROCESS MODELS," *Springer- iceis, Portugal,* 2014.

[85] S. Smirnov, M. Weidlich, J. Mendling, and M. Weske, "Action patterns in business process model repositories," *Computers in Industry,* vol. 63, pp. 98-111, 2012.
[86] R. Dijkman, B. Gfeller, J. Küster, and H. Völzer, "Identifying refactoring opportunities in process model repositories," *Information and Software Technology,* vol. 53, pp. 937-948, 2011.
[87] R. Delbru, S. Campinas, and G. Tummarello, "Searching web data: An entity retrieval and high-performance indexing model," *Web Semantics: Science, Services and Agents on the World Wide Web,* vol. 10, pp. 33-58, 2012.
[88] D. Van Nuffel and M. De Backer, "Multi-abstraction layered business process modeling," *Computers in Industry,* vol. 63, pp. 131-147, 2012.
[89] Z. Yan, R. Dijkman, and P. Grefen, "Fast business process similarity search," *Distributed and Parallel Databases,* vol. 30, pp. 105-144, 2012.
[90] I. Ognjanovic, B. Mohabbati, D. Gaevic, E. Bagheri, and M. Bokovic, "A Metaheuristic Approach for the Configuration of Business Process Families," *2012 IEEE Ninth International Conference on Services Computing,* pp. 25-32, 2012.
[91] L. K. WIVES, "Um estudo sobre Agrupamento de Documentos Textuais em Processamento de Informações não Estruturadas Usando Técnicas de "*Clustering*"," *Instituto de Informática da UFRGS,* vol. Dissertação de mestrado, 1999.
[92] S. Ristov and D. Korenčić, "Using static suffix array in dynamic application: Case of text compression by longest first substitution," *Information Processing Letters,* vol. 115, pp. 175-181, 2015.
[93] S. Gog, G. Navarro, and M. Petri, "Improved and extended locating functionality on compressed suffix arrays," *Journal of Discrete Algorithms,* 2015.
[94] H. Ordonez, J. C. Corrales, and C. Cobos, "MultiSearchBP-Entorno para busqueda y agrupacion de modelos de procesos de negocio," *Polibits,* vol. 49, 2014.
[95] C. D. Manning, P. Raghavan, and H. Schutze, "Introduction to Information Retrieval," 2008.
[96] K. Ulrich and K. Birgitta, "Measures for Benchmarking Semantic Web Service Matchmaking Correctness," pp. 45-59, 2010.
[97] J. C. C. Cristhian Figueroa, "Business Process Retrieval based on Behavioral Semantics," *Revista EIA,* 2012.
[98] B. T. Messmer, H. Bunke, and I. C. Society, "A New Algorithm for Error-Tolerant Subgraph Isomorphism Detection," vol. 20, pp. 493-504, 1998.
[99] H. Liu, H. Bao, and D. Xu, "Concept vector for semantic similarity and relatedness based on WordNet structure," *Journal of Systems and Software,* vol. 85, pp. 370-381, 2012.
[100] D. Petrelli, "On the role of user-centred evaluation in the advancement of interactive information retrieval," *Information Processing & Management,* vol. 44, pp. 22-38, 2008.
[101] P. Petratos, "Information Retrieval Systems : A Human Centered Approach," *Software Eng Application,* 2007.
[102] Y. B. MARIA HALKIDI, MICHALIS VAZIRGIANNIS, "On *Clustering* Validation Techniques," *Journal of Intelligent Information Systems,* 2001.

[103] M. Brun, C. Sima, J. Hua, J. Lowey, B. Carroll, E. Suh, and E. R. Dougherty, "Model-based evaluation of *clustering* validation measures," *Pattern Recognition,* vol. 40, pp. 807-824, 2007.
[104] X. L. Q. Z. G. Wei, "The *clustering* algorithm for Chinese texts based on Lingo," *Fuzzy Systems and Knowledge Discovery (FSKD), Eighth International Conference,* pp. 1187 - 1190, 2011.
[105] M. F. Suneetha, S.S.; Pervez, S.M.Z., "*Clustering* of web search results using Suffix tree algorithm and avoidance of repetition of same images in search results using L-Point Comparison algorithm," *Emerging Trends in Electrical and Computer Technology (ICETECT), International Conference* pp. 1041 - 1046, 2011.
[106] H. X. J. W. J. Chen, "K-Means *Clustering* Versus Validation Measures: A Data-Distribution Perspective," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions* vol. 39, pp. 318 - 331, 2009.
[107] T. R. L. dos Santos and L. E. Zárate, "Categorical data *clustering*: What similarity measure to recommend?," *Expert Systems with Applications,* vol. 42, pp. 1247-1260, 2015.
[108] K. D. a. J. Szyma´nski, "External Validation Measures for Nested *Clustering* of Text Documents," *Springer-Verlag Berlin Heidelberg - Emerging Intelligent Technologies in Industry,,* 2011.
[109] D. G. Ferrari and L. N. de Castro, "*Clustering* algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods," *Information Sciences,* 2015.
[110] M. M. Carlos Cobos, Elizabeth León, Milos Manic, and Enrique Herrera-Viedma, "TopicSearch—Personalized Web *Clustering* Engine Using Semantic Query Expansion, Memetic Algorithms and Intelligent Agents," *Polibits,* vol. 47, 3012.
[111] V. L. Matteo Gaeta, Giuseppina Rita Mangione, Francesco Orciuoli, Pierluigi Ritrovato, Saverio Salerno "A methodology and an authoring tool for creating Complex Learning

Objects to support interactive storytelling," *Computers in Human Behavior,* vol. 31, 2014.
[112] Y. Akpınar, "Conventional and web based reflection tools in learning the design of interactive learning objects," *World Conference on Educational Sciences,* 2009.

Anexos

Actas del IV Congreso Internacional ATICA 2012 - Loja (Ecuador)

MODELO PARA EL DESCUBRIMIENTO DE PROCESOS DE NEGOCIO BASADO EN TRAZAS DE EJECUCIÓN GENERADAS CON BIZAGI

Hugo Ordoñez [1], Manuel Obando [2], Wilmer Mora [3]

[1]Facultad de Ingeniería Electrónica y Telecomunicaciones
Universidad del Cauca Popayán, Cauca, Colombia
hugoordonez@unicauca.edu.co

[2]Universidad Mariana
San Juan de Pasto, Nariño, Colombia
manuelobando5@hotmail.com

[3]Universidad Mariana
San Juan de Pasto, Nariño, Colombia
w_mora159@hotmail.com

Resumen. En este artículo se presenta un modelo para el descubrimiento de procesos de negocio que esta basado en las trazas ejecución de registradas en archivos Log generados por la suite Bizagi (herramienta para la gestión de procesos de negocio). El modelo esta compuesto por 3 componentes: 1) Parser, 2) Indexación y 3) Consulta, se describe cada uno de estos componentes. Además se describe la funcionalidad de la suite Bizagi para ejecutar los procesos de negocio modelados. La herramienta que implementa el modelo también es presentada y finalmente los resultados de la evaluación a la que fue sometido el modelo.

Palabras clave: Procesos de negocio, trazas de ejecución, descubrimiento de procesos.

Abstract. This article presents a model for the discovery of business processes that is based on execution traces recorded in log files generated by the suite Bizagi (tool for business process management). The model is composed of three components: 1) Parser, 2) Indexing and 3) Consultation, describes each of these components. It also describes the functionality of the suite Bizagi to run business processes modeled. The tool that implements the model is presented and finally the results of the assessment to which the model was subjected.

Keywords: business process execution traces, process discovery.

1 Introducción

Hoy en día, las empresas ofertan variedad de productos y servicios para mantener su competitividad en el mercado. Las tareas y funciones que intervienen en las actividades comerciales son representadas por Procesos de Negocios (BP) [1], los cuales constituyen un conjunto de tareas lógicamente relacionadas, que se ejecutan de forma secuencial siguiendo reglas para producir salidas válidas para el negocio [2]. Estas salidas representan toda la información generada por las tareas y procesos que se han ejecutado en la organización, esta es almacenada en archivos

de registros conocidos como LOG[3] , los cuales contienen información acerca de recursos compartidos, participantes, conexiones con otros sistemas y transacciones realizadas, fechas de ejecución [4].

Las tareas que buscan alcanzar los objetivos organizacionales son organizadas en procesos los cuales son modelados por expertos utilizando herramientas para el diseño de BP en donde plasman las operaciones o tareas que necesita ejecutar en la organización. Una de las herramientas más utilizadas y reconocidas en este momento es Bizagi (Suite para modelamiento y ejecución de Procesos de negocio) [5], la cual cuenta con un entorno sencillo de utilizar y muy completo, en relación con la gestión de BP. A pesar de la robustez de esta herramienta y de muchas otras existentes en el mercado en al actualidad no se cuenta con una propuesta que permita explotar la información registrada en los archivos de transacciones LOG con el propósito de que esta información sirva como base para el replanteamiento o (re)modelamiento de un nuevo BP que cumpla con los requerimientos actuales de la organización [6], y además permite que los diseñadores re-utilicen efectivamente los BP desarrollados previamente y se disminuya así el tiempo de desarrollo de los nuevos procesos de negocio[7].

Por esta razón en este artículo se plantea un modelo para el descubrimiento de BP basado en trazas de ejecución registradas en los archivos LOGs generados con la herramienta de gestión Bizagi. En la siguiente sección se presenta algunos componentes teóricos, en la sección 3 se presentan trabajos relacionados con la temática, en la sección 4 se presenta y se describe la funcionalidad de Bizagi, en la sección 5 es presentada el modelo propuesto, en la sección 6 una evaluación parcial del modelo basada en precisión, recuerdo y medida F medidas ampliamente utilizadas en recuperación de información y finalmente en la sección 7 se presenta las conclusiones y trabajo a futuro.

2 Componentes teóricos

Business Process (BP) – Proceso De Negocio

Un proceso de negocio (Business Process BP) es una actividad del mundo real que consta de un conjunto de tareas lógicamente relacionadas, que se ejecutan siguiendo las reglas del negocio para generar una salida válida. Un ejemplo de esto es realizar un pago, realizar una extracción de efectivo de una cuenta bancaria, etc. También especifican el orden de ejecución eventual de las operaciones lógicamente relacionadas de una colección de servicios teniendo en cuenta, los datos compartidos entre los servicios, que socios participan y cómo participan, manejo de excepciones, además estos procesos de negocio pueden ser de dos tipos. 1) Procesos de negocio abstractos especifican el intercambio de mensajes entre las diferentes partes, la cual puede ser visto como una sola organización que revela su comportamiento interno, 2) Procesos de negocio ejecutables, especifican el comportamiento real de un participante.[8]

Notación para el Modelado de Procesos de Negocio (BPMN)

BPMN es un estándar para el modelado grafico de BP creado inicialmente por la BPMI (Business Process Management Initiative) y actualmente mantenido por el grupo OMG. Este estándar permite modelar flujos de BP y WS a través de la coordinación de secuencias de procesos y los mensajes que fluyen entre los participantes de las diferentes actividades o tareas [9]. Una característica importante de BPMN es su capacidad para consolidarse como una notación entendible por diferentes tipos de usuarios, desde analistas de negocios (encargados de crear los borradores iniciales), hasta los desarrolladores de procesos ejecutables (responsables de crear una aplicación que ejecute las tareas del proceso). Adicionalmente, esta notación facilita que los lenguajes diseñados en XML para la ejecución de BP, tales como BPEL4WS

(Business Process Execution Language for Web Services) y BPML (Business Process Modeling Language) puedan expresarse visualmente mediante una notación estándar [10].

Trazas de Ejecución o LOGs

Las trazas de ejecución o también conocidas como "LOGs", contienen información de las instrucciones del trabajo a realizarse, quién debe realizarlo, condiciones de intensificación, conexiones a otros sistemas, sentencias, actividades, información de tiempos de ejecución, optimizador utilizado. Toda esta información de los registros se guardan en un fichero de texto, al cual se adiciona líneas a medida que las tareas de los BPs son ejecutas. Además por medio de estos es posible encontrar información para detectar problemas o errores en las actividades que están en ejecución. [11].

Descubrimiento de Procesos

El descubrimiento de BP esta basado en la capacidad de los sistemas de recuperación de BP para inferir acerca de los conceptos sobre los cuales el cliente está realizando su búsqueda. Para esto, el nivel considera criterios que tienen en cuenta el significado y las relaciones entre los conceptos que describen cada una de las tareas y procesos ejecutados para realizar emparejamiento entre la información que forma el historial de las ejecuciones realizadas anteriormente y lo que el usuario necesita en su consulta o búsqueda. En esta temática existen propuestas basadas en: Semántica las cuales incorporan ontologías para refinar las búsquedas [12]. Estructura que utilizan formalismo par facilitar el análisis estructural de los BP utilizando técnicas matemáticas, como por ejemplo el isomorfismo de grafos [13] . Comportamiento definido por varios aspectos que pueden determinar el comportamiento de un BP, como por ejemplo el intercambio de mensajes dentro de las actividades, los registros de ejecución histórica (LOGs) y el flujo de control [5].

3 TRABAJOS RELACIONADOS

El tema de interés central en esta investigación es el descubrimiento de BP, temáticas de alto interés investigativo en la actualidad. A continuación se presenta un resumen de los trabajos más destacados en esta temática.

En [14] se presenta el problema de asignación de recursos en la ejecución de LOGs, el cual es abordado con técnicas de minería de datos como son las reglas de asociación las cuales se utilizan para descubrir la relación y el potencial de las asociaciones en grandes cantidades de datos, esto con el objetivo de identificar los patrones que se expresan en forma de reglas de asociación, los datos en los conjuntos de transacciones para descubrir los patrones de frecuencia de asignación de recursos en actividades y sus dependencias asociadas, aprovechando la hora de inicio y fin de una ejecución de la actividad.

En [15] se hace una referencia al pasado, presente y futuro de la minería de procesos, en donde destacan la existencia de tres tipos de minería las cuales tienen como objetivos extraer modelos de procesos de negocios, comprobar la integridad del modelo y del registro en lo cual se mide el ajuste de estos para proponer nuevos modelos derivados, por último se tiene que hay técnicas para modificar o extender el modelo. Basándose en un análisis del registro de eventos (LOGs) puede haber sugerencias para cambiar el modelo, aportando que los modelos ampliados pueden usarse en simulaciones, realizando un procedimiento para reportar predicciones, chequeos o modificación de procesos en ejecución o procesos por ejecutar. Aunque solo se trabaja con procesos de negocios modelados con YAUL.

En [16] se presenta una técnica de minería de procesos de negocios en aplicación de reglas para información no estructurada, esta técnica se lleva a cabo utilizando datos no estructurados en lugar de los registros de las aplicaciones. En ausencia de los registros del sistema se analiza la organización de documentos, informes y reportes ejecutivos sobre el proceso de ejecución. Los documentos escritos en lenguaje natural contienen información sobre la ejecución de los procesos exitosos o información acerca de los problemas en torno a la ejecución del proceso, con todo esto la investigación se centra en la detección de reglas, patrones y relaciones de causa-efecto y encontrar pruebas de procesos incompletos o mal ejecutados.

En [17] se presenta un análisis sobre la búsqueda de los BP, dando a conocer las dificultades para encontrar componentes que puedan adaptarse exactamente a las exigencias de los usuarios. Por esta razón el desarrollo de nuevos métodos de búsqueda, intuitivos, dotados de inteligencia artificial, basados en semántica y que reconozcan lo que realmente el usuario necesita recuperar es un área importante de la I+D que permite agilizar el despliegue y configuración de nuevos BP.

4 LA SUITE PARA GESTIÓN DE PROCESOS DE NEGOCIO BIZAGI

En la actualidad es considerada una de las herramientas mas utilizadas para modelar y automatizar un proceso de negocio de forma rápida y flexible. Esta fue diseñada para resolver problemas de negocio reales, además de su poderosa y sencilla suite de BPM. BizAgi ofrece dos (Modelado y Ejecución) complementos para la gestión de procesos de negocio (BP) "El modelador de Procesos y la Suite BizAgi", las cuales se encuentran basada en Business Process Model and Notation (BPMN) notación gráfica que describe la lógica de los pasos de un proceso de negocio, y así proporcionar un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente tal como se muestra en Figura 1.

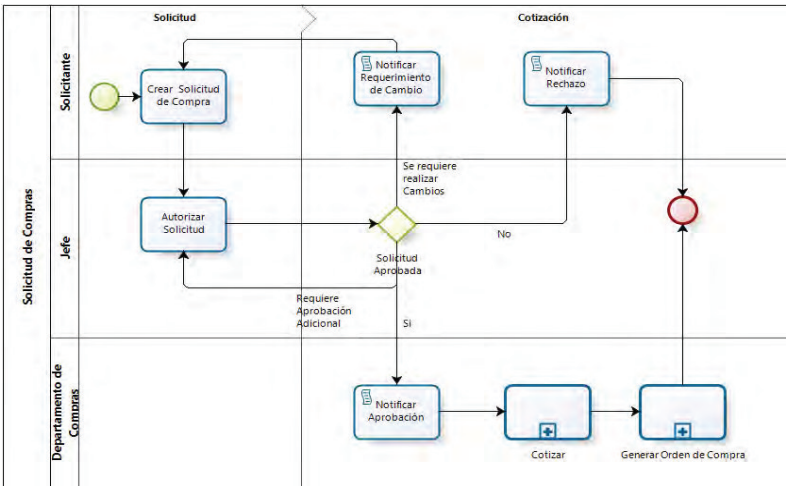


Figura 1. Modelo de Proceso de negocio

La etapa de Ejecución Bizagi la realiza mediante una seria de pasos consecutivos en los cuales están: la creación de formas asociadas al proceso. Las formas son el medio por el cual el usuario registra el resultado de un trabajo particular asociado a un caso del proceso y por lo tanto sólo se asocian a figuras en las cuales exista intervención humana tales como : tareas de usuario y eventos intermedios sin especificar que sean realizados de forma manual. Tal como se muestra en la Figura 2.

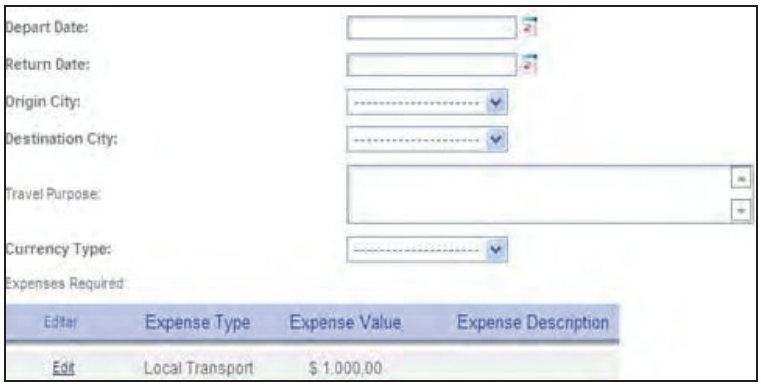


Figura 2. Creación de Formas

También, es importante resaltar que cada uno de los campos mostrados en las formas hace referencia a los atributos de las diferentes entidades.

Continuando con el proceso la siguiente etapa en el asistente de ejecución de Bizagi es crear las reglas de asociación a los flujos de secuencia. Estas reglas se relacionan a los flujos de secuencia que salen de las compuertas en las que el proceso tiene que tomar una decisión, es decir que se asocian a los flujos de secuencia salientes. Para realizar la evaluación en cada uno de los flujos de secuencia se definirá una condición o expresión booleana. Posteriormente se definen los eventos de las actividades las cuales hacen referencia ha acciones que se pueden realizar al entrar, al guardar o al salir de una actividad. Entre las acciones que se tiene están: las expresiones, mensajes (notificaciones), políticas, cartas.

En el siguiente paso se definen los participantes o personas que serán encargadas de realizar cada una de las tareas en las cuales exista intervención humana. Para esto, es necesario definir ciertas características de la organización como cargos, ubicaciones geográficas, áreas de la organización, entre otras características.

Para concluir con el proceso de ejecución, es necesaria la configuración del tracing que es el encargado de generar los archivos de registro o LOGs de cada BP que se ha ejecutado. Para habilitar esta configuración cuenta con la opción de seguimiento o Tracing, que se encuentra en el menú estándar de Bizagi Studio, como se muestra en la Figura 3.

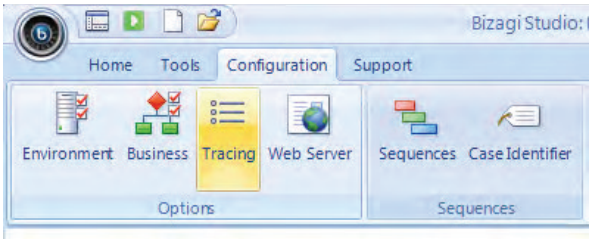


Figura 3. Tracing

Además, en el tracing es posible configurar las diferentes formas de como crear la traza, entre estas tenemos: por WorkFlow, EntityManager, Interfaces, Render, Rules, LDAP, Scheduler, o la combinación de todas. Tal como se muestra en la Figura 4.

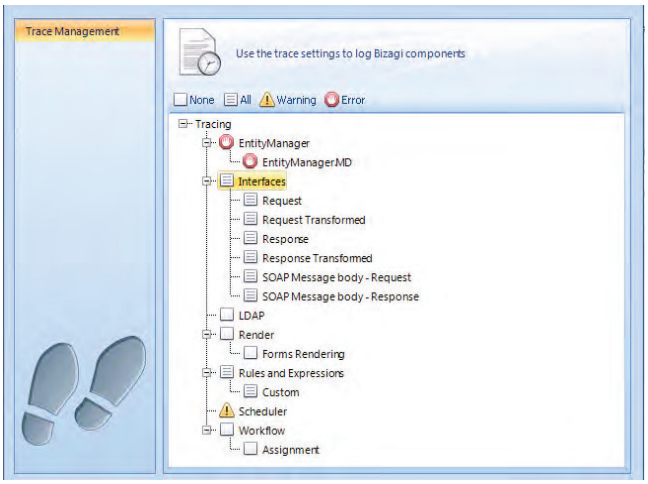


Figura 4. Configurar Tracing

Una vez, se realizo los pasos anteriores, se puede obtener la traza de ejecución de un BP sin errores, dando a conocer sus caminos, actividades, así como los diferentes componentes de un BP.

4 MODELO PROPUESTO

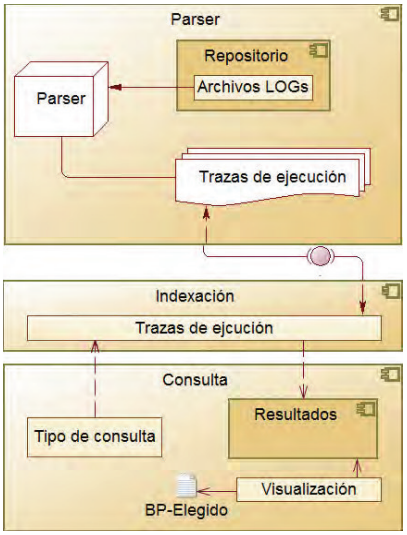


Figura 5. Arquitectura del modelo propuesto

El presente modelo se centra en aplicar una estrategia de búsqueda o descubrimiento de BP centrándose en la información de las trazas de ejecución que generan los BP, con el propósito que los BP recuperados sirvan de base para el replanteamiento re- modelamiento de uno nuevo.

El modelo esta compuesto de tres (3) Elementos como se muestra en la Figura 5 : ***Parser***, ***Indexación***, ***Consulta,*** las cuales son descritos a continuación.

Parser

Esta capa integra el repositorio de archivos LOGs resultado de la ejecución de los procesos de negocios (BP), en el cual se encuentran registradas las sentencias, actividades, información, tiempos de ejecución, optimizador utilizado, mensajes, llamadas a servicios, etc. que generan los BP una vez ejecutados. Además incorpora el algoritmo de parser que procesa los archivos y encuentra las trazas de ejecución. Los componentes y procesos principales se describen a continuación.

- **Repositorio:** Es usado como unidad central de almacenamiento y gestión, con similitud a una base de datos, en la que se encuentran un total de 120 archivos de registro obtenidos con la herramienta Bizagi.

- **Parser:** En este paso se ejecuta el algoritmo (Parser) que identifica el conjunto de trazas de ejecución de cada uno de los LOGs. El algoritmo lee secuencialmente cada archivos Log detectando cada caso de ejecución realizado en el BP_i (Un BP puede tener n casos ejecutados), cada caso detectado es analizado para formar una matriz de casos por BP denominada MC (ver Figura 6) en la cual cada fila representa un BP_i presente en el repositorio y cada columna a un c_j caso encontrado en el BP_i, una ves encontrado cada c_j este es analizado para encontrar las descripciones de las trazas de ejecución te_w que $\in c_j$. Con cada conjunto de trazas encontradas se construye una segunda matriz MTe (ver Figura 6 7) temporalmente donde cada fila representa cada c_j del BP_i y cada columna a te_w de cada c_j.

Indexación.

Este componente crea el índice en la organización de archivos del sistema operativo. En esta estructura se indexan las trazas de ejecución representadas en la MTe haciendo del algoritmo PorterStemming que se encarga de convertir cada uno de las descripciones de las trazas a su raíz léxica (por ejemplo "helping", "helped", en "help"), además de este proceso se remueven caracteres especiales, acentos, palabras vacías. El índice es similar a la matriz de términos por documentos del modelo vectorial en recuperación de información (RI), donde cada celda w_{ij} refleja la importancia del componente textual en su raíz léxica contra los archivos LOGs, basado en la propuesta por Salton, donde $F_{i,j}$ es la frecuencia observada del componente j en el BP_i. $Max\ (F_i)$ es la mayor frecuencia observada en el BP_i. N es el número de BP en la colección y n_j es el número de BP en los que aparece la traza j. ver Ecuación 1.

Bp_i	c_j				
$Bp_{..}$		c_j			
$Bp_{..}$			c_j		
$Bp_{..}$				c_j	
BP_n					c_n

Figura 6. Matriz MC

c_j	te_w				
$c_{..}$		te_w			
$c_{..}$			te_w		
$c_{..}$				te_w	
c_{jn}					te_w

Figura 7. Matriz temporal MTe

$$w_{i,j} = \frac{F_{i,j}}{\max(F_i)} \times \log\left(\frac{N}{n_j + 1}\right)$$

Ecuación 1.

Capa Proceso de Consulta

Esta capa es la encargada de proveer una interfaz para que el usuario realice consultas sobre el espacio que el elija (palabras claves o traza de ejecución completa). Para el proceso de consulta se toma el modelo de vectorial de recuperación de información que en la actualidad es el que mejores resultados aporta a las consultas realizadas por los usuarios. En esta parte tomamos a cada BP como una bolsa de palabras y la colección de BP se representa en la matriz de términos por documentos anteriormente descrita. La cercanía entre BPs contenidos en el repositorio y la consulta realizada por el usuario es calculada por medio de la distancia de cosenos formalizada en la Ecuación 2 ampliamente utilizada en el modelo vectorial de representación de documentos en el área de recuperación de información.

$$Sim\,(d,q) = Cos\,(\theta) = \frac{\sum_{i=1}^{M} W_{i,d} \times W_{i,q}}{\sqrt{\sum_{i=1}^{M} W^{2}_{i,d}}\sqrt{\sum_{i=1}^{M} W^{2}_{i,q}}}$$

Ecuación 2

Una vez los resultados son ordenados y filtrados se listan en orden de acuerdo a la similitud (más similares a menos similares) que presentan con respecto a la consulta realizada por el usuario, el cual puede elegir y visualizar la información procesada de cada archivo LOG recuperado, tomando conocimiento de las actividades o procesos encontrados para el remodelamiento de un BP nuevo.

5 HERRAMIENTA QUE SOPORTA EL MODELO PROPUESTO

La herramienta cuenta con una interfaz centrada en el usuario final, procurando incorporar los principales atributos que componen la Usabilidad. Atributos objetivos como facilidad de aprendizaje, facilidad de memorización, eficacia, eficiencia o tiempo empleado para completar una tarea, operabilidad, y facilidad de comprensión; y atributos subjetivos orientados a la satisfacción del usuario como, Accesibilidad, Funcionalidad, Utilidad, Estética y Credibilidad. Además cuenta con una interfaz sencilla y usable que esta conformada por paneles, que conforman las diferentes funcionalidades que permite el modelo. Por otra parte el usuario esta en la capacidad de elegir un modelo de BP de la lista de resultados para visualizar, y así comprobar la valides de su consulta.

En la Figura 8 se muestra una ejecución de proceso de consulta realizada por palabra clave, en esta los resultados son desplegados (recuadro rojo) una vez el usuario ha ejecutado la consulta, los resultados contienen los modelos de BP mas relevantes según el nivel de similitud entre la consulta y los BP contenidos en el repositorio

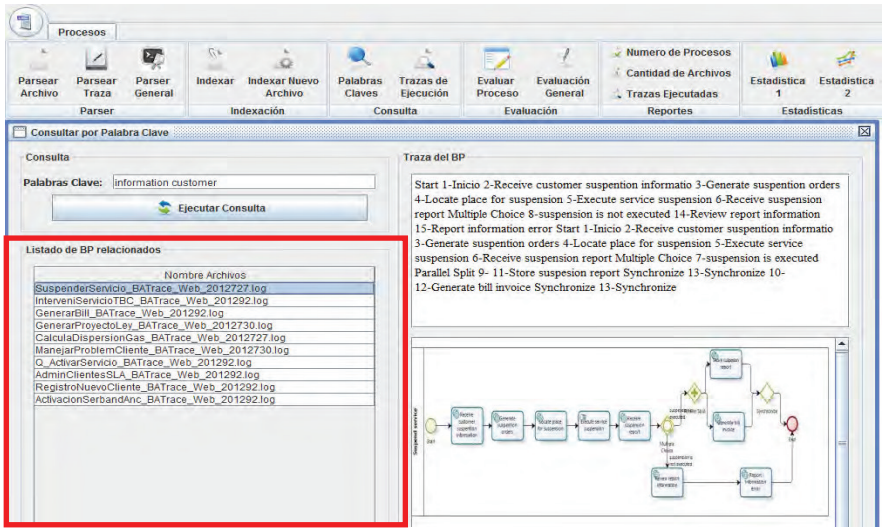


Figura 8. Opción de consulta por palabra clave

6 EVALUACIÓN DEL MODELO PROPUESTO

Para determinar la calidad del entorno fue necesario someterlo a un proceso de evaluación experimental, con el objetivo de verificar la eficiencia en el proceso de descubrimiento de BP con base al modelo de similitud definido para las opciones de consulta que permite el modelo.

La finalidad de la evaluación consiste en generar un ordenamiento (Ranking) de los 10 primeros modelos BP (dispuestos por orden de similitud) retornados para satisfacer una petición definida por medio de una de las opciones 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 evaluación de sistemas de recuperación de información [18]. Estas medidas son las medidas de efectividad: Precisión P, Recall R y Medida F [19]. Para esto se tiene:

- D, conjunto de documentos
- R, conjunto de documentos relevantes
- $\bar{R} = D - R$, conjunto de documentos no relevantes
- A, conjunto de documentos recuperados
- $A \cap R$, conjunto de documentos relevantes recuperados

Entonces:

La precisión mide la porción de documentos recuperados que son relevantes

$$precision = \frac{A \cap R}{A}$$

Recall mide la porción de documentos relevantes que son recuperados

$$recall = \frac{A \cap R}{R}$$

Medida F mide la armonía entre la precisión y el recall

$$F = \frac{2 \times recall \times precision}{recall + precision}$$

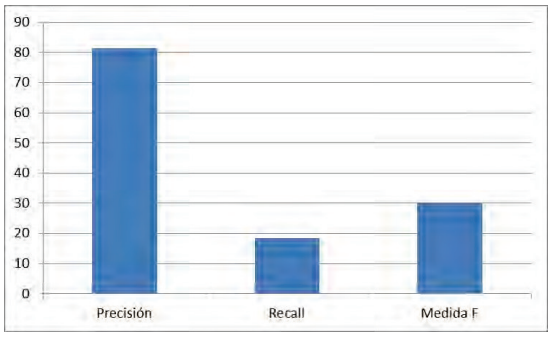


Figura 9. Resultados de evaluaciónFigura 9 muestra los valores obtenidos en la evaluación donde se tiene que el modelo alcanzó 81% de Precisión demostrando validas en las tareas de consulta y descubrimiento de BP. Por otra parte el 19% alcanzado en el Recall demuestra que el modelo deja de recuperar un número elevado de falsos positivos (BP que se recuperar erróneamente), además la Media F muestra que se tiene un 30% de armonía en los resultados retornados por el modelo.

7 CONCLUSIONES Y TRABAJO FUTURO

Se definió un modelo de recuperación de BP que utiliza los archivos de registros Log generados por Bizagi una de las suites de gestión de BP mas utilizadas y difundidas en el momento. El modelo propuesto demuestra que es posible recuperar modelos de BP que sirvan como base para el replanteamiento de uno nuevo, y así disminuir el tiempo de modelado.

La utilización de la trazas de ejecución en el descubrimiento de BP permiten visualizar y determinar si las tereas se desarrollan de manera correcta o incorrecta al interior de una organización o empresa. Los resultados de la evaluación permiten evidenciar el alto grado de validez del modelo. Alcanzando 81% de precisión en las consultas realizadas.

Como trabajo a futuro se pretende incorporar en el modelo la opción de semántica por medio de la incorporación de ontologías de domino especifico, con el propósito de tener una representación mas amplia en el momento de realizar las consultas. Además evaluar el modelo con base a una clasificación echa por jueces humanos donde se tenga una serie de consultas y sus resultados previamente definidos.

8 REFERENCIAS BIBLIOGRAFICAS

1. Y. Gong and M. Janssen, “From policy implementation to business process management: Principles for creating flexibility and agility,” *Government Information Quarterly*, vol. 29, pp. S61–S71, Jan. 2012.
2. R. Škrinjar and P. Trkman, “Increasing process orientation with business process management: Critical practices’,” *International Journal of Information Management*, Jun. 2012.
3. L. E. Mendoza, M. I. Capel, and M. a. Pérez, “Conceptual framework for business processes compositional verification,” *Information and Software Technology*, vol. 54, no. 2, pp. 149–161, Feb. 2012.
4. V. D. Aalst, Wil M, “Auditing 2.0: Using Process Mining to Support Tomorrow ’ s Auditor,” *Auditing*, vol. 32, pp. 713–732, 2007.
5. A. Koschmider, T. Hornung, and A. Oberweis, “Recommendation-based editor for business process modeling,” *Data & Knowledge Engineering*, vol. 70, no. 6, pp. 483–503, Jun. 2011.
6. S. Smirnov, M. Weidlich, J. Mendling, and M. Weske, “Action patterns in business process model repositories,” *Computers in Industry*, vol. 63, no. 2, pp. 98–111, Feb. 2012.
7. T. Raghu and a Vinze, “A business process context for Knowledge Management,” *Decision Support Systems*, vol. 43, no. 3, pp. 1062–1079, Apr. 2007.
8. O. M. G. D. Number and A. S. Files, “Business Process Model and Notation (BPMN),” *Business*, no. January, 2011.
9. O. M. G, “Business Process Model and Notation (BPMN),” *Business*, vol. 2, p. 530, 2011.
10. M. Chinosi and A. Trombetta, “Computer Standards & Interfaces BPMN : An introduction to the standard,” *Computer Standards & Interfaces*, vol. 34, no. 1, pp. 124–134, 2012.
11. C. Bose, R P Jagadeesh and V. D. Aalst, Wil M P, “Trace Alignment in Process Mining:,” *Framework*, vol. 6336, pp. 227–242, 2010.
12. H. a. Reijers, T. Freytag, J. Mendling, and A. Eckleder, “Syntax highlighting in business process models,” *Decision Support Systems*, vol. 51, no. 3, pp. 339–349, Jun. 2011.
13. A. Awad, A. Polyvyanyy, M. Weske, D.- Potsdam, and M. W. De, “Semantic Querying of Business Process Models,” *Process Technology*, pp. 85–94, 2008.
14. H. D. Zhengxing Huang, X.L., “Mining association rules to support resource allocation in business process management.,” *Expert Systems With Applications*, 2011.
15. and J. Z. Zichen Huan, J.H., Xudong Liu, “Business Process Decomposition based on Service Relevance Mining,” *IEE/WIC/ACM international Conference on Web Intelligence and Intelligent Agent Technology*, 2010.
16. N. H.-G. Dafne A. Rosso-Pelayo, R.A.T.-R., Miguel Gonzales-Mendoza, “Business Process MIning and Rules Detection for Unstructured Information,” *Ninth Mexican International Conference on Artificial Intelligence*, 2010.
17. G. Ramirez, “´ MULTINIVEL DE PROCESOS DE.”
18. R. Baeza-Yates A. and B. Ribeiro-Neto, *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999, p. 513.
19. H. Christopher D. Manning, Raghavan, Prabhakar, Schütze, *An Introduction to Retrieval Information*, no. c. 2008, p. 428.

Business Process Indexing based on Similarity of Execution Cases

Hugo Ordoñez
Universidad del Cauca
Calle 5 No. 4 - 70, 190003
Popayán, Colombia
hugoordonez@unicauca.edu.co

Juan Carlos Corrales
Universidad del Cauca
Calle 5 No. 4 - 70, 190003
Popayán, Colombia
jcorral@unicauca.edu.co

Carlos Cobos
Universidad del Cauca
Calle 5 No. 4 - 70, 190003
Popayán, Colombia
ccobos@unicauca.edu.co

Cristhian Figueroa
Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129
Turin, Italy
cristhian.figueroa@polito.it

Maurizio Morisio
Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129
Turin, Italy
maurizio.morisio@polito.it

Leandro Krug Wives
Universidade Federal do Rio Grande do Sul
Avenida Paulo Gama, 110
Porto Alegre, Brasil
wives@inf.ufrgs.br

ABSTRACT

This paper presents *EC-Indexer* a new approach for Business Process indexing based on execution traces extracted from event-log files. Additionally, a tool implementing the proposed similarity mechanism was developed in order to evaluate the effectiveness by common measures as precision, recall, and f-measure. The results showed that even when the *EC-Indexer* approach scored low values of recall, it could reach high values of precision while reducing the execution time.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

Keywords

Event Log, Execution Cases, Business Process Similarity

1. INTRODUCTION

Nowadays, enterprises offer a variety of products and services in order to remain competitive in the market [1], so they describe their internal processes as sets of tasks and functions that are involved in their business activities. Such representation is commonly known as Business Process (BP) [19], which constitutes a set of logically related activities. Those activities are executed following certain rules given by a control-flow described in the BP with the aim to produce valid outputs [17]. The outputs represent all the information generated by the executed processes, and can be stored as log files which contain information about the shared resources, execution times and dates, partners involved, connections with other systems, and executed transactions [1]. Experts commonly model BPs through powerful design tools that let them to describe operations or tasks needed to execute activities in the enterprises. For example, the *Bizagi BPM suite*[1] is a popular BP modeling and execution tool featuring a complete and easy to use environment [19].

But, despite the robustness of the design tools, there is still a lack of tools for studying dependence between activities of real execution cases (EC) of BPs registered in file logs. It is necessary as base for rethinking or remodeling a new BP meeting the current requirements of the enterprise [12], [18] and also allowing designers to effectively reuse previously developed BPs while reducing the design time [14].

In this paper, it is proposed a method to index BPs based on the execution cases generated after the execution of real BPs taking into account not only the textual information represented as names of task and events; but also the causal dependence between the elements of the BPs. Additionally it is computationally inexpensive and therefore can be effectively used as indexing mechanism for BPs similarity in order to reduce the search space.

The rest of the paper is organized as follows. We review the state of the art in section 2; the proposed approach and its architecture is presented in section 3; section 4 describes the effectiveness and performance evaluation of the proposed approach; and section 5 presents conclusions and future work.

EATIS '14 Valparaíso, Chile

[1] http://www.bizagi.com/

2. STATE OF THE ART OF BPS SIMILARITY TECHNIQUES

There are several approaches that study the similarity of BPs depending on different characteristics as linguistics [2], [6], structure [7],[11], [24], and behavior [10], [13]. However, the approach presented in this paper only takes into account the similarity of execution cases, and therefore this section was focused on approaches to calculate similarity of causal dependency between activities, and common sets of execution traces.

This section has been based on the work of Becker and Laue [5], that presents a more complete study about the main approaches to calculate BP similarity.

2.1 Behavior based on causal dependency between activities

Bae et al [4] defines a dependency graph to compare two BPs just as the difference in the number of arcs that links activities having a dependence relationship. However this approach does not take into account the type of gateways.

Weidlich et al [22] defines causal behavioral profiles that represent the dependencies between pairs of activities. The similarity is calculated taking each pair of activities in one BPs for which there are corresponding pairs of activities and analyzing if there are corresponding pairs sharing the same relations.

Dijkman et al [8] capture the precedence relations between activities as loopback links and causal footprints. The causal footprints are represented as vectors of index terms. This approach can build vectors of a high dimension making the method computational expensive.

Other approaches take into account the direct precedence of activities represented as Transition Adjacency Relation [25], n-grams [23], and behavioral profiles [22]. In this case, basically the similarity is calculated analyzing the correspondence between direct precedence of the activities in the trace.

2.2 Common sets of execution traces

Gerke et al [9], Wang et al [20] compare the compliance between BPs calculating the longest common subsequence of traces, i.e., the degree of similarity of ordering rules of activities between two BPs. However, this approach is computational expensive when there are large sets of traces.

Weerdt et al[21] review real execution traces of BPs with the purpose of discovering BPs, i.e., to try to infer which BPs can produce such traces.

Medeiros et al [3] compare BPs by studying the frequency of traces obtained from real executions or from simulations.

The *EC-Indexer* approach integrates the best characteristics of the aforementioned approaches by taking into account the causal dependence between activities of real execution cases (traces).

Additionally it is computationally inexpensive and therefore can be effectively used as indexing mechanism before the execution of more expensive algorithms for BPs similarity in order to reduce the search space.

Next section presents the architecture of the EC-indexer and describes its components.

3. EC-INDEXER: AN APPROACH FOR INDEXING BPS BASED ON SIMILARITY OF EXECUTION CASES

The "*EC-Indexer*" approach indexes BPs with execution cases (traces) retrieved from their log files. A log file is created the first time a BP is executed, and it is updated by adding a new execution case each time the BP is executed again. An execution case registers information about a specific BP execution (i.e. what activities happened in a certain moment in time during the BP execution)[21]. Thus, a BP contains only one log file, but multiple execution cases included in the log file. This paper contributions are twofold: i) it shows the BPs indexing generation based on the execution cases, and ii) it describes how the *EC-Indexer* approach can be used to rank a set of BPs successfully executed in concordance with their degree of compliance with a query representing the execution behavior of a BP. The "*EC-Indexer*" is limited to only real execution cases extracted from logs of BPs stored in a repository. Section 3.1 details the index creation as well as the procedure to query the index so that the stored BPs can be ranked.

3.1 Creating the index of execution cases

Let $R = \{BP_1, ..., BP_i, ...BP_m\}$ be a repository of BPs. Each $BP_i \in R$ contains a log file $l_i = \{ec_{i1}, ..., ec_{ij}, ...ec_{ik}\}$ that is updated each time the BP_i is executed by adding a new execution case ec_{ij}. Each execution case is composed of a sequence of elements of the BP (nodes) that can be activities and gateways (XOR (Join-Split), AND(Join-Split)) which are ordered just as the execution flow followed by the BP.

The first step in the BP indexing mechanism is to collect all the nodes of each execution case $ec_{ij} = \{n_1, ..., n_p\}$ and form pairs of nodes keeping their causal dependence (i.e., adjacent nodes in the execution case). For example, in ec_{ij} the set of node pairs is $PS_{ij} = \{(n_1, n_2), (n_2, n_3), ..., (n_{i-1}, n_i), (n_i, n_{i+1}), ..., (n_{p-1}, n_p)\}$.

After collecting the pairs of the execution cases of the entire repository, a matrix named "execution cases matrix" M_{ec} is created. The columns of this matrix are the node pairs found in the execution cases for the entire repository avoiding those repeated (i.e. there are not two columns representing identical node pairs), and the rows are all the BPs stored in the repository. Therefore the size of the matrix is $m \times k$ where m is the number of BPs in the repository, and k is the number of all pairs found in execution cases minus those which are repeated.

Finally, the matrix M_{ec} is filled by placing the number of times a pair is found in the execution cases of a given BP, e.g., if a pair p_j is found three times in the log $l_i \in BP_i$, then the number 3 is inserted on the cell (i, j) (Figure 1). Thus, the index of execution cases is created and represented by the matrix M_{ec}.

In this paper, this matrix is similar to the "term-document matrix" of the vector space model in the Information Retrieval (IR) field proposed by Salton in 1989.

Therefore the M_{ec} matrix can be normalized in the same way as the "term-document matrix", which is composed of cells w_{ij} representing textual components (in their lexical root) detected in a log-file. Then, each w_{ij} is weighted with the equation 1 [15], where F_{ij} is the observed frequency in the component j of the BP_i; $Max(F_i)$ is the highest ob-

	p_1	p_2	...	p_j	...	p_k
BP_1	0	0	...	2	...	1
BP_2	3	0	...	0	...	0
...	...	...	...	..	...	...
BP_i	2	0	...	3	...	2
...	...	...	...	..	...	...
BP_n	1	5	...	0	...	3

Figure 1: Example of the execution cases matrix

served frequency of the BP_i; N is the number of BP in the repository; and n_j is the number of BP in which the execution case j has been detected.

$$w_{ij} = \frac{F_{ij}}{max(F_i)} \times log\left(\frac{N}{n_j + 1}\right) \quad (1)$$

3.2 Querying the index of execution cases

To query the index of execution cases a query set of node pairs (PS_q) is required. The set PS_q is processed in order to find repeated node pairs, and to create a query vector v_q that registers the number of occurrences of each pair. For example, let $PS_q = \{p_{q1}, p_{q2}, ..., p_{qi}, ...p_{qt}\}$, if $p_{q1} = p_{q2}$ then the number of occurrences of p_{q1} is 2. This value is then inserted in the corresponding cell for the p_{q1}. Figure 2 shows an example of a query vector.

Subsequently, each pair of the vector v_q is searched in the index (matrix M_{ec}) in order to obtain the number times it is found in each BPs stored in the repository. This number is then multiplied by the corresponding value in the vector v_q, and the resulting value is inserted in a new matrix named "query matrix" (M_q) whose rows are the BPs of the repository and columns are the node pairs of the query vector v_q. For example, lets take the vector query of figure 2 and the execution cases matrix of figure 1, and suppose that $p_{q1} = p_1, p_{q3} = p_2, p_{qj} = p_j$ and $p_{qt} = p_k$. The pair p_{qj} with an occurrence of 2 in the vector query v_q is found three times in the execution cases matrix, hence by multiplying those values we get 6; this value is inserted on the cell (i, j) of the query matrix M_q. Finally, in order to rank the BPs of the repository, the values of each row are added obtaining a value of execution cases similarity (*ec-sim*) for each BPs. Accordingly, the BPs are ranked from the greatest value to the lowest one. The complete resulting query matrix of the example is presented in figure 3 where the resulting ranking is $r = \{BP_i(10), BP_n(7), BP_2(6), BP_1(4)....\}$.

3.3 Architecture of the EC-Indexer

The *EC-Indexer* approach allows for indexing and searching BPs stored in a repository according to their similarity to a query represented as a set of node pairs (PS_q). Three kinds of query options are supported in this approach: keywords, minimal behavior, and log-files.

p_{q1}	p_{q3}	...	p_{qj}	...	p_{qt}
2	1	...	2	...	0

Figure 2: Example of the query vector

	p_{q1}	p_{q3}	...	p_{qj}	...	p_{qt}	**ec-sim**
BP_1	0	0	...	4	...	0	**4**
BP_2	6	0	...	0	...	0	**6**
...	...	...	...	...	...	...	...
BP_i	4	0	...	6	...	0	**10**
...	...	...	...	..	...	...	...
BP_n	2	5	...	0	...	0	**7**

Figure 3: Example of the query matrix plus the similarity for each BP

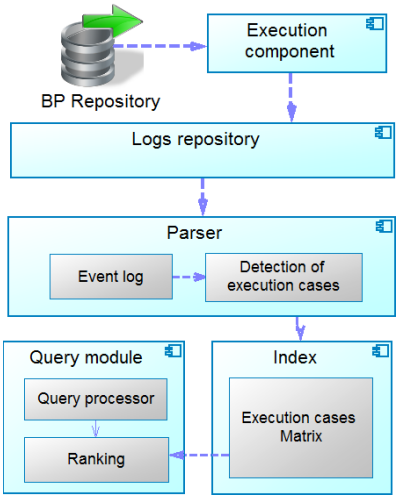


Figure 4: Architecture of the *EC-Indexer* indexing

The *EC-Indexer* works both in the indexing phase and at the querying phase (sections 3.1 and 3.2). In the indexing phase, logs of the BPs stored in the repository are indexed and an execution-cases matrix M_{ec} is generated. In the querying phase, a query (keyword, minimal behavior, or log file) is received and processed in order to it to obtain a set of node pairs. Lastly, when the set of node pairs is obtained, the query matrix is generated M_q and the repository is ranked.

The architecture of the *EC-Indexer* (Figure 4) is described as follow:

3.3.1 BP repository

It stores a set of BPs that are executed and processed in order to extract the log-files containing the execution cases. The current implementation of the repository includes 120 BPs modeled with BPMN (Business Process Modeling Notation). Those BPs were graphically designed by experts of the Telematics Engineering Group of the University of Cauca (Colombia) based on real processes provided by Telco operators in Colombia and examples found in different web sites (e.g the TM Forum[2]). It was not possible to use a real repository of a Telco operator because operators are reluctant to give access to their repositories due to privacy and security policies.

3.3.2 Execution component

This component executes the BPs of the repository and

[2]www.tmforum.org

collects the log files containing the execution cases. The current version of this component is implemented by the *Bizagi BPM suite* that is considered as one of the most used tools to model and to automate BPs[19]. In this fashion, the BPs were executed in the lab (to simulate the real execution scenario) and log files were then stored in a second repository named "logs repository".

3.3.3 Logs repository

This repository stores all the log files obtained in the execution of the BPs of the repository. Each BP contains only one log file, nevertheless one log file contains multiple execution cases. The current implementation of this repository stores the log files in the file system.

3.3.4 Parser

This component extracts and process the execution cases from each log file stored in the "logs repository" (i.e., each execution log of the BPs stored in the BP repository). The extracted execution cases are inserted on vectors (*execution case vectors*) that associate each execution case with the BP that contains it. Afterwards, each execution case vector is processed to form pairs of adjacent nodes in order to keep causal relationships. Once the node pairs are formed, they are arranged together with node pairs of other execution cases in the same BP in order to create a new vector (*node pairs vector*) that relates a BP with its set of node pairs. This procedure is repeated for the entire BP repository, obtaining a one vector of node pairs for each BP.

3.3.5 Indexer

This component processes the vectors of node pairs and generates an index. First at all, the node pairs of each vector are analyzed with the *Porter Stemming*[10] algorithm that transforms the labels of the nodes to their lexical root (e.g. words "helping" and"helped" are transformed to their lexical root "help"), and removes special characters, void words, and accents. Further, the indexer creates a "matrix of execution cases" (M_{ec}) whose rows are the BPs stored in the repository, and the columns are the node pairs of all the BPs of the repository but avoiding the pairs that are duplicated. The matrix M_{ec} is filled by counting the number of times that a pair is found in each BP (i.e. in the vector of node pairs of each BP).

3.3.6 Query module

This module has two functions: the first one is transforming the supported types of queries to node pairs in order to create a query matrix (M_q) containing information about the number of times each pair is repeated in the query; and the second one is ranking the BPs of the repository according to their degree of similarity to the query.

- **Query processor:** this module receives a query so that it is transformed into a set of node pairs. The current implementation of the query module supports the three kinds of queries described bellow:
  - *Execution case:* this option supports a textual string that represents a BPs execution case. Therefore, the string must contain a sequence of nodes (activities and gateways) that are further transformed to a set of node pairs.
  - *Minimal behavior of execution flow:* this option offers a list of node pairs obtained from the execution cases of the BPs in the repository. Then a user can choose a combination of node pairs to build a query.
  - *Log file:* this option allows to introduce a log file that is processed to identify the execution cases and subsequently the sets of node pairs. In this option, the user can choose one of the found sets of execution cases in order to rank the BPs in the repository that have executed similar execution cases.

  Once, the query is transformed, the set of node pairs are processed with the "PorterSteeming" algorithm as explained before. Then, the duplicated pairs are counted and inserted in a query vector which contains the number of occurrences of each pair.

- **Ranking:** the ranking takes the query vector v_q and the execution cases matrix M_{ec} as basis for generating the query matrix (M_q) as described in the section 3.2. The M_q matrix allows to get a similarity value for each BP of the repository with the query, and therefore the BPs are ordered from the greatest degree of similarity to the less one.

4. EXPERIMENTAL EVALUATION AND RESULTS

This section focuses on the evaluation of the effectiveness and performance of the *EC-Indexer* approach. To this aim a java-based implementation of the mentioned approach was developed and measures used in IR filed were considered. The implementation was developed conforming to the reference architecture described in section 3.3 taking into account the main aspects of usability in order to improve the user-experience [16]. Though this implementation it was possible to generate multiple rankings considering the different types of queries the approach supports. The effectiveness of those rankings was evaluated by applying the measures precision (p), recall (r), and f-measure (f) which are widely used in IR systems [15].

4.1 Results

This section presents the results of both the effectiveness and the performance tests.

4.1.1 Effectiveness test

Table 1 shows the results of recall, precision, and f-measure presented as percentages.

Query option/Measure	p	r	f-measure
Execution case	81	19	30.78
Minimal behavior	84	23	36.13
Log-file	90	28	33.42

Table 1: Results for each query option

As can be seen in table 1 the different query options scored values of precision between 81% and 90% demonstrating the *EC-Indexer* approach avoids to retrieve non-relevant BPs (i.e. false positives). Nevertheless, the lower values of recall

from 19% to 28% demonstrate that the approach also loose a high number of relevant BPs (i.e. false negatives).

Regarding f-measure, the approach scored values from 30% to 42% for the different query options showing acceptable values of harmony between the precision and recall measures.

4.1.2 Performance Test

This test evaluates the execution time consumed by the *EC-Indexer* approach to rank the 120 BPs of the repository with multiple queries. Figure 5 shows the results of this test, where can be seen that the execution time is practically low with values between 18 and 39ms, and it does not presents a linear relation to the number of nodes of the BPs. It means that the *EC-Indexer* approach can be used as an initial step in a BPs retrieval system due its low execution times.

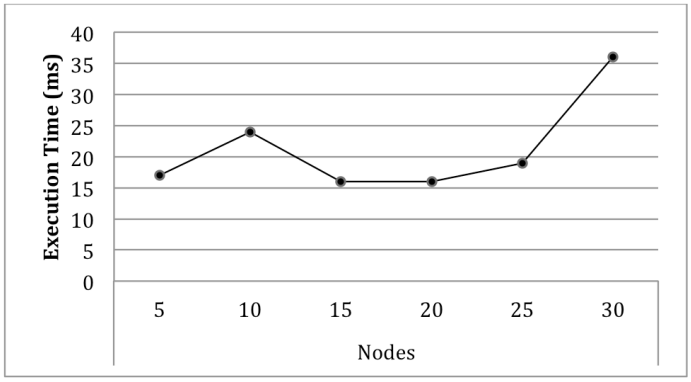


Figure 5: Results of the peformance evaluation

5. CONCLUSIONS AND FUTURE WORK

In this paper it is presented the *EC-Indexer* approach to rank BPs based on execution cases extracted from log files. The analysis of the execution cases allows enterprise users to visualize and study the real execution behavior of their BPs. The *EC-Indexer* approach scored high values of effectiveness,(a precision of 90% for the different query options) while reducing the execution time. Additionally, the f-measure scored values around 42% which is an acceptable value for the relation between precision and recall, i.e. acceptable values of false negatives and false positives. Therefore, it can be concluded that discovering BPs taking into account the execution cases can be used as filter phase for posterior exhaustive analysis of other characteristics of BPs such structure or more advanced behavior-based models. Additionally, *EC-Indexer* approach can be extended by adding new query options. For example, as future work can be incorporate a semantic option by adding domain ontologies in order to represent queries in a broader format; and a multimodal option which can incorporate structural, behavioral, and linguistic information in only one search space.

6. ACKNOWLEDGMENTS

The authors would like to thank Telematics Engineering Group of the Universidad del Cauca[3], Software Engineering Group of the Politecnico di Torino[4], and COLCIENCIAS[5] for supporting the research projects which were the base of this paper.

7. REFERENCES

[1] W. Aalst. *Configurable Services in the Cloud: Supporting Variability While Enabling Cross-Organizational Process Mining*, volume 6426 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2010.

[2] R. Akkiraju and A. Ivan. Discovering business process similarities: An empirical study with sap best practice business processes. In P. Maglio, M. Weske, J. Yang, and M. Fantinato, editors, *Service-Oriented Computing*, volume 6470 of *Lecture Notes in Computer Science*, pages 515–526. Springer Berlin Heidelberg, 2010.

[3] A. Alves de Medeiros, W. M. van der Aalst, and A. Weijters. Quantifying process equivalence based on observed behavior. *Data & knowledge engineering*, 64(1):55–74, 2008.

[4] J. Bae, L. Liu, J. Caverlee, L.-J. Zhang, and H. Bae. Development of distance measures for process mining, discovery and integration. *International Journal of Web Services Research (IJWSR)*, 4(4):1–17, 2007.

[5] M. Becker and R. Laue. A comparative survey of business process similarity measures. *Computers in Industry*, 63(2):148–167, Feb. 2012.

[6] P. Châtel. Toward a semantic web service discovery and dynamic orchestration based on the formal specification of functional domain knowledge. In *International Conference on Software & Systems Engineering and their Applications (ICSSEA)*, 2007.

[7] R. Dijkman, M. Dumas, and L. García-Bañuelos. Graph matching algorithms for business process model similarity search. In *Business Process Management*, pages 48–63. Springer, 2009.

[8] R. Dijkman, M. Dumas, B. Van Dongen, R. Käärik, and J. Mendling. Similarity of business process models: Metrics and evaluation. *Information Systems*, 36(2):498–516, 2011.

[9] K. Gerke, J. Cardoso, and A. Claus. Measuring the compliance of processes with reference models. In *On the Move to Meaningful Internet Systems: OTM 2009*, pages 76–93. Springer, 2009.

[10] S. Goedertier, D. Martens, J. Vanthienen, and B. Baesens. Robust process discovery with artificial negative events. *Journal of Machine Learning Research*, 10:1305–1340, 2009.

[11] D. Grigori, J. C. Corrales, M. Bouzeghoub, and A. Gater. Ranking bpel processes for service discovery. *Services Computing, IEEE Transactions on*, 3(3):178–192, 2010.

[12] A. Koschmider, T. Hornung, and A. Oberweis. Recommendation-based editor for business process modeling. *Data and Knowledge Engineering*, 70(6):483 – 503, 2011.

[13] M. Kunze, M. Weidlich, and M. Weske. Behavioral similarity–a proper metric. In *Business Process*

[3]http://www.git.unicauca.edu.co

[4]http://softeng.polito.it

[5]http://www.colciencias.gov.co

Management, pages 166–181. Springer, 2011.

[14] R. Laue and A. Awad. Visual suggestions for improvements in business process diagrams. *Journal of Visual Languages and Computing*, 22(5):385 – 399, 2011.

[15] C. D. Manning, P. Raghavan, and H. Schütze. *An Introduction to Information Retrieval*. Cambridge Univrsity Press, 2009.

[16] A. Raza, L. F. Capretz, and F. Ahmed. An open source usability maturity model (os-umm). *Comput. Hum. Behav.*, 28(4):1109–1121, July 2012.

[17] H. Reijers, R. Mans, and R. van der Toorn. Improved model management with aggregated business process models. *Data and Knowledge Engineering*, 68(2):221 – 243, 2009.

[18] D. Rosso-Pelayo, R. Trejo-Ramírez, M. Gonzalez-Mendoza, and N. Hernandez-Gress. Business process mining and rules detection for unstructured information. In *Artificial Intelligence (MICAI), 2010 Ninth Mexican International Conference on*, pages 81–85, 2010.

[19] L. Sánchez-González, F. García, F. Ruiz, and J. Mendling. Quality indicators for business process models from a gateway complexity perspective. *Inf. Softw. Technol.*, 54(11):1159–1174, Nov. 2012.

[20] J. Wang, T. He, L. Wen, N. Wu, A. H. Ter Hofstede, and J. Su. A behavioral similarity measure between labeled petri nets based on principal transition sequences. In *On the Move to Meaningful Internet Systems: OTM 2010*, pages 394–401. Springer, 2010.

[21] J. D. Weerdt, M. D. Backer, J. Vanthienen, and B. Baesens. A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs. *Information Systems*, 37(7):654 – 676, 2012.

[22] M. Weidlich, A. Polyvyanyy, J. Mendling, and M. Weske. Efficient computation of causal behavioural profiles using structural decomposition. In *Applications and Theory of Petri Nets*, pages 63–83. Springer, 2010.

[23] A. Wombacher and M. Rozie. Evolution of workflow similarity measures in service discovery. In M. Schoop, C. Huemer, M. Rebstock, and M. Bichler, editors, *Konferenz im Rahmen der Multikonferenz Wirtschaftsinformatik, 2006*, volume P-80, pages 57–71, Bonn, Germany, February 2006. Gesellschaft fuer Informatik.

[24] Z. Yan, R. Dijkman, and P. Grefen. Fast business process similarity search with feature-based similarity estimation. In *On the Move to Meaningful Internet Systems: OTM 2010*, pages 60–77. Springer, 2010.

[25] H. Zha, J. Wang, L. Wen, C. Wang, and J. Sun. A workflow net similarity measure based on transition adjacency relations. *Computers in Industry*, 61(5):463–471, 2010.

Collaborative Grouping of Business Process Models

Hugo Ordonez
University of Cauca,
Sector Tulcán, Popayán, Colombia,
Edifice FIET
Office 422 57-2-8209800x2119
hugoordonez@unicauca.edu.co

Juan Carlos Corrales
University of Cauca,
Sector Tulcán, Popayán, Colombia,
Edifice FIET
Office 418 57-2-8209800x2124
jcorral@unicauca.edu.co

Carlos Cobos
University of Cauca,
Sector Tulcán, Popayán, Colombia,
Edifice FIET
Office 422 57-2-8209800x2119
ccobos@unicauca.edu.co

Leandro Krug Wives
Universidade Federal do Rio Grande
do Sul, Caixa Postal 15.064, Porto
Alegre, RS, Brasil
wives@inf.ufrgs.br

ABSTRACT

This paper presents a collaborative platform that allows a set of judges (evaluators) to form business process (BP) groups based on the relationship of a user's query with the BPs stored in a repository and the relevant results to that query. Queries are expressed as complete BP and are presented to the evaluators in order to allow them to form groups of BP taking into account similarity relationships. Additionally, each evaluator can compare the concordance or discordance of his results with relevance judgments issued by other evaluators; in this way evaluator can collaborate in the global evaluation process or change his evaluations. Results of the evaluation can be used to assess the quality of the results retrieved by an automatic BP similarity tool. The proposed platform was evaluated with a set of 54 users and results are promising.

Keywords

Business Process, Grouping, Platform, Collaborative Evaluation.

1. INTRODUCTION

. Business process (BP) models capture the set of procedures or interrelated activities that collectively develop a common business goal within the context of an organizational structure [1]. Commonly, BPs is stored in closed repositories that may contain large quantities of reusable BP models.

However reusing BP models requires search mechanisms effective capabilities to find similarities between them. Some of these mechanisms apply data mining techniques (e.g, clustering) to build a BP model group hierarchy based on the similarity of operational characteristics found in BP models [2].

Most of the BP Model searching mechanisms for clustering have been evaluated theoretically based on laboratory tests, i.e., the evaluation addresses mainly the quality and performance on automatic BP models clustering [3] .

EATIS'14, April 02 - 04 2014, Valparaíso, Chile

ACM 978-1-4503-2435-9/14/04.
http://dx.doi.org/10.1145/2590651.2590686

In this sense this paper presents a platform that allows a set of judges (evaluators) to manually group BP model groups in a collaborative environment named "CollaborativeGroupBP". Accordingly, the formed groups by evaluators provide a basis to check validity and consistency of groups automatically crated by different BP clustering mechanisms.

2. PROPOSED PLATFORM

The proposed platform, providing an infrastructure integrates a group of judges in a collaborative environment in order to create and retrieve BP model groups defined by them manually in a consensus. The platform functionality is divided into two modules (Figure 1).

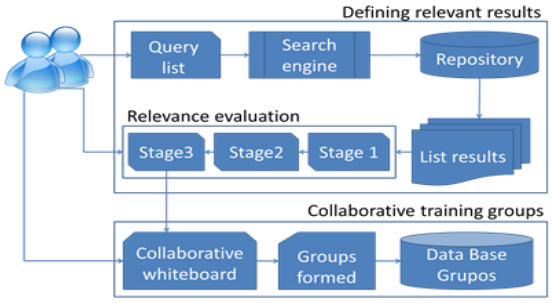


Figure 1. Collaborative Platform Architecture

- **Collaborative group's formation**

This module use the results considered as relevant to each query made by judges in the previous phase. From these results, the platform provides to judges the functionality of BP model group's creation, through a consensus on a collaborative environment (see Figure 2). Every created group is defined by a name and some labels of the elements belonging to it. Judges can make changes in the name of each group. Each judge can select one or more items in the results list and drag them to the group; similarly. A judge can use the remove option to remove one or more items of the group. In addition, this module has a chat room where members can share points of views to define group name and the items within it. All of these functionalities are available in the Collaborative whiteboard. The Groups formed component acts as an intermediary between the collaborative whiteboard and the database; it captures all the attributes of each group created, finally, the Group's database component stores the transactions records and the groups formed by the judges in each one of the platform modules.

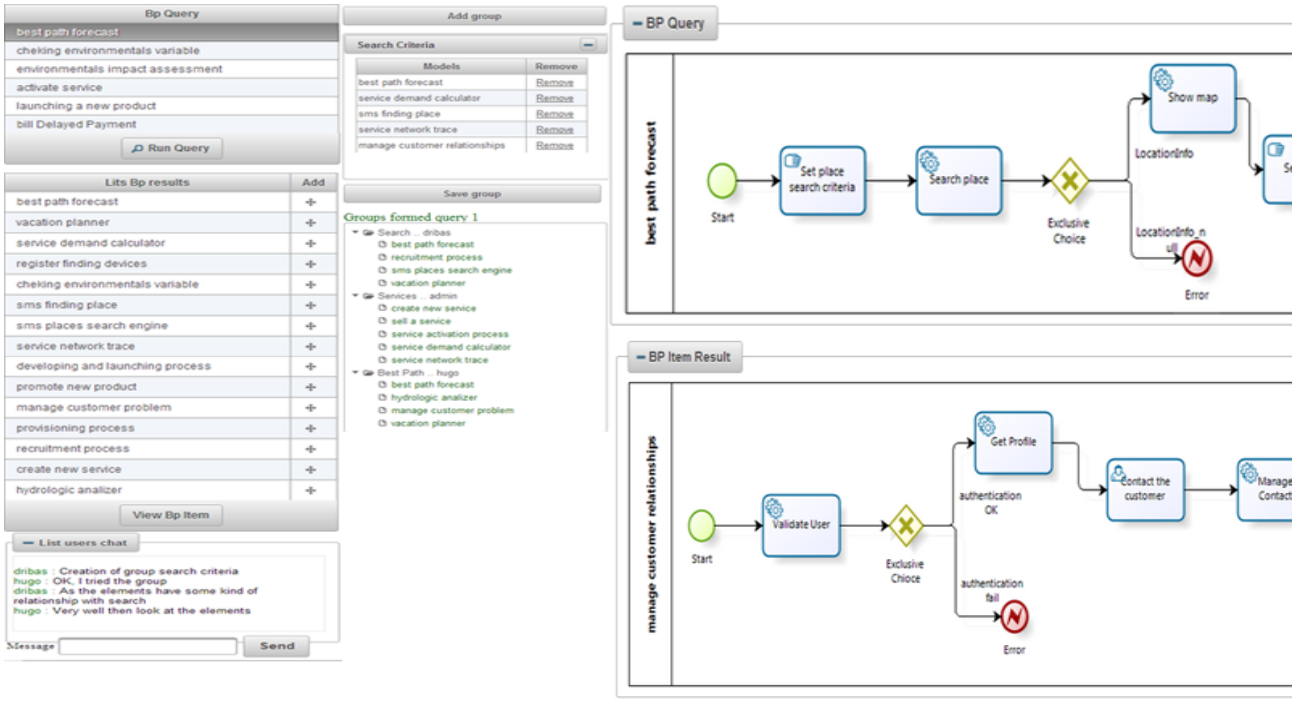


Figure 2. Manual groups' creation interface

3. USE EXPERIENCE

Regarding to group's formation we count with 54 people experts on BP management and modeling topics belonging to the Institute of Informatics (21) and School of Business Administration (30) of the Federal University of Rio Grande do Sul (Brazil) and (3) of Telematics Department of the University of Cauca (Colombia). We defined a set of 6 BP models as query items and lists of 20 results sorted by the search echanisms. Then was explained the platform operation and its goals and finally was executed in a coordinated way the relevant results definition phase. In this phase, each judge performed an average of 360 manual comparisons, for a total number of 19440 comparisons among all judges. Table 1 shows the consensus evaluations made by judges in the relevant results definition phase and in the group's formation. Based on the relevant results we got an average of 12.33 items considered as relevant to each query. Moreover, based on the number of the relevant results, the average of formed groups by query is 3.5; additionally, the average of elements belonging to each group is 3,55. This is due to the agreements number on judges' joint work, in the groups' creation and the elements' choice that belongs to each group.

Table 1. Relevant elements and groups formed by query

Queries	Relevant elements	Formed groups	Elements by group
Q1	10	3	3,33
Q2	12	4	3
Q3	13	3	4,33
Q4	11	3	3,66
Q5	14	4	3,5
Q6	14	4	3,5

The quantity of groups formed by each query allows having a test dataset partitioned in sets, where the elements belong to a specific group that is identified by a unique label. Generating groups by judges helps to know a priori the complete set of possible elements that are part of the cluster for a specific query. Consequently, these groups can be used to evaluate the overall group's formation quality of a model automatic grouping (i.e. accuracy).

4. CONCLUSIONS

A manual creation of BP groups by judges using the collaborative platform allows to add items that truly are in the context of the group they belong. The consensus formation of BP model groups allows the overlap decreasing (existing elements in more than one group), generating thereby coherent groups. Using results considered as relevant in a query allows increasing the consistency in the groups and decreasing the time in group's creation. Finally, formed groups can serve as basis for comparison (precision's evaluation) of any clustering BP mechanism

5. REFERENCES

1. Laue, R. and A. Awad, Journal of Visual Languages and Computing Visual suggestions for improvements in business process diagrams. Journal of Visual Language and Computing, 2011. 22: p. 385-399.
2. Leonardi, S.M.a.G., Retrieval and Clustering for Business Process Monitoring: Results and Improvements. Springer-Verlag Berlin Heidelberg 2012.
3. Qiao, M., R. Akkiraju, and A.J. Rembert, Towards Efficient Business Process Clustering and Retrieval: Combining Language Modeling and Structure Matching. 2011. 57: p. 199-214.

Collaborative Evaluation to Build Closed Repositories on Business Process Models

Hugo Ordoñez[1], Juan Carlos Corrales[1], Carlos Cobos[2], Leandro Krug Wives[3] and Lucineia Thom[3]
[1]*Telematics Engineering department, University of Cauca, Sector Tulcán, Popayán, Colombia*
[2]*Systems engineering department, University of Cauca, Sector Tulcán, Popayán, Colombia*
[3]*Institute of Informatics, Federal University of Rio Grande do Sul, Caixa Postal 15.064, Porto Alegre, RS, Brazil*
{hugoordonez, jcorral,ccobos}@unicauca.edu.co, {wives, lucineia}@inf.ufrgs.br

Keywords: Business Process Relevance, Business Processes Management, Collaborative Methodology, Business Process Search Evaluation.

Abstract: Nowadays, many companies define, model and use business processes (BP) for several tasks. BP management has become an important research area and researchers have focused their attention on the development of mechanisms for searching BP models on repositories. Despite the positive results of the current mechanisms, there is no defined collaborative methodology to create a closed repository evaluation for these search mechanisms. This kind of repository contains some closed BP predefined lists representing queries and ideal answers to these queries with the most relevant BPs based on a set of evaluation metrics. This paper describes a methodology for creating such repositories. To apply the proposed methodology, we built a Web tool that allows to a set of evaluators to make relevance judgments in a collaborative way for each one of the items returned according to predefined queries. The evaluation metrics used can measure the consensus degree in the results, therefore confirming the methodology feasibility to create an open access, scalable and expandable closed BP repository with new BP models that can be reusable in future research.

1 INTRODUCTION

Currently, many companies define, model, and use business processes (BP) for several tasks such as manufacturing, services, purchasing, inventory management and others. With the advances in technology development, the impact of BP management has become an increasingly important research area in academic and business fields. As a result, big effort has been dedicated to the development of mechanisms to search and discover reusable components (Škrinjar and Trkman 2012) for defining new BP adjustable to current requirements of the organization. These efforts are aimed at providing companies a starting point to improve their trading activities.

Therefore, these mechanisms should be evaluated to find their inconsistencies, fix them and ensure the proper implementation of their functional purpose. Besides, there is still a lack of closed repositories in business process evaluation that would allow to compare the performance of two or more BP searching techniques in the same conditions. This also could help to find the shortcomings and to make improvements to these techniques.

This paper presents a collaborative evaluation methodology to build closed repositories. It also presents and discusses the outcomes obtained after applying the proposed methodology. To this end, we have developed and used a tool that implements this methodology and uses a BP searching mechanism to return a smart BPs list created with the BPs to be evaluated on each query. Thus, evaluators do not have to evaluate all existing BPs within the repository.

The methodology is proposed to build closed repositories' evaluation while taking into account the opinion of an expert group from a collaborative perspective. In this sense, each expert makes relevance judgments between BPs reported as results by a searching mechanism and a BP defined as query. Then the BP query mechanisms can use the repository to evaluate the quality in their searching process.

This paper presents two specific contributions: first, an evaluation methodology to create closed repositories of BPs taking into account the opinions

of a group of experts; and, second, an open access BP repository (motivated by the approach proposed in(Kunze and Weske 2012)) with a hundred BP models from the telecommunications and geo-referencing domain.

The rest of the paper is organized as follows: Section 2 describes related work and evaluation methodologies for BP model searching mechanisms. Section 3 presents the proposed methodology for collaborative assessment. Section 4 describes a Web tool specially developed to allow the projected methodology's application. Section 5 describes the repository. Section 6 describes a case study, and Section 7 presents the conclusions and future works that are expected in the short term.

2 RELATED WORK

Despite the progress in the development of tools for searching and discovering BPs (Rosa, Arthur et al. 2010; Kunze 2013), to date there are no formal methodologies to evaluate these mechanisms.

Regarding the above, some related works propose evaluation methodologies and experimental setups centered on the evaluation of tools for discovering Semantic Web Services (SWS).

Consequently, these experimental setups can serve as a starting point to create a formal evaluation methodology for the results reported by BP searching tools.

2.1 Evaluation on BP Searching

Regarding the BP searching task, some metrics have been defined to measure or evaluate the degree of precision and relevance of the results reported by proposals for finding similarities between BPs (Dijkman et al., 2011); (Becker and Laue, 2012). Among those proposals are: linguistic, focused on the name or description of each BP element (Koschmider et al., 2011); association rules, focused on the historical execution of BP tasks which are recorded in log files; and genetic algorithms that integrate more data as inputs, outputs, edges, and nodes in the search process (Turner, 2010). In addition to these proposals, there are further approaches centered on searching BP models within repositories using proprietary languages or methods for excecuting queries (La Rosa et al., 2011); (Yan et al., 2012)

2.2 Evaluation Methodologies

In (Tsetsos et al., 2006), for instance, an evaluation system for Semantic Web Services (SWS) discovery based on information retrieval (IR) theories is proposed. there two similarity schemes are evaluated: 1) A Boolean schema that sets two values, 0 or 1 for similarity degrees, and a correspondence between a query service and a comparison service, where "1" means that two services have some level of affinity, and "0" when they have no affinity; 2) A scale of similarity values (i.e., numerical values in the range [0-1], corresponding to fuzzy terms like "relevant", "irrelevant", and so on) that allows us to sort the results according to similarity levels, which present the query services and a comparison service. In this case, the evaluation is made according to the equivalence between the services sorted by the experts and the result obtained by the tool.

In (Küster and König-Ries, 2009)a services collection is shown. This collection contains three different evaluation scales that were used to classify the relevance of the reported results in a query. They have used three schemes: 1) A binary one, which has been most commonly used, where “1” determines that there is a degree of relevance and “0” that there is no relevance at all; 2) One-dimensional graded relevance that is a multi-valued scale to measure the similarity between two services; 3) A Multi-dimensional graduate importance, which provides a multi-scale to evaluate different aspects (equivalence, scope and interface, among others) between two services.

Moreover, (Dijkman et al., 2011)state that there is a considerable research gap for comparing different approaches for searching BPs because the evaluation process has only been based on similarity metrics evaluation, and therefore it is interesting to evaluate several of these approaches in the same scenario or closed repository.

As noted in previous works, so far there is no method or methodology for BP evaluation that integrates several experts to collaboratively build closed repositories of BPs that could serve as a basis for evaluations involving semantics and structure on BP searching.

Considering the description above, in (Kunze and Weske, 2012)an open library available to all community members is proposed. This library shares the BP's information and repositories following a few guidelines. For this reason, it is important to contribute to the definition of a BP repository based on the ideas expressed in: A successful BP

repository depends on having a good searching engine allowing the retrieval of the desired process models in a short time period. In addition, due to the evaluations made on the repository, it may act as a closed document collection where, for each proposed query, the resulting BPs and their corresponding relevance levels are known.

3 EVALUATION METHODOLOGY

The proposed collaborative evaluation methodology is divided into three stages: individual evaluation, searching for consensus on discordant evaluations, and results refinement. The methodology arises as a consolidation instrument which allows a set of judges to make judgments in relation to relevant results against a BP query in a collection (or list) of BP previously stored.

Indeed, the results considered relevant by the panel of judges will be those that represent the ideal responses for each query in the closed repository built.

The evaluation takes a set of BPs from the repository, defined as $Q=\{bp_1,bp_2,bp_3...bp_n\}$, which represents each of the queries. For each query, a resulting list of items T is evaluated, where $T<=M$ (in order to decrease the workload of judges), and M is all the BP existent in the repository. Each item of the resulting list is evaluated using a Likert scale containing the following concepts: very relevant, relevant, quite relevant, not very relevant, and irrelevant. This scale is defined because two BPs may have different similarity levels in relation to each other. The weight (w) assigned to each concept of relevance is $w=\{1, 0.75, 0.50, 0.25, 0\}$ in the scale and, therefore, the overall relevance level (nr) of each item is defined by the following equation (1):

$$nr = \frac{1}{n} \sum_1^n w \,, \qquad (1)$$

In this equation, n is the number of users who evaluated each item, and w is the weight assigned them to each item. The similarity perspective of the evaluator in relation to the models being compared is determined by taking into consideration what he/she finds in the textual or structural characteristics (or by a combination of both).

3.1 Individual Evaluation

At this stage, each evaluator or judge runs each query Q and the system shows up a list of results. Evaluators then express their judgment of similarity of each result against the query. To express such judgment, judges must consider the complete representation of the two business processes (query and result) and their experience in the subject

3.2 Searching for Consensus on Discordant Evaluations

At this stage, each evaluator reviews one by one the relevance judgments issued in the previous stage, and compares them with the judgments that other judges have stated. Thus, evaluators may confront how concordant or discordant their given judgment is against each item, according to the judgment of other evaluators. If evaluators believe that their judgment regarding the set of evaluators is too discordant, they can change their judgment guided by the collective response of other evaluators. For instance, if an evaluator qualified an item as not very relevant in stage 1, but the rest of evaluators (panel of judges) rated it as very relevant, that assessment can make the evaluator reflect on his/her judgment and change his/her decision. This feedback allows judges to have an overview of the evaluation made of each item by all the evaluators.

3.3 Results Refinement

At this stage, and after the judges have (or not) changed their positions (taking into account the contribution of the other judges), the results of each query are listed, taking into account a pair of thresholds. Results are thus filtered by values of nr ranging from 50% to 60% (these parameters can be adjusted depending on the desired confidence level), which means that so far they are not considered as truly relevant nor irrelevant and there still exists a high disagreement level among the judges. As in the previous step, judges may re-analyze the pair of BPs and alter their assessment based on the evaluations of the other judges

3.4 Methodology Objectives

A fundamental task for building a BP test repository is the definition of an intuitive evaluation process where the evaluators (judges) collaboratively agree to clarify similarity criteria in the results retrieved by a BP search system. It may thereby determine the quality of these BPs through a consensus view, given that it is almost impossible to access a real BP repository from an organization.

3.5 Measures for the Evaluation of Relevance

Measures for assessing relevance calculate the relevance of the retrieved results of a BP similarity tool in decreasing, gradual, and continuous forms. They measure the gain of a result item based on the position of this item in the ranking, recognizing that the most relevant BPs are most useful if they appear in the top positions of the ranking (Ulrich and Birgitta, 2010).

Graded relevance measures (*Pg* and *Rg,* described below) must be applied in the above to provide a classification (*Ti*) of the BPs returned in the repository, those that are considered similar to a query BP (*Q*) according to different levels of relevance. *Pg* and *Rg* (Tsetsos et al., 2006) take into account the sum of degrees of relevance Among the BPs.

In addition, to measure the quality of the ranking of the results generated by the BP searching mechanism applied on the current evaluation, ANDCG (Average Normalized Discounted Cumulated Gain) and GenAveP' (Generalized Average Precision) (Ulrich and Birgitta, 2010) measures were used as presented and improved in the works of Küster and König-Ries (2008). These measures quantify the quality of the ranking produced by Web services´ retrieval tools, but are fully applicable to the BP searching field.

4 DEVELOPED TOOL

The main purpose of the platform is to provide an infrastructure to integrate a group of judges (evaluators) in a collaborative environment to issue relevance judgments regarding the set of results reported for different queries by a BP searching engine. The platform enables the implementation of any BP search engine that integrates the required features to capture data in the indexing and searching interface. All the functionality is provided through a Web user interface. In this sense, the platform allows manual and intuitive comparison of the BPs within a given repository, according to each query. Next we describe the architectural components of the tool.

An architecture composed by three layers was defined for the development of the application (see Figure 1). This architecture provides the following advantages: flexibility, scalability and facilitates the construction and maintenance of the platform. These layers are described below.

Presentation Layer: This layer includes a simple and usable user-centric Web interface that can be accessed using any Web browser. Therefore, this interface provides a visual functionality for evaluators (judges) to execute each query, and additionally specifies the relevance level through a consensus view in a collaborative environment for each one of the searching results classified and sorted sequentially in a list.

Business Logic Layer: this layer comprises business rules and processes related to the functionality offered by the system and that are implemented at this layer. For instance: executing each evaluation phase, running query options in the search engine (which may be a list of the *M* BPs from the repository or a short list of $T <= M$ BPs that relies on a searching tool to reduce the judges efforts), evaluating retrieved items, giving relevant judgment, calculating relevance, providing a chat service for users, among others.

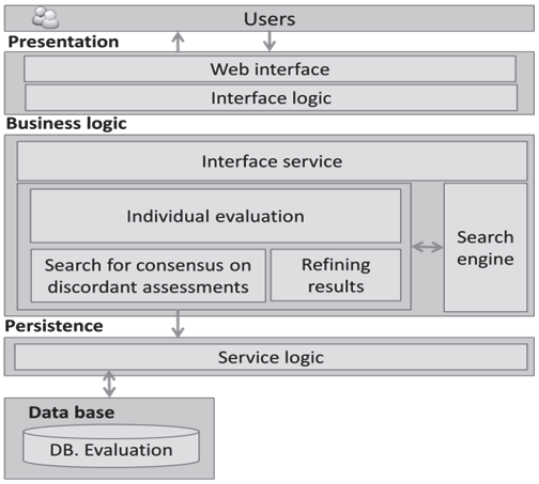


Figure 1: Web Application Architecture.

Persistence Layer: this layer provides the functionality for flexible storing: BP models in an XML representation; BP models to be used as queries; evaluation data of the judges; and evaluation judgments about each of the retrieved items according the queries. Besides, this layer provides agile and efficient mechanisms to retrieve, access and manage the existing BP models in the repository and the collected information throughout the evaluation process.

Figure 2 depicts the individual evaluation interface that was developed for the evaluation step. The tool was implemented with Java technology, additionally PostgreSQL was used as RDBMS for storing the information managed in the evaluation

5 REPOSITORY BUILT

This section presents the results obtained in the manual comparisons made by the judges using the developed platform and the concordance and the evolution of consensus judgments using the proposed methodology.

5.1 Repository

The current implementation of the repository includes 100 BPs modeled with BPMN (Business Process Modeling Notation). Those BPs were graphically designed by experts of the Telematics Engineering Group of the University of Cauca (Colombia) based on real processes provided by Telco operators in Colombia and examples found in different Web sites (e.g., the TM Forum)(Figueroa 2011). It was not possible to use a real repository of a Telco operator because operators are reluctant to give access to their repositories due to privacy and security policies. This is available in the following link:https://drive.google.com/file/d/0B1J2e8JSqOR2QlBQcENPdXlMMTA/edit?usp=sharing.

5.2 Judge's Profiles

In order to evaluate the proposed methodology, we have counted with 59 people (judges or evaluators), which belong to the Institute of Informatics and to the Business Management School, both of the Federal University of Rio Grande do Sul (Brazil), and to the University of Cauca (Colombia), distributed according to Table 1.

Table 1: Kind of Judges or evaluators.

	Dr.	MSc.	Professional
Institute of Informatics/UFRGS	-	7	14
Business Management School/UFRGS	-	-	33
University of Cauca	2	3	-

5.3 Evaluation Phase

For this phase, a set of 6 BP were defined as query elements, and, for each query, the searching mechanism returned a list of 20 results sorted by the similarity defined within the searching model.

Thus, each judge manually compared the similarity between the query models with each item in the results list, and maked a relevance judgment from the ones established in the methodology (i.e., the Likert scale described in Section 3).

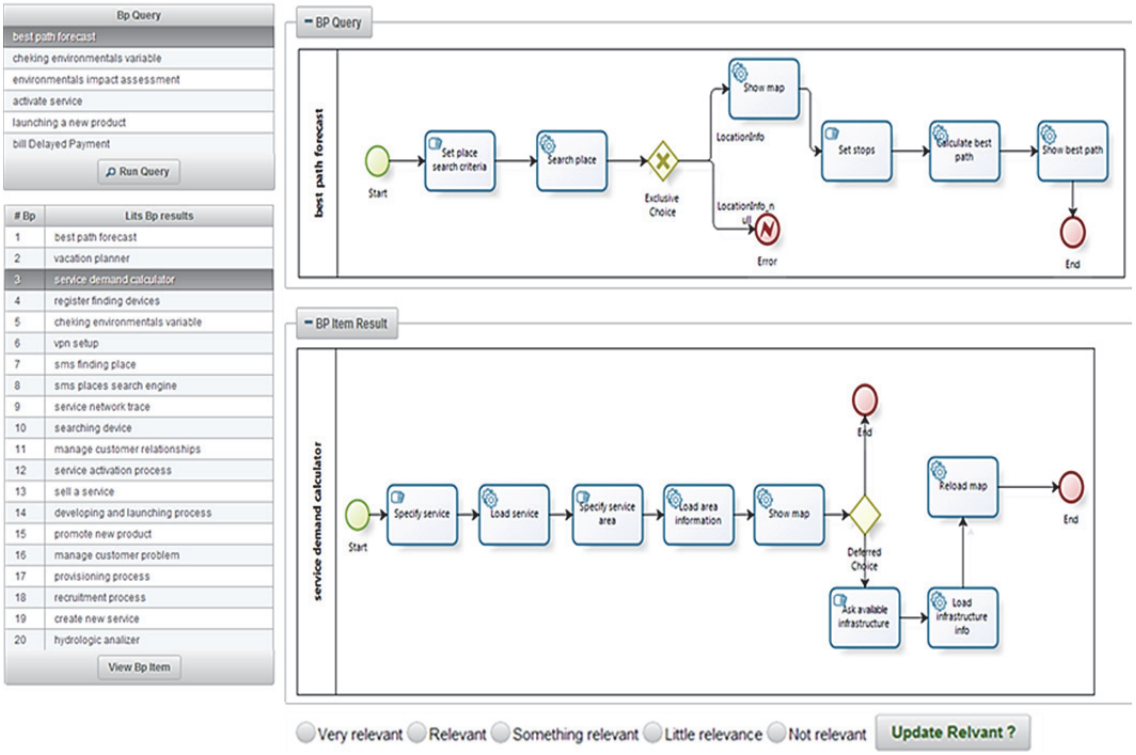


Figure 2: Developed tool, individual evaluation interface.

The evaluation was conducted in this way: each group of judges was gathered to the computers lab at the university they belong to. The evaluation methodology and its aims were explained to the groups once they were met. Subsequently, the operation of the evaluation platform was explained, and the individual evaluation phase was started in a coordinated way. This is because it is necessary to start the searching for consensus on discordant items taking as initial state the whole set of relevance judgments issued by the judges from each group during the evaluation phase

Once the first phase was finished, a period of time was established to complete the other evaluation phases. For this purpose, we have established communication via mail as a reminder element on the completion of the final evaluation stages.

According to the above, each judge provided an average of 360 manual comparisons, in that sense, the total of manual comparisons made by the judges was around 21,240.

5.4 Methodology Application on the Repository

Comparisons made by the judges in a manually way at each one of the stages (St1- Individual, St2- Searching for consensus on discordant evaluations, and St3-Results refinement) based on standard deviation allow an overview of the concordance level between them. In Table 2 we present the concordance values between judges for the items evaluated at each query stage. This value is represented by grouped standard deviation values, which measures the relevance levels dispersion which are classified within the range values previously presented.

In relation to the application of the methodology on the repository, the following average concordance (AVG) values between the judges were obtained: 0.284 for stage 1, 0.256 for stage 2 and 0.250 for stage 3. These values indicate that these relevance judgments are not widely dispersed and therefore do not differ much. When judges progress through the evaluation stage, these values are lower and tend to commonalities showing the force of the proposed methodology.

In addition, it has a 9.7% of concessive improvement in (MCF) between stage 1 and stage 2, and 2.4 % between stage 2 and stage 3 for each query, confirming that stage evaluations allow to better refine the repository (results by each query).

This allows us to perceive that the 59 judges improved their consensus at 11.8%, unlike if they would have done individually. In this sense, the repository gets 11.8% of general concessive improvement (MCG) making it more "ideal" than required at stage 1.

Besides, the collaborative evaluation methodology and the developed tool minimize the re-evaluation work in stages 2 and 3.

Consequently, the collaborative evaluation methodology and this tool improve the repository quality, increasing its usefulness.

In addition, the Pearson correlation coefficient was used to calculate the concordance level between judges in each of the stages (St1 to St3) for each query. For this, we took as population the relevance judgments executed by the evaluators (judges) to each item in the list. The Figure 3 shows that the correlation becomes stronger as the stages advance and evaluation goes forward. Consequently, *Q*1 scored the lowest concordance level between stages 1 and 2, achieving 83%. Similarly, between stages 2 and 3, it scored 87%. Moreover, *Q*6 scored the highest concordance degree between stages 1 and 2,

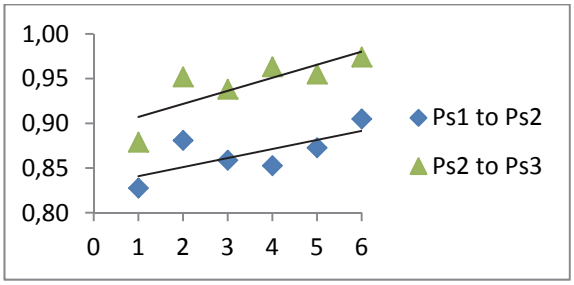


Figure 3: Concordance between evaluators to each stage.

Table 2: Standard deviation value by each relevance judgment per phase.

	Q1			Q2			Q3			Q4			Q5			Q6		
Measure	St 1	St 2	St 3	St 1	St 2	St 3	St 1	St 2	St 3	St 1	St 2	St 3	St 1	St 2	St 3	St 1	St 2	St 3
AVG	0,31	0,27	0,26	0,29	0,27	0,26	0,28	0,25	0,25	0,27	0,24	0,23	0,28	0,26	0,25	0,28	0,25	0,25
MCF		11,2%	2,8%		8,6%	3,7%		9,4%	2,1%		10,3%	1,5%		7,4%	2,3%		11,1%	1,9%
MCG			13,6%			11,9%			11,3%			11,6%			9,6%			12,8%

achieving 90%. In the same way, between stages 2 and 3, it scored 97%, showing that concordance level between judges is a growing correlation (very high and positive).

6 CASE STUDY

This section presents the outcomes of applying the methodology on the repository built using a BP searching mechanism. In our case, we have used a BP model searching mechanism that uses linguistic information (activity name, activity type and description) and structural information; it is called a MultiModalSearBP model that is described as follows.

6.1 BP Searching Model Applied

The discovering process applies a searching strategy that integrates linguistic and structural information contained in the BPs, thus allowing us to increase the effectiveness and relevance of the searching results. The MultimodalSearchBP architecture consists of three layers, described below.

Parsing Layer: This layer has a parser that transforms BPs from its original format XPDL (XML Process Definition Language) to a vector representation, where each BP is considered a term's matrix consisting of a linguistic component and other structural.

Indexing Layer: This layer gives a weight to the linguistic and structural components in order to create a multimodal search index consisting of the linguistic matrix component (*MC*) and the matrix structural component (*MCd*) as follows: $MI = \{MCd \cup MC\}$, and the index stores the physical file location of each of the models stored in the repository.

Query Layer: This layer is responsible for allowing BP's search from three querying options:
linguistic, structural, and multimodal query (Ordoñez 2013).

6.2 Analysis of the Results

In this section, the results obtained using the search engine on the built repository are presented.

For this, it is necessary to create an outcome list with the items considered as relevant by the judges for each query, which is sorted from highest to lowest depending on the relevance level (*nr*), achieved in manual evaluation.

Then, the resulting list generated by this BP searching mechanism is compared to the resulting list considered as relevant by the judges on that query. In Figure 4Figure 4, the evaluated searching model achieves a grated precision (*Pg*) average that ranges from 57% (minimum) to 85.2% (maximum). This model combines structural and linguistic criteria present in the BPs, over text processing algorithms capable of reducing the probability of retrieving irrelevant results (false positives).

Regarding to graded Recall (*Rg*), it ranges between 34% and 56%. This is because the number of results returned by each query is limited to twenty BPs. This limitation is inspired in the Web search domain, where users only are focused on the first ten or twenty results in the answers set. Therefore, this indicates that the model can get false negatives (lose relevant business processes in the ranking), but at the same time increases accuracy by reducing the number of false positives.

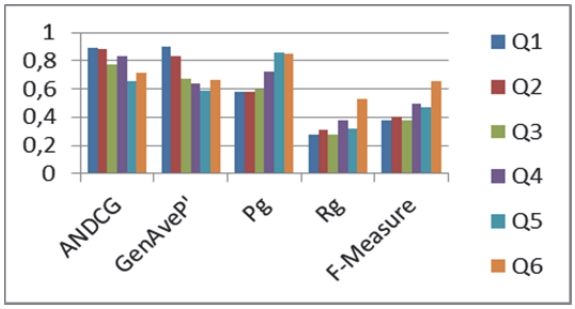


Figure 4: Evaluation measures.

About to the effectiveness of the searching model, it is characterized by the performance obtained in the rankings. In that sense, F-Measure allows observing the harmony of Pg and Rg results, and, in the searching model applied, it obtained average values between 36% and 47%. Regarding to the results ranking, ANDCG demonstrates that the ranking generated by the model used has high quality, because it places a representative number of relevant elements at the beginning of the ranking, reaching an average range between 79% and 88%. As explained before, the difference between GenAveP and ANDCG' measures is that the last one possesses a factor that evaluates the elements retrieved to the bottom of the ranking with a higher value. In these cases, the model reached an average value between 71% and 88. The graded measures provide a more intuitive and flexible evaluation. They also reduce the influence of inconsistent judgments among evaluators

7 CONCLUSIONS AND FUTURE WORK

In this paper, we have established a methodology for the collaborative construction and evaluation of BP repositories. For this purpose, we used a BP searching mechanism applying graded measures to determine the relevance degree of the retrieved elements. Consequently, this allowed the demonstration of the usefulness of the responses and their relationship to queries submitted by users. These responses serve as the most appropriate responses for evaluating and comparing searching mechanisms that use the same repository.

The collaborative evaluation allows judges to have an overview of the relevance judgments issued by each judge on elements retrieved in the results list. As a result, judges can compare the concordance or discordance in the relevance judgment issued for an evaluated item and thus corroborate or change their assessment.

The data shows that there are some differences in the points of view of the evaluators. While most experts considered the items ordered at the top of the result list (1, 2, 3, 4) as relevant or very relevant, a minority (10%) of these were considered as not relevant or irrelevant. This is because the latter took into account only one part of the evaluation process (linguistic or structural), or simply because the comparison between the BP query and each one of these results was performed superficially, which may have been due to fatigue as a result of the huge number of evaluations performed.

The application methodology proposed serves as the basis for the generation of stable evaluations of BP repositories, which are thus more maintainable and reusable. In addition, as a secondary contribution, the BP repository that was used in our evaluation can be seen as an open access repository that will be shared, expanded with new models BP, and can be used in future researches by any actor interested in the area of BP management.

As a future work, it is aimed to expand the evaluation methodology by manually creating groups or families of BPs with those BPs considered as truly relevant in each one of the queries. This allows group representation of thematic topics or structural patterns of the BPs within the repository.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the collaboration of the judges belonging to the Institute of Informatics and of the Business Management School of the Federal University of Rio Grande do Sul (Brazil) and the Department of Telematics of the University of Cauca (Colombia). They also acknowledge the Institute of Informatics for sharing their infrastructure. Finally, we would like to state that this research was partially supported by CAPES and CNPq, Brazil.

REFERENCES

Becker, M. and R. Laue (2012). "A comparative survey of business process similarity measures." Computers in Industry 63(2): 148-167.

Chris J. Turner, A. T., Jorn Mehnen (2010). "A Genetic Programming Approach to Business Process Mining."

Dijkman, R., M. Dumas, et al. (2011). "Similarity of business process models: Metrics and evaluation." Information Systems 36(2): 498-516.

Dijkman, R., M. Dumas, et al. (2011). "Similarity of business process models: Metrics and evaluation." Information Systems 36: 498-516.

Koschmider, A., T. Hornung, et al. (2011). "Recommendation-based editor for business process modeling" *Data & Knowledge Engineering* 70: 483-503.

Kunze, C. R. a. M. (2013). "An Extensible Platform for Process Model Search and Evaluation." Business Process Management Demos 2013: Beijing, China.

Kunze, M. and M. Weske (2012). "An Open Process Model Library." Business Process Management Workshops, BPM 2011 International Workshops Clermont-Ferrand, France, August 29, 2011 Revised Selected Papers, Part II: 26-38.

Küster, U. and B. König-Ries (2009). "Relevance Judgments for Web Services Retrieval - A Methodology and Test Collection for SWS Discovery Evaluation." 2009 Seventh IEEE European Conference on Web Services: 17-26.

La Rosa, M., H. a. Reijers, et al. (2011). "APROMORE: An advanced process model repository." Expert Systems with Applications 38: 7029-7040.

Ordoñez, C. F., Juan Carlos Corrales, Carlos Cobos (2013). "Multimodal Model for Business Process Search." Submitted to : Information-Sciences.

Rosa, L., H. M. Arthur, et al. (2010). "QUT Digital Repository: http://eprints.qut.edu.au/ This is the author version published as:."

Škrinjar, R. and P. Trkman (2012). "Increasing process orientation with business process management: Critical practices'" International Journal of Information Management.

Tsetsos, V., C. Anagnostopoulos, et al. (2006). "On the Evaluation of Semantic Web Service Matchmaking Systems." 2006 European Conference on Web Services (ECOWS'06): 255-264.

Yan, Z., R. Dijkman, et al. (2012). "Business process model repositories – Framework and survey." Information and Software Technology 54: 380-395.

Eliciting Requirements in Extreme Programming (XP) Through Business Process Models

Hugo Ordoñez
Department of Systems Engineering
St. Bonaventure University
Cali, Colombia
haordonez@usbcali.edu.co

Andrés Escobar Villada
Department of Systems Engineering
St. Bonaventure University
Cali, Colombia
anfelesvillada@gmail.com

Lorena Velandia Vanegas
Department of Systems Engineering
St. Bonaventure University
Cali, Colombia
dilove0122@gmail.com

Carlos Cobos
Department of Systems Engineering University of Cauca
Popayán, Colombia
ccobos@unicauca.edu.co

Armando Ordóñez
Intelligent Management Systems Group, Foundation University of Popayan
Popayán, Colombia
jaordonez @unicauca.edu.co

Juan Carlos Corrales
Department of Telematics
University of Cauca
Popayán, Colombia
jcorral@unicauca.edu.co

Abstract

During requirements elicitation some problems in analyst - stakeholders communication may appear; such misunderstanding may cause that final products do not accomplish customer expectations. This proposal aims for improving understanding and comprehension between stakeholder and software development team during requirements phase of XP (eXtreme Programming) methodology. To do so, our approach replaces user stories by business processes models (BP). For analyzing the effectiveness of the present approach, user stories and BP models were used in eleven projects during requirements phase and quantity and quality of data collected was compared. Experiments evidence that the use of BPMN for lifting requirements vs. using user stories from XP methodology helps to improve the quality and quantity of information collected and causes that the users could specify more clearly their needs and business goals to analysts.

1. Introduction

Software is an important part of global capital, because it allows development of business and knowledge management [1]. Moreover, software industry is involved heavily in the "new economy", it is a white industry that does not pollute and generates well-paid jobs [2].In this scenario, competitive software tools with international quality standards require to be connected to the world, to have access to the latest technologies and to be implemented with appropriate methodologies [1], [3]. However, conditions and limitations of each project are different, making necessary to adapt methodologies to each project or situation. Some software methodologies have been specially designed for projects that develop software which have conditions such as: tight deadlines, volatile requirements, and based on emerging technologies [4], [5]. One of these methodologies is extreme Programming (XP), which is focused on enhancing interpersonal relationships as a key point for successful software development, promoting teamwork, learning developers and good working environment; based on continuous feedback between the customer and the development team [6].

The main challenge in XP is to define the system requirements, which are identified and described in meetings or interviews with customers and stakeholders. In XP the identification of system requirements is realized through user stories. User stories describe the functionality of the software to build and customer's needs. However, these user stories are written in natural language, which may guide to misinterpretation, given the ambiguity and uncertainty of the collected information [7], [8].

Meetings between developers and customers for building user stories are usually not productive enough, causing the requirements are not identified clearly; this situation leads to software products that don't accomplish stakeholder expectations [9].

Recent studies show that the graphical representation of requirements can contribute to a clearer understanding of the system requirements. This may reduce the ambiguities, due to the fact that graphical representations provide a good overview of the system ("a picture is worth a thousand words") [10], [11].

In this paper we propose the use of business process models (BP) for representing software requirements. BP can be defined as a set of coordinated activities that aim for achieving a common business goal [9]. In our work, BP are represented using BPMN [12] standard.

To validate the proposal, both strategies (user stories and BP models) were applied for requirements elicitation in eleven software projects. For each project, a group of analysts evaluated the advantages of the two options in terms of potential for better elicitation (common understanding of requirements among stakeholders) and a greater number of requirements.

This paper is organized as follows: in section two some of the most representative works on the subject of research are presented, in section three the proposed method is shown, section four puts forward the validation and results of the proposed technique, finally the conclusions and future work are depicted in section five.

2. Related Work

Agile methodologies are focused on enhancing interpersonal relationships as a key to success in software development; they promote the use of user stories [7] during requirements phase. The latter implies that the constant communication between user and developer is vital to clearly describe the needs and objectives of the project [13]. There are several suggested templates for user stories, but there is no consensus. However, it is common to find that they include one or more of the following parts: name, description, tasks description, and estimate effort in days. However, it is recommended that at least one story for each major feature existed; in addition, it is suggested that each developer develops one or two user stories each month [14].

[15] Affirms that the greatest weakness of user stories is that managers, users and developers describe non-essential elements instead of the main goals of the systems. The user story should only describe external behavior of the system, which can be understood by the stakeholder. As an alternative to the textual requirements elicitation, some studies have used visual models that help to understand to the users how to use the system, equally these graphical models contribute to communicate to stakeholders the development team objectives, keeping them interested and involved during development process. Additionally, these models also facilitate communication, understanding, problem detection and exploration of scenarios.

In the work of Zheng et al. [16] the adaptation of an agile methodology for implementing BPMS (Business Process Management Systems) is presented, the authors analyze the overall project implementation as well as the organizational impact. This work is particularly focused on Scrum methodology. In the proposed approach the team is formed by development staff, and customers, who have the responsibility to establish and define requirements and test. At the end of iterations, stakeholders perform deliverables assessment, so that improvements and solutions are implemented in the following iterations. Finally in the Zheng proposal, an analysis is performed for describing future expected situations with the implementation of BPMS. With this, business analysts draw processes diagram in BPMN notation.

On the other hand, Avnet et al. [17] perform a comparison between use case and BPMN representations for system requirements. This study shows that business analysts and users have different visions of processes that take place in the organization. The experiment concluded that the graphical notation enables faster cognitive understanding and use cases allow viewing many exceptions in the process. Moreover, users and analysts understanding improves when use cases template is used first (it means a textual model), and then a BPMN model.

From the analysis of the works cited above, it may be concluded that requirements analysis is one of the most difficult issues faced by software analysts. Customers or users are not completely sure of what is needed because they do not understand the whole problem domain. The use of natural language between customers and development group makes that sometimes requirements or priorities are not well defined, creating delays in development review and redefinition of the requirements for each user story. Furthermore customers or users usually omit information which is considered "obvious". In spite of the fact that previous approaches propose using BPMN for modelling requirements, they do not focus on agile methodologies and nor they use BPMN to support other process phases such as design.

3. Representation of user requirements in business process model

The present proposal seeks to use of Business process (BP) for requirements elicitation, replacing user stories in agile development methodologies such as XP. BP models allow customers, stakeholders and users to adopt a simple and standard notation, facilitating the capture of requirements in a visual model through a notation that does not require additional training of the project team. BP models representing user requirements support the talks with developers

and stakeholders. BP models also act as a basis for possible improvements in each stage of the project development based on XP methodology. For a description of the proposal, each element of the user story is described and its correspondence is established with BP models in BPMN notation:

3.1 Title of the story

Here the main objective is clearly propounded. Also the title defines the interactions between users and the system. In addition the title of the story contains descriptions of all functions (each interaction) of the System. In BP model the title of the story is represented by the name of the business process model.

3.2 Description of the Story

It describes the sequence of activities identified that make up the flow model for the development process. This description contains the step by step to follow as well as the activities provided by customers, users and stakeholders. This information becomes the starting point for BP model building. In BP model, it is defined by name and description attached to each one of the activities or tasks that make up the logical sequence of the BP.

3.3 Historical Revision

Historical revision provides traceability of changes made to the requirements and establishes responsible actor, and change dates. These changes and its traceability will be represented in BP through the versioning process, which can respond to events in development activities according to circumstances.

3.4 Reference Documents

In user story template, reference documents are elements that serve to refine requirements or as an input or output products. In BP model, reference documents are represented as data objects. However, if documents are not part of the process, but references for requirement elicitation, so this documented is represented as an annotation in BP.

3.5 Actors

This element describes actors involved in the requirement. These actors perform sequences of actions that the system must perform. In BP model, actors are represented by the roles within each process. Roles are defined by Pools and lanes. A pool represents the main participants in a process, and can be divided according to the organizations that they belong. A pool contains one or more lanes. The lanes are groups of tasks by area or participant involved in the process.

3.6 Dependencies

Within the user stories template some dependencies may be established. For example, some user stories must be previous to some activities, this dependence allows establishing order constraints between stories. In BP model, these dependencies are determined by sending messages between processes and creating threads to indicate the sequence of activities between different roles and tasks. Figure 1 provides a graphic representation of the relationship between each of the elements of a user story and BP model.

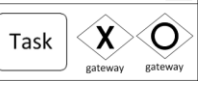
USER STORY	BPMN NOTATION	SYMBOL
Title of the story	Name of Process	
Historical Review	Process versioning	
Reference Documents	Data Objects, but if the document is not part of the process but to elicit reference is treated as annotation.	
Actors	Roles. Task Allocation via Lanes.	
Dependencies	Related threads or processes. Sending messages between processes.	
Description of the Story	Sequence of Activities	

Figure 1. User story vs. BP model

3.7 Adaptation of BP in the XP Methodology

BP generated models are integrated into the later stages of the XP methodology. This section shows how this integration was performed. XP proposes a dynamic life cycle supported on short development cycles (called iterations) with functional deliverables at the end of each cycle [18]. During each iteration a full cycle of analysis, design, development and testing is done (see Figure 2). Next the integration of BP models in the XP methodology is described.

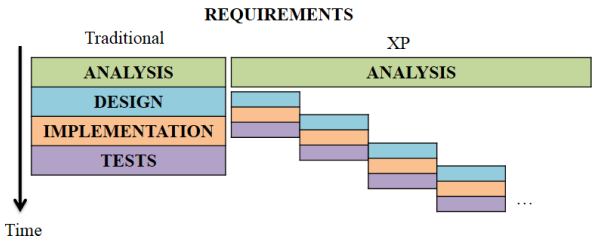


Figure 2. Traditional vs. extreme programing (XP) phases

Exploration: In this phase, the customer defines what he needs by writing "user stories" at a high abstraction level. Based on this information, developers estimate the development time of each activity described in the story [18]. In this phase, the estimates should be preliminary, because requirements may vary when they deeper analyzed in subsequent iterations. With the BP model, more details are specified, because this is created during the interview with the stakeholder. Therefore, in this phase, a low detail level is maintained as methodology proposes. The defined functionality of each BP models should be programmed in a one to three weeks period. If time is greater, the process should be divided in two or more processes.

Planning: In this phase the stakeholder group and the developers agree the order in which user stories should be implemented and its deliverables. These deliverables are specified in a document called the Delivery plan, or Release Plan. Furthermore at this stage, small test programs called "spikes" are developed to validate the stakeholder functionality, spikes are created in order to reduce risks and improve effort estimation. These spikes are used exclusively during requirements phase, and subsequently discarded; however developing of such spikes may be time consuming.

The BP model specifies the order, priority and dependency of the functionality to be developed, therefore, the implementation and deliverables order is defined in advance. Additionally, in this phase, there are some BP tools such as: BonitaSoft and BizAgi that automatically generate small prototypes from defined process models. In these prototypes the developer does not invest more time than the necessary for requirement modelling, he also validates the requirement immediately; reducing the time it might take for validation.

Iterations: in this phase each user story is translated into specific tasks and generating software deliverables. Because user stories are not defined in sufficient detail during exploration phase, then an analysis process is performed with the stakeholder to specify items as data constraints. With the BP data details and restrictions are already defined from the exploration phase, this allows estimation of time for the developer to be more accurate, therefore at this stage the process with the customer is more a validation process that specification process.

Deployment: this phase includes Implementation of functionalities and testing, these tests may require modifications or adjustments, these changes are known as "fine tuning". In Table 1 the input of BP models are generalized when the user story is replaced.

4. Hypothesis

During the research process seven hypotheses where
defined for validating the present approach (see Table 2). Hypotheses aim for identifying the real shortcomings in the use of each technique (user stories vs. BP models). Next each hypothesis and their purpose will be explained.

Table 1. Integration of BP models in XP methodology

User Stories	BP Models
During planning the order in which User stories should be implemented is defined and its deliverables.	The order in which BP should be implemented is specified in exploration phase, so planning should not be defined.
During exploration some aspects such as conditions and data are not defined. "spikes" are required.	BP provides more detail to facilitate the estimation of time and risks, it becomes possible to avoid the generation of spikes.
Release Plan based on the prioritization of the stories, plans are agreed between developers and customers.	Release Plan based on prioritized BP models modelled with stakeholder.
They are written by the stakeholder, in their own language.	Modelled in BPMN, which is an understandable graphic language.
Minimum Detail, programmers make time estimation for development.	Greater detail, it facilitates time estimation, i.e. BP models needs more time for modelling, but less time in subsequent phases.
If estimated effort is higher to 3 weeks per story, so it is divided in 2 or more stories. If one story takes less than a week, it is combined other one.	If the estimation is more than 3 weeks per process, it is divided into two or more processes. The division is done using threads, avoiding the most complex model.
Customers group user stories and define dependencies.	Customers define priority and BP model describes dependencies using pools

	and threads.
Each user story is translated into specific tasks.	The sequence and priority of tasks are already defined in the BP flow.
Spikes help to estimate the risk.	Some BPMN tools generate quick prototypes which permit validate in the moment when a model is defined.

The aim of the first four (H1 to H3) hypotheses is to identify which technique helps to improve understanding between the stakeholder and project analysts with less ambiguous requirements and which accomplish organization objectives. The last three hypotheses aimed to determine if the use of BP models for requirements elicitation, instead of the user story improves other phases of XP methodology. These Hypotheses also validate which of the two techniques has the highest percentage of implemented and approved features.

Table 2. Hypothesis raised

Hypothesis	Purpose
H1: requirements elicitation using BPMN improves domain understanding compared to user stories	To identify if BP models enhance understanding during requirements elicitation for those involved software development.
H2: requirements elicitation using BP models after user stories increases the requirements understanding.	To identify if BP models improves domain understanding when used after user stories.
H3: requirements elicitation using user stories after performing the BPMN model increases the understanding of domination by those involved.	To identify whether the user stories improve domain understanding after using BP models.
H4: Generation of release plan from BP model specifies in greater detail the deliverables compared with plans generated from user stories.	To identify if the requirements modelling with BP models allows specifying better release plans compared to user stories.
H5: Activities of release plan are better when generated from BP models instead of user stories.	To identify whether the tasks in the iterations of the release plan are more complete when they are created from BP models instead of user stories.
H6: percentage of approved tasks in the first iteration of release plan is higher when BP models are used instead of user stories	To compare the percentage of features implemented according to customer needs in the first iteration of the release plan both
	with BP models and User stories.

Hypothesis validation was done through an experimental approach. This validation consists of survey by sampling [19]. For performing the survey each technique (user stories and BP models) where tested separately for requirements elicitation. Then a comparative analysis of the results was done. Finally the BP based approach was tested during planning and iterations phases.

4.1. Experimentation

Both techniques (BP models and user stories) were applied for requirements elicitation in eleven projects in analysis phase. The experimentation included participation of 62 analysts in the software building department of SENA - Latin American Minor Species (LCMS) Valley Regional and 25 different organizations stakeholder from different companies.

Firstly participants were trained in XP methodology, user stories creation, BPM and BPMN Then two groups were created for the purpose of applying the two techniques in each project. In these projects, both techniques were applied in order to compare the quantity and quality of information collected. In each of the techniques, the analysts interacted with stakeholders involved in each project through focus groups that collect data using a semi structured group interviews [20]. The main purpose of the focus group is to know concerns and reactions from stakeholders and analysts. Moreover, the focus group focuses on the user story that is being generated or the BP model being defined.

For each technique, it was considered and interval of one week for its execution and information gathering. Tests were done at the same time in the SENA and they were done with a week of difference between test in order to reduce stakeholder fatigue.

Table 3 shows the projects and the number of participant analysts for each technique. For requirements elicitation using user stories we worked with a template [21]. This was done in order to standardize how the requirements are documented between projects (See Table 4), for the elicitation of requirements under BP models, BonitaSoft tool was used (see Figure 3).

Table 3. Number of projects and stakeholders by hypothesis

Hypothesis	Projects	Number of Analysts		Number of Stakeholder
		User stories	BP Models	
H1	A	2	2	2
	B	2	2	2
	C	4	3	3
	D	4	3	3
H2	E	6	6	2
	F	6	6	5
	G	6	6	4
H3	E	6	6	2
	F	6	6	5
	G	6	6	4
H4	H	4	4	3
	I	4	4	3
	J	4	4	3
	K	4	4	3
H5	H	6	6	3
	I	6	6	3
	J	6	6	3
	K	4	4	3
H6	H	6	6	3
	I	6	6	3
	J	6	6	3
	K	4	4	3

As an example we take one of the user stories and one BP model defined in project A. This project is focused on staff evaluation. The A project aims to create a tool that allows the evaluation of employees performance when a contract ends. This system will help to determine employees performance based on a predetermined performance factors. It also allows the user to register and establish assessment criteria and elements to be considered during the assessment. Table 4 shows the user story created for one of the requirements (R1) Project A. The requirement (R1) consists of recording the performance evaluation of an employee by the manager of human resources.

Table 4. Template of user story for project A, R1

Title: Project A – R1. Register staff evaluation			
Historical review			
Version.	Change description	Author	Date
Reference document			
Document number:		title:	
Word Format – staff evaluation template			
GENERAL INFORMATION			
Actor:	Human capital manager		
Dependencies:	Staff must be registered and the evaluation to apply must be created (State active).		
Priority:	Low		Medium		High	X
Description of the story						
Human talent manager can record performance evaluation of staff. To do so, he must enter the employee's id and if it is registered with a valid contract, the last active evaluation will be displayed; this evaluation includes information for each element with the corresponding criteria. When the evaluation is available the manager shall record the score (1-100). When he finish entering the score (all criteria are required to evaluate), he can enter comments if necessary and save the assessment. At the time of evaluation registration, the system must register the scores per criterion, the date of the assessment and the overall score.						
Observations: None						

Figure 3 shows the BP model created for one of the requirements (R1) of Project A. The manager should select an employee through the identification; the system must consult if the employee exists and has a valid contract with the company. If the user is active in the system, the elements and the evaluation criteria can be evaluated in a range of 1 -100 points.

Eventually the system will calculate and display the scores and the assessment will be recorded. Upon completion of the requirements elicitation stage, a survey for each applied technique was performed. The survey allows evaluating how each analyst experienced information capture. For more detailed and accurate information on the surveys, the sample was defined considering different groups of analysts working on development projects, in which the hypotheses were validated. The end of the sample is to avoid partial or biased results. For example, Table 5 shows the survey to the evaluation process in user stories.

Table 5. Survey for user stories application

Does the template used to describe user stories allow you to define concise and complete customer requirements?
Does the template used to describe the user story allowed you to establish if the requirement outlined by the user was required for the system or the organization?
Does the template used to describe user stories allow you to define whether the requirements were consistent? This means, if it was established that the requirements were not contradictory?
When reviewing user stories written by another analyst, did you find that these were not ambiguous and that its interpretation was the same that the stakeholder had said?
Are the requirements documented in the user stories are verifiable, i.e. could you validate their compliance once implemented?
With the user stories template, could you set the priority or urgency of the requirement according to the objectives of the stakeholder?
It is clear the language used by the stakeholder when the

system requirements were specified?
Indicate percentage of total system information that you could capture using user stories.
It was adequate the customer disposition for specifying their requirements?

Furthermore in the validation of the hypotheses, the stakeholder was active, since he knows the needs and objectives of development projects. Thus a survey for stakeholders was also applied. This survey has the following purposes: 1) To measure the percentage of needs and objectives that the stakeholder consider that are specified correctly with the two applied techniques, 2) to determine how the analyst is dealing with customers and/or users communication in order to unify criteria, 3) to identify how stakeholder are included in the process of requirements elicitation and how the analyst makes use of information collected in the interview with the stakeholders.

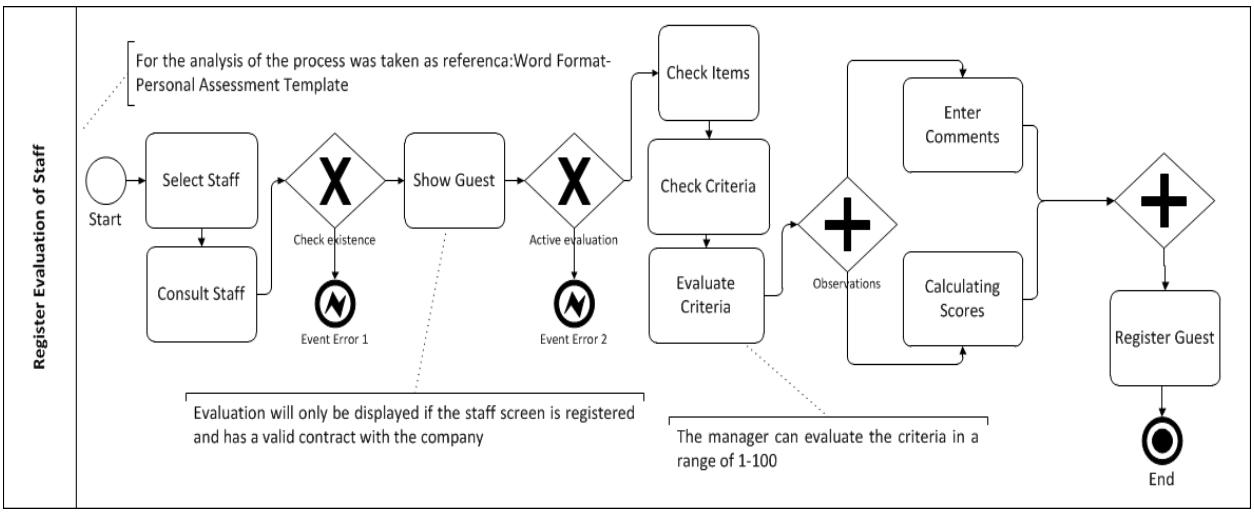


Figure 3. BP Modeling Project A to R1

To validate the hypothesis H5 – H6 both groups used the requirements elicitation techniques and subsequently they created iterations plan and release plans.

For generating the plan from user stories and BPMN models, it was used a template [22]. The release plan in the first iteration includes: priority, that may be low, medium or high, estimated effort at a range of 1-10, where 10 is the largest effort, delivery date. Each process or user story is reflected in specific programming tasks.

Each project module was developed by a couple of developers. At the end of this first iteration, an interview with the developers and stakeholder was performed to verify the status of each task and acceptance by the user. The interview allowed the validation of hypotheses H1 to H6 in the following items: 1) Measure the percentage of requirements that were implemented according to the needs of the stakeholder, after the first iteration with each of the techniques applied, 2) determine whether release plan generation from BP model specifies in greater detail the deliveries, unlike that with the use of user stories, 3) the activities of the iterations plan from the BP model specify more completeness, unlike when using user stories. Table 6 shows the questions asked in the survey to the evaluation.

Table 6. Survey for evaluation of release plan

Does the technique provide enough detail assigned for software development?
The selected technique allows you to know restrictions in some requirements during development time?
Did you take into account other sources of requirements different to the assigned technique? Which one?
Does the assigned technique allow you identify other requirements during development?
Was the release plan modified structurally at the end of the first iteration?
The esteemed effort in the release plan is consistent with what you experienced during development?
Does the assigned technique allowed you to establish a precise number or tasks? During development these tasks were increased / decreased?
Does the estimated delivery date of each user story / BP correctly established?

5. Discussion of Results

5.1. Validation of the hypotheses

H1: According to information collected in the survey, requirements documentation in natural language creates ambiguities and different interpretations by the analyst when other person review the description, furthermore it was not possible to establish whether the requirements contradicted each other, for example when multiple stakeholder were involved in the process of requirements elicitation.

Moreover, in the analyst's survey, the percentage of the total system information that could be captured with this technique was between 50% and 80%, indicating that the requirements are not defined completely in this phase. However it was established that the stakeholder's participation was 75% favoring validation of requirements. In addition language used by stakeholder to specify his needs was clear, but could be improved.

A second survey with the second group of analysts who applied the identification of requirements using BP models in projects was done. This survey allowed establishing the utility of the use of BP models for representing stakeholder's needs. According to information collected in the survey, documentation requirements through BP model generate more clarity and a single interpretation by the analyst when reviewing models made by another analyst, equally, the BP model, allowed to identify consistency between the requirements using relationships / dependencies between processes. It also clearly defines the assignment of responsibilities to each role or an actor through the Lanes. From the results of this survey it was highlighted that 87% of analysts indicated that the requirements modeled through BP allowed validating information with stakeholder, to verify that the sequence of activities was adequate, and besides if the information requested in each activity was complete.

H2: eliciting requirements through BP after performing user stories increases domain understanding by those involved. A group of analysts used user stories for requirements elicitation in 3 projects. It was established that 73.33% of the analysts found that user stories written by another analyst were ambiguous and that its interpretation was not the same of the stakeholder. In these surveys it was established that 56% of the analysts said that they captured between 50 and 70% of the requirements.

After application of user stories, the same group use the BP model to raise the requirements, at this point it was found that 95% of analysts affirm that BP Model allow them to define a concise and complete stakeholder requirements. Now, from the previous survey increased from 22% of analysts who say that when reviewing a model BP has made another analyst team are not ambiguous and its interpretation was that the same that stakeholder had expressed. The rate of requirements capture was between 78% and 89% with this technique, demonstrating that BP model allows to increase by 19% the capture and identification of requirements.

H3: eliciting requirements through user stories after making the BP model increases domain understanding. A group of analysts applied first BP for requirements elicitation in 4 projects, through a survey, I was established that for 80% of the analysts understand the notation, considering they pass first through a training process. Now for the stakeholder, it was found that 83.33% said that the notation was easy to understand, because the analyst generates the diagram with him (stakeholder) in real time, which allows both parties to resolve doubts immediately. The validation of this hypothesis also found that 73.33% of analysts indicated that when reviewing a model of BP was made by another analyst, no ambiguities arise and that his interpretation was that the same that the stakeholder said.

After applying the BP model, the same group used user stories for requirements elicitation. In this process it was found that 83% of analysts used models that BP had done before, to write the story. In addition, it was noted that 85% of analysts indicated that when they conducted the review of user stories written by another analyst, these were not ambiguous and that its interpretation was that the stakeholder had said. The results therefore validate the four hypotheses allowing to evidence that the BP model can increase by 19%, the number of identified requirements, besides the ambiguity in the requirements specification is decreased by 25%. In effect of reducing ambiguity and increasing identification requirements, can demonstrate that using BP models increases the level of understanding between the different actors in the process of building software in XP.

H5 - H6: Generation Release Plan with the projects worked in the previous stage, it was evaluated the result of the first iteration. The results are listed in Table 7. The results of the first iteration allow identifying that plans generated from BP models reach 98.33% in the tasks approved by the stakeholder and implemented by the development team, which exceeds 23% of the requirements elicitation technique with user stories in the first iteration, which achieves 75.85 of the Release Plan.

Table 7. Execution results of the first iteration[1]

P	T	TT	NT	%	NS	%	NI
H	BP	15	15	100	15	100	100%
I	BP	18	17	94,44	17	94,44	100%
J	BP	15	15	100	14	93,33	93,33%
K	BP	14	12	85,77	12	85,71	100%
H	HU	16	13	81,25	13	81,25	92,86%
I	HU	19	14	73,68	16	84,21	100%
J	HU	11	9	81,81	9	75,00	90 %
K	HU	12	8	66,66	6	54,55	75%

Additionally at this stage, it was used some metrics that allowed to establish the value of stakeholder. With these metrics, the stakeholder will know if functionalities are being implemented according to functional and time requirements. For this, a set of different metrics of different related aspects (see Table 8). It can be seen that with the use of BP model, the averaged percentage of requirements that were not approved by the stakeholder at the end of the first iteration was 20% lower than the average of the requirements with user stories.

Table 8. Validation metrics for hypotheses H5-H6

Metrics evaluated by iteration	User story	BP model
Average requirements completed in iteration	12	14,75
Average incorporated changes and added requirements on the initial scope of the project	4	1
Average completed requirements for total iteration requirements.	75,85	98,43
Average conditions not approved by the customer or us	26,25	6,63

The results obtained when evaluating the hypothesis H5 –H6 demonstrate that proposed technique allows 23% more fully developed requirements by iteration, consequently the functionality defined by the stakeholder is widely covered. The latter reduces the time of software development, due to the fact that it is not required new stakeholder feedback. The reduction in changes avoids delays and allows the development times stays in line with customer needs.

5.2. Results of stakeholder and customers surveys

The use of BP model for requirements elicitation instead of using user stories provides some advantages: better communication between analysts and stakeholders, allowing establishing requirements more clearly. BP models allow representing requirements in a unified and standardized way, representing the process sequence just as it works in in the organizations. Furthermore, understand a model through BP notation by the customer does not require much time and additional training. The stakeholder describes the BP model and may validate it at real time. In addition it was found in the study that 87% of stakeholders and customers (see Figure 4) indicated that the notation used by the analyst to specify their requirements were clear and for easier to understand compared to the use of user stories. With BP models, customers can actively participate in the validation requirement, due to the fact that they can add or define clearly the sequence of activities to develop, also assign users to specific tasks as well as the data required in each task.

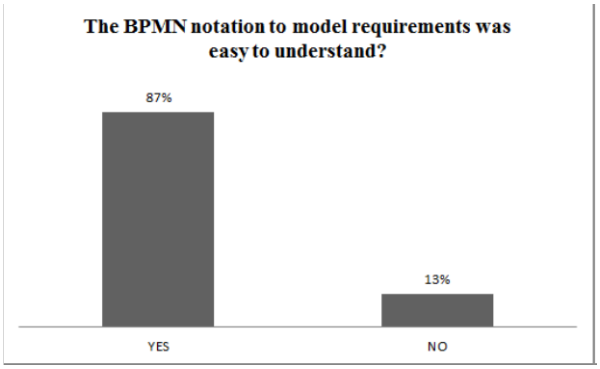


Figure 4. Understanding of BP model by stakeholder.

6. Conclusions and Future Work

Preliminary results allow evidencing the follow issues: Using BP models for requirements elicitation improved the communication between the analyst, stakeholder and customers. BP notation used was easier to understand that natural language. Furthermore, using BP models was much more productive and easy to understand by 80% of analysts, which represents a major improvement in this activity. Analysts also emphasize that BP models can be more productive, because they elicit a greater number of requirements compared to user stories. BP models improve interpretation and understanding, of requirements elicited by other analyst team. This is because the BP models are defined a common and standardized language allowing to clearly identify the sequence of activities and what the user expects the system. In relation to the support among the techniques of elicitation of requirements, we can say that the user story itself did not allow fully clarify requirements compared to BP models. The description of requirements in user stories created from

[1] P: Project T: Technique used, TT: Total specified tasks, NT: Number of implemented tasks, NS: Number of tasks approved by the stakeholder, NI: Number of approved and implemented tasks

BP models increased understanding in 11.67 % due to the fact that BP models allowed making clearer the redaction of the user story by analysts. However, it is convenient to use two techniques for requirements elicitation, this is because this will add an additional activity that does not contribute more to the process would and this will affect one of the agile manifests that supports on simplicity. The generation and implementation of release plan from BP model is faster and generates a higher percentage of implemented and approved requirements at each iteration, this allows that the development to be best fit to the customer needs.

Future work includes further experimentation for verifying the impact of BP models in test phases. The latter is due to the fact that when designing a system with an efficient communication with the stakeholder may lead to a reduction of errors during test phase. Another field future work, it will be explored the use of BP models in other agile methodologies such as SCRUM. The latter with the aim of improve communication between the scrum manager, developers and stakeholders inside the planned sprints.

7. References

[1] N. Paternoster and C. Giardino, "Software development in startup companies: A systematic mapping study," *Inf. Softw. Technol.*, pp. 1200–1218, 2014.

[2] R. C. Nguyen-Duc, A., D.S. Cruzes, "The impact of global dispersion on coordination, team performance and software quality-A systematic literature review," *Inf. Softw. Technol.*, 2014.

[3] H. B. Christensen, "Analysis and design of software ecosystem architectures – Towards the 4S telemedicine ecosystem," *Inf. Softw. Technol.*, vol. 56, pp. 1476–1492, 2014.

[4] F. J. Pino, "Assessment methodology for software process improvement in small organizations," *Inf. Softw. Technol.*, vol. 52, pp. 1044–1061, 2010.

[5] I. F.-C. Losada, B., M. Urretavizcaya, "A guide to agile development of interactive software with a 'User Objectives'-driven methodology," *Sci. Comput. Program.*, vol. 78, pp. 2268–2281, 2013.

[6] P. S. T. a. S. Verma, "A closer look at extreme programming (XP) with an onsite-offshore model to develop software projects using XP methodology," *Bus. Inf. Process.*, vol. 16, pp. 166–180, 2009.

[7] M. Blom, "Is Scrum and XP suitable for CSE Development?," *Procedia Comput. Sci.*, vol. 1, pp. 1511–1517, 2010.

[8] J. Domann, "An Agile Method for Multiagent Software Engineering," *Procedia Comput. Sci.*, vol. 32, pp. 928–934, 2014.

[9] M. and A. T. Chinosi, "Computer Standards & Interfaces BPMN : An introduction to the standard.," *Comput. Stand. Interfaces*, vol. 34, pp. 124–134, 2012.

[10] S. Štolfa and I. Vondrák, "A description of business process modeling as a tool for definition of requirements specification," *Proc. Syst. Integr.*, pp. 463–469, 2004.

[11] K. POHL, "The three dimensions of requirements engineering," *Semin. Contrib. to Inf. Syst. Eng.*, pp. 63–80, 2013.

[12] BPMI.ORG & OMG, "Business Process Modeling Notation Specification. Final Adopted Specification. Object Management Group," 2006.

[13] J. Newkirk, "Introduction to agile processes and extreme programming," *Proc. 24th Int. Conf. Softw. Eng. - ICSE '02*, p. 695, 2002.

[14] K. I. Tong, "Chapter 8 Managing Software Projects with User Stories," in *Essential Skills for Agile Development*, 2010, pp. 217–252.

[15] A. Jaqueira, M. Lucena, E. Aranha, F. Alencar, and J. Castro, "Using i * Models to Enrich User Stories Objectives of the research," no. iStar, pp. 55–60, 2013.

[16] G. Zheng, "Implementing a business process management system applying Agile development methodology : A real-world case study," 2012.

[17] A. Ottensooser, A. Fekete, H. a. Reijers, J. Mendling, and C. Menictas, "Making sense of business process descriptions: An experimental comparison of graphical and textual notations," *J. Syst. Softw.*, vol. 85, no. 3, pp. 596–606, Mar. 2012.

[18] A.-W. Kent Beck, "Extreme Programming Explained: Embrace Change," 2000.

[19] Y. Liu and G.-L. Tian, "A variant of the parallel model for sample surveys with sensitive characteristics," *Comput. Stat. Data Anal.*, vol. 67, pp. 115–135, 2013.

[20] J. Escobar and F. I. Bonilla-Jimenez, "GRUPOS FOCALES : UNA GUÍA CONCEPTUAL Y METODOLÓGICA," vol. 9, no. 1, pp. 51–67, 2009.

[21] M. Qasaimeh, "Extending Extreme Programming User Stories to Meet ISO 9001 Formality Requirements," *J. Softw. Eng. Appl. 04(11)*, pp. 626–638, 2011.

[22] G. van Valkenhoef, T. Tervonen, B. de Brock, and D. Postmus, "Quantitative release planning in extreme programming," *Inf. Softw. Technol.*, vol. 53, no. 11, pp. 1227–1235, Nov. 2011.

MultiSearchBP: Entorno para búsqueda y agrupación de modelos de procesos de negocio

Hugo Ordoñez, Juan Carlos Corrales, Carlos Cobos

***Resumen*—El artículo presenta un entorno para búsqueda y agrupación de procesos de negocio denominado MultiSearchBP. Es basado en una arquitectura de tres niveles, que comprende el nivel de presentación, nivel de negocios (análisis estructural, la indización, búsqueda y agrupación) y el nivel de almacenamiento. El proceso de búsqueda se realiza en un repositorio que contiene 146 modelos de procesos de negocio (BP). Los procesos de indización y de consulta son similares a los del modelo de espacio vectorial utilizado en la recuperación de información, y el proceso de agrupación utiliza dos algoritmos de agrupación (Lingo y STC). MultiSearchBP utiliza una representación multimodal de los BP. También se presenta un proceso de evaluación experimental para considerar los juicios de ocho expertos evaluadores a partir de un conjunto de los valores de similitud obtenidos de comparaciones manuales efectuados con anterioridad sobre los modelos de BP almacenados en el repositorio. Las medidas utilizadas fueron la precisión gradual y el *recall* gradual. Los resultados muestran una precisión alta.**

***Palabras Clave*—Procesos de negocio, recuperación de información, búsqueda multimodal, agrupamiento.**

MultiSearchBP: Environment for Search and Clustering of Business Process Models

***Abstract*—This paper presents a Business Process Searching and Grouping Environment called MultiSearchBP. It is based on a three-level architecture comprising Presentation level, Business level (Structural Analysis, Indexing, Query, and Grouping) and Storage level. The search process is performed on a repository that contains 146 Business Process (BP) models. The indexing and query processes are similar to those of the vector space model used in information retrieval and the clustering process uses two clustering algorithms (Lingo and STC). MultiSearchBP uses a multimodal representation of BPs. It also presents an experimental evaluation process to consider the judgments of eight expert evaluators from a set of similarity scores obtained from previous manual comparisons made between the BP models stored in the repository. The measures used were graded precision and graded recall. The results show high accuracy.**

***Keywords*—Business processes, information retrieval, multimodal search, clustering.**

Manuscrito recibido el 18 de marzo de 2013; aceptado para la publicación el 27 de julio del 2013; versión final 16 de junio de 2014.

Hugo Ordoñez está con la Facultad de Ingeniería, Universidad de San Buenaventura, Cali, Colombia, y el Grupo de Ingeniería Telemática de la Universidad del Cauca, Colombia (correo: hugoeraso@gmail.com).

Juan-Carlos Corrales está con el Departamento de Telemática, Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca, Colombia (correo: jcorral@unicauca.edu.co).

Carlos Cobos está con el Departamento de Sistemas, Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca, Colombia (correo: ccobos@unicauca.edu.co).

I. Introducción

La apertura de los mercados y la globalización del comercio hacen que las empresas centren su atención en la oferta de nuevos productos y servicios con el propósito de atraer más clientes y de esta forma mantener o mejorar el nivel de ventas y su posicionamiento en el mercado [1]. Para lograr lo anterior, aplican estrategias que satisfacen la demanda y los requerimientos de clientes conocedores y expertos que cada día exigen más [2]. Entre estas demandas se encuentran: agilidad y calidad de servicio, rebaja de costos, disminución de tiempos, calidad de productos, agilidad en las transacciones, entre otras. Esto exige que las empresas se organicen entorno a funciones del negocio tales como: mercadeo, ventas, producción, finanzas y servicio al cliente, donde cada una de ellas se ejecutan de forma independiente según su propio modelo de negocio [3]. La aparición de los Business Process Management Systems (BPMS) permiten agilizar estas funciones dentro de la empresa facilitando su organización en torno a procesos de negocio (BP) [4], [5]. 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.

Los lineamientos organizacionales definidos por las empresas se modelan por medio de BP, que son formados por procedimientos o actividades que colectivamente alcanzan un objetivo o política de negocio, definiendo roles y relaciones funcionales [6]. La organización por BP permite a las empresas adaptarse más eficientemente a las necesidades de los clientes, ya que los BP pueden ser modificados en cualquier momento y tantas veces como sea necesario [7].

Los BP en las organizaciones son normalmente modelados o creados por expertos, utilizando herramientas para el diseño de BP en donde plasman las operaciones o tareas que se necesita ejecutar en la organización. Las organizaciones que pretenden diseñar o modelar un nuevo BP tienen que empezar revisando grandes cantidades de información acerca de los BP existentes (normalmente almacenados en repositorios de BP).

Dentro de esta información están las instrucciones del trabajo a realizar, quién debe realizarlo y la descripción de las

conexiones con otros sistemas [8]. Esta información es almacenada en archivos que contienen los registros de transacciones conocidos como "logs" o trazas de ejecución [7], [9]. Posteriormente la información revisada sirve como base para el replanteamiento o remodelamiento de un nuevo BP que cumpla con los nuevos requerimientos de la organización [10]. El éxito en la búsqueda (descubrimiento) de los BP sobre los repositorios empresariales permite a los diseñadores reutilizar efectivamente los BP desarrollados previamente y, así disminuir el tiempo de desarrollo de los nuevos BP.

De acuerdo con lo anterior, es necesario contar con un mecanismo de gestión de información eficiente que permita buscar (descubrir) los datos generados por los BP con el propósito de encontrar aquellos BP que más similitud tienen con el comportamiento de las tareas ejecutadas en la organización y que se esperan usar para definir un nuevo BP, para un área del negocio especifica [11], [12].

En esta investigación se propone un entorno que permite el descubrimiento y agrupación de BP por medio de consultas, que contemplan características estructurales y componentes textuales. El entorno se evaluó con base en un repositorio de BP modelados con Business Process Modeling Notation (BPMN), representado en sintaxis XML, mediante el lenguaje Processing Description Language (XPDL). El entorno se basa en el modelo espacio vectorial para la representación de los BP, incorpora características de representación multimodal (que utiliza información estructural y textual) y usa algoritmos de clustering para realizar agrupaciones con base en la similitud de los BP recuperados en la consulta del diseñador.

El resto del documento está organizado de la siguiente manera. La sección 2 presenta trabajos relacionados. La sección 3 describe el entorno propuesto, sus algoritmos y algunas interfaces. La sección 4 muestra los resultados preliminares de la evaluación del modelo. Finalmente, se presentan las conclusiones y el trabajo futuro que el grupo de investigación espera desarrollar en el corto plazo

II. TRABAJOS RELACIONADOS

El tema de interés central en esta investigación es el descubrimiento de BP y la agrupación (clustering) de los mismos. A continuación se presenta un resumen de los trabajos más destacados y al final de cada sección se hace un resumen de las deficiencias de los enfoques propuestos hasta el momento.

A. Descubrimiento de BP basado en lingüística

En [11] los autores plantean un sistema de búsqueda de BP que extiende semánticamente la consulta. Cuenta con un editor de BP basado en redes de Petri e incorpora un repositorio en el cual todos los BP son etiquetados con metadatos. En este trabajo se crea un índice de búsqueda, se eliminan palabras vacías y se ponderan los términos presentes en actividades y estados del BP. El sistema cuenta con dos opciones de búsqueda, una básica y otra extendida. La búsqueda básica consulta sobre todos los modelos presentes en el repositorio o sobre un modelo en especial e incorpora WordNet como elemento de generación de sugerencias semánticas en las búsquedas. Por otra parte, la búsqueda extendida considera a cada actividad del BP como un vector de términos agregando una función de costo parcial, con la cual se calcula una función de costo total. El ordenamiento de los resultados de la consulta se realiza con los valores de la función de costo total más bajas o de menor peso.

En [13] se propone un método de compresión de lingüística basado en redes de Petri , donde se resaltan dos contribuciones realizadas, a saber: 1) un argumento teórico para establecer el grado de compresión de la lingüística, abordando la semiología (estudio de signos) de los gráficos, en donde identifican ocho variables visuales distintas que pueden ser utilizadas para codificar la información de la gráfica del BP y el color es tomado como una de las variables más eficaces para distinguir los elementos de la notación. 2) la formalización de conceptos en el modelado de flujos de trabajo (*workflows*), para lo cual toma el BP como un grafo dirigido bipartito donde P es un conjunto de nodos llamados lugares, T un conjunto de nodos llamados transiciones y Fp $(P \times T) \cup (T \times P)$ es una relación de flujo binario basado en un operador que mapea cada conjunto de nodos T. Para realizar la búsqueda del nuevo modelo ejecuta un algoritmo denominado (max-flow-min-cut) que realiza emparejamiento de nodos para encontrar el flujo máximo de coincidencias de los operadores de conexión.

En [14] se presenta un método de búsqueda basado en descomposición de BP creando un análisis híbrido entre estructura y relevancia. El algoritmo está basado en un análisis iterativo del grafo que representa al BP. La descomposición crea fragmentos de procesos reutilizables (RPF), los cuales cumplen las siguientes características: 1) Un RPF debe ser conectado de manera que todos los nodos puedan llegar desde una entrada de borde o arista, y 2) Cada RPF debe tener sólo una arista de entrada o de salida o ambos en común interconectados con otro fragmento. En este proceso se tiene como meta de búsqueda extraer la frecuencia de ocurrencia más alta en las tareas de los BP representados por los fragmentos generados.

En [15] los autores proponen un método de búsqueda de BP mediante la aplicación de reglas de asociación para información no estructurada. El proceso es llevado a cabo utilizando datos no estructurados en lugar de los registros de las aplicaciones. La ejecución del algoritmo de detección de reglas está dividida en dos: 1) la obtención de la asociación entre los documentos y procesos, 2) construcción de un modelo de lenguaje estadístico para identificación de normas relacionadas con el proceso y las actividades que se presentan en los documentos. La construcción del modelo está dividida en dos actividades principales: el algoritmo analizador, que detecta frases relacionadas con las actividades del proceso por medio de una ontología de dominio, y la identificación de patrones que utiliza una heurística, basada en los elementos de

la ontología de dominio y las sentencias del documento de búsqueda. En la recuperación de los BP se utiliza la detección de patrones, el cálculo de su frecuencia y las asociaciones de las actividades.

B. Descubrimiento de BP basado en agrupamiento (Clustering)

En [16] los autores plantean un algoritmo de clustering secuencial con el propósito de organizar una serie de objetos en un conjunto de grupos, donde cada grupo contiene objetos que son similares por un tipo de medida. Esta medida depende del tipo de objetos o datos presentes en los BP. Cada grupo está asociado con un modelo probabilístico, por lo general una cadena de Markov (al igual que el presentado en [17], [18]). Si para todos los grupos se conocen las cadenas de Markov, entonces cada secuencia de entrada es asignada a la agrupación que mejor pueda producir tal secuencia. El algoritmo desarrolla los pasos siguientes: 1) Inicializa los modelos de cluster (es decir, la cadena de Markov para cada grupo) al azar. 2) Asigna a cada secuencia de entrada el grupo que es capaz de producirlo con la mayor probabilidad. 3) La estimación de cada modelo de clúster de la serie de secuencias que pertenecen a ese grupo. Finalmente, se repiten los pasos 2 y 3 hasta encontrar los modelos de cada cluster o grupo.

En [19] plantean un enfoque de clustering que agrupa secuencias similares e identifica tópicos temáticos presentes en los BP sin la necesidad de proporcionar información de entrada. La agrupación es realizada con el propósito de encontrar información valiosa sobre el tipo de secuencias que se están ejecutando en los BP. El procedimiento de agrupación incluye: Un algoritmo alfa el cual es capaz de volver a crear el BP a través de una red de Petri, con base en las relaciones encontradas en el registro de ejecución de los BP. Métodos de inferencia que consideran el registro de ejecución como una secuencia simple de símbolos, inspirada en el modelo de Markov (al igual que el presentado en [17]) y que genera un modelo gráfico que considera cadenas de Markov de orden creciente con grafos acíclicos dirigidos. Un algoritmo de Clustering jerárquico que tiene en cuenta un amplio conjunto de trazas de ejecución de un mismo proceso, que separa las trazas en grupos y encuentra el gráfico de dependencias por separado para cada grupo. Un algoritmo genético donde las soluciones candidatas son evaluadas por una función de aptitud y cada solución es representada mediante una matriz causal, es decir, un mapa de las entradas y dependencias de salida para cada actividad.

En [18] presentan un esquema de agrupación de BP (tal como en [20], [21]) para recuperación de esquemas gráficos en grupos similares de (sub) procesos y sus relaciones. Se parte de un macro proceso para llegar hasta las actividades más sencillas, para lo cual se toma un conjunto de grafos dirigidos $G_i = <N_i, A_i>$ donde N_i es el conjunto de nodos y $A_i \subseteq N_i \times N_i$ es el conjunto de arcos posiblemente etiquetados, generando un esqueleto de agrupación típica de subestructuras. Los grafos son iterativamente analizados para descubrir en cada paso un grupo de sub-estructuras isomorfas. El clustering se utiliza para comprimir los grafos sustituyendo a cada ocurrencia de la subestructura con un nodo; este proceso se repite hasta que no haya más compresión posible.

C. Diferencias con los trabajos previos

Las propuestas anteriormente descritas en el descubrimiento lingüístico de BP se limitan al emparejamiento de entradas y/o salidas tomando como base la información textual o gráfica y las relaciones semánticas que se encuentra en la notación de estos elementos, además deja de lado el flujo de ejecución o comportamiento. En el proceso de búsqueda los resultados no tienen en cuenta similitud en patrones frecuentes, tipo de actividades, finalidad de la tarea o actividad. Por otro lado en las propuestas de descubrimiento basado en agrupación se eliminan secuencias que solo ocurren una sola vez sin tener en cuenta que pueden ser relevantes para los modelos que forman cada grupo, además la agrupación de atributos internos se mide separando su comportamiento de las propiedades estructurales y los atributos externos son medidos con datos tales como: tiempo de duración, número de errores, costo de ejecución. Esta medición de atributos hace que el costo computacional del algoritmo sea demasiado elevado.

Para alcanzar mayor relevancia de los resultados reportados en los sistemas de descubrimiento de BP, en esta propuesta se plantea un entorno que unifica en un solo espacio de búsqueda, unidades de comportamiento y características textuales de los BP, en lo que se conoce como una representación multimodal. Adicionalmente, integra el uso de algoritmos de clustering para agrupar los resultados de la búsqueda (descubrimiento) con base en la similitud de las características representadas en los modelos de BP descubiertos y lograr así una forma más efectiva de visualización de los resultados.

III. El entorno propuesto

El entorno propuesto, llamado MULTISEARCH*BP*, esta implementado sobre la tecnología Java y es soportado por una arquitectura organizada en 3 capas como se muestra en la La fig. 1. Está compuesta por: 1) un nivel de presentación desde la cual el usuario puede gestionar los BP (adicionar, eliminar, modificar y buscar BP) almacenados en el repositorio y el índice. 2) un nivel de lógica de negocio que se encarga de gestionar los BP, extraer las características estructurales y los componentes textuales de los BP e indexarlos, también responde a las opciones de búsqueda con dos tipos de respuesta: lista lineal ordenada de BP o grupos temáticos de BP que se relacionan con la consulta del usuario (diseñador) y finalmente, 3) un nivel de almacenamiento que se encarga de dar persistencia a los procesos de negocio y al índice de búsqueda. A continuación se explican cada uno de los componentes de esta arquitectura.

Formas para Adicionar / Actualizar / Eliminar: Corresponde a la interfaz grafica de usuario (GUI) usada para adicionar, modificar y eliminar BP del repositorio y del índice.

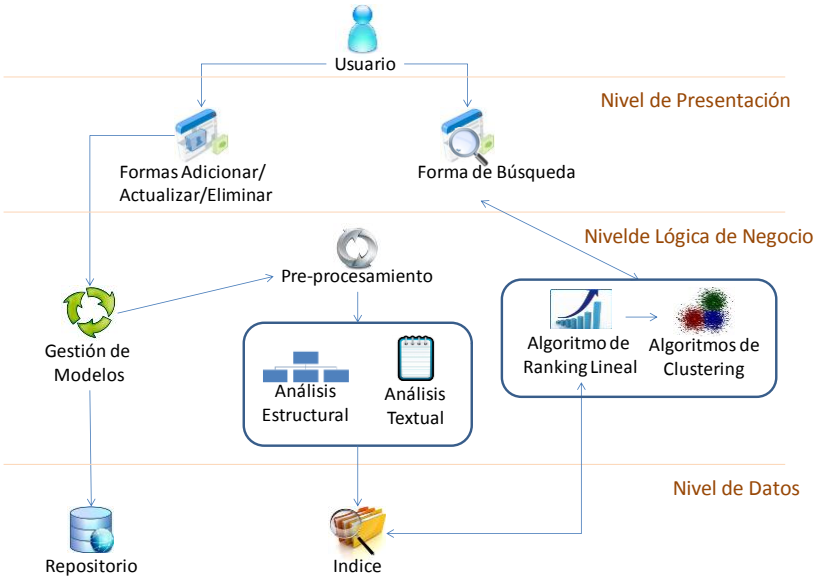


Figura 1. Arquitectura del entorno

Gestion de modelos: Este componente permite hacer gestión sobre los BP, que están en sus formatos originales XML, y representan los modelos de *BP*MN (Business Process Modeling Notation) con sintaxis XPDL (XML Process Definition Language). Estos pueden ser BP de referencia para procesos de dominio específico o BP que ejecutan un conjunto de tareas de una colección empresarial y que pueden ser reconfigurables.

Repositorio: Es la unidad central de almacenamiento y gestión, es similar a una base de datos que comparte información acerca de los artefactos de ingeniería producidos o utilizados por una empresa [10], [22]. Para la evaluación del presente entorno se usó un repositorio con 146 BP. Para cada BP se almacenan las tareas, su*b-p*rocesos y flujos de control.

Cuando la colección de BP se indexa, se realizan tres tareas fundamentales: el pre-procesamiento de cada BP, luego el analisis textual, después el analisis estructural y finalmente la creación del índice completo de la colección. Es preciso tener claro, que el índice se crea para toda la colección, pero también se puede realizar incrementalmente, es decir, uno a uno cada BP.

El **Pre-procesamiento** se encarga de convertir los términos textuales del BP a minúsculas, eliminar caracteres especiales, eliminar palabras vacías, eliminar acentos, y aplicar stemming (algoritmo de porter [27], [28]) para convertir cada uno de los componentes textuales de los BP a su raíz léxica (por ejemplo "fishing" y "fished" en "fish").

En el **Analisis textual** se lee cada uno de los elementos del conjunto $T:\{BP \mid BP\}$ presentes en el repositorio *S*, para lo cual cada uno de los elementos de *T* es representado en forma de árbol (*A*) tal que ($BP_i = A_i \rightarrow (v, x)$ donde *v* es un nodo y *x* representa las aristas). El proceso inicia tomando cada A_i, para extraer las características textuales C_{tj} (nombre de actividad, tipo actividad y descripción) para formar un vector, es decir $\{C_{tj1}, C_{tj2}, \ldots, C_{tjN}\}$, que corresponde a una fila de la matriz MC_{ij} del componente de caracteristicas textuales, donde *i* representa los BP y *j* representa las características textuales de cada uno de estos.

El **Analisis estructural** incorpora una estrategia de formación y uso de libros de códigos (codebooks) para generar unidades estructurales básicas secuenciales de los BP. Estos codebooks son construidos con base en las propiedades de similitud en patrones secuenciales frecuentes en la estructura de cada uno de los BP. Generalmente los codebooks han sido empleados en el dominio de recuperación de imágenes utilizados como histogramas de patrones visuales [29] y como vocabularios o diccionarios visuales [23], [24], [25]. Además se utilizan para analizar y buscar ocurrencias de palabras en transcripciones de texto [26].

En este paso se ejecuta el algoritmo (Parser*BP*toCodebook) que analiza la estructura de los modelos de BP almacenados en el repositorio. En este proceso se recorre de manera secuencial la estructura en árbol de los archivos XPDL donde se describe cada BP, para formar una matriz *MC* de características textuales y una matriz *MCd* de componentes estructurales (usando codebooks). Este paso se realiza tomando cada $A_i \ni vt$ (vector de transiciones), donde $vt=\{t_{j1}, t_{j2},\ldots, t_{jn}\}$, de lo cual $\forall Cd_i = (vt-1, vt); \quad i \geq 2$ con esto se tiene $A_i = \sum_{i=1}^{n} Cd_i$ formando de esta manera la matriz MCd_{ij} de componentes codebook, donde *i* representa los BP y *j* representa los codebook de cada BP. LaFigura 2 hace una representación gráfica de la manera como se forma cada uno de los codebook de un BP. De lo cual es obtenido un vector de codebooks asi: $\{\text{Start\_TaskUser}_1, \text{TaskUser\_ParallelRoute}_2, \text{ParallelRoute\_TaskService}_3, \text{ParallelRoute\_TaskService}_4\}$.

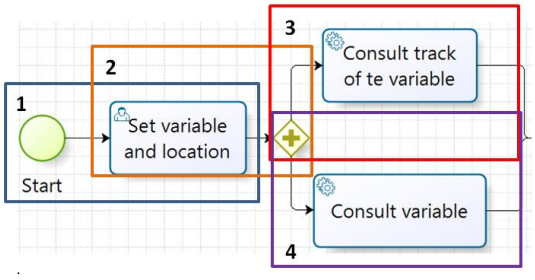


Figura 2. Estructura de cada *codebook*

	1	2	3	4	5	6	7	8	9	10	m
BP_1	1	Cd_1	Cd_2	Cd_3	Cd_4	Cd_5	Ct_1	Ct_2	Ct_3	Ct_4	Ct_m
BP_2	2	w_{ij}					w_{ij}				
BP_3	3		w_{ij}					w_{ij}			
BP_4	4			w_{ij}					w_{ij}		
BP_5	5				w_{ij}					w_{ij}	
BP_n	n					w_{ij}					w_{ij}

Figura 3. Matriz índice (MI)

El **Índice** almacena información de dos tipos: 1) Indexación de las funciones de negocios en la cual se tiene en cuenta la información textual existente en cada BP. 2) indexación estructural la cual está basada en una caracterización entre tipos de tareas, tipos de eventos y tipos de conexiones. Estas dos formas de indexación se unifican (representación multimodal) para tener una representación más exacta del objeto de estudio. El índice almacena eficientemente una estructura conceptual denominada matriz índice (*MI*) de términos por BP (similar al modelo espacio vectorial de recuperación de información [5]), que almacena en cada celda un peso (w_{ij}), el cual refleja la importancia del componente textual en su raíz léxica o codebook contra cada BP. Esta matriz se basa en la ecuación (1) propuesta por Salton [29], [27], donde $F_{i,j}$ es la frecuencia observada del componente textual o del codebook j en el BP_i. Max(F_i) es la mayor frecuencia observada en el BP_i. N es el número de BP en la colección y n_j es el número de BP en los que aparece el componente textual o codebook j. Finalmente la matriz índice $MI = \{MCd_{ij} \cup MC_{ij}\}$ puede ser resumida gráficamente como se muestra en la Figura 3. Esta figura muestra dos zonas o componentes en la *MI*, la primera, muestra el peso de los elementos de cada codebook en cada BP y el segundo el peso de los elementos textuales en cada BP.

$$w_{i,j} = \frac{F_{i,j}}{\max(F_i)} \times \log\left(\frac{N}{n_j + 1}\right) \quad (1)$$

La **Forma de búsqueda** hace referencia a un interfaz gráfica en la cual el usuario puede realizar consultas de tres formas diferentes: 1) por palabras clave (textual), 2) estructural (codebooks), y por 3) combinada de texto y estructura (es decir las dos anteriores en forma conjunta).

La consulta por palabras clave: En estas consultas el usuario puede digitar una o varias por palabras clave representadas en lenguaje natural las cuales forman un vector de consulta $qpc = \{pc_1, pc_2, \ldots, pc_n\}$. El sistema pre-procesa las palabras clave, genera un vector de consulta con los términos registrados en la *MI* y luego compara esta consulta con la parte textual del índice para entregar aquellos BP más similares a la consulta.

La consulta estructural: En esta opción el usuario tiene la posibilidad de elegir uno o varios (codebooks) de una lista de componentes estructurales formados a partir de la colección de BP existentes en el repositorio para formar el vector de consulta $qcd = \{cd_1, cd_2, \ldots, cd_n\}$. Los elementos utilizados en la consulta son comparados con la parte del índice que contiene los componentes estructurales y retorna los BP más similares a dicha consulta.

La consulta combinada de texto y estructura: Este proceso de consulta integra las dos opciones de consulta anteriores. Para realzar este proceso el sistema forma automáticamente un vector de consulta $qmg = qpc \cup qcd$, el cual se compara con cada BP registrado en la matriz *MI*, tomando las dos zonas o componentes.

Para la comparación del vector de consulta con los BP registrados en el índice se parte de los datos introducidos en la consulta, los cuales son representadas en forma de vector de términos $q = \{t_1, t_2, t_3, \ldots, t_n\}$, además se convierten todos los términos de q a minúsculas, se eliminan palabras vacías, acentos, caracteres especiales, finalmente se aplica stemming (algoritmo de porter) para convertir cada uno de los términos de q a su raíz léxica. Con la cadena de consulta procesada se ejecuta la búsqueda en el espacio elegido por el usuario, a continuación se describa cada uno de los componentes de este nivel.

Consulta: En el proceso de ejecución de la consulta el modelo ordena y filtra los BP retornados, implementando la ecuación (2) de calificación conceptual (puntuación) definida en LUCENE [28].

$$Puntuacion(q,d) = coord(q,d) \times Qnorma(q) \sum_{t \in q} \left(tf(t \in d) + idf(t)^2 \times t.getBoost() \times norm(t,d) \right) \quad (2)$$

En la ecuación anterior t es un término de la consulta q y d es el documento consultando, $tf(t \in d)$ es la frecuencia del término en el documento, definida como el número de veces que el término t aparecen en el BP d. En esta medida los documentos de mayor puntuación son los que contiene mayor frecuencia del término, $idf(t)$ es la frecuencia inversa del término t en un BP (número de BP en los que aparece el termino t), $coord(q, d)$ es un factor de puntuación basado en el número de términos de la consulta que se encuentran en el BP consultado, los BP que contienen más términos de la consulta obtienen mayor puntuación, $Qnorma(q)$ es un factor de normalización utilizado para hacer las puntuaciones (para este modelo es tomado con el valor de 1 ya que no afecta la

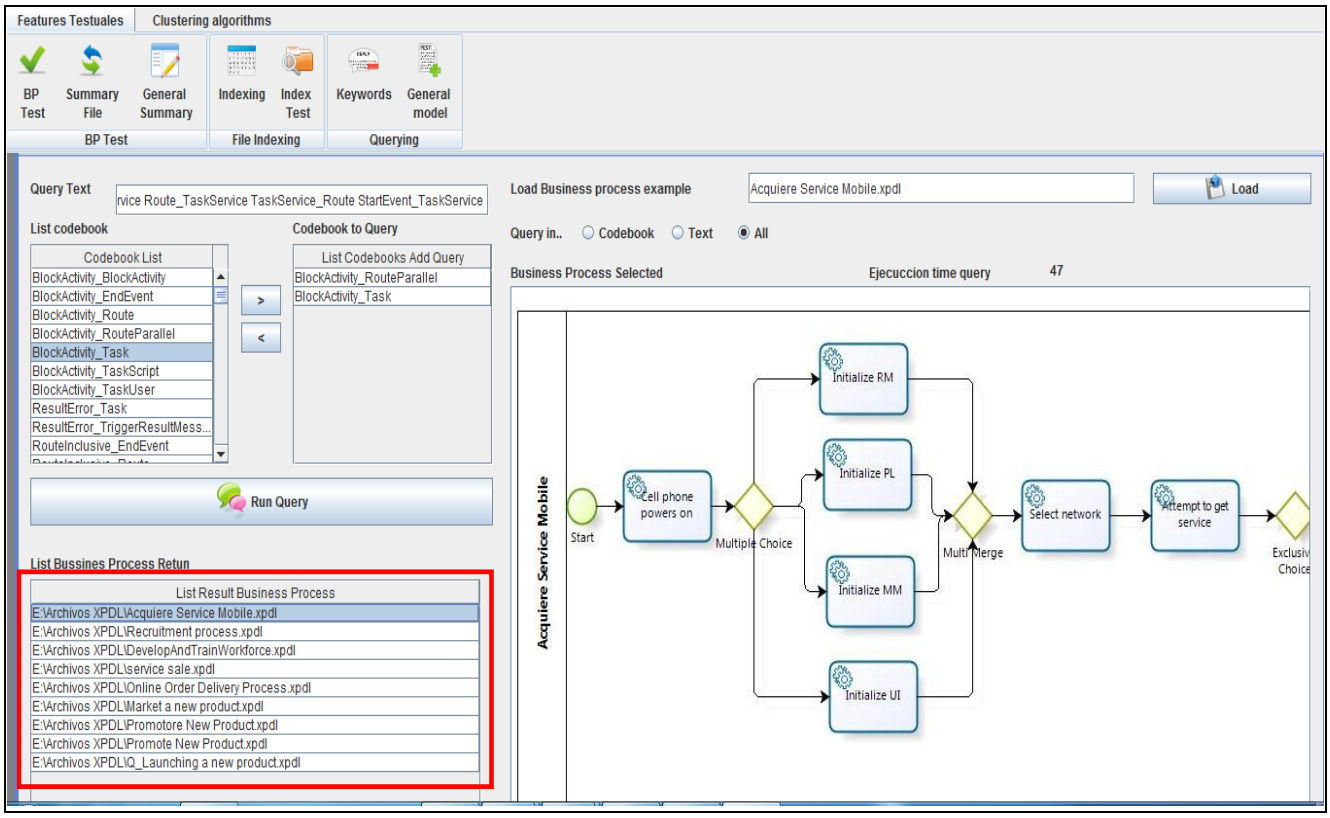


Figura 4. Opciones de consulta y despliegue de resultados en lista lineal ordenada

puntuación de cada BP evaluado). *t.getBoost*() es la ponderación del termino *t* en la consulta en este caso es igual a 1 debido que todos los términos de la consulta tienen la misma ponderación. *norm*(*t*,*d*) es un factor de ponderación en la indexación, tomado de w_{ij} en la indexación.

Una vez los resultados son ordenados y filtrados se listan en orden de acuerdo a la similitud (más similares a menos similares) que presentan con respecto a la consulta realizada por el usuario, quien puede elegir y visualizar cada uno de los modelos de BP recuperados.

Lista de resultados: Los resultados se despliegan al usuario en una lista ordenada dependiendo del nivel de relevancia, el cual es asignado obedeciendo a la puntuación definida por (2). En esta lista, el usuario puede elegir cada uno de los modelos de BP recuperados, para visualizarlos y analizarlos completamente. LaFigura 4 hace una representación gráfica de las opciones de consulta (parte izquierda central) y la lista de resultados (parte izquierda abajo enmarcada en rojo).

Nivel de agrupación: En este nivel se ejecutan los algoritmos de agrupamiento por afinidad o algoritmos de clustering [18,30] basado en las opciones de consulta explicadas en el nivel anterior, con el propósito de estructurar los resultados en grupos o familias de BP que contienen correlación en características textuales, estructurales o en ambas. Los algoritmos adaptados para este nivel son: LINGO y STC (Sufix Tree Clustering). A continuación se describen brevemente cada uno de ellos.

STC: Toma cada BP como una secuencia ordenada de términos que pueden ser textuales o estructurales, de lo cual se utiliza la información sintáctica de la secuencia para realizar la agrupación. Originalmente este algoritmo consta de tres pasos, 1) Limpiar BP, 2) Identificar clusters base y 3) Combinar clusters base. En este proyecto para aumentar el rendimiento y evitar el desarrollo de tareas redundantes del algoritmo, se eliminó el paso uno 1) Limpieza de BP, debido a que este paso se realiza previamente en el proceso de indexación.

El proceso de agrupación empieza realizando un árbol de sufijos a partir del vector que contiene todos los componentes textuales y de estructura de cada BP, se detecta una raíz, cada nodo al menos tiene dos hijos internos, las aristas entre nodos se etiquetan con una parte del texto resumen, las etiquetas de los nodos se forman uniendo el texto de las aristas, la clasificación del cluster base es realizado con la función $s(B)$, del cluster base B con frase P es: $s(B) = |B| \times f(|P|)$, donde $|B|$ = número de documentos en el cluster base B, $|P|$ = número de palabras en P que no tienen calificación 0, f = función que penaliza a las frases de una sola palabra y es lineal para frases de 2 a 7 palabras, además constante para frases mayores.

En la combinación de cluster base se tiene que en dos cluster base Bn y Bm, con tamaños $|B_m|$ y $|B_n|$. Sea $|B_m \cap B_n|$ el número de documentos comunes, La similitud entre B_n y Bm está definida como: 1 si $|B_m \cap B_n| / |B_m| > 0.5$ y $|B_m \cap B_n| / |B_n| > 0.5$ y 0 en cualquier otro caso.

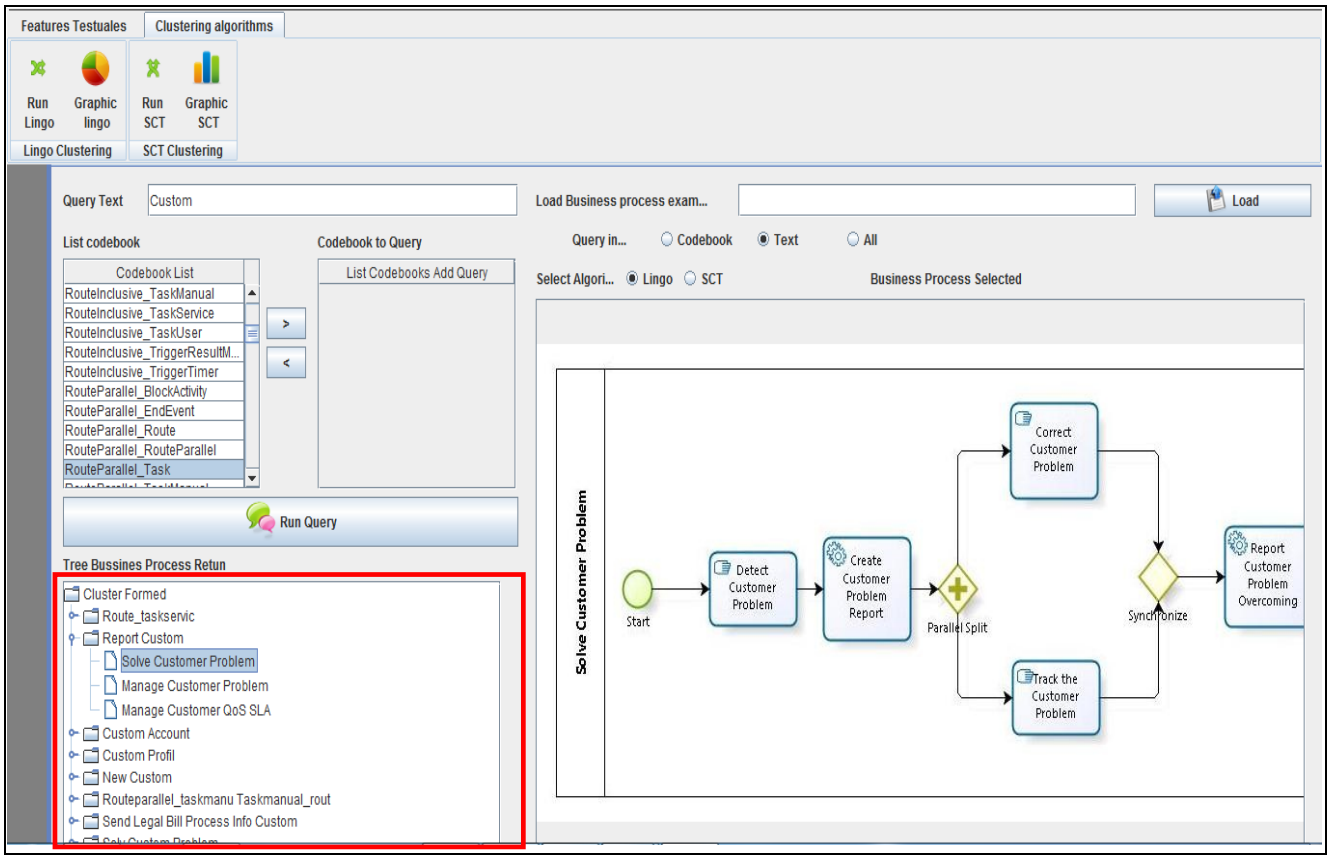


Figura 5. Opciones de consulta y despliegue de resultados en grupos temáticos

Lingo: En este algoritmo se realiza un resumen (Snippet) con los términos textuales y estructurales contenidos en cada BP descubierto en la consulta.

El algoritmo consta de cinco fases, 1) filtrado de texto, 2) extracción de características, que tiene como objeto identificar frases o términos que pueden ser candidatos para etiquetas de grupo, esto se realiza calculando el número de veces que aparecen dichas características en los BP recuperados, 3) inducción de etiquetas de cluster: en esta fase se forman descripciones significativas de grupo tomando la información de la matriz de términos por BP. Esta consta de cuatro pasos: valor del término en la matriz, el descubrimiento del concepto abstracto, la concordancia de la frase y el etiquetado, poda y evaluación, 4) descubrimiento de contenido de cada cluster: se comparan fragmentos de texto con todas y cada una de las etiquetas de grupo, para esto se forma una matriz Q en la que cada etiqueta de cluster es representada como un vector columna. De tal forma que $C=Q^TA$, donde A es el termino original de la matriz de términos por BP. De esta manera, el elemento c_{ij} de la matriz C indica el peso de adhesión del *BP j* en el grupo i, 5) formación final de clusters: se calcula con la formula valor-cluster = etiqueta-score × numero-veces, esta formación se ordena con base a la puntuación obtenida.

Al igual que en el algoritmo anterior se aumenta el rendimiento realizando la primera fase de filtrado de texto en el proceso de indexación. La Figura 5 muestra una representación gráfica de la agrupación de una consulta desplegada en forma de árbol (sección izquierda abajo enmarcada en rojo).

IV. Evaluación del entorno propuesto

Para determinar la calidad del entorno fue necesario someterlo a un proceso de evaluación experimental, con el objetivo de verificar la eficiencia en el proceso de descubrimiento de BP con base al modelo de similitud definido para las opciones de consulta que permite el entorno. Es preciso aclara que en la actualidad no se cuenta con la evaluación del proceso de agrupación. La experimentación se realizó teniendo en cuenta una colección cerrada de prueba elaborada con el juicio de ocho (8) evaluadores expertos en la temática de descubrimiento de procesos de negocio. Esta colección de prueba se realizó comparando manualmente los BP del repositorio con cada una de las consultas. En este proceso se realizaron un total de 1168 comparaciones manuales entre parejas de procesos de negocios, los cuales fueron comparados por los 8 evaluadores.

Para la evaluación se le solicitó a MultiSearchBP generar un ordenamiento (Ranking) de los 10 primeros Modelos BP (dispuestos por orden de similitud) retornados para satisfacer una necesidad definida por medio de una de las opciones 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 evaluación de sistemas de recuperación de información [27], [29]. Estas medidas son la Precisión gradada (*Pg*) y el Recall gradado (*Rg*) [31], las cuales proporcionan una clasificación de los BP_t considerados similares a un BP_q de acuerdo a diferentes niveles de relevancia. De esta manera, mientras precisión y recall solo consideran la cantidad de elementos relevantes recuperados, Pg y Rg tienen en cuenta la suma total de grados de relevancia entre la consulta y los BP. En el presente trabajo se utilizaron las ecuaciones (3) y (4) [32] para evaluar *Pg* y *Rg*, relacionando el ordenamiento de los BP obtenidos por el entorno (*fe*) y el ordenamiento de las evaluaciones manuales de los expertos (*fr*). En estas ecuaciones se midió la efectividad de la recuperación de una herramienta al comparar una consulta BP_q con cada elemento de una colección BP_{ti}. Por simplicidad se considera que $BP_q = Q$ y *que* $BP_t = T$:

$$Pg = \frac{\sum_{T_i \epsilon T} \min\{f_r(Q,T_i), f_e(Q,T_i)\}}{\sum_{T_i \epsilon T} f_e(Q,T_i)}, \quad (3)$$

$$Rg = \frac{\sum_{T_i \epsilon T} \min\{f_r(Q,T_i), f_e(Q,T_i)\}}{\sum_{T_i \epsilon T} f_r(Q,T_i)}. \quad (4)$$

LaFigura 6 presenta el nivel de precisión del entorno en el descubrimiento de BP. En este proceso se desarrollaron consultas tomando como consulta 8 modelos de BP del repositorio. Los resultados de evaluación de la *Pg* en el tipo de consulta basada en la estructura alcanzaron un 41%, mientras que para las consultas realizadas por palabra clave, el entorno alcanzó un porcentaje de 76%. Finalmente las consultas realizadas con el modelo general (estructura y texto) alcanzaron el 89% de *Pg*, lo que demuestra que las consultas por modelo general (características estructurales y componentes textuales) son mucho más precisas.

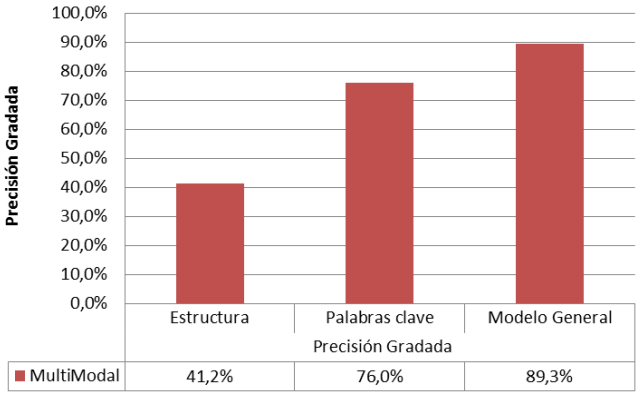


Figura 6. Grafica de precisión gradada

LaFigura 7 muestra niveles de *Rg* bajos en cada uno de los tipos de consulta, estos se encuentran en el 30% para consulta de estructura y por palabra clave (textual) mientras que el 22% para consulta por modelo general. Esto se debe a que solo se están evaluando los primeros 10 resultados y no toda la lista de resultados relevantes.

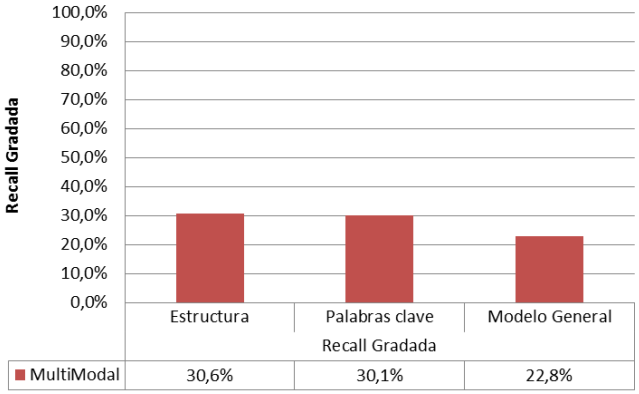


Figura 7. Grafica de recall gradada

V. Conclusiones y trabajo a futuro

En este trabajo se presentó un entorno para la búsqueda (descubrimiento) y agrupación de BP, el cual permite realizar varios tipos de consulta para ampliar el proceso de descubrimiento. Las opciones de consulta aportan flexibilidad al usuario ya que es posible replantear las búsquedas para aprovechar más el espacio de consultas y de esta forma aumentar la relevancia y pertinencia en los resultados retornados.

Los resultados obtenidos en la evaluación del entorno propuesto demuestran la eficiencia y relevancia en el proceso de descubrimiento de BP, ya que estos presentan similitud con la evaluación hecha por los expertos humanos. Alcanzando niveles de Precisión gradada que se encuentran entre el 41% como punto mínimo y 89% como punto máximo. Los resultados obtenidos en la medida de Recall gradada son bajos debido a que en el proceso de descubrimiento solo se están evaluando los primeros 10 resultados y no toda la lista de resultados relevantes, por ende no son tenidos en cuenta los BP clasificados como falsos positivos.

En el nivel de agrupación. Los grupos son formados mediante correlación y similitud directa entre características textuales, estructurales o ambas. La estructura de árbol formada permite al usuario revisar las categorías y seleccionar el grupo de mayor similitud a su consulta.

Como trabajo a futuro se propone realizar una clasificación manual de grupos de BP para poder evaluar la opción de agrupación del entorno. Evaluar la formación de grupos y comparar los resultados con otros entornos que se encuentren en el estado del arte. Incorporar ontologías de dominio específico con el propósito de realizar enriquecimiento semántico a los BP y las consultas, desarrollar un módulo de evaluación automática que genera graficas de relevancia. Ampliar la evaluación aplicando nuevas medidas para el descubrimiento de BP propuestas en [33].

AGRADECIMIENTOS

Los autores agradecen a la Universidad del Cauca y la Universidad de San Buenaventura – Cali, Colombia, por el apoyo dado al estudiante de Doctorado en Ingeniería Telemática Hugo Armando Ordóñez.

REFERENCIAS

[1] C. Cho, S. Lee, "A study on process evaluation and selection model for business process management," *Expert Systems with Applications*, vol. 38, no. 5, 2011, pp. 6339–6350

[2] Y. Gong, M. Janssen, "From policy implementation to business process management: Principles for creating flexibility and agility," *Government Information Quarterly*, vol. 29, 2012, pp. S61–S71

[3] H. Reijers, R. S. Mans, and R. van der Toorn, "Improved model management with aggregated business process models," *Data & Knowledge Engineering*, vol. 68, no. 2, 2009, pp. 221–243

[4] J. Lee, K. Sanmugarasa, M. Blumenstein, Y.-C. Loo, "Improving the reliability of a Bridge Management System (BMS) using an ANN-based Backward Prediction Model (*BP*M)," *Automation in Construction*, vol. 17, no. 6, 2008, pp. 758–772

[5] L. Xu, L. Chen, T. Chen, Y. Gao, "SOA-based precision irrigation decision support system," *Mathematical and Computer Modelling*, vol. 54, no. 3–4, 2011, pp. 944–949

[6] S. Inês, D. D. Pádua, R. Y. Inamasu, "Assessment Method of Business Process Model of EKD," vol. 15, no. 3, 2008

[7] D. Greenwood, R. Ghizzioli, "Goal-Oriented Autonomic Business Process Modelling and Execution," in: S. Ahmed and M. N. Karsiti (eds.), *Multiagent Systems*, 2009, p. 18

[8] S. Narayanan, V. Jayaraman, Y. Luo, J. M. Swaminathan, "The antecedents of process integration in business process outsourcing and its effect on firm performance," *Journal of Operations Management*, vol. 29, no. 1–2, 2011, pp. 3–16

[9] H. H. Chang, I. C. Wang, "Enterprise Information Portals in support of business process, design teams and collaborative commerce performance," *International Journal of Information Management*, vol. 31, no. 2, 2011, pp. 171–182

[10] S. Smirnov, M. Weidlich, J. Mendling, M. Weske, "Action patterns in business process model repositories," *Computers in Industry*, vol. 63, no. 2, 2012, pp. 98–111

[11] A. Koschmider, T. Hornung, A. Oberweis, "Recommendation-based editor for business process modeling," *Data & Knowledge Engineering*, vol. 70, no. 6, 2011, pp. 483–503

[12] R. Dijkman, M. Dumas, B. van Dongen, R. Käärik, J. Mendling, "Similarity of business process models: Metrics and evaluation," *Information Systems*, vol. 36, no. 2, 2011, pp. 498–516

[13] H. a. Reijers, T. Freytag, J. Mendling, A. Eckleder, "Syntax highlighting in business process models," *Decision Support Systems*, vol. 51, no. 3, 2011, pp. 339–349

[14] Z. Huang, J. Huai, X. Liu, J. Zhu, "Business Process Decomposition Based on Service Relevance Mining," *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2010, pp. 573–580

[15] D. Rosso-Pelayo, R. Trejo-Ramirez, M. Gonzalez-Mendoza, N. Hernandez-Gress, "Business Process Mining and Rules Detection for Unstructured Information," *2010 Ninth Mexican International Conference on Artificial Intelligence*, 2010, pp. 81–85

[16] D. R. Ferreira, "Applied Sequence Clustering Techniques for Process Mining," *Science*, April, 2009, pp. 492–513

[17] M. Qiao, R. Akkiraju, A. J. Rembert, "Towards Efficient Business Process Clustering and Retrieval: Combining Language Modeling and Structure Matching," *Lecture Notes in Computer Science*, vol. 6896, 2011, pp 199–214

[18] C. Diamantini, D. Potena, E. Storti, "Clustering of Process Schemas by Graph Mining Techniques (Extended Abstract)," *SEBD 2011*, 2011, p. 49

[19] D. Ferreira, M. Zacarias, M. Malheiros, P. Ferreira, "Approaching Process Mining with Sequence Clustering: Experiments and Findings," *Lecture Notes in Computer Science*, vol. 4714, 2007, pp 360–374

[20] J.-Y. Jung, J. Bae, L. Liu, "Hierarchical clustering of business process models," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 12, 2009, pp. 613–616

[21] J. Melcher, D. Seese, "Visualization and Clustering of Business Process Collections Based on Process Metric Values," *10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 2008, p. 572–575

[22] Z. Yan, R. Dijkman, P. Grefen, "Business process model repositories – Framework and survey," *Information and Software Technology*, vol. 54, no. 4, 2012, pp. 380–395

[23] H.-L. Luo, H. Wei, F.-X. Hu, "Improvements in image categorization using codebook ensembles," *Image and Vision Computing*, vol. 29, no. 11, 2011, pp. 759–773

[24] Y.-C. Hu, B.-H. Su, C.-C. Tsou, "Fast VQ codebook search algorithm for grayscale image coding," *Image and Vision Computing*, vol. 26, no. 5, 2008, pp. 657–666

[25] M. Wu, X. Peng, "Spatio-temporal context for codebook-based dynamic background subtraction," *AEU - International Journal of Electronics and Communications*, vol. 64, no. 8, 2010, pp. 739–747

[26] M. E. Fonteyn, M. Vettese, D. R. Lancaster, S. Bauer-Wu, "Developing a codebook to guide content analysis of expressive writing transcripts," *Applied nursing research: ANR*, vol. 21, no. 3, 2008, pp. 165–168

[27] C. D. Manning, P. Raghavan, H. Schütze, *An Introduction to Information Retrieval*, 2008, p. 428

[28] G. Bordogna, A. Campi, G. Psaila, S. Ronchi, "Disambiguated query suggestions and personalized content-similarity and novelty ranking of clustered results to optimize web searches," *Information Processing & Management*, vol. 48, no. 3, 2012, pp. 419–437

[29] R. Baeza-Yates, B. Ribeiro-Neto, *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999, p. 513

[30] F. Aiolli, A. Burattin, A. Sperduti, *A Metric for Clustering Business Processes Based on Alpha Algorithm Relations*, Technical report, 2011

[31] U. Küster, B. König-Ries, "On the Empirical Evaluation of Semantic Web Service Approaches: Towards Common SWS Test," *2008 IEEE International Conference on Semantic Computing*, 2008

[32] D. A. Buell, D. H. Kraft, "Performance measurement in a fuzzy retrieval environment," *SIGIR Forum*, pp. 56–62, 1981

[33] M. Becker, R. Laue, "A comparative survey of business process similarity measures," *Computers in Industry*, vol. 63, no. 2, 2012. pp. 148–167

Business Processes Retrieval Based on Multimodal Search and Lingo Clustering Algorithm

H. Ordoñez, J. C. Corrales and C. Cobos

***Abstract*— This paper presents a model for searching and grouping of business process models. To search business process models, the model contains a module for multimodal indexing that takes into account textual and structural information of models. To group models, an adaptation of the Lingo algorithm was used; it is based on singular value decomposition and frequent phrases extracted from the description (textual and structural information) of the business process models retrieved according to a query. The evaluation of model was conducted by executing the search process on a closed test-collection built collaboratively, that containing 146 business process models, and comparing the results with a set of relevant business process models obtained from an evaluation issued by 59 experts. Measures of relevance such as graded precision and recall shown promising results in the search process, as well as the quality assessment of the sets formed by the grouping process.**

***Keywords*— Business processes, repository, multimodal search, clustering, evaluation.**

I. INTRODUCCIÓN

LA REUTILIZACIÓN de componentes como servicios Web y procesos de negocio facilita a las empresas el despliegue de nuevos servicios para garantizar la permanencia de sus clientes, y la creación de servicios de valor agregado que induzcan diferenciación competitiva. Esto exige que las tareas dentro de una empresa sean organizadas en torno a funciones del negocio tales como: mercadeo, ventas, producción, finanzas y servicio al cliente, las cuales generalmente se ejecutan de manera independiente [1].

Por otro lado, las funciones organizacionales se pueden describir a través de modelos de procesos de negocio (PN), los cuales representan procedimientos o actividades que colectivamente alcanzan un objetivo de negocio en común, definiendo roles y relaciones funcionales. Es decir, la información de las instrucciones del trabajo a realizarse, quién debe realizarlo y descripción de las conexiones con otros sistemas [2]. Para esto, los PN están descritos o representados a través de lenguajes de definición de PN como por ejemplo BPEL, BPMN, XPDL, YAWL, entre otros.

En la actualidad las empresas cuentan con repositorios que normalmente almacenan cientos o incluso miles de PN que pueden ser reutilizables y expandibles con nuevas funcionalidades de acuerdo a los requerimientos específicos de la empresa. En consecuencia encontrar manualmente un PN que se asemeje a dichos requerimientos es una tarea compleja que demanda tiempo y esfuerzo considerable [3].

En éste contexto, diferentes autores han desarrollado y modelado mecanismos para buscar y recuperar PN que puedan servir como punto de partida para la creación de nuevos modelos de PN [4]. Estos mecanismos parten de una consulta expresada como un PN completo o una fracción del mismo, y posteriormente encuentran un conjunto de PN similares dentro del repositorio.

A pesar de los resultados obtenido por dichos mecanismos los cuales denotan niveles aceptables de relevancia, el presente trabajo propone un modelo computacional para búsqueda y agrupación de PN, con el fin aumentar estos niveles aplicando un enfoque nuevo de indexación y búsqueda multimodal (integrado información textual y estructural), la cual es una técnica muy difundida en el campo de la recuperación de la información multimedia [5].

Adicionalmente, el modelo propuesto está centrado en el usuario garantizando una presentación visual clara y categorizada de los resultados obtenidos en cada consulta [6]. Para lograr esto, se usan técnicas de agrupamiento por afinidad o clustering, logrando que el usuario en sus consultas gaste menos tiempo y revise de forma organizada un mayor número de resultados [7].

Para el proceso de evaluación del modelo propuesto se desarrolló una herramienta software denominada"MultiSearchBP" presentada en [8], la cual permite la interacción con usuarios en el proceso de consulta de PN. La evaluación se realizó en dos fases, a) evaluación relevancia en la lista de resultados, utilizando una colección cerrada de modelos de PN, b) evaluación de la agrupación de resultados aplicando métricas de evaluación interna y medidas de evaluación externa. Los resultados obtenidos en el proceso de evaluación son promisorios y permiten verificar la eficacia del modelo propuesto.

El documento está organizado de la siguiente manera. La sección 2 el escenario de motivación, la sesión 3 presenta los trabajos relacionados., la sesión 4 describe el entorno propuesto, los componentes y sus algoritmos. La sección 5 expone los resultados de la evaluación del modelo propuesto. Finalmente, se presentan las conclusiones y el trabajo futuro que el grupo de investigación espera desarrollar en el corto plazo.

II. ESCENARIO DE MOTIVACIÓN

Las organizaciones a menudo cuentan grandes repositorios

H. Ordoñez, Universidad de San Buenaventura, Cali, Colombia, haordonez@usbcali.edu.co

J. C. Corrales, Universidad del Cauca, Popayán, Colombia, jcorral@unicauca.edu.co

C. Cobos, Universidad del Cauca, Popayán, Colombia, ccobos@unicauca.edu.co

de modelos de procesos de negocio (PN). Un PN puede ayudar a la gente a entender actividades empresariales complejas con facilidad, mediante una descripción abstracta del negocio [9]. Un desafío particular en este contexto es el mantenimiento del repositorio y la gestión de los PN existentes en este. Esto es debido a que las organizaciones por su capacidad producen líneas de productos o servicios basados en conjuntos de características dentro de una familia de productos dada. Como alternativa, a este desafío se han modelado mecanismos para búsqueda y detección automática de las diferentes versiones de PN existentes en el repositorio, que puedan explicar el comportamiento de la empresa [10]. A pesar que los resultados de los mecanismos propuestos tienen relevancia, estos solo proporcionan una lista con PN independientes que describen de manera específica algunas de las actividades desarrolladas en la organización. Como complemento, a estos mecanismos se han incorporado algoritmos de agrupación o clustering, con el propósito de juntar en un mismo grupo un conjunto coherente de PN, que pueden compartir características comunes determinadas por el flujo de control, estructura, finalidad, función del proceso o producto que representan. De esta manera cada grupo formado puede ser utilizado para generar un modelo de proceso más comprensible [11]. Además, permite a los ingenieros (modeladores de PN) explorar organizadamente los resultados agrupados y, así poder plantear posibles sugerencias sobre cómo rediseñar los PN, a fin de incorporar los cambios más frecuentes y significativos de una vez para todos los elementos de cada grupo.

III. Trabajos relacionados

El presente artículo está enfocado en la búsqueda y agrupación (clustering) de modelos de PN, por lo tanto a continuación se presenta un resumen de los trabajos más destacados en éstas dos temáticas

A. Propuestas basadas en búsqueda de modelos de PN

Estas propuestas se enfocan en un conjunto de elementos o tipos de datos presentes en los PN, por ejemplo, las basadas en lingüística se enfocan en el nombre o la descripción de las actividades, eventos y compuertas lógicas existentes en los PN. En el proceso de búsqueda utilizan técnicas de recuperación de información como la representación espacio-vectorial con un valor (TF) de frecuencia de términos por PN y distancia de cosenos para armar un ranking de resultados relevantes [12,13].

Las propuestas basadas en reglas de asociación están enfocadas en ejecuciones previas de los PN, las cuales son registradas en archivos Log. En el proceso de búsqueda se detectan frases relacionadas con las actividades del proceso negocio por medio de una ontología de dominio, adicionalmente se identifican patrones de actividades. Para definir la lista de resultados utilizan un componente heurístico que determina la frecuencia de aparición de los patrones detectados [14,15].

Las propuestas basadas en algoritmos genéticos transforman los PN a una representación formal (por ejemplo grafos o máquinas de estado) e integran otros datos adicionales para la búsqueda, datos como: número de entradas y salidas por nodo, etiquetas de aristas, nombre o descripción de los nodos. Aunque estos datos aportan mayor precisión a las consultas, el tiempo de ejecución es lento [16,17].

Finalmente se encuentran propuestas centradas en la búsqueda sobre repositorios de PN almacenados con anotaciones en archivos XML. Para el proceso de búsqueda, algunas propuestas utilizan un lenguaje de consulta, denominado IPM Process Query Language (IPM-PQL), el cual soporta consultas específicas tal como la búsqueda de procesos que contienen una determinada actividad, transición o conexión entre actividades [6, 18, 19].

B. Propuestas basadas en agrupación (Clustering) de PN

En este enfoque se encuentran propuestas que utilizan algoritmos de clustering jerárquico [7,20], los cuales construyen una jerarquía de grupos con base en la similitud de las características estructurales y de comportamiento de los PN. En estas propuestas los usuarios revisan la jerarquía y seleccionan el grupo que tiene mayor similitud con sus requerimiento de información [21,22].

El clustering secuencial de procesos de negocio, es otra opción, en ellos se toma como datos de entrada archivos Logs de ejecuciones previas de los PN. En estas propuestas el algoritmo agrupa PN con el mismo tipo de comportamiento basado en el flujo de ejecución y el flujo de datos en un mismo grupo [23,24].

C. Modelo propuesto Vs Trabajos anteriores

Las propuestas basadas en búsqueda de PN se limitan al emparejamiento de entradas y/o salidas, para lo cual toman como base la información textual de los elementos pertenecientes a cada PN. El proceso de búsqueda en estas propuestas deja de lado el flujo de ejecución del PN, así como el comportamiento, estructura, tipo de actividades, tipo de compuertas, tipo de eventos.

Por otro lado en las propuestas basadas en el proceso de agrupación utilizan los datos textuales de cada PN. Entre estos datos están: nombre de las actividades, tiempo de duración de cada actividad, número de errores. Además la agrupación elimina secuencias de actividades que ocurren una sola vez, sin tener en cuenta que estas secuencias pueden compartir información de tipo estructural o textual, la cual puede ser relevante en el momento de la selección de los modelos que forman cada grupo.

A pesar de los aportes ya realizados por los enfoques basados en clustering anteriormente nombrados, los resultados pueden ser ampliados abarcando un número mayor de características de información como: descripción de las actividades, tipo de tareas, tipo de compuertas, estructura, comportamiento, entre otras presentes en los PN. Centrarse en un solo tópico, por ejemplo el textual, solo permite realizar agrupación de PN mediante la comparación de información correspondiente a los nombres y la descripción de cada uno de los elementos pertenecientes a cada PN. En los grupos formados sobre este tópico se dejan de lado PN que pueden tener similitud en su estructura, tipo de tareas o comportamiento.

A partir del análisis anterior, en éste artículo se propone un modelo de búsqueda y agrupación de PN, que unifica en un solo espacio de búsqueda unidades estructurales de comportamiento y características textuales existentes en los PN, en lo que se conoce como una representación de búsqueda multimodal. Adicionalmente, se integra el uso de algoritmos de clustering que utilizan varios tipos de información, tales como: textual, estructural y de comportamiento para agrupar los resultados de la búsqueda. La agrupación es realizada con base en la similitud de los tipos de información contenida en cada uno de PN recuperados, para lograr así una forma más efectiva en el despliegue de los resultados.

IV. MODELO PROPUESTO

El modelo propuesto, define una arquitectura en 2 niveles como se presenta en la Fig. 1; El primero es el *nivel de gestión* el cual se encarga del pre-procesamiento de los PN, es decir, extraer sus características estructurales y los componentes textuales e indexarlos, para dar respuesta a las consultas de PN; además permite gestionar los PN (adicionar, eliminar, modificar PN del repositorio y su respectiva indexación). El segundo *nivel de búsqueda* permite buscar PN a través de tres opciones de consulta: por palabras clave, por estructura, y búsqueda multimodal (combinación de los dos anteriores); además permite el despliegue de resultados en dos formas: Listas ordenada de resultados y lista categorizada de grupos formados por afinidad en los resultados. Cada uno de estos niveles y sus diferentes bloques o módulos se describen a continuación:

A. Nivel de gestión

Este nivel provee las reglas de negocio necesarias para gestionar, pre-procesar los modelos de PN antes de ser almacenados en el repositorio y, además los módulos para su indexacion. Este nivel incluye el procesamiento textual, la generación del índice de búsqueda, A continuación se describen los módulos encargados de ejecutar las reglas mencionadas.

Gestion PN: este componente es utilizado por la interfaz grafica de usuario (GUI) de Gestión, que permite interactuar con el módulo de Gestión de PN, para adicionar, modificar y eliminar los modelos de PN del repositorio y del índice. Los PN del repositorio se encuentran almacenados en sus formatos originales XML, y representan los modelos a traves de la notacion BPMN (Business Process Modeling Notation) usando sintaxis XPDL (XML Process Definition Language).

Pre.procesamiento*:* este módulo genera un índice de búsqueda constituido por dos matrices: una matriz *MC* de características textuales y una matriz *MCd* de componentes estructurales. Este se compone por los siguientes módulos:

***Analisizador Textual*:** En este módulo, una matriz denominada *MC* es generada por la extracción de las etiquetas (por ejemplo, nombres, tipos y descripciones) de las actividades de los PN almacenados en el repositorio. Estas etiquetas o características textuales son entonces procesados previamente aplicando (algoritmo de Porter [12]) con el fin de crear vectores Ct_i que constituyen la *MC*. De esta manera, *MC* se compone de vectores $Ct_i = \{Ct_{i,1}, CT_{l,j}, CT_{l,m}\}$, para cada PNi almacenado en el repositorio. Cada componente $C_{i,j}$ representa el peso de una característica *j* en un PNI (m es el número total de características textuales de PN).

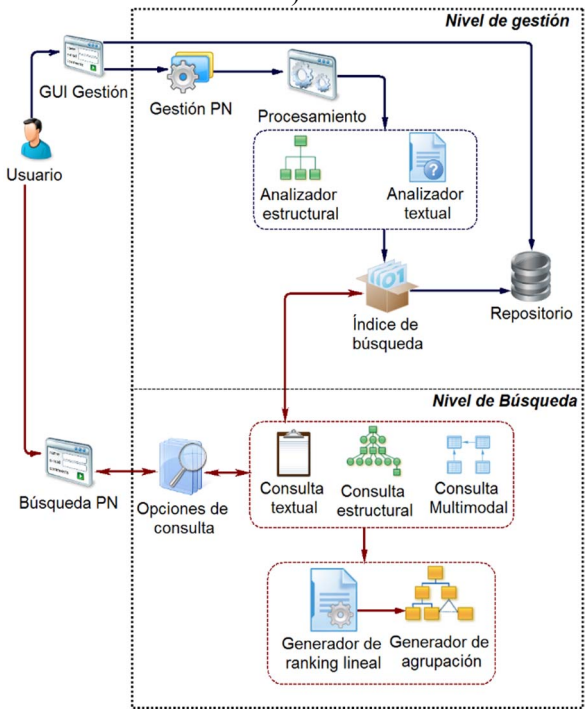


Figura 1. Arquitectura del modelo propuesto

Analizador estructural: Este módulo crea una matriz *MCd* compuesta de codebooks formados por la unión de n componentes estructurales basados en la secuencia del flujo de control de cada PN_i almacenado en el repositorio. Estos componentes describen las transiciones de dos o más nodos de los PN en forma de cadenas de texto que representan patrones secuenciales frecuentes en la estructura del PN_i. Los codebooks son la base para crear vectores $Cd_i = \{Cd_{i,1}, Cd_{i,k}, Cd_{i;p}\}$ donde Cd_{ik} es el peso de un elemento específico de PN_I en comparación con todos los elementos de PN en el repositorio, p es el número total de posibles transiciones de BP_I, y *k* es un índice para codebook específico almacenado en el repositorio. Por último, cada vector Cd_i es una fila *i* de la matriz de *MCd* del componente de codebook.

***Índice de búsqueda*:** este módulo crea un índice compuesto de dos espacios de búsqueda: 1) indexación textual de las funciones de negocio, y 2) indexación estructural (caracterización entre los tipos de: tareas, eventos, y enlaces). Estos dos espacios de búsqueda se unifican en una estructura multimodal para crear un índice más amplio que permite tener una representación precisa de los PN del repositorio con respecto a su conjunto de categorías. Índice de búsqueda almacena de manera eficiente una estructura conceptual llamada matriz índice $MI_{i,j} = \{MCd_{i,j} \bigcup MC_{i,j}\}$ (similar al modelo vectorial de recuperación de información [25]). Esta matriz contiene en cada celda un peso ($w_{i,j}$) que refleja la importancia de cada componente textual/estructural en su raíz léxica o codebook para cada PN.

La matriz *MI* se crea con base en la ecuación (1) propuesta

por Salton [25], donde $F_{i,j}$ es la frecuencia observada del componente textual o del codebook j en el PN_i. Max(F_i) es la mayor frecuencia observada en el PN_i, N es el número de PN en la colección (repositorio), y n_j es el número de PN en los que aparece el componente textual o codebook j.

$$w_{i,j} = \frac{F_{i,j}}{\max(F_i)} \times \log\left(\frac{N}{n_j + 1}\right) \quad (1)$$

***Repositorio**:* es la unidad central de almacenamiento de los archivos físicos de los modelos de PN. Es similar a una base de datos y cuenta con 146 modelos de PN incluyendo tareas, sub-procesos y flujos de control para cada PN.

B. Nivel de búsqueda

Este nivel permite la integracion de componentes visuales que proveen la interacion del usuario con las opciones de busqueda soportadas por el modelo propuesto. Este módulo está compouesto por los siguientes submódulos

Opciones de consulta: este componente es utilizado por la GUI de búsqueda (Implementada en la herramienta): hace referencia a un interfaz gráfica en la cual el usuario puede realizar consultas de tres formas diferentes: 1) textual (por palabras clave), 2) estructural (por codebooks presentes en los PN), y 3) multimodal (textual y estructural al tiempo).

- *Consulta textual*: en esta opción el usuario puede digitar una o varias palabras clave representadas en lenguaje natural, las cuales forman un vector de consulta $qpc=\{pc_1, pc_2, \ldots. pc_k\}$. El sistema pre-procesa las palabras clave, genera un vector de consulta con los términos y luego compara esta consulta con el componente textual del índice (MI) con el fin de entregar aquellos PN con mayor similitud respecto a la consulta planteada por el usuario.
- *Consulta estructural*: en esta opción el usuario tiene la posibilidad de elegir uno o varios (codebook) de una lista de componentes estructurales formados a partir de la colección de PN existentes en el repositorio, con ello se crea un vector de consulta $qcd=\{cd_1,cd_2,\ldots cd_n\}$. Los elementos utilizados en la consulta son comparados con el componente del índice que contiene los elementos estructurales con el propósito de retornar los modelos de PN con mayor similitud.
- *Consulta Multimodal:* esta opción integra las dos opciones de consulta anteriores. En ésta, el usuario escribe palabras de consulta y selecciona coodebooks, en este sentido el sistema crea un vector de consulta $qmg = \{qpc\ U\ qcd\}$, el cual es comparado con cada uno de los modelos representados en la matriz *MI* teniendo en cuenta los dos espacios de búsqueda (información textual e información estructural).

Finalmente, el proceso de comparación del vector de consulta con los PN registrados en el índice ejecuta un mecanismo de correspondencia y refinamiento de resultados, es decir la comparación de los datos de la consulta con los modelos de PN almacenados en el repositorio. Para esto, los datos introducidos en estas opciones de consulta son representadas en un vector de términos $q= \{t_1,t_2,t_3,\ldots t_n\}$ al cual se le aplica el pre-procesamiento de texto (el mismo que se aplica en la etapa de indexación). Una vez se obtiene la cadena de consulta procesada, se ejecuta la búsqueda en el espacio elegido por el usuario a través de la ecuación (2).

***Generador de ranking lineal**:* este módulo genera una clasificación de acuerdo con una clasificación conceptual (puntuación) expresada por ecuación (2) (usado por la librería Lucene) [15]. Entonces se aplica la ecuación (2) con el fin de devolver una lista de clasificación de resultados con un conjunto de BP con mayor puntuación conceptual dentro del índice multimodal.

$$Puntuacion\ (q,d) = coord\ (q,d) * \sum_{t \in q} (tf\ (t \in d) + idf\ (t)^2 * norm\ (t,d)) \quad (2)$$

La clasificación del ranking de resultados es realizada en forma decreciente (mayor a menor), con base en el nivel de puntuación alcanzada por los PN al efectuar la ejecución de la consulta. Para esto en la ecuación (2), se tienen las siguientes consideraciones:

- t es un término de la consulta q.
- d es el PN consultado.
- $tf\ (t \in d)$ es la frecuencia de aparición (número de veces) del término t en el $PN\ d$ especifico.
- *idf(t)* define la importancia relativa del término t en el repositorio de PN como un todo, expresada como la frecuencia de aparición del término t en la colección de PN (expresa numéricamente cuán relevante es el termino t para los PN de la colección).
- *Coord* (q, d) es un factor de puntuación que está determinado por el número de términos pertenecientes a la consulta y que además existen en el *PN* consultado.
- *norm(t,d)* es el factor de ponderación en la indexación, tomado de w_{ij} matriz *MI*.

Una vez realizados los cálculos de ponderación, los resultados son ordenados filtrados y listados en orden descendente de acuerdo con la similitud que presentan respecto a la consulta ingresada por el usuario. En este punto el usuario tiene además la posibilidad de elegir, visualizar y analizar completamente cada uno de los *PN* recuperados.

***Generador de agrupación**:* En este nivel se agrupan los resultados obtenidos en la ejecución de cada una de las opciones de consulta soportadas por el modelo propuesto. El agrupamiento se realiza por afinidad aplicando dos técnicas de clustering [21]: STC (Sufix Tree Clustering) y LINGO. De esta manera los resultados son organizados en grupos o familias de PN que contienen correlación en características texturales, estructurales o una combinación de las dos. A continuación se presentan los algoritmos de agrupamiento conforme fueron adaptados.

***STC**:* Toma cada PN como una secuencia sintáctica ordenada de términos textuales o estructurales para generar el agrupamiento. Este algoritmo consta de dos pasos principales.

Paso 1: identificar grupos base, en este paso el algoritmo crea un árbol de sufijos a partir del vector que contiene todos los componentes textuales y estructurales de los PN. A partir de este vector detecta una raíz de tal manera que se garantiza que cada nodo contiene al menos dos hijos internos (un par). Luego, las aristas entre nodos se etiquetan con una parte del texto resumen con el propósito de formar la etiqueta de dicho nodo.

Paso 2: combinar grupos base. En este paso el algoritmo asigna una clasificación a cada grupo base, teniendo en cuenta el número de PN que el grupo contiene y que están relacionados con una serie de elementos textuales o estructurales P. Para esto se usa una función de grupo base (s(B)), en la cual está contemplado un grupo B con elementos P así: $s(B) = |B| * f(|P|)$, donde |B| es el número de PN en el grupo base B; |P| es el número de elementos en P que no tienen calificación 0 (es decir que estén conectados a al menos algún nodo del árbol de sufijos); f es una función que penaliza los P de un solo elemento.

***LINGO**:* En este algoritmo se construye un resumen (Snippet) con los términos textuales y estructurales contenidos en cada PN retornado por una consulta. El algoritmo consta de cuatro pasos principales.

Paso 1: la extracción de características. En este paso se identifican frases o términos frecuentes que pueden ser candidatos para etiquetas de grupo. Esto se realiza calculando el número de veces que los términos o frases identificadas aparecen en los PN contenidos en el repositorio.

Paso 2: inducción de etiquetas de grupo. Este paso forma descripciones significativas de los grupos tomando la información de la matriz de términos por cada modelo de PN.

Paso 3: descubrimiento de contenido de cada grupo. En este paso se comparan fragmentos de los PN con todas y cada una de las etiquetas de grupo, haciendo uso de SVD (Singular Value Decomposition). Para esto se forma una matriz Q en la que cada etiqueta de grupo se representa como un vector columna. De forma que $C=Q^TA$, donde A es el termino original de la matriz de termino PN. De esta manera, el elemento $c_{i,j}$ de la matriz C indica el peso de adhesión del PN_j en el grupo i.

Paso 4: la formación final de grupos, calcula la ponderación de la etiqueta dependiendo del número de veces que los términos de la etiqueta aparecen en cada uno de los PN asignados al grupo identificado por dicha etiqueta. Al igual que en el algoritmo STC, el modelo propuesto incrementa el rendimiento ya que realiza la primera fase de filtrado de texto en el proceso de indexación.

V. EVALUACIÓN DEL MODELO PROPUESTO

Para determinar la calidad de los resultados generados por el modelo propuesto en cada opción de consulta fue necesario someterlo a un proceso de evaluación experimental. El proceso de evaluación contemplo dos fases, a) fase de evaluación de relevancia en las lista de resultados y b) fase de evaluación de la agrupación de PN.

A. Evaluación de relevancia en lista de resultados

En esta fase, los resultados obtenidos por el modelo propuesto (MultiSearchBP) fueron comparados con los resultados de la evaluación manual realizada en una colección cerrada de prueba, la cual se presenta en [26]. Esta colección cerrada de prueba fue creada con una estrategia colaborativa que contó con 59 evaluadores expertos en gestión de PN. Los perfiles de los evaluadores se pueden observar en la Tabla I.

TABLA I. PERFILES DE JUECES EVALUADORES

	DR.	MSC.	PROFESSIONAL
Instituto de Informática/UFRGS	-	7	14
Escuela de administración /UFRGS	-	-	33
Universidad del Cauca	2	3	-

Adicionalmente a la comparación con la evaluación manual, los resultados entregados por el modelo propuesto fueron comparados con los resultados entregados por una implementación del algoritmo A* [27], ampliamente utilizado en búsqueda y descubrimiento de PN. En este sentido, es posible evaluar la calidad de los resultados obtenidos en la ejecución de consultas a partir de la aplicación de medidas estadísticas ampliamente utilizadas en evaluación de sistemas de recuperación de información [28]. Estas medidas son las medidas de efectividad gradadas: Precisión gradada (Pg), Recall gradado (Rg) y Medida F gradada.

La Fig. 2 presenta el nivel de precisión gradada del modelo propuesto en la búsqueda de PN. Los resultados de evaluación de la *Pg* en el tipo de consulta multimodal alcanzaron el 94%, lo que demuestra que este tipo de consultas (textual y estructural al mismo tiempo) son mucho más precisas que las opciones de consulta por separado. En comparación con los resultados en la consulta de A*, MultiSearchBP es 30% más precisó.

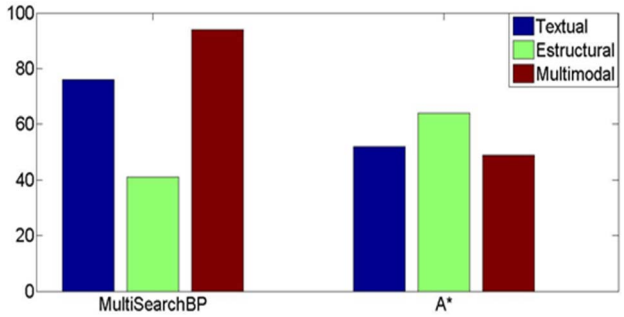


Figura 2. Grafica de precisión gradada

De la Fig. 3 se puede concluir que MultiSearchBP y A*, obtuvieron valores bajos de Recall gradado. A* obtuvo 26% y 33% para la búsqueda Multimodal. Esto se debe a que los dos métodos, generan rankings limitados a un máximo de diez resultados, dejando por fuera otros PN que pueden ser relevantes a la consulta. También se puede observar que en esta medida, MultiSearchBP supera a A* en un 11% en su consulta multimodal.

La Medida F gradada que se obtuvo para los dos métodos se presenta en la Fig. 4; Esta medida representa la armonía de los

resultados de Pg y Rg. En promedio MultiSearchBP logra una mejora del 30% sobre A*.

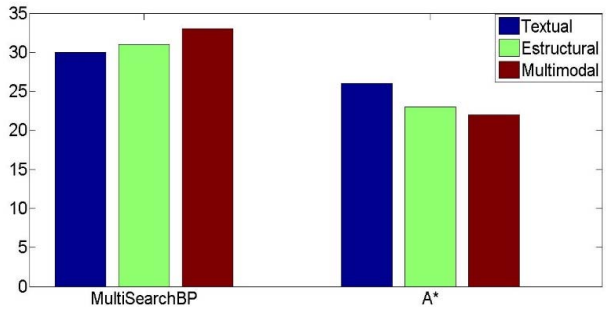


Figura 3. Grafica de recall gradada

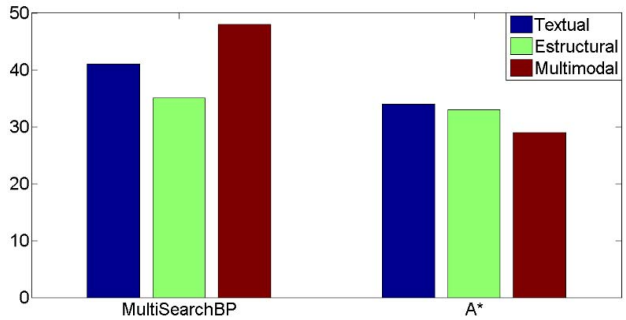


Figura 4. Grafica Medida F

B. *Evaluación de la formación de grupos*

Medir el rendimiento de una agrupación o clustering no es una tarea trivial, y a pesar de que existen varias propuestas, no se ha desarrollado una metodología estándar. En este trabajo, la evaluación de las agrupaciones está basada en métricas de evaluación interna y medidas de evaluación externa [29].

La evaluación interna: mide densidad, distancia entre los objetos en los mismos grupos (si los grupos son más pequeñas se espera que sean más compactos) y separación entre los grupos (distancias más grandes indican mayor separación) [30]. Las medidas utilizadas para la evaluación interna fueron tres, cohesión, acoplamiento y silueta.

Cohesión: expresa el promedio de similitud entre los elementos de un grupo. Por lo tanto, a mayor similitud en los elementos de un mismo grupo, se tiene un grupo más cohesionado. La ecuación (3) muestra el cálculo de la cohesión, donde sim (c_i,c_j) es el grado de similitud entre los elementos c_i y c_j existentes en el grupo c, y m es el número de elementos existentes en el grupo.

$$cohesion(c) = \frac{\sum_{i>j} sim(c_i,c_j)}{\frac{m(m-1)}{2}} \quad (3)$$

Acoplamiento: es utilizada para medir la similitud media entre todos los pares de elementos, donde para cada par un elemento puede pertenecer al grupo *C* y el otro a un grupo externo a *C*. Idealmente, el acoplamiento debe ser bajo. La ecuación (4) muestra el cálculo del acoplamiento, donde *sim* (c_i, q_j) es la similitud entre el elemento c_i del grupo *C* y el elemento q_j de otro clúster; *m* es el número de elementos en *C* y *n* es el número de elementos externos a *C*.

$$Acoplamiento\ (c) = \frac{\sum_{i,j} sim(c_i,q_j)}{m*n} \quad (4)$$

Silueta: es una medida derivada de la cohesión y el acoplamiento; determina cuáles elementos están bien ubicados en el grupo y cuáles tienen una posición intermedia. Entre mayor grado de silueta mejor es la distribución de los grupos. La ecuación (5) representa el cálculo de la silueta, donde a_i es la disimilitud promedio (es decir, la distancia) entre el i^{th} elemento del grupo y los otros objetos del mismo agrupo, y b_i es la disimilitud promedio mínimo entre el i^{th} elemento de cualquier grupo que no contiene al elemento.

$$Silueta\ (i) = \frac{b_i - a_i}{\text{MAX}(a_i,b_i)} \quad (5)$$

En la Fig. 5 se observan los promedios obtenidos para cada una de las medidas aplicadas. En relación con la cohesión Lingo obtiene mayor valor que STC, esto demuestra que los grupos generados por Lingo están más cohesionados, es decir que los elementos existentes en cada grupo guardan mayor similitud o cercanía entre ellos. STC tiene un nivel más bajo de cohesión ya que permite solapamiento, es decir un modelo de PN puede pertenecer a varios grupos, debido a la combinación de los grupos base.

Respecto al acoplamiento, al igual que en la medida anterior Lingo obtiene mayor valor que STC, esto se debe a que los elementos considerados como similares entre los grupos formados comparten relativamente un subconjunto de características comunes, las cuales pueden ser textuales o estructurales.

Finalmente, para el coeficiente de silueta, al igual que en las medidas anteriores Lingo alcanza mejores resultados que STC, lo que permite identificar que los grupos formados por Lingo cuentan con elementos que están bien ubicados, generando así grupos mejor distribuidos. Por lo tanto el valor menor del coeficiente silueta para STC se debe, porque en los grupos formados por este algoritmo existen elementos intermedios, los cuales pertenecen al mismo tiempo a varios grupos.

Las medidas de evaluación interna permiten identificar que el entorno propuesto genera grupos más coherentes con el algoritmo de Lingo. Esto se debe a que el algoritmo de Lingo toma información de la matriz de términos por cada PN, lo que permite identificar con claridad qué los PN comparten mayor número de elementos textuales o estructurales, facilitando así su agrupación.

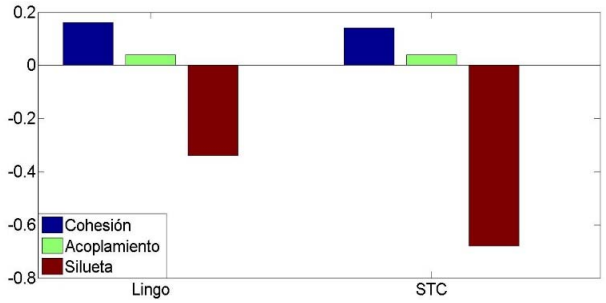


Figura 5. Medidas de evaluación interna

Evaluación externa: evalúa la calidad de una agrupación

mediante la comparación de los grupos producidos por una técnica de agrupación automática, contra los grupos generados manualmente en una etapa previa, realizada por usuarios expertos en la temática inmersa en los datos. Como métricas de evaluación externa se utilizaron las medidas de precisión ponderada, recall ponderada y medida F ponderada (mide la armonía entre precisión y recall), medidas clásicas en el campo de recuperación de información [25,29].

Para evaluar precisión ponderada, recall ponderada y medida F ponderada, se parte de la formación de grupos $\{C_1, C_2 \ldots\ldots C_k\}$ generada automáticamente por MultiSearchBP y se compara con la colección ideal de grupos $C_1^i, C_2^i, \ldots . C_h^i$ presentada en [31], la cual fue construida colaborativamente por 56 usuarios expertos. En la evaluación se ejecutaron los siguientes pasos: (a) encontrar para cada grupo ideal, C_n^i el grupo distinto C_m que más se aproxime en la colección que se está evaluado (grupos formados por MultiSearchBP) y calcular $P(C, C^i)$, definida en la ecuación (6), $R(C, C^i)$, definida en la ecuación (7), $F(C, C^i)$, definida en la ecuación (8). (b) Calcular la precisión ponderada a través de la ecuación (9), recall ponderada con la ecuación (10) y medida F ponderada utilizando la ecuación (11).

$$P(C, C^i) = \frac{|C \cap C^i|}{|C|} \quad (6)$$

$$\boldsymbol{R(C, C^i)} = \frac{|\boldsymbol{C \cap C^i}|}{|\boldsymbol{C^i}|} \quad (7)$$

$$F(C, C^i) = \frac{2P(C,C^i)R(C,C^i)}{P(C,C^i)+R(C,C^i)} \quad (8)$$

$$P = \frac{1}{T} \sum_{j=1}^{h} |C_j^i| \, P(C_m, C_j^i) \quad (9)$$

$$R = \frac{1}{T} \sum_{j=1}^{h} |C_j^i| \, R(C_m, C_j^i) \quad (10)$$

$$F = \frac{2PR}{P+R} \; ; \; T = \sum_{j=1}^{h} |C_j^i| \quad (11)$$

En la ecuación 9, C es un grupo de modelos de PN, C^i es un grupo ideal de modelos de PN.

En la Fig. 6 se presentan los promedios alcanzados en la evaluación externa por los algoritmos de clustering implementados en MultiSearBP. Con base a la precisión ponderada (P), Lingo obtiene el mejor valor 65% de similitud con la formación ideal generada por los expertos humanos debido al número elevado de elementos compartidos entre las dos agrupaciones, superando en 10% a STC. Esto se debe a que Lingo agrupa un porcentaje mayor de verdaderos positivos (VP) aquellos PN que fueron ubicados por el algoritmo en el mismo clúster que indicaba la formación ideal. Para racall ponderada (R) al igual que la medida anterior Lingo logra mejor valor 43%, superando a STC en 6%. Esto es debido a que STC en la agrupación omite PN que pueden ser relevantes para cada grupo evaluado, haciendo así que aumente el valor de falsos positivos (FN). PN que fueron ubicados por STC en el clúster j y que en realidad pertenecían a otro clúster de la formación ideal. En relación al rendimiento determinado por la Medida F ponderada (F) Lingo supera a STC en 9%. Esto permite definir que los grupos creados por Lingo son más relevantes y coinciden en un mayor grado con los grupos formados manualmente por los expertos en el ambiente colaborativo.

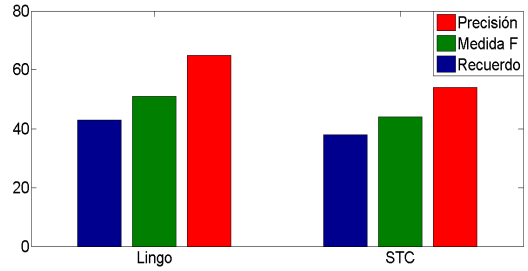


Figura 6. Promedios evaluación externa

VI. CONCLUSIONES Y TRABAJO A FUTURO

En este trabajo se presentó un modelo para la búsqueda y agrupación de PN, el cual implementa tres tipos de consulta, a saber: textual, estructural y multimodal. La flexibilidad del modelo permite adicionar algoritmos de clustering, los cuales pueden ser para agrupación de datos, basados en teoría de grafos, entre otros.

La utilización de información textual e información estructural en el índice multimodal, permite mayor flexibilidad y precisión en las consultas realizadas.

El modelo propuesto fue evaluado con una colección cerrada de PN, el cual fue desarrollado en forma colaborativa por un grupo de expertos humanos. El modelo alcanza niveles de Precisión gradada entre el 41% y el 94%.

La comparación del modelos propuesto con el algoritmo A*, demuestran que MultiSearchBP mejora en un 30% el nivel de relevancia en los resultados, en relación a la armonía de Pg y Rg determinada por la Medida-F gradada.

En el nivel de agrupación. La estructura de árbol formada con los algoritmos STC y LINGO, permite al usuario revisar las categorías y seleccionar el grupo de mayor similitud a su consulta.

En la evaluación interna de la agrupación, el algoritmo de Lingo obtiene mejores grupos, con relación a las medidas empleadas, a saber: cohesión, acoplamiento y silueta. Con base a la evaluación externa los grupos formados por Lingo tienen más similitud con la formación ideal de grupos realizada colaborativamente por los expertos, debido al mayor número de PN clasificados como VP. PN que fueron ubicados por Lingo en el mismo clúster que indicaba la formación ideal.

Como trabajo a futuro se propone implementar más algoritmos de clustering al modelo, entre estos están K-means, Cliques, Start, C-means. Evaluar la formación de grupos con otros resultados reportados en el estado del arte. De la misma forma, se sugiere incorporar ontologías de dominio específico con el propósito de realizar enriquecimiento semántico a los PN y las consultas. Por otro lado, se propone desarrollar un módulo de evaluación automática que genere gráficas y medidas de relevancia, en especial medidas externas. Finalmente, ampliar la evaluación aplicando nuevas medidas para la búsqueda de PN tal como las propuestas en [30].

AGRADECIMIENTOS

Los autores agradecen a la Universidad del Cauca y a la Universidad de San Buenaventura sede Cali por el apoyo dado al estudiante de Doctorado en Ingeniería Telemática Hugo Armando Ordóñez

REFERENCIAS

[1] H. a. Reijers, R. S. Mans, and R. a. van der Toorn, "Improved model management with aggregated business process models," *Data & Knowledge Engineering,* vol. 68, pp. 221-243, 2009.

[2] X. Zhao and C. Liu, "Version management for business process schema evolution," *Information Systems,* vol. 38, pp. 1046-1069, 2013.

[3] M. Kunze and M. Weske, "An Open Process Model Library," *Business Process Management Workshops, BPM 2011 International Workshops Clermont-Ferrand, France, August 29, 2011 Revised Selected Papers, Part II,* pp. 26-38, 2012.

[4] T. Schlegel, K. Vidačković, S. Dusch, and R. Seiger, "Management of interactive business processes in decentralized service infrastructures through event processing," *Journal of King Saud University - Computer and Information Sciences,* pp. 137-144, 2012.

[5] J. C. Caicedo, J. BenAbdallah, F. A. González, and O. Nasraoui, "Multimodal representation, indexing, automated annotation and retrieval of image collections via non-negative matrix factorization," *Neurocomputing,* vol. 76, pp. 50-60, 2012.

[6] M. Kunze, A. Meyer, and M. Weske, "A Platform for Research on Process Model Collections," 2012.

[7] I. S. Engineering and U. States, "Hierarchical clustering of business process models," *Computer,* vol. 5, pp. 613-616, 2009.

[8] H. Ordonez, J. C. Corrales, and C. Cobos, "MultiSearchBP-Entorno para busqueda y agrupacion de modelos de procesos de negocio," *Polibits,* vol. 49, 2014.

[9] W. Dai, D. Covvey, P. Alencar, and D. Cowan, "Lightweight query-based analysis of workflow process dependencies," *Journal of Systems and Software,* vol. 82, pp. 915-931, 2009.

[10] Z. Yan, R. Dijkman, and P. Grefen, "Fast business process similarity search," *Distributed and Parallel Databases,* vol. 30, pp. 105-144, 2012.

[11] I. Ognjanovic, B. Mohabbati, D. Gaevic, E. Bagheri, and M. Bokovic, "A Metaheuristic Approach for the Configuration of Business Process Families," pp. 25-32, 2012.

[12] A. Koschmider, T. Hornung, and A. Oberweis, "Recommendation-based editor for business process modeling," *Data & Knowledge Engineering,* vol. 70, pp. 483-503, 2011.

[13] H. a. Reijers, T. Freytag, J. Mendling, and A. Eckleder, "Syntax highlighting in business process models," *Decision Support Systems,* vol. 51, pp. 339-349, 2011.

[14] J. H. Zichen Huan, Xudong Liu, and Jiangjun Zhu, "Business Process Decomposition based on Service Relevance Mining," *IEE/WIC/ACM international Conference on Web Intelligence and Intelligent Agent Technology,* 2010.

[15] D. a. Rosso-Pelayo, R. a. Trejo-Ramirez, M. Gonzalez-Mendoza, and N. Hernandez-Gress, "Business Process Mining and Rules Detection for Unstructured Information," *2010 Ninth Mexican International Conference on Artificial Intelligence,* pp. 81-85, 2010.

[16] A. T. Chris J. Turner, Jorn Mehnen, "A Genetic Programming Approach to Business Process Mining," 2010.

[17] C. Li, M. Reichert, and A. Wombacher, "Mining business process variants: Challenges, scenarios, algorithms," *Data & Knowledge Engineering,* vol. 70, pp. 409-434, 2011.

[18] C. R. a. M. Kunze, "An Extensible Platform for Process Model Search and Evaluation," *Business Process Management* vol. Demos 2013: Beijing, China, 2013.

[19] Z. Yan, R. Dijkman, and P. Grefen, "Business process model repositories – Framework and survey," *Information and Software Technology,* vol. 54, pp. 380-395, 2012.

[20] F. Aiolli, A. Burattin, and A. Sperduti, "Metric for Clustering Business Processes Based on Alpha Algorithm Relations," *Business,* p. 17, 2011.

[21] C. Diamantini, D. Potena, and E. Storti, "Clustering of Process Schemas by Graph Mining Techniques (Extended Abstract)," *Methodology,* vol. 4, p. 7, 2011.

[22] J. Melcher, D. Seese, and I. Aifb, "Visualization and Clustering of Business Process Collections Based on Process Metric Values," *Measurement,* vol. 8, p. 9, 2008.

[23] D. R. Ferreira, "Applied Sequence Clustering Techniques for Process Mining," *Science,* pp. 492-513, 2009.

[24] D. Ferreira, M. Zacarias, M. Malheiros, and P. Ferreira, "Approaching Process Mining with Sequence Clustering: Experiments and Findings," *Engineering,* vol. 7, pp. 1-15, 2008.

[25] R. Christopher D. Manning, Prabhakar, Schütze, Hinrich, "An Introduction to Retrieval Information," p. 428, 2008.

[26] O. Hugo, J. C. Corrales, C. Cobos, L. Krug Wives, and L. Thom, "COLLABORATIVE EVALUATION TO BUILD CLOSED REPOSITORIES ON BUSINESS PROCESS MODELS," *Springer-iceis, Portugal,* 2014.

[27] H. B. B.T. Messmer, "A New Algorithm for Error-Tolerant Subgraph Isomorphism Detection," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 20 . pp. 493–504, 1998.

[28] C. D. Manning, P. Raghavan, and H. Schutze, "Introduction to Information Retrieval," 2008.

[29] M. M. Carlos Cobos, Elizabeth León, Milos Manic, and Enrique Herrera-Viedma, "TopicSearch—Personalized Web Clustering Engine Using Semantic Query Expansion, Memetic Algorithms and Intelligent Agents," *Polibits,* vol. 47, 3012.

[30] K. D. a. J. Szyma´nski, "External Validation Measures for Nested Clustering of Text Documents," *Springer-Verlag Berlin Heidelberg - Emerging Intelligent Technologies in Industry,,* 2011.

[31] H. Ordonez, J. C. Corrales, C. Cobos, and L. K. Wives, "Collaborative grouping of business process models," *EATIS 2014, Valparaiso -Chile,* pp. 1-2, 2014.

Ordoñez Hugo, Ph.D. (c) en Ingeniería Telemática, Universidad del Cauca, Profesor Titular Facultad de Ingeniería, Universidad de San Buenaventura- Cali, Colombia, Investigador Grupo de Ingeniería Telemática de la Universidad del Cauca, Áreas de investigación: Minera de Procesos de Negocio, Recuperación de Información, Clustering de Documento Web.

Corrales Juan Carlos, Ph.D. en Ciencias de la Computación, Profesor Titular del Departamento de Telemática, Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca, Colombia, Áreas de investigación: Composición Automática de Servicios Web, Descubrimiento de Procesos de Negocio.

Cobos Carlos, Ph.D**.** en Ingeniería de Sistemas y Computación, Profesor Titular del Departamento de Sistemas, Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca, Colombia. Áreas de investigación: Recuperación de Información, Minería de Datos, Optimización de sistemas complejos usando Meta-heurísticas y Educación en Línea.

Dynamic Reconfiguration of Composite Convergent Services Supported by Multimodal Search

Armando Ordóñez[1], Hugo Ordóñez[2,3(✉)], Cristhian Figueroa[3,4], Carlos Cobos[5], and Juan Carlos Corrales[3]

[1] Intelligent Management Systems, Fundación Universitaria de Popayán, Popayán, Colombia
armando.ordonez@docente.fup.edu.co
[2] Research Laboratory for Software Engineering, Universidad de San Buenaventura, Cali, Colombia
haordonez@usbcali.edu.co
[3] Telematics Engineering Group, Universidad Del Cauca, Popayán, Colombia
jcorral@unicauca.edu.co
[4] Software Engineering Group, Politecnico di Torino, Turin, Italy
cristhian.figueroa@polito.it
[5] Information Technology Research and Development Group, Universidad Del Cauca, Popayán, Colombia
ccobos@unicauca.edu.co

Abstract. Composite convergent services integrate a set of functionalities from Web and Telecommunication domains. Due to the big amount of available functionalities, automation of composition process is required in many fields. However, automated composition is not feasible in practice if reconfiguration mechanisms are not considered. This paper presents a novel approach for dynamic reconfiguration of convergent services that replaces malfunctioning regions of composite convergent services considering user preferences. In order to replace the regions of services, a multimodal search is performed. Our contributions are: a model for representing composite convergent services and a region-based algorithm for reconfiguring services supported by multimodal search.

Keywords: Convergent services · Dynamic reconfiguration · Multimodal search

1 Introduction

A composite convergent service (CCS) may be defined as a structured set of services (telecommunication and Web services) that works in a coordinated manner to achieve a common goal [1]. CCS achieve the integration and composition of services offered by IT providers with Telecom operators towards the Web Telecom convergence [2].

W. Abramowicz (Ed.): BIS 2015, LNBIP 208, pp. 1–13, 2015.
DOI: 10.1007/978-3-319-19027-3_11

One example of convergent process is a service that manages environmental early warnings. Environmental manager is in charge of decision making about environmental alarms and crops. In order to do so, it is required information from sensor networks, Telecommunication and Web services that process data and that send information to farmers. One typical requirement of such systems is: to emit an alarm to every farmer within a radius of 2 miles from the river if the river flow is greater than 15 % of average. For solving this request, the sensor data are evaluated and if necessary, an emergency map is generated. This map is created drawing a radius of 2 miles from the sensor. To do so, the system may use geographic services and maps from internet. Finally, the system informs about the alarm to farmers inside the emergency area, in this case the best way to send the information is selected (SMS or call). In both cases, services from Web and Telecommunications work in coordination.

Nowadays, composition of convergent services has been increasingly adopted in telecommunication companies in order to create and offer new CCS that integrate different functionalities of existing convergent services. It helps them to avoid developing new CCS from the scratch and duplication of functionalities. However, composing convergent services is not trivial and is time consuming due to the complexity of the integration of different and heterogeneous messages, operations, and data-types in these services [3]. Due to the latter the automation of this composition has been studied actively. Furthermore, the demanding requirements of the telecommunications environment regarding to performance, availability and accuracy, make it necessary that composed convergent services can replace services (nodes) that may be unavailable or malfunctioning during execution and that can impact negatively in the user experience [4].

Malfunctions or unavailability of convergent services may be originated from diverse factors such as network failures or server overload. In such scenarios, CCS must be fixed or reconfigured in order to continue providing their functionalities. Accordingly, reconfiguration and dynamic composition have been identified as leading challenges in Service Oriented Architectures [5].

Previous works in the literature have addressed the reconfiguration of CCS based mostly in replacement of individual failing services [6]. However, existing proposals leave aside Telecom considerations such as service deployment, real-time requirements or event-based interactions [5]. In addition, replacement of failing services for other ones with different non-functional features cannot maintain initial user constraints [7].

This work proposes a dynamic reconfiguration approach supported by a multimodal model for searching services or sets of services that can be used to reconfigure a CCS where there is a set of services (regions of services) that are malfunctioning or unavailable. The multimodal search is a branch of the information retrieval (IR) that aims for efficiently discovering different kinds of content [8].

The region-based reconfiguration is inspired by the work of Lin et al. [5] which describes a mechanism for selecting a replacement for a region surrounding the failing services instead of replacing the whole CCS. A failing region may

be defined as a subset of services from the CCS that needs to be replaced in order to continue the execution of the service without recreating all the CCS. Previously the region-based reconfiguration mechanism was tested in the AUTO platform using automated planning [9–11], in this paper we focus on the multimodal search instead of the automated planning. Multimodal search uses textual and structural information in order to cover more information dimensions for querying during the search [12], this makes that the retrieval of services to be more precise than traditional information retrieval techniques.

The rest of this paper is organized as follows. Section 2 depicts a general overview of AUTO. Section 3 presents the modeling of CCS. Section 4 shows the service adaptation and the multimodal search applied to reconfiguration process. Section 5 presents the experimentation. Finally, Sect. 6 includes the conclusions as well as the relevant related work.

2 The Approach for Dynamic Reconfiguration of Composite Convergent Services

Figure 1, shows the architecture the main modules of AUTO [11]. AUTO, defines sequential phases for service composition namely: creation, synthesis and execution.

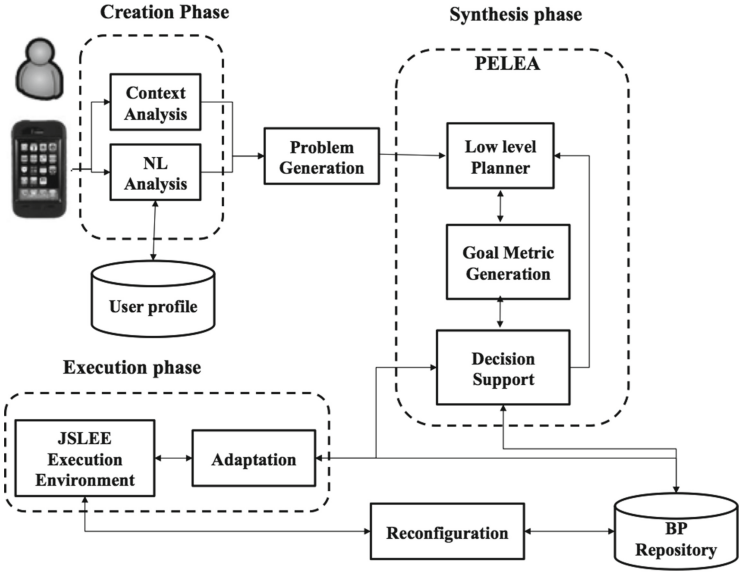


Fig. 1. Architecture of the approach for dynamic convergent service composition

The creation phase accepts user queries (requests) expressed either as voice or as text strings in natural language (NL). The NL Analysis module deals with each query in order to extract a set of significant words, i.e. to delete "stop words", reduce words to their lexical root etc. The Context Analysis module

enriches the user queries with information about the user preferences, device capabilities and situational context. The Problem Generation module translates the processed user query into a problem file expressed in the planning language PDDL (Planning Domain Definition Language) [13]. Thus, this problem file contains the goals (user objectives) and metrics (preferences of the user about the composition). Additionally, it can also contain information about the initial state such as location, device information, etc.

The synthesis phase, receives the problem file as input and generates the structure of the composite convergent service CCS. To do so, this phase uses automated planning. In automated planning the generated structures are known as plans. A plan contains a set of services that solves user queries, the obtained plan represents the CCS. This phase is framed in the Planning and Learning Architecture (PELEA) [14]. *PELEA* is an architecture that contains modules for plans generation, monitoring and reconfiguration. Specifically, the low level planner performs the plan generation, the Goal Metric Generator module associates the goals and metrics, obtained from the input problem, with services available for the composition. Finally the decision support performs the monitoring and reconfiguration of the process. The synthesis phase requires that the user receives the plan immediately, therefore the elapsed time spent building the plans must be minimized. Nevertheless, in order to get the plan that represents the best solution, a big computational effort is necessary. On the average case finding the optimal solution requires exploring a very significant part of the search space, which makes such an endeavor impractical. Therefore, the best solution given a time window may be acceptable to begin the execution, and afterwards the planner may refine the plan while the first services are invoked and the associated tasks are performed in the real world. These new plans are potential replacements for the initial CCS, so these CCS are stored in the *BP repository*, which also contains simple convergent services (i.e. those that are not composed with others into a CCS).

The execution phase is mainly performed in the execution environment for telecommunication applications named Java Service Logic Execution Environment (JSLEE). The Adaptation module integrates *PELEA* and JSLEE in order to take the CCS (synthesized plans) and create executable CCS. To do so, the adaptation module associates the planning operators to Java Snippets and generates a CCS using JSLEE Service Building Blocks (SBB). SBBs are the basic components of the JSLEE architecture and are able to call external Web services or Telecom functionalities AUTO monitors CCS execution and repairs plans.

3 Modeling of Services for Multimodal Search

Our approach integrates multimodal search principles into CCS similarity detection during reconfiguration. Multimodal search is a branch of information retrieval (IR) that allows to efficiently discovering different kind of content - such as text, images, and video. Thus, the multimodal model for searching CCS unifies in single search space textual information and the structure of the CCS.

In this case, linguistics represents names and descriptions of the CCS elements (e.g. activities, interfaces, messages, gates, and events), and the structure is defined as codebooks formed of n-structural components ruled by the sequentially of the CCS control-flow (i.e. the union of two or more control-flow components. It has the following advantages: greater speed and precision, diversity in query types, and a good representation of CCS that allows delivery of results that correspond more precisely to user requirements [12].

Our reconfiguration algorithm that is based on the multimodal approach for searching service regions, a CCS must be appropriately represented so it can be dynamically composed and reconfigured. Therefore we have defined a model for representing a CCS as text strings considering users preferences, context conditions and devices capabilities. We have selected some dimensions associated with the devices and the information that can be obtained about the situation of the users using the analysis of their requests.

These dimensions are based on the context analysis module (Fig. 1). It classifies the information according to three dimensions: Device, User Profile and Situation (See Fig. 2). Each one of them obtains the information from different sources and defines the selection of domains. These dimensions allow us to analyze the capability of the system for responding to user preferences.

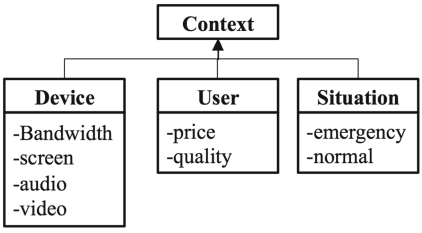


Fig. 2. Dimensions of user Context: Device, User and Situation

The Device dimension gathers the information from the device and the network. To access this information, the device ID or model is consulted in capabilities repositories like the Wireless Universal Resource File (WURFL) and the Composite Capability/Preference Profiles (CC/PP). These repositories store technical specifications of different commercial communication devices.

The User Profile dimension gathers the information using the user ID to look up his/her preferences in the preferences repository in the system. The Situation dimension analyses the Natural Language request identifying specific words such as “urgently” or “emergency”. The relationship between these dimensions is established using a preferences function, which assigns weights to each one of them. For instance, considering a user connected that employs a smartphone with video capabilities. This user has registered low cost preference; therefore, cheaper services like SMS or regular voice calls are more likely to be selected because the user price preference has a higher weight in the preferences calculation. On the other hand, if the system detects a situation of emergency, the user preferences from the repository are overridden and the most reliable services are selected instead no matter the cost of the service.

Table 1. Mapping between user context information and service properties

User criteria	Service property	Type
Network	Payload size	Bytes
Device	Payload size	Bytes
Location	Voice, text	Integer
Data subscription	Require subscription	Boolean
Only free services	Cost	Value
Voice subscription	Voice, text	Boolean
Delivery quality	Delivery warranty	Integer

To map the relationship between context and services, the Context Analyzer maps context data to services properties. This information is used in subsequent stages of Service composition, as described further on. Table 1 shows context criteria identified from the request. It additionally, shows how these user criteria are mapped to service properties. These properties are later considered during CCS reconfiguration.

4 Service Adaptation

4.1 Automatic Deployment

As mentioned before, in automated composition it is required that the user to be served immediately, so the best solution given a time window is selected whereas additional plans generated after the initial solution comprise the ranking of alternative plans that are possibly used if required (see Fig. 3). This approach is described in [11]. During the synthesis of the input problem, the best plan that satisfies the user request and preferences in a defined time is selected. The computation and execution of alternative plans is done in background so the user gets an instant response. In order to establish an association between dynamic synthesized plans and JSLEE components, the synthesized CCS are translated into Java components. To do so, the abstract CCS are integrated with execution patterns (conditional, fork, join, ...) defined as Java snippets [11].

The reconfiguration process uses an alarm-based approach. These alarms are described in the JSLEE standard. In case of failure, one or more alarms are activated. The alarms are encrusted in the Java code during the translation between abstract CCS into SBB Components. The black circles in the Fig. 3 represent such alarms.

The alarms encrusted in the executable CCS allow monitoring the execution during the workflow. Monitoring the CCS may lead to three courses of action: if the error is caused by an atomic service, the system selects another service from the implementation services repository and continues on; second, if the problem is caused by the whole CCS and the CCS itself can be modified, a new plan from

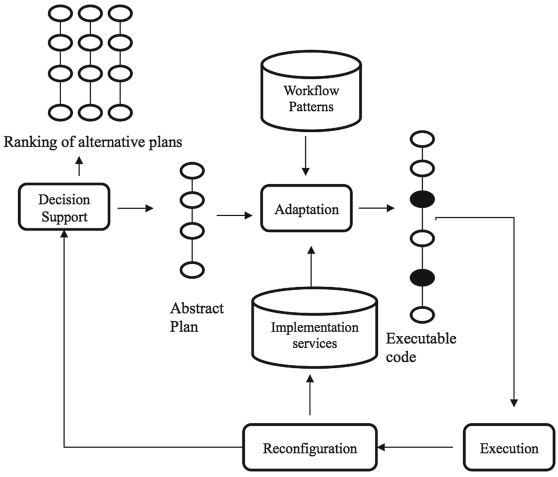


Fig. 3. Reconfiguration schema

the previously generated ranking is selected; finally, if there is a problem in the middle of the executing CCS and no alternatives are available, a new planning CCS is initiated in the Decision Support module to complete the task, starting from the actual state of the world, i.e., a set of values of the variables that define a system in a given moment.

To solve these issues, our approach incorporates a region-based reconfiguration. Next we present the region-based service reconfiguration algorithm.

In the traditional reconfiguration, if the failure is raised in more than one service, it could be necessary to re-plan and/or re-adapt the entire CCS. Besides, it could be necessary to undo all the performed tasks in the previous plan that does not match with the new generated plan. Consequently, the less reconfiguration is performed; the better would be the performance [12]. It was important to clarify that undo actions aren't feasible in some situations when services have been invoked. However, in some scenarios such as early warning management, the services or actions are to configure the sensing time of some devices, or turn on other device. In the latter situations the undo actions may be performed. Additionally, in some scenarios, the reconfiguration of telecommunication services may be done in execution time using techniques such as code injection [15]

In this paper we model the CCS as text strings and perform a novel search algorithm for discovering regions of services based on the multimodal approach. Therefore, it is important to find CCS or fragments thereof that can be reusable for defining new adjustable CCS to meet different requirements of the initial composition. One of the current challenges in this field is precisely the improvement in efficiency in the search for such CCS or fragments that normally is done manually or automated planning, involving heavy demands on time and resources.

4.2 Reconfiguration Algorithm

The Algorithm 1 for convergent service reconfiguration is called when an error is detected during execution, the algorithm is shown below:

Algorithm 1. Reconfiguration Algorithm

Require: faulty CCS si, threshold c
Ensure: replaced sub process $Rf = ri$
1: Set region $ri = si$; $Rf = \emptyset$, $counter = 0$
2: **while** $Rf \neq \emptyset$ and $counter < threshold \quad c$ **do**
3: $counter++$
4: $ri = \{ri - counter, ri, ri + counter\}$
5: Select a created plan ri' that meets QoS of ri
6: **if** $ri' \neq \emptyset$ **then**
7: add ri' to Rf
8: **end if**
9: **if** $Rf = \emptyset$ **then**
10: Search a Rf similar to si using multimodal search
11: **if** $Rf \neq \emptyset$ **then**
12: add ri' to Rf
13: **end if**
14: **end if**
15: **end while**
16: **return** Rf

Let us suppose that a plan is generated and a CCS si presents malfunctions. Then, a replacement for si in the repository must be found. The new CCS should contain the same non-functional properties as to the initial constraints of the plan defined by the user's preferences.

If such a CCS cannot be found, it is necessary to try to replace the surrounding services, so we increase the counter and expand the region. To do so, we include the previous and the next service. Replacing a set of services together gives us more flexibility as long as the replacing services can meet the combined constraints needed. In this way, the reconfiguration region can be extended or reduced in order to include services that accomplish with the aforementioned constraints. It is worth noting that if a region includes many services for replacement, selecting alternative plans from the ranking is probably a better option.

Given p faulty services and a reconfiguration threshold c ($0 < c < 1$) on the maximum number of services to be repaired, the algorithm first starts a loop (lines 2–15) to repeatedly expand the region. Inside the loop, the algorithm first tries to find a replacement region ri' for the failing service and adds it to the response Rf (lines 5–8). If the required plan does not exist, the algorithm search for similar CCS stored in the repository in order to recompose the region ri. The CCS are found using a multimodal model search approach (lines 9–13), which receives as input a text string representing the faulty services to be replaced.

4.3 Multimodal Search for Convergent Services

The multimodal search of CCS is intended to search for text string representing a convergent service or a fragment thereof. This model transforms the CCS into a matrix for two components: a linguistic component and a structural component. The linguistic component contains all the convergent services and gates that compose the CCS, and the structural component contains pairs of sequential nodes (service-service or service-gate) maintaining the order they appear in the CCS. Next the formation of the linguistic and the structural components is detailed.

Formation of the Linguistic Component: suppose that we have a repository T of CCS: $T = \{CCS_1, CCS_2, \ldots, CCS_i, \ldots, CCS_l\}$, where l is the total number of CCSs that it contains. The first step of the algorithm is to read each CCS_i and to represent it as a tree A. Then the algorithm takes each A_i, extracts the textual characteristics (activity name, activity type, and description) and forms a vector $Vtc_i = \{tc_{i,1}, tc_{i,1}, \ldots, tc_{i,l}, \ldots, tc_{i,L}\}$, where L is the number of textual characteristics $tc_{i,l}$ found in A_i. For each vector Vtc_i, which represents a CCS_i, a row of a matrix of textual components MC is constructed. This row contains the linguistic component for the CCS stored in the repository. In this matrix, i represents each CCS and l a textual characteristic for each of them.

Formation of the Structural Component: a codebook Cd is a set of N structural components describing one or more nodes of the CCS in the form of text strings. The set of codebooks formed from the whole repository is called the codebook component matrix. This matrix is formed by taking each tree A_i, which all contain a vector of codebooks $Vcb_i = \sum_{k=1}^{p} Cd_{i,k}$. Therefore, the codebook component matrix MCd_{ik} is formed where i represents the current CCS and k represents its correspondent codebook.

Indexing Process: the linguistic and codebook components are weighted to create a multimodal search index MI composed of the matrix of the linguistic component MC and the codebook component matrix MCd, i.e. $MI = \{MCd \cup MC\}$. The index also saves the reference to the physical file of each of the models stored in the repository. These models correspond to CCS stored in the BP Repository.

– Weighting: Next, the indexing layer applies a weighting scheme of terms, similar to that proposed in information retrieval (IR), via the document representation vector model to form the term document matrix. This weighting scheme is based on Eq. 1, initially proposed by Salton [16].

$$W_{i,j} = \frac{F_{i,j}}{max(F_i)} \times \log\left(\frac{I}{I_j + 1}\right) \tag{1}$$

In Eq. 1, $F_{i,j}$ is the observed frequency of a component j in CCS_i, the component j may be a linguistic (l) or codebook component k. $max(F_i)$ is the greatest frequency observed in CCS_i, I is the number of CCSs in the collection. Each cell $W_{i,j}$ of the multimodal index matrix reflects the weight of a specific element j of a CCS_i, compared with all the elements of the BP in the repository.

– Index: The multimodal index is stored in a physical file within the file system of the operating system, in which each CCS is indexed through a pre-processing mechanism. This mechanism consists in converting all the terms of the linguistic and codebook component matrices to lowercase and removing stop words, accents and special characters. Subsequently, the stemming technique (Porter algorithm [16]) is applied, which enables each of the matrix elements to be transformed into their lexical root (e.g. the terms "fishing" and "fished" become the root, "fish") and physical file of each CCS to be stored in the repository.

Search Similar CCS: when a CCS fails and there are not other services or CCS that accomplish their IOPE and QoS requirements, the multimodal search process is called. It receives the malfunctioning CCS and forms the linguistic and the structural component. Next, a conceptual rating (score) is expressed expressed by Eq. 2 (based on the *Lucene practical scoring function*[1]) is applied in order to return a ranking list of results containing a set of convergent services with the highest conceptual punctuation within the multimodal index.

$$score\,(q,d) = coord\,(q,d) \times \sum_{t \in q} \left(tf\,(t \in d) + idf\,(t)^2 \times norm\,(t,d) \right) \qquad (2)$$

In Eq. 2 t is a term of query q; d is the current CCS with q; $tf(t \in d)$ is the term frequency defined as the number of times the term t appears in d so that CCS are ranked according to the values of term frequency scores; $idf(t)$ is the number of convergent services in which term t appears (inverse frequency); $coord(q,d)$ is the scoring factor based on the number of query terms found in the queried CCS (those CCS containing the most query terms are scored highest); $norm(t,d)$ is the weighting factor in the indexing, taken from $w_{i,j}$ in the multimodal index

Eventually, the goal of the planning algorithm is to create a new plan. Furthermore, the planning algorithm considers user preferences trough the cost function that assigns a cost value to each generated plan. This process is described in detail elsewhere [10].

The algorithm continues and, if the repaired regions include too many nodes (as defined by c), the whole plan will be replaced from the existing ranking of plans.

[1] http://lucene.apache.org/core/3_6_2/api/core/org/apache/lucene/search/Similarity.html.

5 Experimentation

The experimentation was focused on computing the performance of the proposed approach which is based on the execution time evaluated in two parts: the first part verifies the performance of the proposed approach in a telecommunications environment and the second part verifies the performance of the multimodal search using test plans with different number of nodes.

For first part, an hypothetical composite convergent service was defined by running in it a Telecom Server that invokes some telecom features: send SMS, invoke a Web Service and performs a voice call. The composite service uses a set of services in a sequence. The traditional method reconfigures the whole plan from scratch, to do so; this method must undo the tasks performed before to the error occurs. Only then the new plan can be started. If the error is presented at the beginning, then there is no need for undo any task. Conversely, if the error is present at the last node it is necessary to remake the entire plan or select other plan from the ranking. For representing undo tasks in our experiment, we performed a call establishments and web service invocations. As would be expected, time increases linearly with the number of tasks that must be undone; the higher the node of the error, the higher is the time consumed to undone the previous tasks (continuous line in Fig. 4).

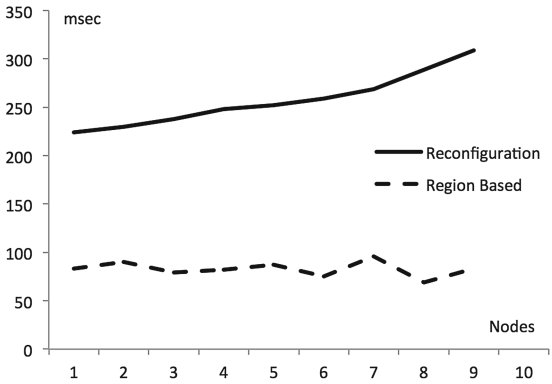


Fig. 4. Performance comparison of traditional vs. Region based reconfiguration using automated planning

Figure 4 shows the performance of the reconfiguration algorithm using automated planning. Figure 4 shows a test a plan with 10 nodes. It is assumed than an error occurred at different nodes of the path from 1st to 10th node (X axis in Fig. 4).

As can be seen in Fig. 4, the algorithm for region-based reconfiguration using automated planning has a better performance that the traditional reconfiguration processes. Next, the aim of the experimentation is to test if the multimodal search presents better results that the region based reconfiguration using automated planning.

Table 2. Performance of the multimodal search for CCS with 10, 20 and 30 nodes

10 Nodes runtime (ms)	20 Nodes runtime (ms)	30 Nodes runtime (ms)
17	40	60

For the second part, the performance of the multimodal search was evaluated finding CCS with 10, 20, and 30 nodes taking into account that most of the CCS contain only few nodes. Table 2 shows, that the Multimodal search approach performing a simple text extraction algorithm (complexity $O(n2)$) takes just a few milliseconds to retrieve the results (17–60 ms) in the CCS with 10–30 nodes. Therefore the multimodal algorithm is suitable to the process of reconfiguration of CCS as it shows a low computational complexity.

6 Conclusions

Convergent composition requires that CCS can be efficiently recovered from failures. This work presents the results of our ongoing work towards the definition of a mechanism for planning based reconfiguration in convergent domains. Furthermore, we present an algorithm based on multimodal model to find services that can be used to reconfigure troublesome regions of a CCS instead of remaking the whole CCS or selecting other plan. "Replanning" the whole CCS or selecting other plan from the ranking involves a big effort in undoing the actions of services previous to the failing service.

Multimodal search is based on text and structural information; due to the latter the quality of the CCS retrieval is higher than using automated planning. Besides, the inclusion of structural information helps to get better plans that sequential plans obtained from traditional planners. Finally the experimental results show that the performance of the multimodal search may reduce in many orders of magnitude the time of execution.

The future work will be focused in using the algorithm for performing the reconfiguration of different failing regions at the same time. Equally we are interested in perform further testing that evaluates performance and quality of the approach in Cloud based platforms for convergent services measuring real user experience.

References

1. Object Management Group: Uml profile for advanced and integrated telecommunication services (TelcoML). Standard, OMG, August 2013. http://www.omg.org/spec/TelcoML/1.0/
2. Ambra, T.: Description and composition of services towards the web-telecom convergence. In: Lomuscio, A.R., Nepal, S., Patrizi, F., Benatallah, B., Brandić, I. (eds.) ICSOC 2013. LNCS, vol. 8377, pp. 578–584. Springer, Heidelberg (2014)

3. Wang, D., Yang, Y., Mi, Z.: A genetic-based approach to web service composition in geo-distributed cloud environment q. Comput. Electr. Eng. (2014) AQ1
4. Jula, A., Sundararajan, E., Othman, Z.: Cloud computing service composition: a systematic literature review. Expert Syst. Appl. **41**(8), 3809–3824 (2014)
5. Lin, K.J., Zhang, J., Zhai, Y., Xu, B.: The design and implementation of service process reconfiguration with end-to-end QOS constraints in SOA. Serv. Oriented Comput. Appl. **4**(3), 157–168 (2010)
6. Pernici, D.A.B.: Adaptive service composition in flexible processes. IEEE Trans. Softw. Eng. **33**, 369–384 (2007)
7. Kaldeli, E., Lazovik, A., Aiello, M.: Continual planning with sensing for web service composition. In: AAAI, pp. 1198–1203 (2011)
8. Ordóñez, H., Corrales, J.C., Cobos, C.: Multisearchbp-entorno para busqueda y agrupacion de modelos de procesos de negocio. Polibits **49**, 29–38 (2014)
9. Ordóñez, A., Corrales, J.C., Falcarin, P.: Natural language processing based services composition for environmental management. In: 2012 7th International Conference on System of Systems Engineering (SoSE), pp. 497–502. IEEE (2012)
10. Ordonez, A., Alcázar, V., Borrajo, D., Falcarin, P., Corrales, J.C.: An automated user-centered planning framework for decision support in environmental early warnings. In: Pavón, J., Duque-Méndez, N.D., Fuentes-Fernández, R. (eds.) IBERAMIA 2012. LNCS, vol. 7637, pp. 591–600. Springer, Heidelberg (2012)
11. Ordóñez, A., Alcázar, V., Corrales, J.C., Falcarin, P.: Automated context aware composition of advanced telecom services for environmental early warnings. Expert Syst. Appl. **41**(13), 5907–5916 (2014)
12. Ordoñez, H., Corrales, J.C., Cobos, C.: Business processes retrieval based on multimodal search and lingo clustering algorithm. IEEE Latin Am. Trans. **13**(9), 40–48 (2015)
13. Gerevini, A.E., Haslum, P., Long, D., Saetti, A., Dimopoulos, Y.: Deterministic planning in the fifth international planning competition: PDDL3 and experimental evaluation of the planners. Artif. Intell. **173**(56), 619–668 (2009). Advances in Automated Plan Generation
14. Guzmán, C., Alcázar, V., Prior, D., Onaindia, E., Borrajo, D., Fdez-Olivares, J., Quintero, E.: PELEA: a domain-independent architecture for planning, execution and learning. In: Proceedings of the ICAPS, vol. 12, pp. 38–45 (2012)
15. Adrada, D., Salazar, E., Rojas, J., Corrales, J.C.: Automatic code instrumentation for converged service monitoring and fault detection. In: 2014 28th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 708–713. IEEE (2014)
16. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval, vol. 1. Cambridge University Press, Cambridge (2008)

Multimodal Model for Business Process Search

Hugo Ordóñez[a], Cristhian Figueroa[a,b,*], Juan-Carlos Corrales[a], Carlos Cobos[a], Enrique Herrera-Viedma[c,d]

[a] *Universidad del Cauca, Calle 5 No. 4 - 70, Popayán, Colombia*
[b] *Politecnico di Torino, Corso Duca degli Abruzzi, 24 - 10129, Turin, Italy*
[c] *Universidad de Granada, Avda. del Hospicio, s/n C.P. 18071, Granada, Spain*
[d] *King Abdulaziz University, Jeddah 21589, Saudi Arabia*

Abstract

Currently companies abstract their business complexity by defining and modeling business processes to suit a variety of tasks. However, the problem arises when searching existing business processes in large repositories in order to rethinking or re-modeling of a new business process to meet the current requirements of the companies. In this paper the above problem is addressed from a novel perspective based on the principles of multimodal search, therefore our approach unifies in a single search space the linguistics and behavior features. Linguistics represents names and descriptions of the business process elements, and behavior is defined as codebooks formed of n-structural components ruled by the sequentiality of the control-flow of the business process. To validate the multimodal approach we present an evaluation of performance and relevance based on common measures taken from information retrieval systems. Then the relevance of multimodal model results was estimated with respect to the results obtained in a manual evaluation carried out by experts on a test set. The results demonstrate that our approach scored high values of recall precision and performance.

Keywords: Business Process Evaluation, Business Process Retrieval, Business Process Similarity, Multimodal Search

*Corresponding author: Phone: +39 011090 7087
Email addresses: hugoordonez@unicauca.edu.co (Hugo Ordóñez), cristhian.figueroa@polito.it (Cristhian Figueroa), jcorral@unicauca.edu.co (Juan-Carlos Corrales), ccobos@unicauca.edu.co (Carlos Cobos), viedma@decsai.ugr.es (Enrique Herrera-Viedma)

1. Introduction

Business process management (BPM) has become an important research area that combines insights from diverse domains such as business administration, computer science and organizational theory(Corradini et al., 2010). BPM has received considerable attention in recent years both in the industry and academia due to it is central to modern organizations for increasing productivity and saving costs (van der Aalst, 2013; vom Brocke et al., 2014).

Currently companies define and use business processes (BPs) for a variety of tasks, including product manufacture, provision of services, procurement, inventory control, and so on (van der Aalst et al., 2010; Dijkman et al., 2012). The use of BPs helps companies to be focused on improving the management of operational resources to make their processes mature and repeatable, ensuring that they have scalable operations, and improving performance (Chang and Wang, 2011; Koschmider et al., 2011; La Rosa et al., 2011). BPs are usually defined by experts using graphic design and modeling tools, in order to represent the business tasks of an company through a range of activities linked together by a common business goal.

Accordingly, many companies with a high maturity level have generated large collections or repositories of BPs. For example, Wang et al. (2014) and Dijkman et al. (2011a) report that there are companies with repositories ranging from hundreds to thousands of BPs. Therefore, the management of these repositories requires effective search techniques.

In this scenario, companies must resort to manual reviews of large amount of information on repositories in order to find reusable BPs or fragments thereof for defining new adjustable BPs to meet different requirements of the company (Smirnov et al., 2012; Dijkman et al., 2011b). Furthermore, it is preferable to find reusable BPs to customize, rather than building a new one from scratch (Wang et al., 2014).

However, due to the heterogeneity of BPs, searching, extracting and analyzing them from these repositories is a tedious task and time consuming, that can hampers the creation of new BPs(Eid-Sabbagh et al., 2012). For that reason, BPs search is one of the current challenges in BPM as it allows companies to offer new products and more competitive services in the market (Gacitua-Decar and Pahl, 2009; Li et al., 2011; Wang et al., 2014).

Most of the current approaches for BP search are based on typical measures as linguistics, structure, and behavior, however there are other approaches from the Information Retrieval (IR) field that should be studied in

order to improve the existing techniques.

Among those approaches reporting the most accurate results for users are multimodal search systems, which involves the combination of different types of data to extend and complement the information to be retrieved (Revuelta-Martínez et al., 2012). Moreover, multimodal search supports different types of queries that leads to improve the performance when combining diverse search strategies with the input query (Xie et al., 2007). However, they have not yet been applied for searching BPs.

This paper proposes a multimodal search model for BP similarity, which unifies in a single search space the linguistics and behavior features. In this case, linguistics represents names and descriptions of the BP's elements (e.g. activities, interfaces, messages, gates, and events), and the behavior is defined as *codebooks* formed of n-structural components ruled by the sequentiality of the BP's control-flow (i.e. the union of two or more control-flow components). In order to validate the multimodal model, an evaluation of performance and relevance was developed, using measures taken from IR systems - such as *precision*, *recall*, *F-measure*, *ANDCG* and *GenAveP* (Zhang et al., 2014; Manning et al., 2008). Using these measures the relevance of multimodal model results was estimated with respect to the results obtained in a manual evaluation carried out by 59 experts in BPM over a closed collection presented in Ordóñez et al. (2014) basen on a collaborative strategy. Finally, the results on relevance were compared with the results obtained by other BP similarity proposals such as the *BeMantics* Figueroa and Corrales (2012) tool and a suitable implementation of the A* algorithm (Grigori et al., 2008) using the same test set.

This paper is organized as follows: section 2 presents work related to the research topic; section 3 focuses on the proposed search model; in section 4, the experiments are presented, in which relevance is evaluated along with performance; and Section 5 outlines conclusions and future work.

2. Related Work

BP search involves finding a set of BP structures that are similar to a query (the query can be a complete BP or a fragment of it). Until now the BP similarity has been addressed by three main metrics: linguistic similarity that compares labels and attributes attached to the elements of the BPs; structural similarity that compares the topology of the BP; and behavioral

similarity that compares the causal relations detected within the BP (Dijkman et al., 2011a). In the following, some of the most representative works in this field are described.

The BP search based on linguistic similarity infers about the concepts related to names and types of activities, control-flow connectors, and interfaces (inputs/outputs)(Awad et al., 2008). In this way it is possible to determine conceptual relationships between BP models stored in a repository and a query that a user enters. Some of the most relevant works on this topic are described below:

The BP search proposed in (Koschmider et al., 2011) makes semantic recommendations that extend the query. This method is based on the term frequency (TF) measurement that evaluates the occurrence of the terms in each label attached to the activities of the BPs. In this way it is possible to apply IR techniques such as index creation, elimination of empty words, and weighting of terms. The search has two options basic, or extended. The basic option queries on a BP repository or on one particular BP by incorporating the *WordNet* ontology as an element of generation of semantic suggestions. In the extended option, each BP activity is considered as a vector of terms, adding a function of partial cost (fp), with which a total cost function (ft) is calculated. The ranking of the results is performed with the lowest ft or those weighing least.

In (Reijers et al., 2011) a syntax compression method is presented, based on Petri nets in the execution flow modeling of the BP. First, a theoretical argument to establish the degree of syntax compression is presented, addressing the semiotics (study of signs) of the graphs, in which eight different visual variables are identified that can be used to encode the information in the BP, and color is taken as one of the most effective variables for distinguishing the elements of notation. Secondly, a formalization of concepts in the modeling of execution flows is developed, for which the BP is taken as a bipartite directed graph. In the search for the new model, an algorithm called *max-flow-min-cut* is run, which performs a pairing of nodes to find the maximum flow of connection operator coincidences.

The structural similarity based BP search takes into account the topology of the BPs generally represented as BP graphs BPG. Dijkman et al. (2011a) define a BPG as a graph that captures node and edge types of different notations as attributes. Formally, a BPG is a tuple $(N, E, \tau, \lambda, \alpha)$ where N is a finite set of nodes that represent the activities of the BPs; $E : N \times N$ is a finite set of edges that represent links between nodes; $\tau : (N \cup E) \rightarrow T$

associates nodes and edges with types; $\lambda : (N \cup E) \rightarrow \Omega$ associates nodes and edges with labels; and $\alpha : (N \cup E) \rightarrow (T \rightarrow \Omega)$ associates nodes and edges with attributes (combination of a type and a label). BPG is a representation that allow the application of the graph isomorphism mathematical technique for a similarity comparison.

Grigori et al. (2008) compare the BPs via a matching algorithm (called error-correction algorithm) in which a set of editing operations is applied to each of the graphs in the repository in order to find an edited graph that resembles as closely as possible a query graph. Furthermore, during the structural comparison, this algorithm performs a lexical comparison between each of the activities of the BPs, matching the labels of their names and calculating a lexical distance.

In (van der Aalst et al., 2010) two important measures are put forward for evaluating the similarity of two BPs. The first measure uses a linguistic distance known as the Hamming distance, which allows calculation of the minimum number of symbol editing operations (substitutions, insertions and deletions) within the text strings that identify the activities. The second measure evaluates a structural distance that is calculated by way of graph isomorphism.

In (Kunze et al., 2011) a similarity measure based on behavioral profiles is presented. This measure uses the Jaccard coefficient to match activities in order to find behavioral relations between pairs of activities of BPs; in this way it can be estimated a behavioral overlap between BPs.

As can be seen the related works revised in this paper are based on typical measures as linguistics, structure, and behavior, however there are other approaches from the IR field that should be studied. Among those approaches reporting the most accurate results for users are multimodal search systems, which involves the combination of different types of data to extend and complement the information to be retrieved (e.g. the relationship between visual content and text) (Pedronette et al., 2014; Revuelta-Martínez et al., 2012). This type of search has advantages: greater effectiveness, precision in results returned, and performance; it also allows users to customize their preferences in various contexts and for different search types.

Furthermore, the multimodal systems have been enhanced through the implementation of codebooks in the domain of recovery of images used as visual pattern histograms (He et al., 2008), visual vocabularies or visual dictionaries (Caicedo et al., 2012; Fonteyn et al., 2008; Lazaridis et al., 2013). However, they have not yet been used in BP similarity.

In this paper a multimodal search model for BP similarity is proposed, which unifies in a single search space the linguistics and behavior features. In this case, linguistics represents names and descriptions of the BPs elements, and the behavior is defined as codebooks formed of n-structural components ruled by the sequentiality of the BPs control-flow (activity-activity, activity-gate, activity-event-gate, and so on).

The multimodal model generates more accurate results sorted according to the similarity of a query with the BPs stored in a repository, and it improves the search time involved. Unlike previous proposals where the search process is executed in diverse sequential steps, in this model the search is performed in a single unified process.

3. A Multimodal Approach for Business Process Search

The multimodal model described here represents a new approach that exploits the efficiencies of multimodal search models by applying them in the field of business process similarity. The reference architecture for the proposed multimodal model and each of its components are described below.

3.1. Architecture of the multimodal model

The multimodal model allows the management (search, add, and delete) of the BPs stored in the repository, where the search is according to their similarity to the query BPs. The repository contains the physical files of the BPs modeled with BPMN (Business Process Modeling Notation). Similarity applies a search strategy that integrates linguistic and structural information contained in the BPs, thereby facilitating an increase in the effectiveness and relevance of search results from the queries defined by the user. The multimodal model architecture consists of three layers (see Figure 1): Parser, Indexing, and Query, which are described below.

3.1.1. Parser layer

This layer consists of a parser that transforms the BPs from their original BPMN format to a vector representation in which each BP is considered a matrix of terms comprising a linguistic and a codebook component.

Parser: The parser contains an algorithm that takes a BP in BPMN notation and builds the codebook of the BP and the linguistic component. The algorithm is described below:

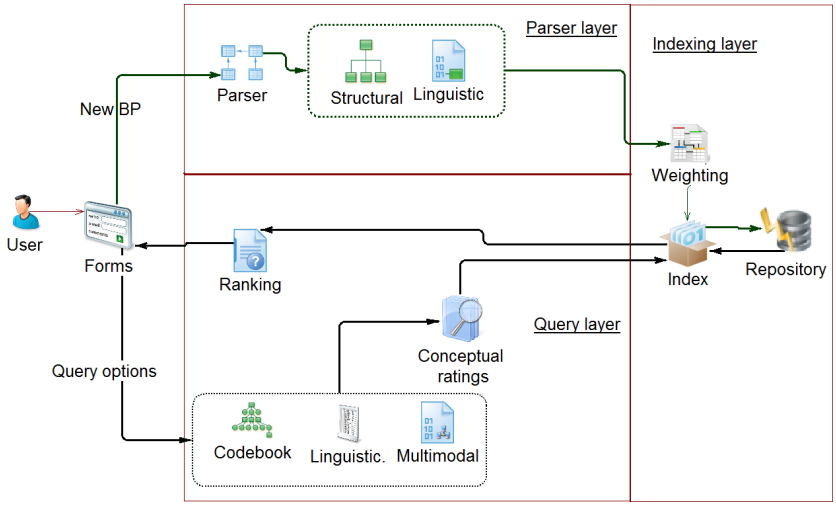


Figure 1: Multimodal model architecture

Formation of linguistic component (*Linguistic*): suppose that we have a repository T of BPs: $T = \{BP_1, BP_2, \ldots, BP_i, \ldots, BP_I\}$ where I is the total number of BPs that it contains. The first step of the algorithm is to read each BP_i and to represent it as a tree A such that: $BP_i = A_i \rightarrow (v, x)$, where v is a node of tree A_i and x represents its edges. Then the algorithm takes each A_i, extracts its textual characteristics Ct (activity name, activity type, and description) and forms a vector $Vtc_i = \{Ct_{i,1}, Ct_{i,2}, \ldots, Ct_{i,l}, \ldots Ct_{i,L}\}$, where L is the number of textual characteristics found in A_i. For each vector Vtc_i, which represents a BP_i, a row of matrix MC_{il} is constructed. This row contains the linguistic component of all BPs stored in the repository. In this matrix, i represents each BP and l a textual characteristic for each of them.

Formation of codebook component (*Structural*): a codebook Cd is a set of N structural components describing one or more nodes of the BP in the form of text strings. The set of codebooks formed from the whole repository is called the codebook component matrix. This matrix is formed by taking each tree A_i, which all contain a vector of edges $vt_i = \{t_{i,1}, t_{i,2}, \ldots, t_{i,k}, \ldots, t_{i,K}\}$ where $t_{i,k}$ is an edge and K is the number of edges in A_i. Then, a vector of codebooks $Vcb_i = \sum_{k=1}^{p} Cd_{i,k}$ is created $\forall Cd_{i,k} = (vt_{i,k} - 1, vt_{i,k})$. Therefore, the codebook component matrix MCd_{ik} is formed where i represents the BP and k represents the codebook for each BP. For example, Figure 2 shows a fragment of a BP_i with its activities. Each activity is represented with a text string defining the node type (*StartEvent*, *TaskUser*, *TaskService*). The

node type refers to the functionality of each activity within the BP.

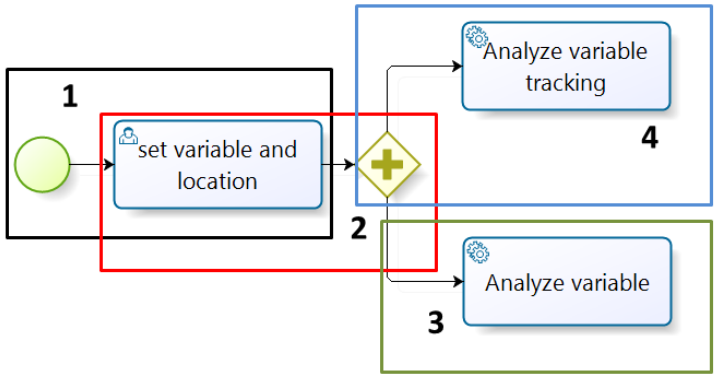


Figure 2: Codebook formation example

Table 1 shows the correspondence between the activities of the BP_i in figure 2 and their node types.

Activity	Type Representation
Start	StartEvent
Set variable and location	TaskUser
Route	RouteParallel
Analyze variable tracking	TaskService
Analyze Variable	TaskService

Table 1: Example of the activities of the BP in figure 2 and their types

In the example of figure 2, pairs of sequences of nodes or components (node types) are taken to represent the set of codebooks. These pairs of nodes are represented as sequential text strings (i.e. adding the text string of the preceding node):
$Vcb_i = \{StartEvent\_TaskUser_{i,1}, TaskUser\_RouteParallel_{i,2}, RouteParallel\_TaskService_{i,3}, RouteParallel\_TaskService_{i,4}\}$. Where $StartEvent\_TaskUser_{i,1}$ corresponds to the activity $Start$ in Figure 2 ($StartEvent$) concatenated with the activity $Setvariableandlocation$ in Figure 2 ($TaskUser$), and similarly for the other components of Vcb_i.

The codebooks are formed simulating the technique of N-grams (sequence of N characters forming a gram in text chains). Unlike the n-grams, which are simple linear sequences of characters, the codebooks are formed by joining

n-structural components using the sequence and semantics of the control-flow represented by the node types. According to Wang et al. (2014) the behavior semantics of BPs describe the activities that are involved and their execution order. The execution order in our work is represented by codebooks or pairs of sequential activities. It is important to note that codebooks respect the sequentiality and semantics of the execution flow of the BPs.

3.1.2. Indexing layer

In this layer the linguistic and codebook components are weighted to create a multimodal search index MI composed of the matrix of the linguistic component (MC) (the blue box on the right in Figure 3) and the codebook component matrix (MC_d) (the green box on the left in Figure 3), i.e. $MI = \{MC_d \cup MC\}$. The index also saves the reference to the physical file of each of the models stored in the repository.

	MCd						MC					
	$Cd1$	$Cd2$		Cdk		CdK	$Ct1$	$Ct2$		Ctl		CtL
BP1	$W_{1,1}$						$W_{1,1}$					
BP2		$W_{2,2}$						$W_{2,2}$				
...			...						...			
BPi				$W_{i,k}$						$W_{i,l}$		
...					...						...	
BPI						$W_{I,K}$						$W_{I,L}$

Figure 3: Multimodal index matrix

Weighting: Next, the indexing layer applies a weighting scheme of terms, similar to that proposed in information retrieval (IR), via the document representation vector model to form the term document matrix. This weighting scheme is based on Equation 1, initially proposed by Salton, that can be found in the work of Manning et al. (2008).

$$W_{i,j} = \frac{F_{i,j}}{max(F_i)} \times \log\left(\frac{I}{I_j + 1}\right) \tag{1}$$

In Equation 1, $F_{i,j}$ is the observed frequency of a component j in BP_i, the component j can be a linguistic (l) or codebook component (k). $max(F_i)$ is the greatest frequency observed in BP_i, I is the number of BPs in the

collection, and N_j is the number of BPs in which the linguistic or codebook component j appears. Each cell $w_{i,j}$ of the multimodal index matrix reflects the weight of a specific element j of a BP_i compared to all the elements of the BP in the repository.

Index: The multimodal index is stored in a physical file within the file system of the operating system, in which each BP is indexed through a preprocessing mechanism. This mechanism consists in converting all the terms of the linguistic and codebook component matrices to lowercase and removing empty words, accents and special characters. Subsequently, the stemming technique (Porter algorithm (Manning et al., 2008)) is applied, which enables each of the matrix elements to be transformed into their lexical root (e.g. the terms "fishing" and "fished" become the root, "fish") and a reference in the physical file of each BP to be stored in the repository.

Repository: The repository is the central BP storage unit, similar to a database, that shares information about the engineered artifacts produced or used by a company (Dijkman et al., 2011b; Smirnov et al., 2012). It is responsible for storing and representing all the attributes of information present in the BP (roles, description of activities, timers, messages, and service calls) (Reimerink et al., 2010). Our repository is composed of 100 BPs obtained from real processes of diverse companies of the telecommunications and georeferencing domains.

The repository was built using diverse definition languages such as OWL-S (Semantic Markup for Web Services), BPEL (Business Process Execution Language), BPMO (Business Process Modeling Ontology) and BPMN (Business Process Model Notation). Unlike the other languages the BPs modeled with BPMO were semantically enriched with concepts of the SeTOM (semantic enhanced Telecom Operations Map) and SSID (Semantic Shared Information/Data Model) ontologies from the telecommunications domain. Then semantic concepts from these ontologies were manually added by experts to each element of the BPs.

In our approach, as as the semantic annotations are not needed, we only used the BPs modeled with BPMN, as it is the most common language for the process definition and automation. However, we are planning to use these semantic annotations for future extensions of our prototypes.

3.1.3. Query layer

This layer is responsible for allowing users to conduct BP searches beginning with three query options: linguistic, codebook, and multimodal. Each

query is represented through a terms vector $q = \{t_1, t_2, t3, \ldots, t_j, \ldots, t_J\}$. The same pre-processing mechanism applied in the indexing layer is applied to this vector, thus obtaining the terms of the query vector reduced to their lexical root.

Conceptual ratings: Once the vector is preprocessed, the search is executed according to the query option chosen by the user. The conceptual rating (score) expressed by Equation 2 (based on the *Lucene practical scoring function* [1]) is applied in order to return a ranking list of results containing a set of BPs with the highest conceptual punctuation within the multimodal index.

$$score\,(q,d) = coord\,(q,d) \times \sum_{t \in q} \left(tf\,(t \in d) + idf\,(t)^2 \times norm\,(t,d) \right) \qquad (2)$$

In Equation 2 t is a term of query q; d is the current BP being evaluated with q; $tf(t \in d)$ is the term frequency defined as the number of times the term t appears in d so that BPs are ranked according to the values of term frequency scores; $idf(t)$ is the number of BPs in which term t appears (inverse frequency); $coord(q,d)$ is the scoring factor based on the number of query terms found in the queried BP (those BPs containing the most query terms are scored highest); $norm(t,d)$ is the weighting factor in the indexing, taken from $w_{i,j}$ in the multimodal index

List of results (Ranking): Once the results are sorted and filtered, they are listed according to the level of similarity found with respect to the BP in the query option selected by the user.

In addition to the multimodal model components, the user has forms that correspond to the graphical user interface (GUI), used to add new BP models to the repository, to execute each of the search options and to display the lists of the results of each of the searches performed.

3.2. Example of execution

This section presents an example of the multimodal model that describes the steps performed: parsing, indexing, and querying. In the example, a repository with three BPs is considered (BP_1, BP_2, and BP_3) (Figure 4),

[1] `http://lucene.apache.org/core/3_6_2/api/core/org/apache/lucene/search/Similarity.html`

which have previously been processed by applying algorithms to build the linguistic and structural components. These algorithms weigh each element of the BPs in order to build the multimodal index.

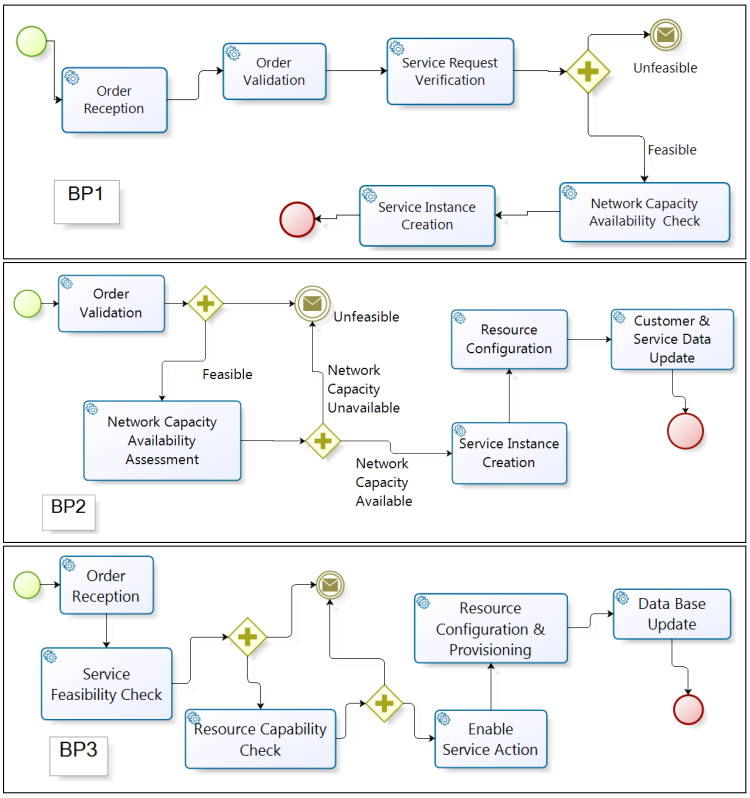


Figure 4: Repository with three BPs

The search phase receives a query BP that may or may not be stored in the repository. If it is not, then it is processed by applying the aforementioned algorithms, added to the index, and stored in the repository. Thereafter, the user can choose between the following query options: linguistics, codebook, and multimodal, and after selecting an option a result list ordered according to the similarity between the query BP and the BPs stored in the repository is presented. The following describes the steps executed in the model:

Parsing: Parsing starts by taking each of the BPs in the repository (BP_1, BP_2, and BP_3). It then creates the codebook component (MC_d) and linguistic (MC) matrices. The formation of each of these matrices can be seen below.

MC_d = { [StartEvent_TaskService, TaskService_TaskService, TaskService_TaskService, TaskService_RouteParallel, RouteParallel_TaskService, RouteParallel_TriggerResultMessage, TaskService_TaskService, TaskService_EndEvent], [StartEvent_TaskService, TaskService_RouteParallel, RouteParallel_TaskService, RouteParallel_TriggerResultMessage, TaskService_RouteParallel, RouteParallel_TaskService, TaskService_TaskService, TaskService_TaskService, TaskService_EndEvent], [StartEvent_TaskService, TaskService_TaskService, TaskService_RouteParallel, RouteParallel_TaskService, RouteParallel_TriggerResultMessage, TaskService_RouteParallel, RouteParallel_TaskService, TaskService_TaskService, TaskService_TaskService, TaskService_EndEvent] } .

MC = { [Start, Order Reception, Order Validation, Service Request Verification, Exclusive Choice, Network Capacity Availability Check, Service Instance Creation, End],[Start, Order Validation, Exclusive Choice, Network Capacity Availability Assessment, Exclusive Choice, Service Instance Creation, Resource Configuration, Customer & Service Data Updation, End],[Start, Order Reception, Service Feasibility Check, Exclusive Choice, Send Message, Resource Capability Check, Exclusive Choice, Enable Service Activation, Resource Configuration & Provisioning, Data Base Updation, End] } .

Indexing: In this phase, each BPi found in the repository is taken and used to form the multimodal matrix $MI_i = \{MCd_{I,j} \cup MC_{I,j}\}$ comprising the elements $w_{i,j}$, whose values are calculated using Equation 1. The index that forms the query space is created in the file system by uniting the matrices that result for each BP in the repository. Therefore, $M_{I \times M} = \{MI_I \cup MI_M\}$ where I is the number of BPs in the repository and $M = J + K$ is the total number of components including linguistic (J) and codebook (K). Table 2 shows the weighting calculation of $w_{i,j}$. For example, $w_{1,1} = \frac{1}{3} \times \log\left(\frac{3}{1+1}\right) = 0.14$.

	MCd										**MC**									
BP_1	0,14	0,30	0,30	0,60	0,60	0,30	0,30	0,20	0,00	0,00	0,20	0,30	0,30	0,60	0,20	0,60	0,60	0,20	0,00	0,00
BP_2	0,14	0,39	0,39	0,60	0,39	0,39	0,26	0,20	0,00	0,00	0,20	0,30	0,26	0,60	0,39	0,60	0,30	0,60	0,20	0,00
BP_3	0,14	0,30	0,39	0,39	0,30	0,39	0,39	0,30	0,30	0,20	0,20	0,30	0,60	0,26	0,60	0,60	0,26	0,60	0,60	0,20

Table 2: Example of Weighting calculation[2] of $w_{i,j}$

Query: In this phase the user introduces a BP model as a "*query BP*". This query BP is pre-processed to form a vector that includes the linguistic

[2]Numbers are rounded to two decimal places merely for simplicity

and codebook components. Later, the query vector is taken and the weight of each of its components is calculated using Equation 1. Finally, the vector is compared with the weights in the multimodal index matrix, and the BPs that most closely resemble the query BP are recovered in an ordered fashion (using Equation 2). For example, a query BP is shown in Figure 5, then its corresponding query vector is shown in Table 3, and finally rankings obtained are shown in Table 4.

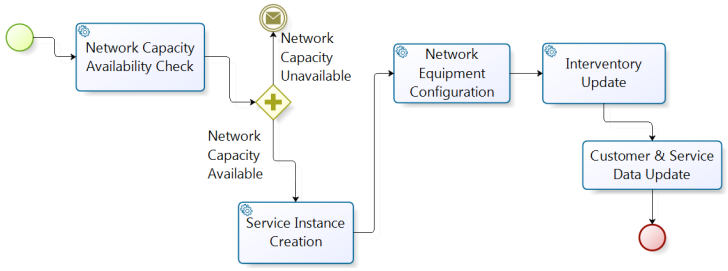


Figure 5: Example of a query BP

	MCd										MC									
BPq	0,14	0,20	0,30	0,60	0,60	0,30	0,60	0,30	0,20	0,00	0,20	0,20	0,30	0,60	0,20	0,60	0,60	0,20	0,60	0,00

Table 3: Example of query BP weighting

Query Options	BP_1		BP_2		BP_3	
	Similarity	Ranking	Similarity	Ranking	Similarity	Ranking
Codebook query	0.9506	1	0.9036	2	0.8926	3
Linguistic query	0.5304	2	0.7094	1	0.1735	3
Multimodal query	0.9537	2	0.9693	1	0,9233	3
Ideal list		2		1		3

Table 4: Example of results based on query BP, repository, and query options

Given that ideally the BP with the greatest similarity to the query BP is BP_2, it can be seen that the best results were achieved with the multimodal model, followed by the results using the linguistic approach and finally the codebook. This shows that the multimodal model results have a higher level of relevance, because the ranking of the result list is organized in a similar way to the ranking generated by the human judges (ideal list). This approach recovers a set of BPs with the highest similarity values according to the query BP.

4. Experimentation

This section presents the evaluation methodology used, and the assessment of the results of the multimodal model.

4.1. Evaluation methodology

To determine the efficiency of the model, it required to undergo: an experimental evaluation to calculate the relevance of the results, and the performance of the approach based on the time spent in the execution of the query. The results obtained by our approach were compared with the judgements pronounced by experts in the field among a closed collection composed of 100 BPs from real environments and a subset of six queries (also BPs).

In the experimental evaluation, the multimodal model was used to generate BP rankings in which lists of the first 6, 8, 10, 15, and 20 BPs in the repository were considered, according to their similarity with the query BP. In this sense, it is possible to assess the relevance of the results obtained in the execution of each search, starting from the metrics widely used in the evaluation of information retrieval systems (Petratos, 2007): Precision graded (*Pg*), Recall graded (*Rg*), *F-measure*, *ANDCG*, and *GenAveP'* (Küster and König-Ries, 2010), which provide a classification of those BPs in the repository considered similar to a query BP, given different levels of relevance. These measures are described in Section 4.3.

4.2. The closed test collection

One of the main problems with regard to BP search is the lack of free and open-access closed test collections. This makes it difficult to benchmark BP search approaches developed by the scientific community(Dijkman et al., 2011b). For this reason, we have developed a closed test collection composed of: a repository of 100 BPs described in BPMN language based on the real environments (telecommunications and geo-referencing); a predefined list of 6 BPs as queries; and a list of ideal results for each query (set of relevant BPs).

To generate the relevant set of BPs, we based our evaluation on a collaborative strategy, where 59 expert reviewers from different institutions executed comparisons between pairs of BPs (a query BP and each BP of the repository). The profiles of the expert reviewers and their affiliations can be seen in Table 5.

Institution/Degree	PhD	MsC	Bachelor
Information Systems Group Federal University of Rio Grande do Sul(Brazil)	-	7	14
Management School Federal University of Rio Grande do Sul(Brazil)	-	-	33
Telematics Engineering Group Universidad del Cauca(Colombia)	2	3	-

Table 5: Profiles of evaluators

Our closed test collection was subsequently used to evaluate the effectiveness, precision and performance of our approach. The process we followed for building our closed test collection and the values achieved in the different relevance metrics can be found in Ordóñez et al. (2014). The closed test collection including relevance judgements is available on the web at: `http://artemisa.unicauca.edu.co/~cfigmart/bpcollection/`

4.3. Measures for the evaluation of relevance

These measures calculate the relevance of the retrieved results of a BP similarity tool in decreasing, gradual, and continuous forms. They measure the gain of a result item based on the position of the item in the ranking, recognizing that the most relevant BPs are most useful if they appear in the top positions of the ranking (Küster and König-Ries, 2010).

Graded relevance measures (Pg and Rg) were applied in the above, to provide a classification (T_i) of the BPs returned in the repository, those which are considered similar to a query BP (Q) according to different levels of relevance. Pg and Rg take into account the sum total of degrees of relevance between the BPs. In our evaluation we considered an adaptation of the equations for Pg and Rg found in the work of Kekäläinen and Järvelin (2002).

In addition, to measure the quality of the ranking of the results generated by the multimodal model, *ANDCG* (Average Normalized Discounted Cumulated Gain) and *GenAveP'* (Generalized Average Precision) measures were used, which were presented and improved in the work of Küster and König-Ries (2010). These measures quantify the quality of the ranking produced by the retrieval tools of the Web services, but are fully applicable to the business processes search field. These measures are described below.

Precision graded (Pg): assesses the system's ability to retrieve only the relevant elements (i.e. those elements considered to be similar to a query

according to the expert evaluators), avoiding the retrieval of irrelevant items (i.e. false positives), relating the minimum values between the automatic tool and results of expert assessors with the summation of all the results. Pg is evaluated using the Equation 3.

$$P_g = \frac{\sum T_i \in T \times min\left\{f_r(Q,T_i), f_e(Q,T_i)\right\}}{\sum T_i \in T \times f_e(Q,T_i)} \tag{3}$$

Recall graded (Rg): evaluates the ability to retrieve all the relevant elements (i.e. false negatives) considered relevant by the expert evaluators, taking as a relationship the sum of all the minimum values between the results of an automated tool (f_e) and the results of the experts (f_r), with the sum of all the results of the expert evaluation. Equation 4 was used to calculate Rg.

$$R_g = \frac{\sum T_i \in T \times min\left\{f_r(Q,T_i), f_e(Q,T_i)\right\}}{\sum T_i \in T \times f_r(Q,T_i)} \tag{4}$$

Graded F-measure: evaluates the combination Pg and Rg in a single harmony value, as can be seen in Equation 5 (Pérez-Castillo et al., 2011). This equation determines the percentage of truly relevant documents retrieved.

$$Mf_g = \frac{2R_g \times P_g}{R_g + P_g} \tag{5}$$

ANDCG and GenAveP': The *ANDCG* and *GenAveP'* measures, unlike precision and recall that estimate the quality of the results in terms of the number of relevant elements obtained, take into account the quality of the ranking generated by the tool. In this case, if a tool delivers more relevant items at the top of the rankings then it will be classified higher. These measures can be evaluated using the Equations 6 and 7 defined by Küster and König-Ries (2010).

$$ANDCG = \frac{1}{|R|}\sum_{i=1}^{L}\frac{DCG(i)}{IDCG(i)} \tag{6}$$

$$GenAveP' = \frac{\sum_{i=1}^{L}\frac{CG(i)}{i}}{\sum_{i=1}^{R}\frac{ICG(i)}{i}} \tag{7}$$

In Equations 6 and 7 $CG(i) = \sum_{j=1}^{i} g(r_j)$ is the cumulative gain in the ranking i, i.e. the gain (g) that an automated tool assigns to the first i items in a ranking. The $ICG(I)$ measure evaluates the gain a user assigns to the first i items in a ranking (in this case they correspond to the elements deemed relevant). The $DCG(i) = \sum_{j=1}^{i} \frac{g(i)}{disc(i)}$ is similar to the CG but uses a discount factor ($disc(i)$) which assigns a greater value to the first elements and reduces the value of the tail end elements of the ranking. In this article, the discount factor used is: $disc(i) = max(1, \log_b i)$.

4.4. Evaluation of relevance

The evaluation of relevance of the results involves three phases. The first is run to calculate the best structural component of the codebook (value of N in structural components) that would achieve the greatest relevance and quality of ranking in the results of the multimodal model (tuning of codebook size); the second, based on the above, assesses the measures of relevance and quality of ranking to determine which is the best search option (between linguistic, codebook, and multimodal) of the multimodal model; and the third compares the best results of the multimodal model with other BP search tools.

4.4.1. Tuning of codebook size

In the first instance the formation of the codebook is evaluated to determine the structural value of the codebook (*N-component*) that allows the best results according to relevance measures, and with that value to form the multimodal model. The evaluations were performed by results intervals, taking into account rankings of 8, 10, 15 and 20 BPs returned by the tool.

For assessment of the search by codebook, several queries were performed deploying rankings with the number of items mentioned above. In addition, the codebooks were constructed by component sequences with values between 1 and 8. Figure 6 shows the average percentages of each N value obtained in the different rankings of this evaluation.

In Figure 6, it can be seen that as the number of components (N) of the codebook increases, Pg, Rg, and consequently *F-measure* is reduced considerably, achieving a slight balance between codebooks from $N = 5$ to $N = 8$. In this case the highest values in these steps (69%, 40%, and 51% respectively) are reported in component $N = 2$.

As for the *ANDCG* and *GenAveP'* measures, it can be seen that they remain more stable than the previous measures, which allows us to conclude

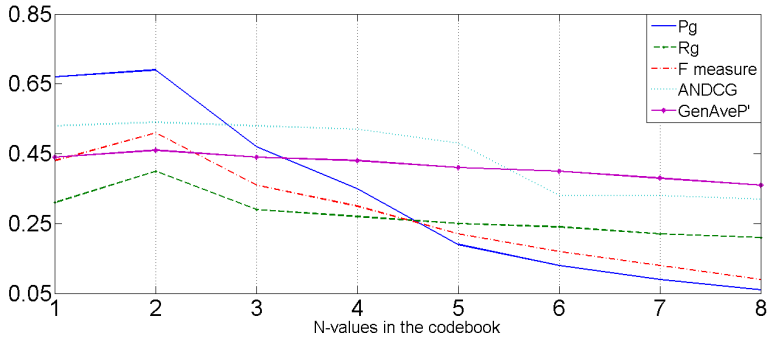


Figure 6: Graph of precision against N-values in the codebook

that the quality of the ranking shows some independence from the N-value defined for codebooks maintaining an average quality of 55%.

From the above it can be concluded that the codebook formed by sequences of two components ($N = 2$) achieved the highest levels amid the results of all measures.

4.4.2. Definition of the best search option

This section presents the assessment of relevance of the results obtained by the multimodal model in each of search options it offers. For this, it is noted that the best codebook component of the model is $N = 2$. The query options evaluated were: linguistic search, codebook search, and multimodal search (the latter corresponds to the combination of the previous two). Below are the results for each measure of relevance.

Figure 7 shows the results in rankings with 6 - 20 items for each of the query options of the proposed model. In the results obtained, the rankings with 6 results show the highest levels of Pg, in this case the linguistic option reached 86.3%, codebook 74.86%, and multimodal 92.24%. This shows that the multimodal model (integrating both linguistic and codebook options) increases the level of Pg to 92.24%, leaving less than 8% as false positives (i.e. irrelevant BPs retrieved).

Figure 8 displays Rg levels for each of the multimodal model options. Similar to the Figure 6, for each one of the query options the rankings with 10 items obtain the highest levels of Rg: 31.41% for the linguistic option, codebook 30.74%, and multimodal 32.25%. These results show that the multimodal option obtains the least number of false negatives (67.75% of relevant BPs unrecovered) when compared with the other options

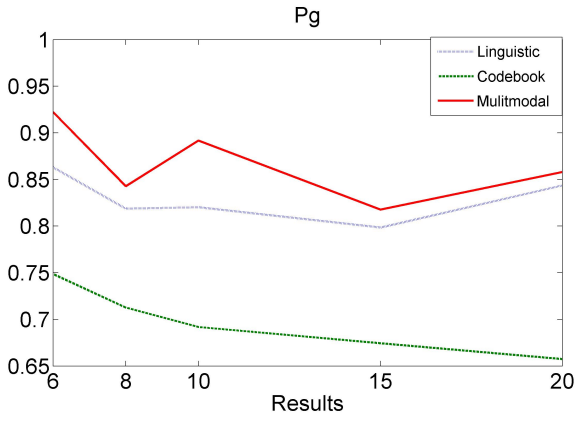


Figure 7: Graph of precision graded

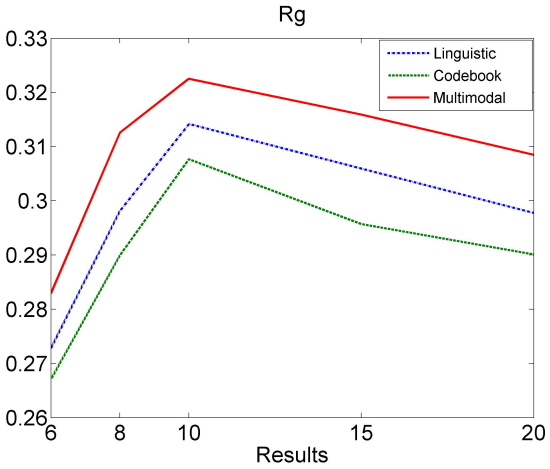


Figure 8: Graph of Recall graded

Figure 9 shows the results of assessment using *F-measure*. In these, it can be seen that, just as with Rg above, the ranking with 10 items shows better levels of harmony between Pg and Rg for each of the multimodal model options: 45.43% for the linguistic option, codebook 42.59%, and multimodal 46.63%. From which we can conclude that the search option with the best harmony is the multimodal, clear from the mere fact that it achieved the best levels in the two previous steps.

Figure 10 presents the results of the *GenAveP* measure, in which it can

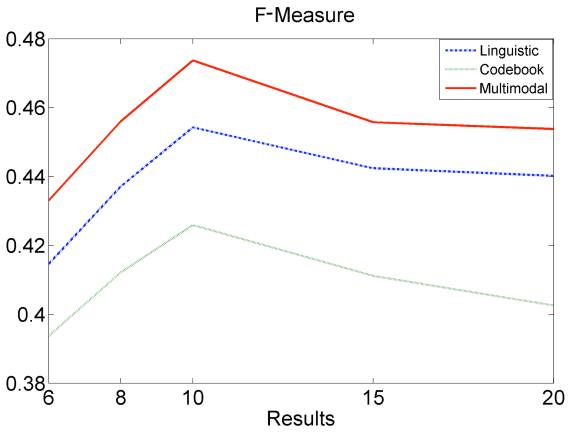


Figure 9: Graph of *F-measure*

be seen that the multimodal option shows slightly higher values than the other two options, indicating that this option offers a better ranking quality.

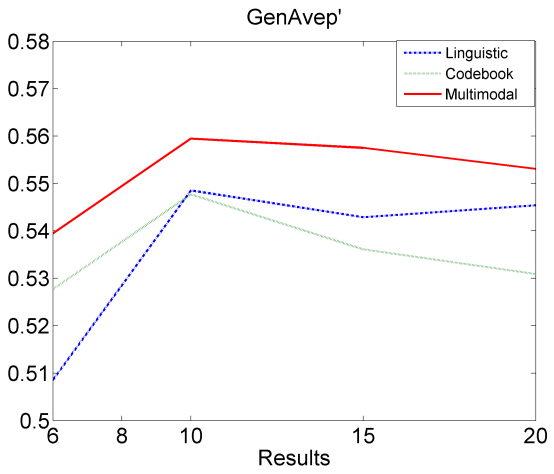


Figure 10: Graph of *GenAveP'*

Figure 11 shows the results of the *ANDCG* measure which as explained above is very similar to the *GenAveP'* measure in the sense that it measures the quality of the ranking. However, this figure shows that the ANDCG results differ from the *GenAveP'* results, because the first uses a discount factor that penalizes the results obtained at the end of the ranking. Figure 11 therefore demonstrates that the multimodal option gives a better quality

of ranking, classifying a greater quantity of relevant elements in the upper reaches of the ranking than the linguistic and codebook options do separately.

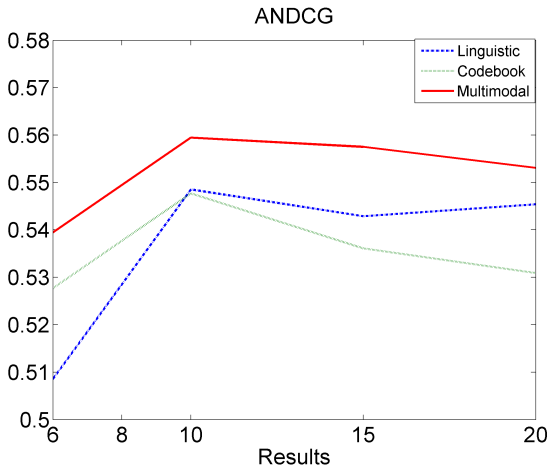


Figure 11: Graph of *ANDCG*

On the other hand it is worth noting that, as in the graphs for *Rg*, *F-measure*, *ANDCG*, and *GenAveP'*, the highest values were obtained in the rankings of 10 items. Therefore, in making comparisons with other models, this number of items was chosen for the rankings.

4.4.3. Comparative assessment with other models

The proposed multimodal model is also evaluated in this paper comparing the relevance of its results with two other BP retrieval methods. The first corresponds to a tool called "*BeMantics*" (Behavioral Semantics Business Process Retrieval) (Figueroa and Corrales, 2012), which allows queries to be run by way of semantic, structural, and behavioral characteristics. The second involves the implementation of the *A** algorithm (Grigori et al., 2008), which only compares BPs structurally. For the comparison of the Multimodal model against the *BeMantics* tool and the A* algorithm, the same test set comprising 100 BPs was used, and the queries that were run enabled the multimodal option of the proposed model to be compared against the structural features of the other two methods, because these gave the greatest relevance values.

The following describes in general the two methods compared with the multimodal model: *BeMantics*: this method comprises two main modules;

an indexing module based on behavioral semantics, and a structural and semantic analysis module (Figure 12).

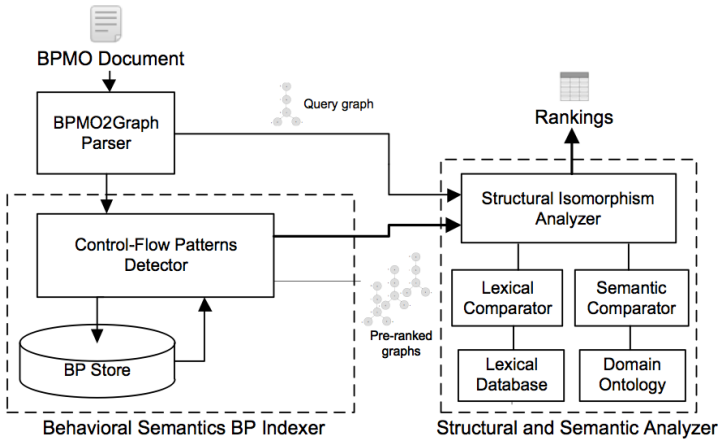


Figure 12: Architecture of *BeMantics* tool (Taken from (Figueroa and Corrales, 2012))

The indexer module seeks sub-structures within a query BP represented as a BPMO (Business Process Modeling Ontology) document. This document is converted to a graph-based formal model and delivered to the control-flow patterns detector module, which encounters a set of control flow patterns in the query model and proceeds to retrieve a list of BPs from the repository (or BP store) containing a set of similar patterns (pre-ranked graphs).

Next, the list of pre-ranked graphs is entered into the structural and semantic analyzer, which performs a set of editing operations (replace or delete nodes, delete or add edges) in order to ensure that the pre-ranked graphs from the repository are as similar as possible to the query graph.

The editing operation values are predefined by the users, but in the specific case of the node replacement operation, the cost is calculated by a linguistic analyzer, which computes a lexical distance or semantic distance value. The lexical distance is evaluated by the similarity between words defined in a lexical database called *WordNet* (Rinaldi, 2014); the semantic distance is calculated by counting the number of hops between concepts in two domain ontologies which contain the concepts in the telecommunications environment.

A*: This algorithm is focused on making structural comparisons between graphs to find a level of similarity between them. To do this, it performs the following steps:

- Given a set of graphs $(G_1, \ldots, G_n)$ and a query input graph G_q, the algorithm starts with the off-line recursive decomposition of each of the graphs in small subgraphs until they represent a single vertex. All these subgraphs are stored in a compact data structure, in order to reduce the runtime.

- Similarity is calculated by taking each of the matching subgraphs to form the distance measure, which is given by the greatest common subgraph between a graph from the set and a graph from the query.

Figure 13 presents the results of the comparison of relevance between the *Multimodal model*, *BeMantics* and *A**. Comparing the precision graded, *BeMantics* obtained the highest average precision value (92.85%), indicating that it reduced the number of false positives to only 7.15%. This is because *BeMantics* carries out a comparison between each node of the query BP and each node in the business process repository using a graph isomorphism algorithm that facilitates higher precision values.

However, the multimodal model obtains similar values in the results by combining the sequential and linguistic criteria present in the BPs, which are processed using text extraction algorithms, being able to reduce the probability of retrieving irrelevant results (false positives) by 8%. On the other hand, the *A** algorithm produces the lowest level since it uses only graph isomorphism algorithms as a function of similarity.

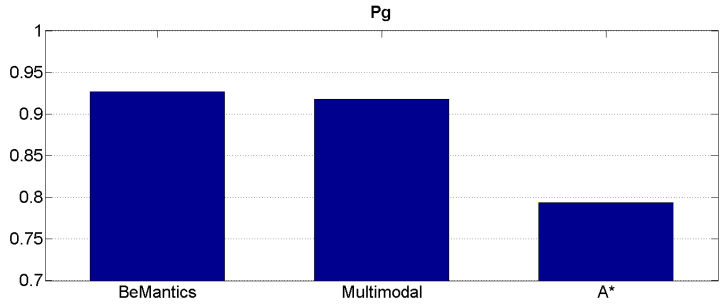


Figure 13: Precision graded average for *BeMantics*, *A**, and the *Multimodal* proposals

As for Recall graded (Figure 14), the tools scored low values 13.9% for *BeMantics*, 16.6% for *A** and 30.6% for *Multimodal*. Although these values are low, Multimodal outperforms *BeMantics* by 120% and *A** by 84%.

This is because the three tools have limited rankings, with a maximum of ten BPs in accordance with many IR applications, especially in web environments, where users focus on only the first eight or ten results returned in a set of responses (Petrelli, 2008). Therefore, *BeMantics*, *A**, and *Multimodal* can return false negatives (i.e. miss relevant BPs in the ranking), while at the same time their greater precision tends to reduce the number of false positives.

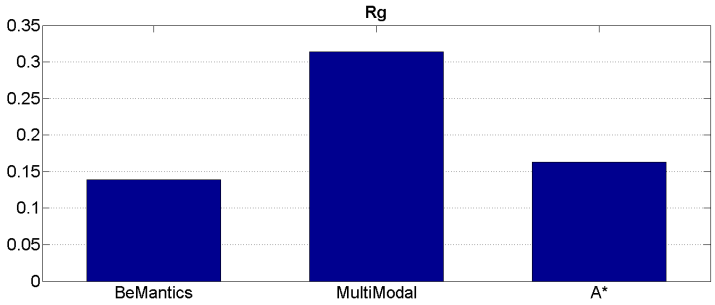


Figure 14: Recall graded average for *BeMantics*, *A**, and the *Multimodal* proposals

With regard to the effectiveness of the tools, these are characterized by the performance of the classification in the results rankings. For this, the F-measure (Figure 15) allows us to look at the harmony of the Pg and Rg results. The comparison tools obtained the following values: *BeMantics* reached 24.14% and *A** 26.31%. This shows that the order and the quality of the rankings of the results have low degrees of harmony.

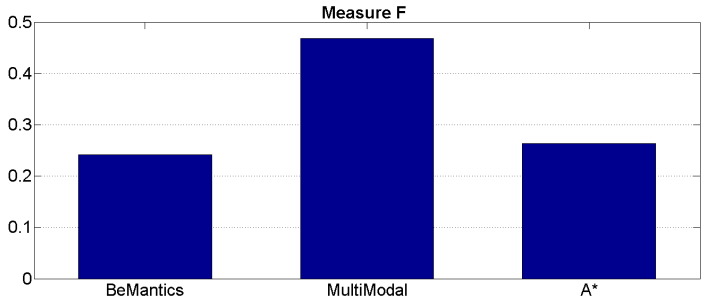


Figure 15: *F-measure* average for *BeMantics*, *A**, and the *Multimodal* proposals

In contrast, the 46.81% achieved by the *Multimodal* shows that the proposed model obtains a higher performance in the classification of retrieved

results in the conducted queries. On average *Multimodal* model achieved an improvement of 94% compared with *BeMantics* and 78% compared with *A**, in ranking classification.

Figure 16 shows the *ANDCG* measure, which indicates that the multimodal model demonstrates a better quality in ranking compared to *BeMantics* and *A**, which fail to locate as many relevant elements at the beginning of the ranking as does the *Multimodal* model, which outperforms *BeMantics* by 50% and *A** by 260%.

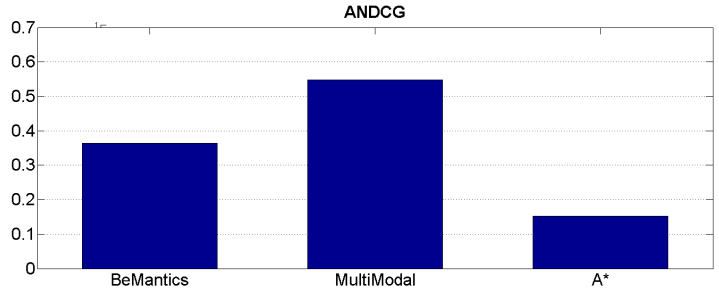


Figure 16: Average ANDCG measure for *BeMantics*, *A**, and the *Multimodal* proposals

As explained above, the difference in the *ANDCG* and *GenAveP'* measures (Figure 17) is that the latter features a discount factor that penalizes with a higher value the items returned at the end of the ranking, in this case, as before, the *multimodal* model obtains the highest values that range from 64% compared to *BeMantics* and 53% against *A**.

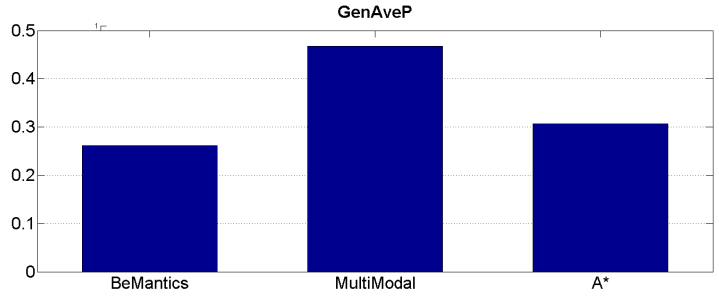


Figure 17: Average GenAveP' measure for *BeMantics*, *A**, and the *Multimodal* proposals

4.5. Evaluation of performance

Comparing the performance analysis run on the BP retrieval tools it was found that *Multimodal* showed the best time response for all business

processes (with different numbers of nodes), outperforming in a range from 588 times faster than *BeMantics* and 555 times faster than *A** for a BP with 10 nodes. As regards the BPs with 20 to 30 nodes, Multimodal is 2500 times faster than *BeMantics* and 312 times faster than *A**. The number of times is calculated based on the time (Table 6) used in the query execution with a BP with a determined number of nodes.

Tool	**10 Nodes Runtime (ms)**	**20 Nodes Runtime (ms)**	**30 Nodes Runtime (ms)**
BeMantics	10000	100000	1000000
A*	9450	12500	18360
Multimodal	17	40	60

Table 6: *Multimodal* vs *BeMantics* and *A** performance comparison and analysis

The above takes into account that while *BeMantics* runs a comprehensive structural comparison based on an error correction mechanism that seeks to make a BP repository as similar as possible to the query BP, *A** performs structural analysis based on subgraphs to find the BPs that have the most subgraphs to retrieve them, while the Multimodal model approach performs a simple text extraction algorithm (complexity $O(n2)$) that takes just a few milliseconds to retrieve the results (17-60 ms) in the BPs with 10-30 nodes.

5. Conclusions and Future Work

In this paper a multimodal model to BP search is proposed, based on information retrieval methods that integrate structural and linguistic components. The relevance of the multimodal model is evaluated with precision graded (Pg), recall graded (Rg), *F-measure*, *ANDCG* and *GenAveP'* measures. In addition, the proposal is compared with other BP search tools, based on structural, semantic and behavioral characteristics, *BeMantics*, and an implementation of the *A** algorithm.

Analysis of relevance for the multimodal model shows that the combination of linguistic and codebook (structural) information as search elements allows for a high level of accuracy and low execution time because it does not require complex and comprehensive graph isomorphism algorithms as in the case of the *BeMantics* and A* tools.

Comparing Pg values, it is found that the semantic method used by *BeMantics* obtained the highest value. However, the multimodal model ob-

tained similar results in a shorter execution time. Furthermore, the levels of quality ranking for multimodal show that the model recovers better positioned and better organized BPs in the results list. Moreover, considering the overall results it is clear that the multimodal (linguistic and codebook) search model achieved the highest values while still giving low execution times.

The multimodal search model shows that textual representation of the linguistic and structural components of the BPs produces better results than the models that represent the BPs in the form of graphs. Thus, the proposed multimodal model permits more precise results as it retrieves only BPs with a high degree of similarity, combining components that make up the BP models, which increases the quality of the ranking in the queries conducted by the user.

As future work, the authors aim to complement the multimodal model with specific domain ontologies that allow the addition of semantics to the searches for more accurate results, and in addition, to incorporate a clustering algorithm that allows groupings of BPs based on the topics found in the codebook and on the linguistic components within the BPs.

Acknowledgement

This work is partially supported by a Ph.D grant from the department of science, technology and Innovation (COLCIENCIAS) (Grant call: 511-2010). Furthermore, we want to thank the *Information Systems Group* and the *Management School* of the Federal University of Rio Grande do Sul (Brazil) because of their collaboration to evaluate our test collection.

References

van der Aalst, W., 2013. Business process management: a comprehensive survey. ISRN Software Engineering 2013, 1–37. doi:10.1155/2013/507984.

van der Aalst, W., van Hee, K., van Werf, J., Verdonk, M., 2010. Auditing 2.0: Using process mining to support tomorrow's auditor. Computer 43, 90–93. doi:10.1109/MC.2010.61.

Awad, A., Polyvyanyy, A., Weske, M., 2008. Semantic Querying of Business Process Models. doi:10.1109/edoc.2008.11.

vom Brocke, J., Schmiedel, T., Recker, J., Trkman, P., Mertens, W., Viaene, S., 2014. Ten principles of good business process management. Business Process Management Journal 20, 2–2. doi:10.1108/BPMJ-06-2013-0074.

Caicedo, J.C., BenAbdallah, J., González, F.A., Nasraoui, O., 2012. Multimodal representation, indexing, automated annotation and retrieval of image collections via non-negative matrix factorization. Neurocomputing 76, 50–60. doi:10.1016/j.neucom.2011.04.037.

Chang, H.H., Wang, I.C., 2011. Enterprise Information Portals in support of business process, design teams and collaborative commerce performance. International Journal of Information Management 31, 171–182. doi:10.1016/j.ijinfomgt.2010.05.010.

Corradini, F., Polzonetti, A., Re, B., Falcioni, D., 2010. An eclipse plug-in for formal verification of bpmn processes, in: Communication Theory, Reliability, and Quality of Service (CTRQ), 2010 Third International Conference on, pp. 144–149. doi:10.1109/CTRQ.2010.32.

Dijkman, R., Dumas, M., van Dongen, B., Käärik, R., Mendling, J., 2011a. Similarity of business process models: Metrics and evaluation. Information Systems 36, 498–516. doi:10.1016/j.is.2010.09.006.

Dijkman, R., Gfeller, B., Küster, J., Völzer, H., 2011b. Identifying refactoring opportunities in process model repositories. Information and Software Technology 53, 937–948. doi:10.1016/j.infsof.2011.04.001.

Dijkman, R., Rosa, M.L., Reijers, H.A., 2012. Managing large collections of business process modelsCurrent techniques and challenges. Computers in Industry 63, 91–97. doi:10.1016/j.compind.2011.12.003.

Eid-Sabbagh, R.H., Kunze, M., Meyer, A., Weske, M., 2012. A platform for research on process model collections, in: Mendling, J., Weidlich, M. (Eds.), Business Process Model and Notation. Springer Berlin Heidelberg. volume 125 of *Lecture Notes in Business Information Processing*, pp. 8–22. doi:10.1007/978-3-642-33155-8_2.

Figueroa, C., Corrales, J.C., 2012. Business Process Retrieval based on Behavioral Semantics. Revista EIA , 105–120.

Fonteyn, M.E., Vettese, M., Lancaster, D.R., Bauer-Wu, S., 2008. Developing a codebook to guide content analysis of expressive writing transcripts. Appl Nurs Res 21, 165–168. doi:10.1016/j.apnr.2006.08.005.

Gacitua-Decar, V., Pahl, C., 2009. Automatic Business Process Pattern Matching for Enterprise Services Design. doi:10.1109/services-2.2009.28.

Grigori, D., Corrales, J.C., Bouzeghoub, M., 2008. Behavioral matchmaking for service retrieval: Application to conversation protocols. Inf. Syst. 33, 681–698. doi:10.1016/j.is.2008.02.004.

He, D., Ritchie, G., Lee, J., 2008. References to graphical objects in interactive multimodal queries. Knowledge-Based Systems 21, 617–628. doi:10.1016/j.knosys.2008.03.023.

Kekäläinen, J., Järvelin, K., 2002. Using graded relevance assessments in ir evaluation. J. Am. Soc. Inf. Sci. Technol. 53, 1120–1129. doi:10.1002/asi.10137.

Koschmider, A., Hornung, T., Oberweis, A., 2011. Recommendation-based editor for business process modeling. Data & Knowledge Engineering 70, 483–503. doi:10.1016/j.datak.2011.02.002.

Kunze, M., Weidlich, M., Weske, M., 2011. Behavioral Similarity A Proper Metric , in: Rinderle-Ma, S., Toumani, F., Wolf, K. (Eds.), Business Process Management. Springer Berlin Heidelberg. volume 6896. chapter 15, pp. 166–181. doi:10.1007/978-3-642-23059-2_15.

Küster, U., König-Ries, B., 2010. Measures for Benchmarking Semantic Web Service Matchmaking Correctness. doi:10.1007/978-3-642-13489-0_4.

La Rosa, M., Reijers, H.A., van der Aalst, W., Dijkman, R.M., Mendling, J., Dumas, M., García-Bañuelos, L., 2011. APROMORE: An advanced process model repository. Expert Systems with Applications 38, 7029–7040. doi:10.1016/j.eswa.2010.12.012.

Lazaridis, M., Axenopoulos, A., Rafailidis, D., Daras, P., 2013. Multimedia search and retrieval using multimodal annotation propagation and indexing techniques. Signal Processing: Image Communication 28, 351–367. doi:10.1016/j.image.2012.04.001.

Li, C., Reichert, M., Wombacher, A., 2011. Mining business process variants: Challenges, scenarios, algorithms. Data & Knowledge Engineering 70, 409–434. doi:10.1016/j.datak.2011.01.005.

Manning, C.D., Raghavan, P., Schütze, H., 2008. Introduction to information retrieval. volume 1. Cambridge university press Cambridge. URL: http://nlp.stanford.edu/IR-book/.

Ordóñez, H., Corrales, J.C., Cobos, C., Krug Wives, L., Thom, L., 2014. Collaborative evaluation to build closed repositories on business process models, in: International Conference on Enterprise Information Systems (ICEIS), pp. 311 –318. doi:10.5220/0004881203110318.

Pedronette, D.C.G., Almeida, J., da S. Torres, R., 2014. A scalable re-ranking method for content-based image retrieval. Information Sciences 265, 91 – 104. doi:http://dx.doi.org/10.1016/j.ins.2013.12.030.

Pérez-Castillo, R., de Guzmán, I.G.R., Piattini, M., Weber, B., Places, A.S., 2011. An empirical comparison of static and dynamic business process mining, in: Proceedings of the 2011 ACM Symposium on Applied Computing, pp. 272–279. doi:10.1145/1982185.1982249.

Petratos, P., 2007. Information Retrieval Systems: A Human Centered Approach. Interdisciplinary Journal of Information, Knowledge, and Management 2, 17–32.

Petrelli, D., 2008. On the role of user-centred evaluation in the advancement of interactive information retrieval. Information Processing & Management 44, 22–38. doi:10.1016/j.ipm.2007.01.024.

Reijers, H.A., Freytag, T., Mendling, J., Eckleder, A., 2011. Syntax highlighting in business process models. Decision Support Systems 51, 339–349. doi:10.1016/j.dss.2010.12.013.

Reimerink, A., García de Quesada, M., Montero-Martínez, S., 2010. Contextual information in terminological knowledge bases: A multimodal approach. Journal of Pragmatics 42, 1928–1950. doi:10.1016/j.pragma.2009.12.008.

Revuelta-Martínez, A., Rodríguez, L., García-Varea, I., Francisco, M., 2012. Multimodal interaction for information retrieval using natural language.

Computer Standards & Interfaces 35, 428–441. doi:10.1016/j.csi.2012.11.002.

Rinaldi, A.M., 2014. A multimedia ontology model based on linguistic properties and audio-visual features. Information Sciences 277, 234 – 246. doi:http://dx.doi.org/10.1016/j.ins.2014.02.017.

Smirnov, S., Weidlich, M., Mendling, J., Weske, M., 2012. Action patterns in business process model repositories. Computers in Industry 63, 98–111. doi:10.1016/j.compind.2011.11.001.

Wang, J., Jin, T., Wong, R., Wen, L., 2014. Querying business process model repositories. World Wide Web 17, 427–454. doi:10.1007/s11280-013-0210-z.

Xie, L., Natsev, A., Tesic, J., 2007. Dynamic multimodal fusion in video search, in: Multimedia and Expo, 2007 IEEE International Conference on, pp. 1499–1502. doi:10.1109/ICME.2007.4284946.

Zhang, J., Wei, Q., Chen, G., 2014. A heuristic approach for -representative information retrieval from large-scale data. Information Sciences 277, 825 – 841. doi:http://dx.doi.org/10.1016/j.ins.2014.03.017.

Available online at www.sciencedirect.com

ScienceDirect

Knowledge-based Systems 00 (2015) 1–12

Knowledge-based Systems

www.elsevier.com/locate/procedia

Improving Business Process Retrieval Using Categorization and Multimodal Search

Cristhian Figueroa[a,b,*], Hugo Ordóñez[b,d], Juan-Carlos Corrales[b], Carlos Cobos[c], Leandro Krug Wives[e], Enrique Herrera-Viedma[f,g]

[a]*Software Engineering Group, Politecnico di Torino, Corso Duca degli Abruzzi, 24 - 10129, Turin, Italy*
[b]*Telematics Engineering Group, Universidad del Cauca, Calle 5 No. 4 - 70, Popayán, Colombia*
[c]*Information Technology Research and Development Group, Universidad del Cauca, Calle 5 No. 4 - 70, Popayán, Colombia*
[d]*Engineering Faculty, Universidad de San Buenaventura, Cali, Colombia*
[e]*Informatics Institute, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil*
[f]*Department of Computer Science and Artificial Intelligence, Universidad de Granada, Avda. del Hospicio, s/n C.P. 18071, Granada, Spain*
[g]*Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia*

Abstract

Enterprises tend to standardize and adapt their operations through Business Processes in order to reuse them for new functional requirements. However a disorganized growth of repositories of Business Processes have made difficult to search or recover Business Process models. In this paper, we propose an approach to retrieve Business Processes that, unlike other approaches, reduces the search space through an automatic semantic categorization approach and applies a multimodal search strategy to rank Business Process based on structural and textual features. The approach proposed was tested in an evaluation over a closed repository built collaboratively by 20 expert evaluators. Firstly, the experts were asked to rate the categories assigned by our approach to each Business Process in order to assess our results against user perspective. Subsequently, these results were used to estimate the ranking quality of the categorizer applying the Normalized Discounted Cumulated Gain measure. Later, the experts were asked to compare six queries against Business Processes stored in the closed repository in order to obtain a set of relevant Business Processes for each query. Finally, well known graded measures, i.e., precision (Pg), recall (Rg) and F-Measure, were applied to evaluate the relevance of the results presented by the multimodal approach. The results obtained demonstrate the effectiveness of our approach for categorizing and retrieving Business Processes.

Keywords: Business Process Categorization, Semantic Categorization, Multimodal Search.

1. Introduction

Reusing software components as Web Services or Business Processes (BP) helps companies to deploy new and value-added services in order to attract and retain customers. In this way, these companies can

*Corresponding author: Phone: +39 011090 7087
Email addresses: cristhian.figueroa@polito.it (Cristhian Figueroa), hugoordonez@unicauca.edu.co (Hugo Ordóñez), jcorral@unicauca.edu.co (Juan-Carlos Corrales), ccobos@unicauca.edu.co (Carlos Cobos), wives@inf.ufrgs.br (Leandro Krug Wives), viedma@decsai.ugr.es (Enrique Herrera-Viedma)

introduce a competitive differentiation and increase the level of service offered to their customers [1].

This requires the tasks within each company to be organized around business functions such as marketing, sales, production, finance and customer service that usually run independently [2]. Moreover, the different functions of companies can be described by models which represent BP procedures or activities that collectively reach a common business goal and define roles and functional relationships [3].

Because of the volume and importance of this information contained in BP models, these are commonly stored in repositories that can contain hundreds or even thousands of models. Therefore, these models are available to be reused and extended with new features in order to accomplish business requirements.

BP repositories have grown in a significant and disordered manner [4], without taking care about classification and organization regarding the purposes of each BP [5]. For this reason, BP management has become a complex and time-consuming task [6]. Additionally, other problems arise such as the complexity for finding BPs with specific functionalities, and keeping coherence between different BPs versions when various users try to edit the same BP [7].

With regard to the aforementioned problems, several authors have developed mechanisms to search and retrieve BP that can be used as starting point for creation of new BP [8]. The aim of these mechanisms is to find a set of BP from a repository that are similar to a query expressed in different ways as for example: a set of keywords, a complete BP or a fraction thereof. This paper proposes an approach for automatic categorization of BPs, which can help to organize repositories and reduce the search space to a subset of BPs belonging to the same category. This approach forms a structure of categories that links each BPs to the context in which they were created within the organization. Searching BPs based on a specific category let users to obtain lists with coherent results regarding the textual information within related functionalities. Furthermore, automatic categorization approaches have been used in different fields, for example managers for software quality models [9], recommender systems for digital libraries [10], textual collections [11, 12] among others.

The following contributions are highlighted in this paper: 1) an automatic semantic categorization mechanism and 2) a multimodal search model that integrates textual and structural information to rank BP retrieved for a specific category. To validate our categorization approach we have integrated it with a multimodal search model (that search BPs based on textual and structural information) and conducted a comparative evaluation of the results of Multimodal with categorization and without categorization. The paper is structured as follows: section 2 presents the architecture of our approach, section 3 describes experiments and results obtained, section 4 summarizes related works, and section 5 discuss the main conclusions and future directions.

2. Architecture of the Approach for Automatic Categorization and Recovering of BPs

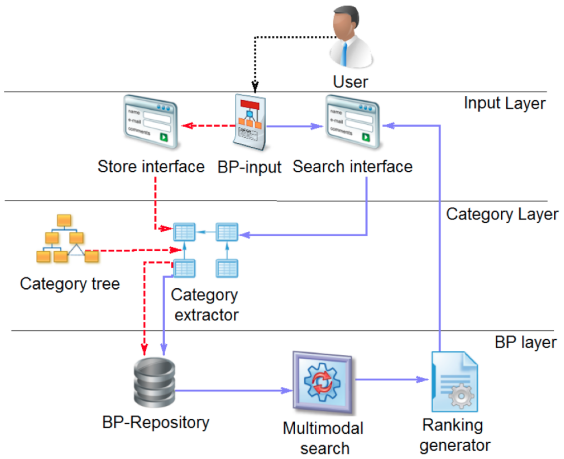


Fig. 1. Architecture of the categorization and multimodal search approach

This section describes the architecture of the approach for automatic categorization of BPs that stores categorized BPs and allows users to search them based on structural and textual features (based on a multimodal model that is described in [13]). Each BP is stored in a repository and classified according to sets of categories obtained from its activities. In our approach, a user (a BP modeler or developer) can enter as input a BP (*BP-input*) to be stored (indexed) or used as query to search for similar BPs depending on user choice.

Figure 1 shows the architecture of the proposed approach, where dashed lines represent an indexing phase, and solid lines a search phase. In the indexing phase, a BP-input is preprocessed to detect the categories to which each of its activities belongs. Then, a set of high-level categories (more general categories) for sets of activities is found. Finally, the BP and its high-level categories are stored in a repository. In the search phase, similarly to the storage phase a set of high-level categories for a BP-input is detected. However, in the search phase this is done in order to find BPs in the repository containing a similar set of categories as the BP-input.

In this way, our approach can be considered as a filter that can be used for reducing the search space that a further searching algorithm will use to produce a ranking of BPs according to their similarity with the BP-input. For example, in this paper we have used a multimodal search algorithm that uses structural and textual information of BPs. In Figure 1, the architecture of the proposed approach follows a three-tier model, which consists of an input layer, a category layer, and a BP layer.

2.1. Input Layer

This layer provides interfaces to interact with users. It receives a BP-Input modeled in BPMN (Business Process Management Notation). Depending on the user choice, the BP-Input can be stored or used as query to find similar BPs. In the first case, a Store Interface allows the user to enter the BP to be stored in the repository. In the second case, a Search Interface gives the user the possibility for entering a BP as query to retrieve similar BPs from the repository. Additionally, this interface shows the final rank of similar BPs.

2.2. Category Layer

Category Tree A category tree is a taxonomy of topics hierarchically organized, which can help users to better specify their search requirements and contribute at improving resources retrieval [14]. In our approach, the aim of the category tree is to classify the activities of a BP. Activities are assigned to one or more specific topics, and further a set of high-level categories covering more specific topics are found to provide a set of categories for each BP. Although in our approach the category tree may be obtained from different category schemas, in this paper we present our experiences with the category schema known as SKOS (Simple Knowledge Organization for the Web) [15], which is an RDF vocabulary for expressing the basic structure and content of concepts schemes. In our case, it can be seen as a large set of categories, hierarchically ordered.

Category Extractor This module extracts categories from the textual information (labels, description) of each activity of a *BP-Input*. These categories are obtained from the category tree and related with each activity of the BP. Furthermore, the *BP-Input* is categorized (labeled) with a set of common high-level categories (generic categories that are common to the categories detected for individual activities of the BP-Input).

This module implements a "categorization algorithm" (see Algorithm 1) that receives a *BP-Input*. The algorithm firstly transforms the *BP-Input* into a graph $GBP = \{N, E, L\}$, where N is a set of nodes that represents: activities, events and logical gates of the BP; E is a set of edges that links these nodes; and L is a set of labels (names or descriptions of activities). Next, labels L are extracted from GBP to form a vector $VL = \{l_1, l_2, \ldots, l_i, \ldots, l_n\}$ where $l_i \in L$ are the labels of activities belonging to the *BP-Input* (Lines 1 - 2). Then, each label ($l_i \in V_l$) is preprocessed to eliminate stop words (obtaining a label l'_i). Next, l'_i is divided in tokens (tokenized) T_i (Lines 3 - 5). Afterwards the algorithm uses two external software tools for extracting the categories. The first one is the lexical database WordNet, and the second one is the semantic annotator *Zemanta* [16].

Algorithm 1 Algorithm for BP Categorization

Require: A *BP-Input*
Ensure: A category set C_{BP} for *BP-Input*
1: $G_{BP} = createGraph(G_{BP})$
2: $V_L = extractActivityLabels(BP\text{-}Input)$
3: **for all** $l_i \in V_L$ **do**
4: $l'_i = removeStopWords(l_i)$
5: $T_i = extractTokens(l'_i)$
6: **for all** $t_k \in T_i$ **do**
7: **if** t_k is a verb **then**
8: $Synsets_k = getVerbSynsets(t_k)$
9: **else**
10: $Synsets_k = getNounSynsets(t_k)$
11: **end if**
12: $relevantTerms_k = computeTF.IDF(Synsets_k)$
13: $H_k = getHyperonyms(relevantTerms_k)$
14: $C_i = addHyperonyms(H_k)$
15: **end for**
16: $HC_i = getCommonHighLevelCategories(C_i)$
17: $C_{BP} = addCategories(HC_i)$
18: **end for**
19: **return** C_{BP}

The *WordNet* database allows the system to obtain the type (name or verb) for each token $t_k \in T_i$, and a set of synonyms ($Synset_k$) also known as synset (Line 7). The same *WordNet* database is used to obtain the text of the lexical definitions (d_k) for each term in the $Synsets_k$ (Line 8). Then, using the semantic annotator *Zemanta* a set of relevant words, from the whole set of lexical definitions for all the words in the Synset, are extracted and linked to semantic concepts (Line 9). Finally, a set of categories (c_k) for the concepts (corresponding to the token t_k) is obtained from the SKOS schema and added to a global set of categories C_g for the whole graph (Lines 10 - 11). Subsequently, a set of common high-level categories HC_i is calculated to obtain more general concepts than concepts in C_g, i.e., super categories which are common to most of the activities of the *BP-Input* (Line 13). Finally, these categories are added to the global set of categories (CBP) for the *BP-Input* (Lines 13 - 16).

2.3. BP Layer

This layer, depending on the user choice, can store or search BPs. In the first case, a BP repository was developed containing physical files of BP and the corresponding categories detected on the category layer. In the second case, a Multimodal Search was implemented to rank BPs according to their similarities with a sub-set of BPs from the repository (those BPs belonging to same category sets as the BP query). Multimodal Search: a Multimodal algorithm [17] allows to search BPs based on textual and structure features of BPs. Linguistics considers textual information (labels and description) attached to different elements of BPs (e.g. activities, interfaces, messages, gates, and events). Structure takes into account basic units of the control flow of BPs to create a set of activities, related in a codebook style [18], i.e. union of two or more control-flow components. These codebooks are formed by the union of n-structural components ruled by the sequentiality of the BP control-flow. Our multimodal search approach is composed of three modules: textual analyzer, structural analyzer and ranking generator.

- *Textual Analyzer*: in this module a matrix named *MC* is generated by extracting labels (e.g., names, types, and descriptions) from activities of BP stored in the repository. These labels or Textual features are then preprocessed applying stemming (Porter's Algorithm [19]) in order to create vectors Ct_i that constitutes the *MC*. In this way *MC* is composed of vectors $Ct_i = \{Ct_{i,1}, \ldots, Ct_{i,j}, \ldots, Ct_{i,m}\}$ for each

BP_i stored in the repository. Each component $C_{i,j}$ represents the weight of a feature j in a BP_i (m is the total number of textual features of BP).

- *Structural Analyzer*: this module creates a MCd matrix composed of codebooks formed by the union of n-structural components ruled by the sequentiality of the control-flow for each BP_i stored in the repository. These components describe transitions of two or more nodes of the BP in the form of text strings representing frequent sequential patterns in the structure BP_i. Codebooks are the base to create a vector $Cd_i = \{Cd_{i,1}, \ldots, Cd_{i,k}, \ldots, Cd_{i,p}\}$ where $Cd_{i,k}$ is the weight of a specific element of BP_i compared to all the elements of BP in the repository; p is the total number of possible transitions of BP_i, and k is an index for a specific codebook stored in the repository. Finally, each vector Cd_i is a row i in the MCd matrix of codebook components.

- *Search Index*: this module creates an index composed of two search spaces: 1) a textual indexing of business functions, and 2) a structural indexing (characterization between types of: tasks, events, and links). These two search spaces are unified in a multimodal structure to create a broader index to have an accurate representation of BP regarding their category set. Search index efficiently stores a conceptual structure named term-index matrix ($MI = \{MC_{i,j} \cup MCd_{i,j}\}$) that is calculated in a similar way as the *Vector Space Model* for information retrieval [20]. In each cell $((i, j))$ MI matrix contains a weight w_{ij} with a value of relevance of each textual/structural component in their lexical root or codebook for each BP_i.

  Values of w_{ij} are calculated based on equation 1 proposed by Salton et al., [21] where $F_{i,j}$ is the observed frequency of a component or codebook j in BPi; $Max(F_i)$ is the greatest frequency observed in BPi; m is the number of BP in the repository; and m_j is the number of BP in which the linguistic or codebook component j appears.

$$W_{i,j} = \frac{F_{i,j}}{max(F_i)} log\left(\frac{m}{m_j + 1}\right) \tag{1}$$

2.3.1. Ranking Generator:

this module generates a ranking according to a conceptual rating (*score*) expressed by equation 2 (used by Lucene library) [22]. Equation 2 is applied in order to return a ranking list of results containing a set of BP with highest conceptual punctuation within the multimodal index.

$$score(q, d) = coord(q, d) * \sum_{t \in q} (tf(t \in d) * idf(t)^2 * norm(t, d)) \tag{2}$$

Where t is a term of a query q; d is a BP_i to be evaluated with q; $tf(t \in d)$ is the term frequency defined as the number of times the term t appears in d so that BP are ranked according to the values of term frequency scores; $idf(t)$ is the number of BP in which term t appears (*inverse frequency*); $coord(q, d)$ is a scoring factor based on a number of query terms found in a queried BP (these BP containing the most query terms are scored highest); and $norm(t, d)$ is the weighting factor in the indexing, taken from w_{ij} in our multimodal index.

Once results are sorted, first 10 are filtered according to the level of similarity found with respect to the BP in the query. We selected only the first 10 results in accordance with many IR applications, especially in web environments, where users focus on only the first eight or ten results returned in a set of responses [23]. In addition to the multimodal model components, the user has forms that correspond to the graphical user interface (GUI), used to add new BP models to the repository, execute each of the search options, and display the lists of the results of each of the searches performed.

3. Experimentation and Results

The evaluation of our proposal was conducted into three phases involving 20 expert evaluators in BPMN modeling and a repository with 100 BP. Phase 1 considered the relevance of the categorization of the BPs

stored in the Repository. Phase 2 addressed the relevance of the whole search process, i.e., the search of BPs plus their categorization. Phase 3 considered performance, i.e., the time needed for indexing and searching BPs.

3.1. Relevance of the categorization approach (indexing phase)

This phase evaluated the relevance of the categories that the algorithm detected for each BP. Firstly, the 20 expert evaluators were asked to review the categories detected for each BP of the repository. To do this, the evaluators were provided with a questionnaire for each BP containing: a BPMN diagram, a description of its functionality, and a set of categories detected by our algorithm. The questionnaire used a *Likert* scale-based approach where each evaluator was asked to choose a value from 1 5 to issue a relevance judgment about the categories detected for each BP. In this scale, values above 3 means that a category is relevant to the BP evaluated.

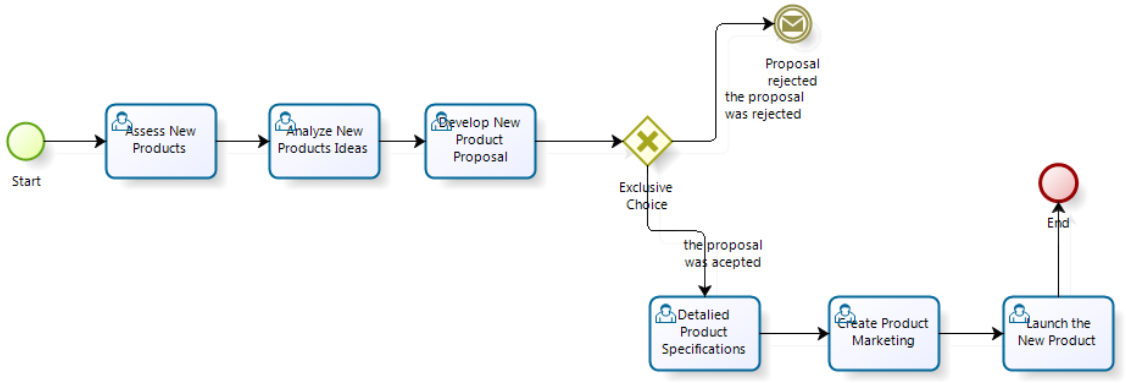


Fig. 2. Example of a BP

Therefore, for each BP we sum the number of times the users selected scores above 3 and divide it to the total number of evaluations. Finally, we calculated the average value for the 100 BPs and found that 73% of the categories detected were relevant for these users. For example, the BP presented on Figure 2 contains 6 activities, a gate and three events (start, end, and send message). In this example the category extractor extracted the categories shown in Table 1 with their scores.

Category	**Score**
Business	0.94
Products	0.92
Business terms	0.86
Sales	0.76
Goods	0.62
Commodity markets	0.47

Table 1. Categories extracted from the example of Figure 2

3.2. Relevance of the search of BPs

In this phase evaluators manually compared a set composed of 6 BP ($Q = \{BP_1, BP_2, \ldots, BP_6\}$) used as queries against the total number of elements of the BP repository. Each evaluator compared the BPs of the repository with each query according to a scale based on the DoM (Degree of Match) model proposed by [24]. In this model, the relevance degree can be assigned to weights of relevance: very relevant (1), relevant (0.75), somewhat relevant (0.50), little relevant (0.25) and not relevant (0). Therefore, the weight for each degree of relevance is the set $w = 1, 0.75, 0.50, 0.25, 0$. Subsequently, an ideal ranking for each

query (and for all users) is generated ordered decreasingly based on the total relevance level for each BP. The total relevance level is defined by equation 3. It is necessary to create an ideal ranking with the items considered as relevant by the evaluators for each query, which is sorted from highest to lowest depending on the relevance level (nr), achieved in manual evaluation. Then, the resulting list generated by this BP searching mechanism is compared to the resulting list considered as relevant by the judges on that query. The total relevance level is defined by equation 3:

$$nr = \frac{1}{n}\sum_{1}^{n} w \tag{3}$$

Where n is the number of evaluators and w is the weight assigned by them for each BP evaluated.

In order to verify the relevance of our results, the rankings generated by our approach (one for each query) are compared with the ideal rankings for each query generated by the evaluators. To do this, graded relevance metrics (Pg, Rg and $F-Measure$) were used, which provide a classification (T_i) for each BP of the results. Hence, these BP are considered similar to a query (Q) according to different levels of significance. The metrics were presented and improved by Küster and König-Ries [25] and were initially defined to quantify the quality of the rankings produced by Web services retrieval tools [26]. Nevertheless, these metrics are fully applicable to the BP recovery domain. Furthermore, the ranking generated by the evaluators and the ranking automatically generated by the proposed approach for BP retrieval were evaluated using the measure $A(Rq)$ proposed by [27]. The A measure was used to determine the degree of coincidence of the position of the BPs in the rankings evaluated.

Following, a discussion of results obtained applying our BP Categorization and Multimodal Search approach to find BPs in the repository that are similar to a BP query is presented. For this, it is necessary to know an outcome list with items considered as relevant by expert evaluators for each query, which is sorted from highest to lowest depending on the relevance level (nr). Then, rankings generated by our approach are compared to a ranking considered as relevant by expert evaluators for diverse queries.

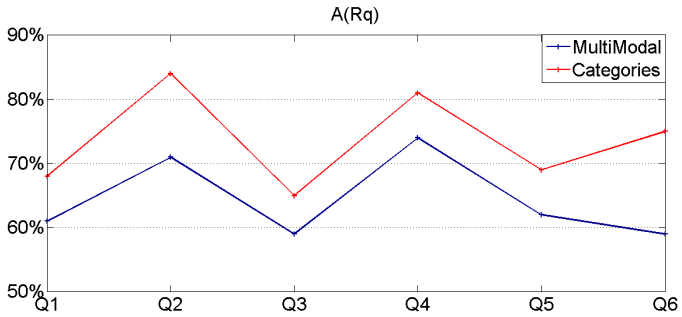


Fig. 3. Ranking concordance for each query ($A(Rq)$)

Figure 3 shows the level of agreement $A(Rq)$ between the ideal ranking generated by the evaluators, and the automatic ranking generated by our method. It worth noting that for each query our approach (the categorization plus multimodal) generated rankings that match considerably with those generated by humans (ideal rankings). For example, in query 2 ($Q2$) the similarity of the ranking for the categorization method scored 84%, while the multimodal model (without categorization) scored 73%. Overall average similarity ranking (considering all queries) for the categorization approach was 74% and for the multimodal approach 66%, indicating an increase in the quality of the rankings generated once the method of categorization was applied in the repository. This is because our method retrieves those BP belonging to the same category, avoiding not relevant processes in the list of results, which were not interesting for the user's query.

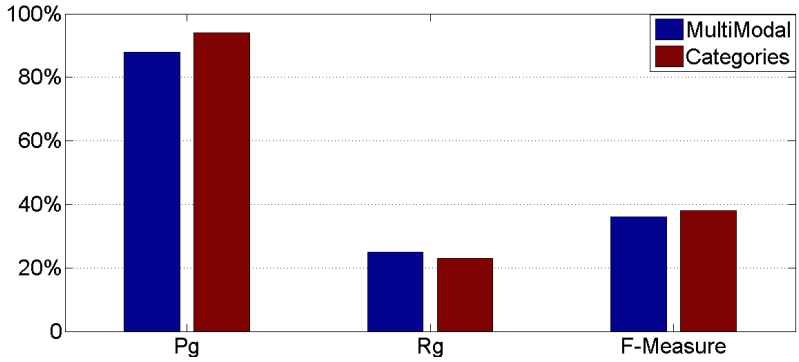


Fig. 4. Measures evaluated

Figure 4 shows the levels of Precision (*Pg*), Recall (*Rg*), and F-Measure scored by our categorization approach. With regard to *Pg*, the categorization approach scored 94% while multimodal model scored 87%. This result demonstrate that both approaches scored high values of *Pg*, nevertheless the categorization approach was slightly higher indicating that it reduced the number of false positives because it reduced the number of results showing only results that belong to the same category. Moreover, the multimodal approach retrieves more false positives because it is only based on structural and textual similarities but not on the categories to which the BP belongs.

With regard to *Rg*, the categorization approach scored 23% and the multimodal approach 25%. These values of *Rg* are low because the multimodal level is limited to show only the first 10 results, and additionally because the categorization approach filters the results showing only those that belong to the same category (which usually did not exceed 6 - 8 results). The multimodal approach was limited to 10 results because in the domain of Web search is commonly known that users focus their attention to only the first portion of results [26].

Finally, the armory of the results for both measures (*Pg* and *Rg*) is synthesized in the F-Measure. In this measure, the category approach scored 38% and multimodal 35%. These low values of the F-Measure show that there is still a need to increase the Rg in order to reduce the number of false negatives (i.e. the number of relevant BP that were not retrieved).

Measure	**Pg**		**F-Measure**	
Query	**Categories**	**Multimodal**	**Categories**	**Multimodal**
Q1	91	83	34	29
Q2	97	89	42	39
Q3	89	82	32	32
Q4	96	92	41	38
Q5	95	91	40	38
Q6	96	85	39	34

Table 2. Pg and F-measure used in *Wilcoxon* signed-rank test

In order to evaluate the statistically significance of the results shown in Figure 4, we applied the Wilcoxon signed-rank test over results reported by each query (see Table 2). In these tests, the categories-based approach scores improve results of *Pg* over multimodal approach with a confidence of 95%. With regard to the *F-Measure*, the categories-based also obtained better results with a confidence of 90%.

3.3. Evaluation of Performance

In this phase we evaluated the execution time for the process of categorizing BP (classify the BPs of the repository in different categories), as well as the execution time for queries with categorization and without categorization.

The categorization process lasted 1800 sec on average on each BP. This is because the categorization uses external components such as *WordNet* to recover the definitions of the keywords extracted from the activities of the BP; and the *DBpedia*[1] endpoint where the semantic categories were obtained. Nevertheless, we are working on download a dump of the category schema *SKOS* in order to execute it locally and improve the performance of the whole categorization process.

Method	**Time(ms) for 10 Nodes**	**Time(ms) for 20 Nodes**	**Time(ms) for 30 Nodes**
Multimodal	58	62	69
Categories	38	41	44

Table 3. Execution time for the Multimodal model vs multimodal with categorization filtering

Table 3 shows the execution time for the multimodal approach without categorization, and the multimodal approach after the BPs were filtered by categories. The execution time was calculated taking into account queries (BPs) with different number of nodes (10, 20, and 30 nodes). As it can be seen, the execution time achieved by the multimodal model after the categorization approach is reduced almost a 20 ms in each case. This was expected because the number of BPs that the multimodal model needed to compare after the filtering the BPs by categories is less than before (only those BPs that belongs to the same set of categories as the query BP).

4. Related Works

BP categorization is considered as a good practice in BPM (Business Process Management), because of the benefits that can be derived from the rigorous analysis of business functions and its activities [28]. This principle has motivated some proposals where categorization is based on manual descriptions that explain how final users (BP modelers) can use each activity of the BP in different granularity levels [29, 30, 31], and taking into account users' needs when they interchange information through various organizational functions represented in BP models [31]. Additionally, other proposal presented in [8, 29] describes a tool that allow to categorize BP based on a collaborative environment which aim is to classify BP taking into account collaborative descriptions of specific activities. Web Services (WS) as well as BP are widely involved in various companies operations such as finance, marketing, entertainment and others. As consequence the number of WS and BPs, available on companies repositories as well as on the Internet, is continuously growing and hinders the search for them.

In this regard, different alternatives based on categorization of these items have been proposed in order to administrate and facilitate their search and retrieval. Categorization mechanisms assign WB and BPs to suitable categories according to their functionality of other features.

For example, [32], describes a semantic categorization mechanism for WS based on their WSDL (Web Service Description Languages) descriptions. The mechanism was designed for categorizing WS based on: a) domain ontologies to take into account knowledge from various domains (e.g. tourism, business, transport etc.), and b) operation ontologies to analyze the structure of the WS. The categorization mechanism works in four phases: 1) extraction of information represented as tokens, 2) preprocessing of tokens in order to identify operations and inputs and outputs, 3) features reduction determined by a impact criteria measure, and 4) classification of WS performed via a LMT algorithm that combines regression trees, and probability analysis for each class.

In other hand [33] proposed a categorization algorithm for WS based on the model for vectorial representation commonly used in information retrieval. The algorithm considers the features found in the WSDL descriptions such as structure and semantic information. The algorithm was designed in three phases. 1) Preprocessing to remove XML labels and to extract information about service name, service description,

[1] `http://dbpedia.org/sparql`

and operations; 2) Construction of feature vectors with each of one the elements extracted in the previous step; and 3) WSs categorization through two evaluation functions: similarity between test documents and category vectors determined by the term frequency inverse document frequency (tf/idf) function for each category in order to specify categories with higher level of similarity. Moreover [34] presents an approach to categorize WS according the whole set of their structural elements, namely operations and input/output parameters according to a specific application domain. The classification process is defined in five phases: 1) extraction of the information of the WS; 2) division of the information in different keywords; 3) detection and removal of stop words and semantic enrichment of the most significant words with WordNet synonyms; 4) formation of term vectors according to a weight that determines the relevance of the term for the WS; 5 all produced vectors are provided as training data to a learning classification algorithm (e.g. Naïve Bayes) for the generation of an accurate classification model.

Proposals for searching BP use diverse sets of elements or data types contained in BP models to compare or find similar BP. For example, these taking into account the name or description of activities, events and logical gates are known as linguistic-based approaches. Other approaches that can also be considered as linguistic-based are these using techniques from the IR (Information Retrieval) field, such as the vector space model (VSM) with a term-frequency value (TF) for each BP and the cosine distance, to rank relevant results [35]. Other approaches are based on association rules that analyze the historical execution logs of BP to detect related descriptions to the activities of the BP through a domain ontology, and to identify common patterns of activities. These approaches to generate the results list use a heuristic component to determine the frequency of occurrence of the detected patterns [36]. In other hand, some proposals are based on genetic algorithms to search BP. These proposals transform BP to a formal representation (e.g., graphs or state machines) and integrate additional data to the search such as: number of inputs and outputs for each activity, edge labels, and node names or descriptions. Even though these proposals score high precision values, their main drawback is they are time-consuming [37, 38]. Finally, other approaches are centered on search BP on repositories, where BP are stored with annotations in XML files. To search, these proposals use a query language named IPM Process Query Language (IPM-PQL), which allows specific queries as for example searching for BP containing a specific activity, transition or link between activities [7, 39, 40]. As can be seen, the proposals in the categorization are based on manual information provided by users when the BP is modeled. Regarding proposals for search BPs are limited to the matching of inputs/outputs, taking as basis the textual information of the BP's elements. These proposals do not take into account: control-flow, behavior, structure, and type of activities, gates and events. Unlike the studied proposals, our approach is focused to develop a categorization schema that can help to reuse, discovery and extend BP models. Categorization provides also some advantages because limits the search space only to BP of similar category sets avoiding searching for BP in completely different domain. The categorization schema is then followed by a multimodal search, which offers effective and precise results by integrating in the search process the structural and behavioral features of BP.

5. Conclusions and Future Work

This paper presents an approach for improving Business Process (BP) retrieval using a categorization and multimodal search approach. Categorizing BP allowed us to create an organized repository according to the categories covering the functionality of BP. In this sense, it was possible to identify BP sharing similar sets of categories regarding to a BP query to generate a consistent ranking. BP in the ranking not only shares textual and structural information, but also functional purposes represented as sets of categories. This allowed reducing the overall execution time because the search space was limited to only these BP sharing the same set of categories as the BP queries. Because categories were extracted from a general-purpose category schema SKOS it was possible to obtain sets of categories for each BP regardless of its application domain. Even though the execution time lasted to create the category-based index is long it is compensated because the increasing of the precision in 10% for each query. Additionally, the execution time for the search over categorized BPs it is reduced in 35% with respect to the multimodal model without the categorization approach. Formation of sets of categories allowed us to create an organized repository in the functionalities of each BP. In the same way, identifying sets of categories of a BP query can help to

obtain a list where results not only share textual or structural information, but also information or functional purposes. Besides the search process on the categorized repository showed a significantly reduction on the execution time because the limitation of the search space to only BP belonging to a similar set of categories and not the entire repository. The proposed approach was evaluated using a closed collection of BP that was issued by a set of human experts in a collaborative way. Our approach scored good levels of graded precision (between 79% to 98%), but poor levels of Recall because the multimodal search only ranks the first ten most relevant results. Future work is about extending our approach with clustering algorithms to create groups or families of BP for each category, and using bigger category sources that allow us to classify BP in a wider set of categories.

References

[1] H. Bae, S. Lee, I. Moon, Planning of business process execution in business process management environments, Information Sciences 268 (2014) 357–369. doi:10.1016/j.ins.2013.12.061.

[2] X. Zhao, C. Liu, Version management for business process schema evolution, Information Systems 38 (8) (2013) 1046 – 1069. doi:http://dx.doi.org/10.1016/j.is.2013.03.006.

[3] A. Jiménez-Ramírez, B. Weber, I. Barba, C. del Valle, Generating optimized configurable business process models in scenarios subject to uncertainty, Information and Software Technologydoi:10.1016/j.infsof.2014.06.006.

[4] Z. Yan, R. Dijkman, P. Grefen, Business process model repositories –framework and survey, Information and Software Technology 54 (2012) 380–395. doi:10.1016/j.infsof.2011.11.005.

[5] R. Dijkman, M. L. Rosa, H. a. Reijers, Managing large collections of business process models—current techniques and challenges, Computers in Industry 63 (2) (2012) 91–97. doi:10.1016/j.compind.2011.12.003.

[6] R. Dijkman, B. Gfeller, J. Küster, H. Völzer, Identifying refactoring opportunities in process model repositories, Information and Software Technology 53 (9) (2011) 937–948. doi:http://dx.doi.org/10.1016/j.infsof.2011.04.001.

[7] R.-H. Eid-Sabbagh, M. Kunze, A. Meyer, M. Weske, A platform for research on process model collections, in: J. Mendling, M. Weidlich (Eds.), Business Process Model and Notation, Vol. 125 of Lecture Notes in Business Information Processing, Springer Berlin Heidelberg, 2012, pp. 8–22. doi:10.1007/978-3-642-33155-8_2.

[8] A. Koschmider, T. Hornung, A. Oberweis, Recommendation-based editor for business process modeling, Data & Knowledge Engineering 70 (6) (2011) 483–503. doi:10.1016/j.datak.2011.02.002.

[9] G. Abaei, A. Selamat, H. Fujita, An empirical study based on semi-supervised hybrid self-organizing map for software fault prediction, Knowledge-Based Systems 74 (2015) 28–39. doi:10.1016/j.knosys.2014.10.017.

[10] C. Porcel, E. Herrera-Viedma, Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries, Knowledge-Based Systems 23 (1) (2010) 32–39. doi:10.1016/j.knosys.2009.07.007.

[11] Q. Zhu, R. R. Freimuth, Z. Lian, S. Bauer, J. Pathak, C. Tao, M. J. Durski, C. G. Chute, Harmonization and semantic annotation of data dictionaries from the pharmacogenomics research network: a case study., Journal of biomedical informatics-doi:10.1016/j.jbi.2012.11.004.

[12] Q. W. H. Zhang, M. K. Ng, S.-S. Ho, ForesTexter, Y. Ye, An efficient random forest algorithm for imbalanced text categorization, Knowledge-Based Systems Volume 67 (2014) 105–116.

[13] H. Ordonez, J. C. Corrales, C. Cobos, Multisearchbp-entorno para busqueda y agrupacion de modelos de procesos de negocio, Polibits 49.

[14] H. Hage Ameur, E, Topic tree: Increasing the accuracy of item retrieval, World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education Volume 2007.

[15] A. Miles, Skos core: simple knowledge organisation for the web., International Conference on Dublin Core and Metadata Applications.

[16] D. G. Ferrari, L. N. de Castro, Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods, Information Sciencesdoi:10.1016/j.ins.2014.12.044.

[17] H. Ordoñez, J. C. Corrales, C. Cobos, Business processes retrieval based on multimodal search and lingo clustering algorithm, IEEE Latin America Transactions 13 (9) (2015) 40–48, in press.

[18] Q. Zhong, Z. Qingqing, G. Tengfei, Moving object tracking based on codebook and particle filter, Procedia Engineering 29 (2012) 174–178. doi:10.1016/j.proeng.2011.12.690.

[19] M. F. Porter, An algorithm for suffix stripping, Program: electronic library and information systems 14 (3) (1980) 130–137.

[20] C. Manning, P. Raghavan, H. Schütze, An introduction to information retrieval (2007).

[21] G. Salton, A. Wong, C. S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (11) (1975) 613–620. doi:10.1145/361219.361220.

[22] Y.-C. Hu, B.-H. Su, C.-C. Tsou, Fast vq codebook search algorithm for grayscale image coding, Image and Vision Computing 26 (5) (2008) 657–666. doi:10.1016/j.imavis.2007.08.001.

[23] D. Petrelli, On the role of user-centred evaluation in the advancement of interactive information retrieval, Information Processing & Management 44 (1) (2008) 22 – 38. doi:http://dx.doi.org/10.1016/j.ipm.2007.01.024.

[24] R. S. Tetsuya Sakai, Evaluating diversified search results using per-intent graded relevance, SIGIR'11, Beijing, China (2011) 1043–1052.

[25] U. Kster, B. Knig-Ries, Measures for benchmarking semantic web service matchmaking correctness, in: L. Aroyo, G. Antoniou,

E. Hyvnen, A. ten Teije, H. Stuckenschmidt, L. Cabral, T. Tudorache (Eds.), The Semantic Web: Research and Applications, Vol. 6089 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 45–59. doi:10.1007/978-3-642-13489-0_4.

[26] V. Tsetsos, C. Anagnostopoulos, S. Hadjiefthymiades, On the evaluation of semantic web service matchmaking systems, 2006 European Conference on Web Services (ECOWS'06) (2006) 255–264doi:10.1109/ECOWS.2006.28.

[27] M. Guentert, M. K. Mathias Weske, Evaluation measures for similarity search results in process model repositories, Springer-Verlag Berlin Heidelberg.

[28] I. Alberts, J. Schellinck, C. Eby, Y. Marleau, Bringing together functional classification and business process analysis : Growing trends in records management, Cogniva Information Research Institute, Gatineau, Québec Bringing.

[29] C. P. Qinyi Wu Akhil Sahai, Roger Barga, Categorization and optimization of synchronization dependencies in business processes, Data Engineering, ICDE, IEEE 23rd International Conference.

[30] C. Armistead, S. Machin, Implications of business process management for operations management, International Journal of Operations & Production Management 17 (9) (1997) 886–898. doi:10.1108/01443579710171217.

[31] B. B. Stephan Roser, A categorization of collaborative business process modeling techniques, E-Commerce Technology Workshops, Seventh IEEE International Conference.

[32] E. Mavridou, G. Hassapis, D. Kehagias, D. Tzovaras, Semantic categorization of web services based on feature space transformation, in: Informatics (PCI), 2012 16th Panhellenic Conference on, 2012, pp. 162–167. doi:10.1109/PCi.2012.41.

[33] J. He, P. Li, X. Hu, X. Wu, A new automatic categorization algorithm for web services, in: Granular Computing (GrC), 2010 IEEE International Conference on, 2010, pp. 200–205. doi:10.1109/GrC.2010.124.

[34] D. Kehagias, E. Mavridou, K. Giannoutakis, D. Tzovaras, A wsdl structure based approach for semantic categorization of web service elements, in: S. Konstantopoulos, S. Perantonis, V. Karkaletsis, C. Spyropoulos, G. Vouros (Eds.), Artificial Intelligence: Theories, Models and Applications, Vol. 6040 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 333–338. doi:10.1007/978-3-642-12842-4_39.

[35] Z. Huang, J. Huai, X. Liu, J. Zhu, Business process decomposition based on service relevance mining, Sciencedoi:10.1109/WI-IAT.2010.21.

[36] R. A. T. R. Dafne A. Rosso-Pelayo Miguel Gonzales-Mendoza, Neil Hernandez-Gress, Business process mining and rules detection for unstructured information, Ninth Mexican International Conference on Artificial Intelligence.

[37] C. J. Turner, A. Tiwari, J. Mehnen, A genetic programming approach to business process mining, in: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO '08, ACM, New York, NY, USA, 2008, pp. 1307–1314. doi:10.1145/1389095.1389345.

[38] C. Li, M. Reichert, A. Wombacher, Mining business process variants: Challenges, scenarios, algorithms, Data & Knowledge Engineering 70 (5) (2011) 409–434. doi:10.1016/j.datak.2011.01.005.

[39] C. R. Kunze, Matthias, An extensible platform for process model search and evaluation, Business Process Management Demos 2013.

[40] S. Smirnov, M. Weidlich, J. Mendling, M. Weske, Action patterns in business process model repositories, Computers in Industry 63 (2) (2012) 98–111. doi:10.1016/j.compind.2011.11.001.

GROUPING OF BUSINESS PROCESSES MODELS BASED ON INCREMENTAL CLUSTERING ALGORITHM USING FUZZY SIMILARITY AND MULTIMODAL SEARCH

Hugo Ordoñez[1], Juan Carlos Corrales[1], Carlos Cobos[2], Leandro Krug Wives[3], Lucineia Thom[3]
[1] Telematic Engenieering Department ,University of Cauca, Sector Tulcán, Popayán, Colombia
[2] System Engenieering Departament,University of Cauca, Sector Tulcán, Popayán, Colombia
[3]Institute of Teleinformatics, Universidade Federal do Rio Grande do Sul (UFRGS), Caixa Postal 15.064, Porto Alegre, RS, Brazil
{hugoordonez,jcorral,ccobos}@unicauca.edu.co, {wives, lucineia}@inf.ufrgs.br

Abstract. Today, companies standardize and adapt their operations through Business Process (BP) to reuse them when new functionalities are required. This situation has generated a growth of BP repositories which makes difficult to recover BP from repositories. This paper presents MultiModalGroup, a model for grouping and searching for BP. The search is based on a multimodal representation which integrates textual and structural information of BP; by its part, the grouping mechanism is built upon a clustering algorithm which uses a similarity function based on fuzzy logic. This grouping is performed using the results of each user request. The evaluation of the proposed model was carried out in two phases: 1) internal quality evaluation of created groups and 2) External evaluation of created groups compared with the ideal of groups. Evaluation was done using a closed BP collection created collaboratively by 59 experts. Experimental results in each phase are promising and prove the validity of the proposal model.

Keywords: Business process, multimodal, search, Fuzzy, Clustering, evaluation.

1 Introduction

Model Business Processes (BP) gather procedures and interrelated activities in organizational structures that work together to accomplish a business objective. BP may be related to: product manufacturing, provision of services, procurement of goods and Inventory Management, among others [1-3]. Furthermore, BP models include information of shared data, process participants and how these participants interact [4, 5]. Usually, BP repositories may contain hundreds or even thousands of BP models [6, 7] . These BP models are stored in order to be used when new functionalities are required.

Unfortunately, BP repositories have grown in a in a disorganized and unsystematic way. Therefore manual search of BP may become a cumbersome and time consuming task. In this connection, some approaches to search BP models have been proposed, these found BP may be used as a starting point to create new BP models which fulfills new company requirements. Existing approaches are mainly based on: logs or execution traces [8-10], BP structure [11, 12], BP behavior [13, 14], textual similarity [15, 16]. All these proposals present results in mixed and unorganized lists.

Grouping techniques based on affinity or clustering may solve the problem of mixed results list. These techniques build a hierarchy of groups based on similarity of BP features [17, 18]. In these approaches, users verify the created hierarchy and select the most similar group to their search [19]. Despite contributions done by these approaches, results can be improved by including additional information from BP models such as: activities description, task type, gates type, among others.

This paper presents a grouping method for BP models based on multimodal search and fuzzy logic called MultiModalGroup. The indexation and multimodal approach integrates textual (tasks names and description, events and gates) and structural information (defined by codebooks containing information of BP behavior such as: task-task, task-gate, task- event-gate). Multimodal approach aims for get a more extensive representation of the subject of study (Business process), consequently, search options and relevance level of search results may be expanded.

Furthermore, MultiModalGroup groups results in order to provide a most clear and categorized representation of the results obtained in each request [20]. To achieve this, a grouping technique based on fuzzy logic is used. This technique identifies common features in the BP models obtained as a result of each user search and calculates its similarity level. The grouping model aims for reduction of search time as well as the provision of an organized way of reviewing results; consequently, users can select the BP models in a better way.

MultiModalGroup evaluation was done in two phases: i) evaluation of created groups using MultiModalGroup over other clustering algorithms found in state of the art, ii) the results were compared with other BP groups created manually by experts using a collaborative platform. The results of manual formation of groups BP model are presented elsewhere [21] .

The rest of the paper is organized as follows. Section 2 presents the motivating scenario, Section 3 presents the proposed model. Next, evaluation model is described in Section 4, Section 5 is dedicated to related works and Section 6 present conclusions and future work.

2. Motivating scenario

Nowadays organizations have big repositories of BP models. BP models help people to understand complex business activities by abstract business description [4] . A special challenge in this context is repositories maintenance as well as management of BP

models within these repositories. The latter is due to the fact that organizations produce product and services based on features set Inside a given product family. As an alternative to this challenge, some mechanisms are proposed to search and automatically detect different versions of BP models inside the repository that can explain Company behavior [22].

In spite of the relevance of proposed mechanisms, they only provide a list with independent BP models which describe specifically some of the activities carried out in organizations. To complement this, some mechanisms incorporate grouping or clustering algorithms which gather in the same group a coherent set of BP models, BP models in each group share common features given by: flow control, structure, purpose, and function of represented process or product. In this way, each created group may be used for generating more comprehensive process models [23]. Besides, created groups allow engineers (BP modelers) to explore grouped results in an organized way, so that they can make suggestions on BP redesign in order to incorporate the most frequent and significant changes to elements of a group all at once.

3 Description of MulTimodalGroup

BP models contains different type of information: syntactical, structural and behavioral. Therefore queries on these models may lead to biased results based on one or more type of information [24]. In this sense, the proposed model here combines textual and structural information to create groups of BP models based on multimodal search and fuzzy logic. The model comprises three layers (see Figure 1), i) Parser ii) Multimodal search, iii) cluster formation. The main components of the method are described below.

3.1 *Parser*

This layer comprises two processes: analyzer and indexer. Analyzer takes each BP model in the repository and pre-processes it to create two components: one textual and one structural. The indexer takes the created components (textual and structural) and weighs its elements (based on the TF-ID Standard formulae for information retrieval) to form multimodal search index. Both processes of this layer are described down below.

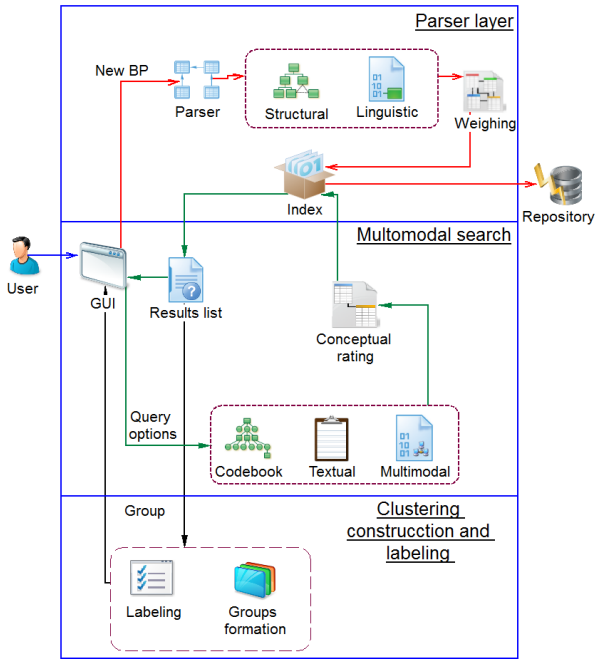


Figure 1. Components method proposed

Pre-Procesing: This process generates two componets represented by two matrices: *MC* matrix contains textual features and *MCd* matrix contains structural components. In order to generate these matrices, this process execute two analysis: textual and structural.

Textual Analizer : In this phase, each BP model (BP_i) in the repository is represented as a tree (A_i) taking into account the execution flow. Next, textual features (name of activity, activity type and description) are extracted from each tree. Each one of the extracted textual features is transformed to lowercase; then special characters, empty words and accents are removed. Afterwards, stemming is applied(Porter algorithm [26] to convert each one of the textual components to its root form., e.g. "Running" and "Runner, into" Run ") . Additionally, a new vector Ct_i is created with the processed textual features $Ct_i=\{Ct_{i,1},Ct_{i,2},\ldots,.Ct_{i,M}\}$, where M is the number of textual features in the repository. Then, vector Ct_i is used to create the matrix *MC* (that represents the component of textual features). in the matrix *MC*, each component $Ct_{i,j}$ represents the weight of textual feature j in BP_i.

Structural Analyzer: This task includes a training strategy and uses codebooks to generate sequential basic structural units from BP models. Codebooks are constructed based on frequent sequential patterns found in the structure of BP models. These codebooks are used to create the matrix MCd of structural components, in order to do this, the Structural Analyzer extracts a transitions vector

(*vt*) from each tree *Ai*, *vt* contains pairs of nodes which are interconnected sequentially in the BP model structure. Formally, let P be the total number of possible transitions in the repository, for each BP_i a vector *Cdi*={*Cdi*,1,*Cdi*,2,…,.*Cdi*,P} is created, where each codebook is formed by union of two or more transitions.

Vector *vt* represents a row *i* in matrix *MCd* of codebook components, where *N* is the total number of BP models in the repository, *i* is an specific BP, *j* is the index for an specific codebook of the repository and *P* is the total of codebooks in the repository. The figure 2 shows an example of a graphical representation of codebooks composing the BP model. As is seen in Figure 2 there are 4 codebooks: {Start_TaskUser$_1$, TaskUser_ParallelRoute$_2$, ParallelRoute_TaskService$_3$, ParallelRoute_TaskService$_4$}. In this case, TaskUser refers to each one of BP User task or activity (e.g. Set Variable and location).

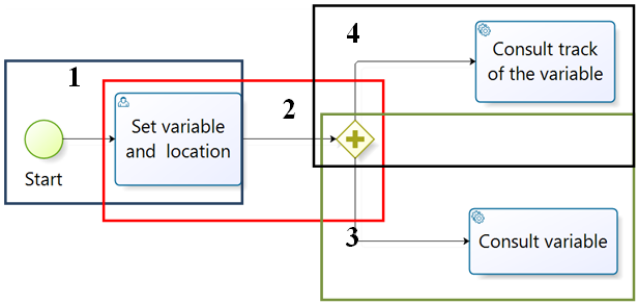


Figure 2. Codebook formation

Indexing: This module creates an index composed of two search spaces: 1) textual indexing of business functions in each BP model and 2) structural indexing (characterization of task types, event types, and connection types). These two search spaces are unified in a multimodal structure to create a broader index which offers a more accurate representation of the subject of study (BP models repository). In this process the textual and codebook components are weighted in the index of multimodal search *MI* composed by textual matrix (*MC*) and the codebook component matrix (*MCd*), in other words, $,IM = \{MCd \cup MC\}$. *MI* contains in each cell a weight (w_{ij}) that represents the importance of textual components (in its lexical root) or codebooks in each BP model.

Matrix *MI* is created based on equation (1) given by Salton where $F_{i,j}$ is observed frequency of component *j* in $BP_{i,}$ model. Component *j* can be textual (L) or codebook (K). Max (F_i) represents most observed frequency in BP_i model, *N* is the number of BP models in the collection (repository), and n_j is the number of BP models in which textual component or codebook is found. Figure 3 shows graphically the index matrix composed of two spaces or components of *MI*: the first one (green box at left) represents the weight of elements in each codebook in each BP model; and the second one (blue box) shows the weight of textual elements in each BP model. Furthermore, this index also store the physical reference of BP models found in the repository.

(1)

$$w_{i,j} = \frac{F_{i,j}}{\max(F_i)} \times \log\left(\frac{N}{n_j + 1}\right)$$

	MC					MCd				
BP1	**Ct1**	**Ct2**	**Cti**		**CtL**	**Cd1**	**Cd2**	**Cdk**		**CdK**
BP2	$W_{1,1}$					$W_{1,1}$				
BPn		$W_{2,2}$					$W_{2,2}$			
BP4			$W_{n,l}$					$W_{n,k}$		
BP5				$W_{n,l}$					$W_{n,k}$	
BPn					$W_{N,L}$					$W_{N,K}$

Figure 3. Matriz índice (MI)

3.2 Multimodal search

This layer allows searching based on three options: Textual, structural (codebook) and multimodal (combination of the previous two). Each request BP is represented by a vector $q = \{t1, t_2, t_3, t_j\}$. The same pre-processing mechanism of parser layer is applied to this request vector, so the terms in the request vector are reduced to root forms.

Similarities identification: after request vector preprocessing, this module generates a linear ranking taking into account one of the three query options selected by user, retrieved BP models are ordered and filtered using equation (2) (Lucene practical scoring function[1]).

(2)

$$score\ (q,d) = coord(q,d) * \sum_{t\ \epsilon q} (tf(t\ \epsilon\ d) + idf(t)^2 * norm(t,d))$$

In Equation 2 t is a term of query q; d is the current BP being evaluated with q; $tf\ (t \in d)$ is the term frequency defined as the number of times the term t appears in d so that BPs are ranked according to the values of term frequency scores; $idf(t)$ is the number of BPs in which term t appears (inverse frequency); $coord(q; d)$ is the scoring factor based on the number of query terms found in the queried BP (those BPs containing the most query terms are scored highest); $norm(t; d)$ is the weighting factor in the indexing, taken from $w_{i;j}$ in the multimodal index. Once the score calculations are performed, results are sorted and filtered in descending order according to similarity to user query.

List of results: in this component, a matrix called *Mtop-k* containing textual and structural elements of all BP models retrieved from the query. These BP models are sorted and filtered based on the equation (2). BP models stored in the *Mtop-k* matrix will be used in the process of group formation in the next layer. Furthermore, this approach allows to view an ordered list of elements stored in matrix *Mtop-k.*

Repository: This is the central BP storage unit, and it is similar to a database which shares information about artifacts produced by engineers or used by the company [23]. Repository is responsible for storing and displaying information on BP attributes (roles, activities description, timers, messages, and service calls) [25]. The repository used in this research consists of 100 BP obtained from real processes of telecommunications companies.

3.3 Clustering construction and labeling

In this layer, BP models retrieved as relevant for each query are used to form groups based on the level of similarity between the terms (linguistic and structural) of each one of these BP models. Levels of similarity are calculated by a fuzzy logic function (Equation (3). Weights of similarity between terms are stored in matrix *Mterc* shown in Figure 4. Matrix *Mterc* stores in each cell the similarity degree between two BP, that is, each cell shows how similar two BP are. Values stored in matrix *Mterc* are between 0 and 1, where 0 means two BP are completely different and 1 means that two BP are very similar. As BP models are very similar to itself, so elements of main diagonal contains 1. The latter makes that matrix *Mterc* to be symmetric, therefore the lower half of the matrix can be ignored and the upper half (a triangular matrix is formed) is used to the creation of groups. Once formed *Mterc*, an incremental clustering algorithm called BestStart [26] (based on Start [27, 28]) is executed. This algorithm creates groups for BP visualization. Next, processes related to this layer are described.

	BP1	BP2	BP3	BP4	BP5	BP6
BP1	***1***	**0.4**	**0.2**	**0.7**	**0.8**	**0.9**
BP2	0.4	***1***	**0.5**	**0.3**	**0.6**	**0.7**
BP3	0.2	0.5	***1***	**0.3**	**0.4**	**0.3**
BP4	0.7	0.3	0.3	***1***	**0.7**	**0.6**
BP5	0.8	0.6	0.4	0.7	***1***	**0.5**
BP6	0.9	0.7	0.3	0.6	0.5	***1***

Figure 4. Similarity Matrix (Mterc)

Calculation of similarity between retrieved BP models: this process calculates degree of similarity (*gs*) between BP models in a cluster using fuzzy logic as described in Equation (3).

(3)

$$\boldsymbol{gs(X,Y) = \left(\sum_{h=1}^{k} gi(a,b)\right)/n}$$

In equation (3) X and Y are vectors representing BP models, k is the number of common elements between BP models, *n* is the number of elements in X and Y, *gi* defines the degree of equality (*gi*) between weights of element h^{th} (*a* in *X* and *b* in *Y*). The Equation (4) shows (*gi*) represented using fuzzy logic.

[1] http://lucene.apache.org/core/3_6_2/api/core/org/apache/lucene/search/Similarity.html

(4)

$$gi(a,b) = 1/2\ \left[(a \rightarrow b) \wedge (b \rightarrow a) + (\bar{a} \rightarrow \bar{b}) \wedge (\bar{b} \rightarrow \bar{a})\right]$$

In equation (4) $1 - x; a \rightarrow b = \max\{c \in [0,1] \mid a * c \leq b\}; and \wedge = min.$ In this equation an attribute may have different importance grades in different vectors (BP models). In this sense, instead of estimate average or product between two BP, the function determines equality degree between them. Therefore, when calculation of similarity between all models is carried out, similarity values are stored in matrix Mterc (see Figure 4).

BP models Grouping: In this step, an incremental clustering algorithm based on graph theory called BestStar is applied to the matrix *Mterc* which improves Star algorithm presented in [28]. Algorithms based on graph theory use a graphical notation to facilitate group's identification (Clusters), each element (BP model) represents a node, links between nodes represent groups and nodes interconnected by lines belong to the same group. These algorithms have low quality of formed groups because they depend on a previously established thresholds. Besides, the order in which elements are selected as star centers also influences grouping results. BestStart (Algoritm 1) solves these disadvantages, as user does not need to set a threshold previously, besides BestStart performs an analysis of already assigned elements each time a new centroid is selected in order to reallocate elements to the closest star as needed, even if they are already grouped.

Algorithm 1 takes matrix *Mterc* as input, *Mterc* contains similarity of textual and structural information of BP models which where returned as relevant to a query. In this matrix, not assigned BPs are selected and gathered in a new group (line 1-3); once the group is created using the current BP as centroid, then other BP is selected and the similarity level is checked, if the selected BP is not assigned so it is added to the created group, otherwise the most similar group is found and the BP is added to this group (lines 6-10). This is the way how groups are created, using unassigned BP as group centroid. When a BP is already assigned, so similarity of original cluster is obtained and this similarity is compared with the current BP in order to create a new group (lines 14-17). If similarity level between current BP and assigned BP is higher than the similarity with the original group, so the BP is assigned to the new group it is removed from the original group (lines 21-24), finally groups with one single element are eliminated and the formed groups are shown to the user.(lines 29 and 30).

Algorithm 1. Algorithm for BP Group

Require: A Matrix (*Mterc*)

Ensure: A group create (G_{BP}) for the query *BP-Input*

```
1   for all BPi ϵ MtrecL do
2   If BPi not assigned then
3     star = CreatNnewCluter(BPi)
4         for all BPi ϵ MtrecJ do
5               BPj =getCurrentBP(MtrecJ )
6               Similarity = getSimilarity(BPi, BPj)
7                 if BPj not in star then
8                    addToSatr(BPj)
9                 else
10                  ClusterGreaterSimilarityFound(Similrity)
                    AddToMostSimilarCluster(BPj)
11                endIf
12            endfor
13            AddGBP(star)
14  else
15  originalCluster= GBP(i)
16  BPprevious = getBpOriginalCluster(0)
17  star = CreatNnewCluter()
18  similarityOriginal= getSimilarityOriginalCluster(BPprevious)
19     for all BPi ϵ MtrecJ do
20       BPj =getCurrentBP(MtrecJ )
21       Similarity = getSimilarity(BPprevious, BPj)
22        if Similarity > similarityOriginal then
23          addToSatr(BPj)
24           originalClusterRemove(BPprevious)
25        endIf
26   endfor
27  endif
28  endfor
29  removeClusterWithOneBP()
30  Return GBP
```

Labeling: in clustering algorithms based on graph theory, created groups are not identified with labels defining its content. Therefore the present approach adapted a labeling method based on Suffix Arrays to identify the content of each created group

(purpose or functionality of BP models) and to ease user's interaction with the results. Thus users may get a better idea of group or groups to review.

The labeling process begins by creating a summary or snnipet (*S*) using tasks names, these tasks describe functionalities of BP models that compose the group to label. Subsequently, chain *S* is pre-processed, converted to lowercase; latter special characters and empty words are removed from *S*. Finally an array of suffixes *As* is created, this array is ordered lexicographically to find most common phrases in *S* that identify group content.

In labeling algorithm, *S* is processed as a character set $S = \{s_1, s_2, s_3, s_n\}$, from this set, it is formed a new set *S´ [i ,j]* an array of sub-string of S, which runs from index *i* to index *j*. after that, an array of integers *As* is created containing initial positions of suffixes in S ordered lexicographically. Then, *As[i],* stores the starting position of the *i*-th smallest suffix in S. Afterwards, array of substrings *S´* is traveled using a binary search that aims to find the most common and with higher length suffix. The search starts with separator of terms character $ and subsequently found suffix is returned for labelling the BP group.

3.4 Example of grouping execution

For the example, it will be used a BP called Active service. In order to run a consultation on the repository (see Figure 5), multimodal search model retrieves BP with higher similarity to the query BP and generates a result list filtered and sorted in descending order (see Figure 6). Later results are grouped using BestStar algorithm on the basis of similarity levels stored in the Matrix *Mterc*. BP Groups are formed based on the relationship rules implemented in the algorithm. After the grouping phase, it start the labeling phase. In labeling phase a summary or Snippet is generated using BP labels inside each group, to do so the most representative phrase in the group is found. Finally labels and groups are displayed as shown in figure 7.

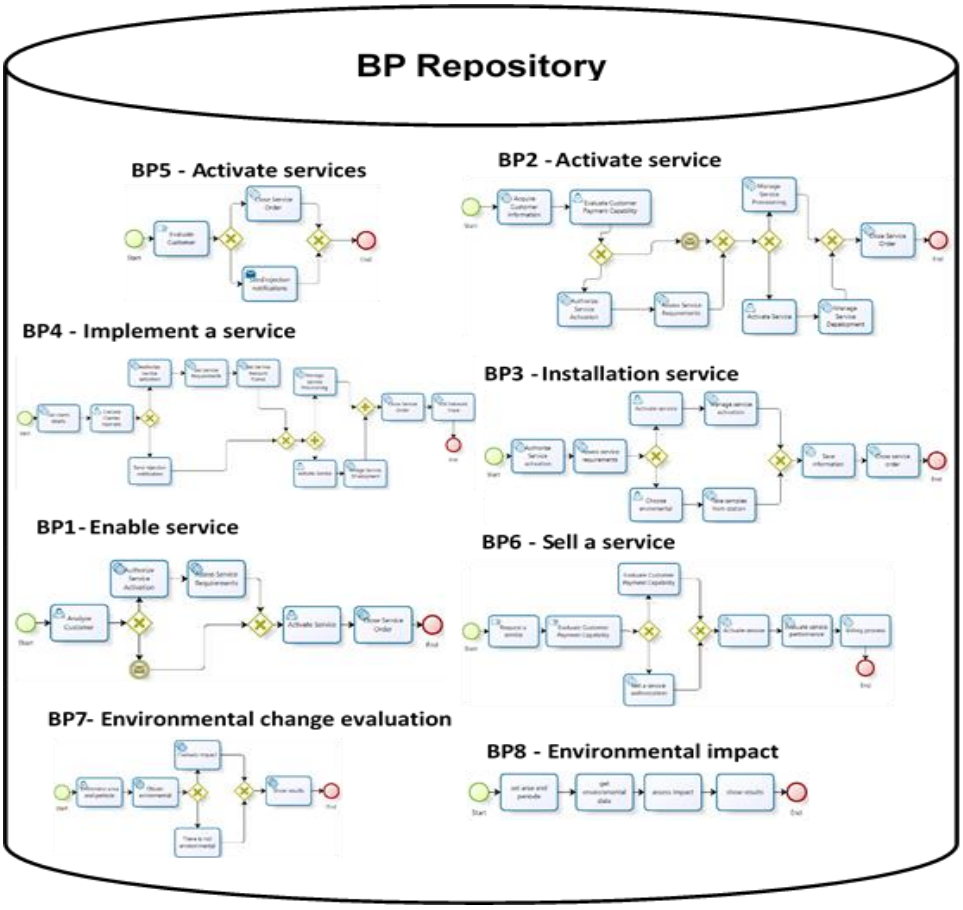


Figure 5. BP Repository

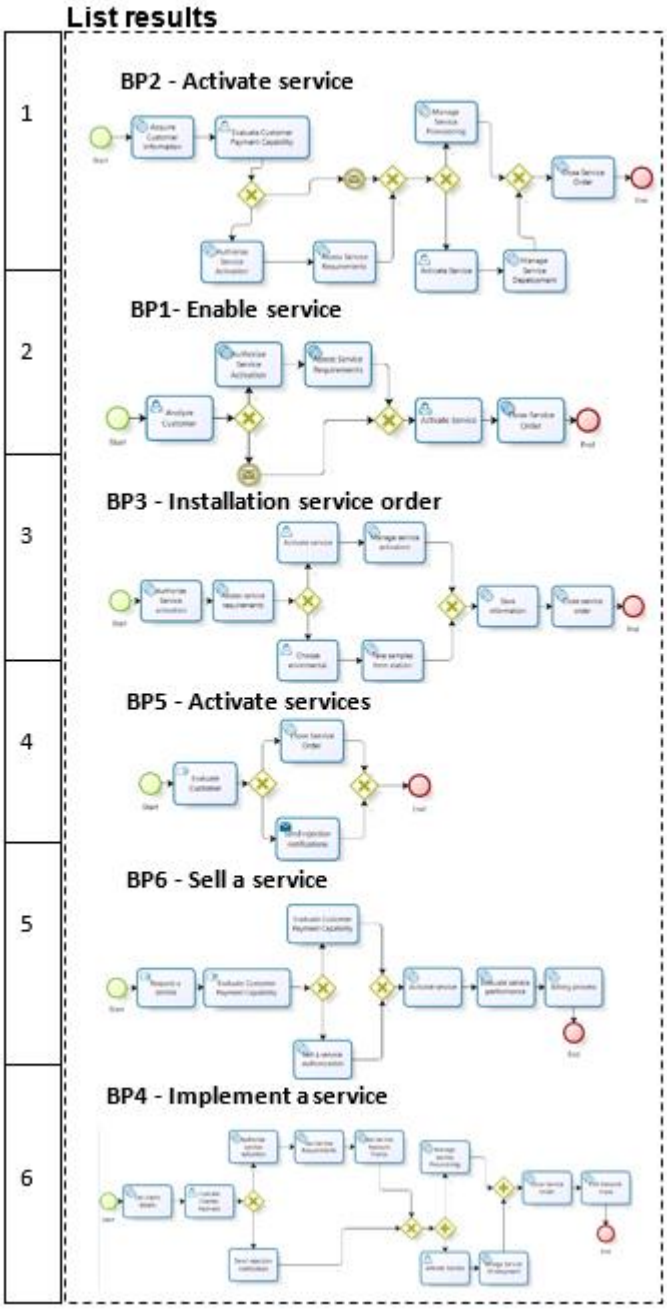


Figure 6. result List

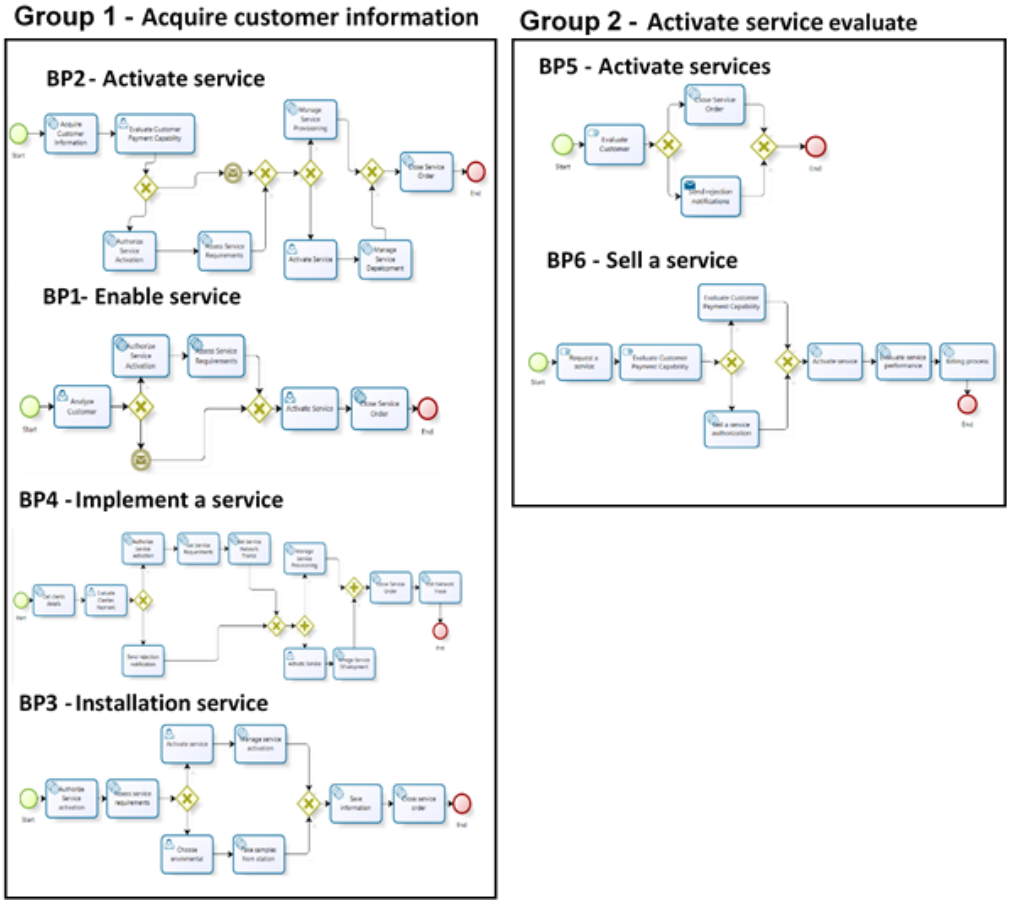


Figure 7. Created groups

4. Evaluation of the proposed method

Measuring grouping or clustering performance is not a trivial task, because there is no standard methodologies for this purpose. Consequently, clustering evaluation is based on diverse metrics that evaluate internal and external quality [25, 29] of created groups.

Internal evaluation is based on the fact that groups of objects that are compact and well separated from other groups are better. This evaluation measures density, distances between objects in the same group (smaller distances means more compact groups) and separation between clusters (larger distances indicate greater separation, which is better) [30].

External evaluation is used when an "ideal" training data sets exists. The class (or classes) of data objects are known and they are compared with groups created by the evaluated algorithm. Consequently, external validation is more accurate than internal one. This is an important type of validation when it is necessary to find the best clustering method for a specific task and usually involves comparing a variety of algorithms on specific datasets [31].

Consequently, to validate the present approach it was designed an evaluation process which includes two phases: i) Internal assessment using internal metrics to measure proximity of groups formed by the proposed method (BestStartBP) and other algorithms of state of arte: Start, Cliques, K-means, Lingo, STC, and ii) external evaluation that compares groups formed by the proposed method and the groups created collaboratively by experts, this latter grouping is considered “ideal”.

4.1 Compared algorithms

- **STC**

STC takes each BP and extracts the text of all components and creates an ordered syntactical sequence of textual terms to generate the BP grouping based on extracted information. This algorithm consists of two steps:

Step 1: to identify base groups. In this step the algorithm creates a suffix tree from the array containing all textual components of the BP. Based on this matrix, the algorithm detects a root in such way that it is ensured that each node contains at least two internal sons (a pair). Then, edges between nodes are labeled using a part of the summary text with the aim of forming the label of this node.

Step 2: to combine base groups. In this step the algorithm assigns a classification to each base group, taking into account the number of BP inside the group and which are related with a set of textual elements. For this purpose it is used a base group function (s(B)), in which it is included a group B with elements P like this: s(B) = |B| * f(|P|) , where B| is the number of BP in the base group B; |P| is the number of elements in P that has no qualification 0 (i.e. that are connected at least with one node of the suffix tree); f is a function that penalize P with only one element.

- **LINGO**

This algorithm builds summary (Snippets) with textual terms of each BP's component retrieved by queries. The algorithm involves 4 steps:

Step 1: Feature extraction, this step identifies phrases or terms candidates to label the group. The latter is done by calculating the number of occurrences of identified phrases and words in the list of BP resulting from the request.

Step 2: group labels inducement: this step forms significant group descriptions using information of the terms matrix for each BP.

Step 3: identification of BP belonging to each group: this step compares text fragments with each and every group labels. To do so a matrix Q is created. In Q each group label is represented as a column vector in such a way that C = QTA, where A is the original term of the BP term matrix. This, element ci,j of the matrix C indicates adhesion weight of BP j in Group i.

Step 4: final formation of BP groups: this step calculates label information weighing depending on the occurrences of label terms in each one of the BP assigned to the group.

- **K-Means**

In this algorithm the number of BP groups (k -clusters) to form must be specified in advance. Then k BP are randomly selected which will represent centroid of media of each BP group. Later, each one of the BP in the result list are assigned to the closest cluster centroid according to a distance function (the most used function is Euclidean). For each of the formed groups the centroid of all their BP is calculated. These centroids are taken as new centers of their respective groups. K-means steps are described next:

Step 1: Choose k BP acting as centroids (k determines the number of BP groups to form)
Step 2: each BP, is added to the group with the highest similarity or proximity
Step 3: Calculate the centroid of each BP group, in order to become the new centroids.
Step 4: if a convergence criterion is not reached (for example, two iterations do not change classification of BP groups) so return to step 2).

- **Stars**

The algorithm is executed iteratively, each iteration selects a BP from the result list to form a new group, then the Stars assesses all remaining BP to find the most similar ones in order to add them to the created group (Stars considers that a BP is similar to another when its similarity level is greater than or equal to a predefined minimum acceptance threshold). Thus, at each executed iteration formed group is represented as a graph, in which initial BP references to the main node (central) and the BP conforming the group are connected to the main node through edges. This graph forms a figure very similar to a star (hence the name of Stars). The algorithm performs the following steps:

Step 1: To select a BP that is not part of any group and create a new group.
Step 2: Add to the created group all BP with greater possible similarity.
Step 3: If there are still ungrouped BP, repeat steps 1 and 2.

- **Cliques**

This algorithm is also executed iteratively, at each iteration all retrieved BP are compared with each other, in order to assign them to the same group depending on the similarity between them. A BP is similar to another, if similarity degree is greater than or equal to the acceptance minimum threshold defined previously. The steps of the algorithm are:

Step 1: To select a BP that is not part of any group and create a new group.
Step 2: To select a new BP and compare it with the BP from the previous step.
Step 3: If the selected BP is greater than or similar to all BP of the current group, then it is added to that group.
Step 4: If the selected BP is already assigned to a group, then return to step 2.
Step 5: while there are ungrouped BP, return to Step 1

- **FullStart**

This algorithm considers that BP are already grouped, FullStart assigns BP to all groups where similarity is greater than or equal to the threshold without taking them out of the groups which they are already part. This algorithm performs the following steps:

Step 1: To select a BP that is not part of any group and adds it to a new group.
Step 2: To select a new BP and adds it to all groups which is greater or similar.
Step 3: Perform steps 1 and 2 until all BP are added to at least one group.

4.*1 Measures of internal evaluation*

As an important validation aspect of the proposed method, a performance study was done. This study involves the application of some internal metrics for clustering analysis that do not require human intervention. These metrics are used to identify how close or distant BPs are from each other in the formed groups. Additionally this study is important to know how relevant are the elements belonging to each group. The used metrics are described below.

Cohesion: It expresses similarity average between elements of a cluster. With greater degree of similarity the group is more cohesive. Equation 5 allows measuring cohesion.

(5)

$$cohesion(c) = \frac{\sum_{i>j} sim(c_i, c_j)}{\frac{m(m-1)}{2}}$$

Where sim (c_i,c_j) is similarity degree between elements c_i and c_j which exist in the cluster C, and m is the number of existing elements in the cluster.

Coupling: It is used to express the average similarity between all pairs of elements, where an element belongs to group C and the other does not. Ideally, the coupling must be low. The coupling is calculated with Equation 6

(6)

$$coupling(c) = \frac{\sum_{i,j} sim(c_i, q_j)}{m * n}$$

Where *sim* (c_i, q_j) is the similarity between the c_i element of group C and q_j is another cluster member; m is the number of elements in C and n is the number of external elements from C.

Silhouette: This is derived from cohesion and coupling, they show which items are appropriate within a group and which have an intermediate position between the group they were assigned and other groups, the greater silhouette degree better the distribution of groups is . This can be measured with the Equation 7.

(7)

$$silhouette(i) = \frac{b_i - a_i}{\max(a_i, b_i)}$$

Where a_i is average dissimilarity (ie, distance) between the i^{th} element of the group and the other objects of the same group, and b_i is the minimum dissimilarity average between the i^{th} element of any group that does not contain the element.

Sum of squares Between clusters (SSB): this measures separation between clusters, in equation 8 k is the number of clusters, n_j is the number of elements in the cluster j, c_j is the centroid of cluster *j and x* is the mean of the data set evaluated through Equation 8:

(8)

$$SSB = \sum_{i=1}^{k} n_j \, dist^2(c_j - \bar{x})$$

Sum-of-squares within cluster (SSW): measures the variance within groups, based on each of the existing elements in each group, calculated on Equation 9

(9)

$$SSW = \sum_{i=1}^{k} \sum_{x \in c_i} dist^2(m_i, x)$$

Where k is the number of clusters, *x is* a point in the cluster c_i and m_i is the centroid from cluster c_i.

Davis building: Measures the relation of the dispersion within the cluster, and the separation between clusters, a lower value means that the group is good, this measure is useful to evaluate the formation of unique groups. The Equation 10 allows the calculation of this index.

(10)

$$DB = \frac{1}{k} \sum_{i=1, i \neq j}^{k} max\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right)$$

Where k is the number of clusters, σ_i is the average distance between each point in the cluster i and the centroid of the cluster, σ_j is the average distance between each point in the cluster j and the centroid of the cluster, and $(c_i,)$ is the distance between the centroids of the two clusters.

4.2 internal evaluation of the Grouping

This section presents the average results obtained by each algorithm in each one of the evaluated measures, see Table 1.

Table 1. *Results of internal evaluation of the grouping*
(Cohesion **Ch**, coupling **Ac**, Silhouette **Si**, bulding Davis index **Db**)

Algorithm	Ch	Ac	Si	SSB	SSW	Db
MulTimodalGroup	**1,382**	**0,167**	**0,80**	**0,33**	**0,004**	**0,167**
Cliques	1,088	0,283	0,56	0,133	0,069	0,913
FullStars	1,035	0,726	0,25	0,138	0,087	0,725
Stars	0,948	0,274	-0,008	0,282	**0,063**	0,920
K-means	0,620	0,246	0,097	**0,051**	**0,048**	**0,583**
Lingo	0,147	0,340	-0,043	**0,129**	0,070	**0,527**
STC	0,164	0,676	-0,036	0,148	0,079	0,817

Regarding the ***Cohesion*** MulTimodalGroup obtains the highest value compared to other algorithms. MulTimodalGroup improves an average of 0,51 in relation to the evaluated algorithms. Improvement is due to the fact that MulTimodalGroup does not allow overlapping, i.e. elements that exist in multiple groups simultaneously. In that way, the groups formed by MulTimodalGroup contain elements with more closeness or similarity between them. Besides BestStar algorithm implemented in MulTimodalGroup has the advantage of re-grouping BP models to greater similarity clusters, even when they have already been grouped, consequently closeness level of elements (BP model) within group increases.

Concerning ***Coupling,*** like in the previous measure, the smallest value is obtained by MulTimodal Group, reaching an average improvement of 0,45 in relation to the algorithms evaluated. This is because elements considered being similar between the formed groups shares relatively a common subset of features, which may be textual or structural. Those specifications are not taken into account by the other algorithms because these are focus on a single type of information (textual) for clustering.

The value 0,8 achieved by MultimodalGroup in the ***silhouette*** coefficient expresses that the grouping is good, it also confirms the average improvement of previous measures ***(Cohesion and Coupling*)**. This shows that the groups formed by MulTimodalGroup contain well-placed elements, generating better distributed groups. Therefore the lower value of coefficient for the rest of algorithms is because in the formed groups there are intermediate elements which belong to several groups at once, this situation makes that items within groups are dispersed or have low similarity.

Regarding **SSB,** the low average 0,33 reached by multimodal Group shows that separation of formed groups is good, since the elements are assigned to the group having higher similarity, and eliminating the existence of intermediate elements in groups.

In relation to **SSW**, elements variation between formed groups by MulTimodalGroup is low, due to the fact that existing elements in each group must share textual and structural information in the group with higher similarity with them. Moreover ***Db***, expresses that elements are well placed within each group, in other words, there are not dispersed, based on information shared between them.

In Multimodal Group, elimination of overlapping (items that can exist in several groups at the same time) and reallocation of items to a group with the highest similarity is a determining factor in the quality of the formed groups. As is shown in Table 1, the quality of the formed groups by the algorithms that overlap is lower, because they produce too many groups with excessive duplication of elements belonging to each group.

4.2 Measures of external evaluation

An external measure assesses quality of clustering by comparing groups created by automatic grouping techniques with groups generated by domain experts. External evaluation metrics used in this evaluation are: pondered precision measures, weighting precision, weighting recall and weighting -F measure (which calculates relation between precision and recall). The latter measueres are taken from recovery information fiel [32, 33].

In order to evaluate , weighting precision, weighting recall and weighting -F measure, the groups set {C1, C2…..Ck} automatically created with MulTimodalGroup was compared to the ideal group collection $\{C_1^i, C_2^i, \ldots . C_h^i\}$ generated collaboratively by 56 expert users and presented in [21]. During evaluation, the following steps were performed: (a) to find for each ideal group C_n^i, a different group C_m that relates the most within the collection that is being evaluated, and to assess $P(C, C^i)$, defined in the Equation 11, later, to asses $R(C, C^i)$ defined in Equation 12, and to asses $F(C, C^i)$ defined on equation 13. (b) to calculate the weighting precision, weighting recall and weighting –F measure based on Equation 14.

(11)

$$P(C, C^i) = \frac{|C \cap C^i|}{|C|}$$

(12)

$$R(C, C^i) = \frac{|C \cap C^i|}{|C^i|}$$

(13)

$$F(C, C^i) = \frac{2P(C, C^i)R(C, C^i)}{P(C, C^i) + R(C, C^i)}$$

(14)

$$P = \frac{1}{T} \sum_{j=1}^{h} |C_j^i| \, P(C_m, C_j^i);$$

$$R = \frac{1}{T} \sum_{j=1}^{h} |C_j^i| \, R(C_m, C_j^i);$$

$$F = \frac{2PR}{P + R} \; ;$$

$$T = \sum_{j=1}^{h} |C_j^i|.$$

In Equation 14 C is a group of BP models, C^i is an ideal group of BP models.

Additionally they were evaluated, the *Formed cluster number (Nc):* measures the number of groups formed, based on number of returnees results as relevant in each query done by a user. And *Number of items per cluster (Ne*): measures the number of items within each formed group

4.3 Analysis of external evaluation

At this stage, groups created with MulTimodalGroup and evaluated algorithms were compared with groups formed manually by experts. During manual group formation, the same 6 queries of the previous phase were executed. With the results considered relevant by the experts in each query some groups were formed and labels were assigned. These are described in Table 2.

Table 2. Relevant elements and groups formed manually for experts on each query

Query	Relevant elements	Formed groups	Elements by group
Q1	10	3	3,33
Q2	12	4	3
Q3	13	3	4,33
Q4	11	3	3,66
Q5	14	4	3,5
Q6	14	4	3,5

Table 3 shows average scores of Precision, Recall and F-Measure for the evolution of groups formed using MultimodalGroup and using the evaluated algorithms. Table 4 also includes the comparison with the groups formed manually by experts.

Regarding precision, the best results are achieved by MulTimodalGroup followed in second position by Lingo (0,502) and STC (0,401), in third place , remaining algorithms keeps an average between (0.34 and 0.38). Overall, MulTimodalGroup increases precision in 30% compared to Lingo and 40% compared with STC and 42% compared with other algorithms. Average precision allows to conclude that groups generated by MulTimodalGroup have a high similarity with the groups formed manually ("ideal") due to the high number of shared elements. Moreover, combination of structural and textual information used by MulTimodalGroup allows creating groups with higher similarity to groups formed by experts who consider several information types presented in BP. In this context Lingo and STC obtained high values due to the number of groups formed and the number of elements per group, this makes that items considered relevant exist in some form in one or more groups.

In relation to remember the best values are achieved by MulTimodalGroup (0.638), STC reports the second highest average (0.581) and K-means gets the third best value (0.424), the rest of algorithms maintain a Recall average between (0.134 and 0.30). Consequently MulTimodalGroup increases Recall 6% compared to STC, 19% compared to K-Means and an average of 59% compared to other algorithms. The Recall value reached shows that some elements in the formed groups by MulTimodalGroup, are scattered in the groups of manual creation. Besides elimination of factors such as overlapping, a threshold value and the number of groups to form allow that MulTimodalGroup to reduce the value of false negatives (FN), i.e. those elements of group j which were placed in a group different to the one that was indicated by its label. The amount of formed groups by STC and the number of elements per group makes that this algorithm increases the value of true positives (TP), that is, those elements that were placed by the algorithm in the same group indicated by the manual grouping performed by experts, however STC also increases the values of (FN), as a result recall value decreases slightly. Moreover the value of the groups to be formed and the number of iterations to be performed is a determining factor for K-means, consequently precision decreases due to the increment of number of false positives (FP), i.e., those elements that they were placed by the algorithm in the group j, but manual grouping by experts assign these elements to other groups.

Higher values of F-measure are obtained by MulTimodalGroup that achieves a 23 % more than STC and 33 % higher than K-means. MulTimodalGroup F-Measure (0.7065) determines the performance of the grouping performed by the proposed method. This allows to infer that created groups are relevant and matched closely with groups formed manually by experts.

Table 1. Precision, Recall and F-Measure in external evaluation

Algorithm	Measure	Q1	Q2	Q3	Q4	Q5	Q6	Average	Nc	Ne
	P	**0,857**	**0,786**	**0,764**	**0,730**	**0,844**	**0,823**	**0,801**		
MulTimodalGroup	R	**0,714**	**0,614**	**0,543**	**0,763**	**0,610**	**0,581**	**0,638**	6,3	**2,9**
	F-measure	**0,779**	**0,690**	**0,635**	**0,746**	**0,708**	**0,681**	**0,707**		
	P	0,684	0,385	0,632	0,018	0,373	0,183	0,379		
Cliques	R	0,158	0,121	0,174	0,036	0,132	0,175	0,133	2,3	6,9
	F-measure	0,257	0,184	0,272	0,024	0,195	0,179	0,185		
	P	0,449	0,385	0,621	0,030	0,441	0,267	0,366		
FullStars	R	0,241	0,158	0,174	0,071	0,173	0,250	0,178	19,6	17,9
	F-measure	0,314	0,224	0,271	0,042	0,248	0,258	0,226		
	P	0,449	0,385	0,632	0,018	0,373	0,183	0,340		
Stars	R	0,168	0,121	0,174	0,036	0,132	0,175	0,134	4	8,4
	F-measure	0,245	0,184	0,272	0,024	0,195	0,179	0,153		
	P	0,759	0,450	0,292	0,037	0,421	0,303	0,377		
K-means	R	0,462	0,400	0,338	0,083	0,583	0,675	**0,424**	**3**	**3,6**
	F-measure	0,575	0,424	0,313	0,051	0,489	0,418	0,378		
	P	0,583	0,483	0,411	0,487	0,502	0,545	**0,502**		
Lingo	R	0,279	0,300	0,235	0,346	0,331	0,309	0,300	12,3	2,6
	F-measure	0,378	0,370	0,299	0,405	0,399	0,395	0,374		
	P	0,405	0,341	0,338	0,432	0,441	0,446	0,401		
STC	R	0,530	0,565	0,533	0,708	0,523	0,628	**0,581**	12	7,3
	F-measure	0,459	0,426	0,413	0,537	0,478	0,522	**0,473**		

Figure 5, present the results of the external evaluation, the algorithm based on graph theory that gets better gender results is Fullstart which achieves a performance (F-measure) of 0.2263 based on the manual creation of experts, this is because FullEstar allows overlapping and generates a considerable number of groups, which contain common elements among them, one of the disadvantages presented in these algorithms is the threshold value for determining the similarity of the elements, which in turns determines the assignation to each group, the latter makes that during grouping some items that may be relevant to each group to be put aside (FN).

Moreover, the algorithm for clustering web documents with better performance was STC with 0.4725, STC like FullStar allows overlapping and generates a high number of groups with common elements, although this algorithm does not require a threshold or pre-define the number of groups to be formed, it is based on a suffix tree built from the textual information present in BP, which makes that groups elements sharing only this type of information. Regarding K-means, algorithm for partially clustering of data, this achieves a performance of 0.3783 compared to the ideal group formation, although K-means does not allow overlapping, the value of (K) groups to form makes that the number (VP) decreases due to this parameter.

Finally, the values Nc and Ne from MulTimodalGroup show that number of groups and elements belonging to them are proportional with the number of results returned as relevant in each query.

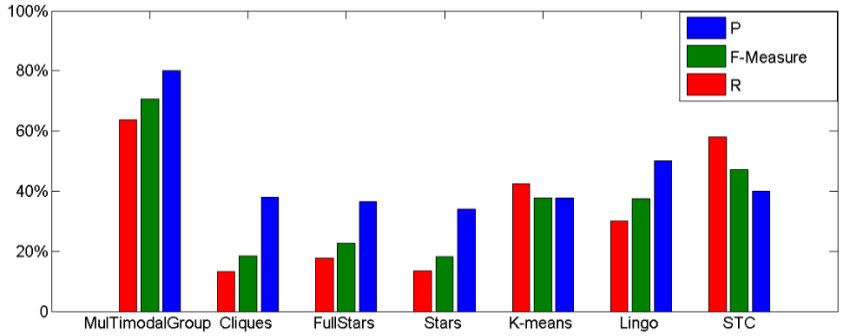


Figure 5. Average values of Precision, Recall and F-Measure in external evaluation

To evaluate statistical significance of results shown in Table 4, non-parametric Friedman test (Average rankings) were applied to results obtained using the algorithms in each of one the queries. Friedman test generated an Classification sorted ascending according to the average value of performance considering a chi-square distribution with 6 degrees of freedom. Table 1 shows results of precision, recall and F-Measure and it also includes the values of the Friedman statistic and P value for each test. Algorithms classification according to Friedman test on the values of Recall and F measure, shows that the best grouping is obtained using MulTimodalGroup in first place. STC is in second place and k-means in third place. This order corroborates results obtained of external evaluation, where best results were achieved by these algorithms and in the same Order. Regarding Precision, the best results were obtained by MulTimodalGroup followed by Lingo and k-means.

Table 2. Classification quality of algorithms according to Friedman test

Algorithm	Precision		Recall		F-Measure	
	Test	Order	Test	Order	Test	Order
MulTimodalGroup	1	1	1.3333	1	1	1
STC	4.9167	5	2.1667	2	2.3333	2
K-means	4.1667	3	2.6667	3	2.8333	3
Lingo	2.8333	2	3.8333	4	3.8333	4
FullStars	4.6667	4	2.1667	5	5.3333	5
Stars	5.4167	6	6.3333	6	6.4167	6
Cliques	5.0	7	6.5	7	6.25	7
Value P of test	0.004		1.2676 E-5		9.4605 E-6	
Friedman statistical	18.8750		32.5714		33.2321	

5. Related work

MulTimodal Group is focused on searching and grouping (clustering) of BP models, therefore, a summary of the most important works in these two topics are presented.

5.1 Approaches based on PN model search

These proposals are focused on a set of elements or data types present in BP models. For example, approaches based on linguistic are focused on activity name or description, events and existing logic gates in BP models. These approaches use during search process some techniques to recover information such as: space-vector representation with a frequency value (TF) of terms by PN and cosine distance to create a ranking of relevant results [34, 35].

Approaches based on association rules are focused on previous executions of BP, which are stored in log files. During search process some phrases related with business process activities are identified through domain ontologies; besides activities patterns are detected. To define the list of results, these approaches use heuristic components that determines appearance frequency of detected patterns [12, 36].

Approaches based on genetic algorithms transform BP models into a formal representation (for example graphs or state machines). These approaches also integrate additional data for the search process such as: number of inputs and outputs per node, edge labels, nodes name or description. Although these data provide more precise queries, the execution time is slow [37, 38].

Finally there are proposals focuses on searching in PN models repositories which use annotations in XML files. To search these proposals use a query language called IPM Process Query Language (IPM-PQL), which supports specific queries, such as: finding processes that contain particular activities, transitions or connections between activities [15, 39, 40].

5.2 Approaches based on grouping (Clustering) of PN models

These approaches use hierarchical clustering algorithms [17, 41], which build groups hierarchy based on similarity of structural and behavioral features of BP. In these proposals users check the created hierarchy and select the group that has greater similarity with their request [19, 42].

Secondly, BP sequential clustering takes as input data logs files of previous BP executions. In these proposals the algorithm groups BP models with the same type of behavior based on the execution flow and data flow [43, 44].

5.3 Differences with previous works

Approaches based on BP model search are limited to matching inputs and / or outputs, which use textual information of BP elements. The search process in these proposals leaves out execution flow, behavior, structure, type of activity, type of gates, and type of events.

On the other hand proposals based on grouping process use BP textual data. These data include: activities name and duration and number of errors. Besides, clustering eliminates activity sequences that occur only once without considering that these sequences may contain structural or textual information which may be relevant when selecting models that compose groups.

Despite contributions done by previously mentioned works, results can be extended if a larger number of information features is covered. These features are: activity description, task type, gates type, structure, behavior, among other BP models features. Focusing on a single topic, such as textual information, allows grouping of BP models by comparing names and descriptions without considering BP models that may share structures, task types or behavior.

From the previous analysis, the present work proposes an approach that unifies in a single search space: behavior structural units and textual features existing in BP models. The latter is known as multimodal search representation. Additionally, a clustering algorithm based on fuzzy logic is integrated. This algorithm uses different types of information (textual, structural and behavioral) for results grouping. Clustering is performed based on similarity of information types contained in retrieved BP models. The latter offers more effective way of results display.

6 Conclusions and future work

This paper proposes Multimodal Group a model for searching and clustering BPs models.

Based on the analysis of internal clustering in the compared algorithms, MulTimodalGroup gets best grouping regarding the measures employed: cohesion, coupling, silhouette, Sum-of-squares, Sum of squares between clusters, *Davis Bulding* index. Results show that using textual and structural information offers more compact BP groups due to the fact that grouped elements share relatively a subset of common features based on information types used for clustering. Moreover, overlapping elimination (BP Model elements that may exist in several groups at a time) allows having more similar elements within each group, which also offers greater separation between created groups.

Dinamic reassignment of elements (BP models) to the group with highest similarity, even after they have been initially assigned, allows more homogeneous groups. External evaluation of Multimodal Group shows a 70% of performance determined by grouping F-measure. The latter allows inferring that created groups are relevant and that these groups highly agree with groups created manually by experts. Accuracy levels achieved in each query shows that MulTimodalGroup creates relevant groups to user queries automatically without human intervention. in the grouping level. The list of groups created by MulTimodalGroup allows user s to browse categories and to select the most similar group to the request

The search is based on a multimodal presentation (it integrates structural and linguistic components existing in BP models) well-known in multimedia information retrieval field. Grouping is performed by an interactive and iterative clustering algorithm which applies a similarity function based on fuzzy logic on the set of results retrieved by a query.

As future work, the research group has as objective to complement the proposed model with specific domain-ontologies which allow including semantics to search, in order to have more precise results. Equally, future work will be focused on evaluation of labeling method to determine if created labels help users to identify more easily information and functionality defined in BP models existing in the groups to review. Finally, it will be incorporated a hierarchical clustering method to create categories and subcategories of BP models existing in the repository.

References

1. Zhao, X. and C. Liu, *Version management for business process schema evolution.* Information Systems, 2013. **38**: p. 1046-1069.
2. Reijers, H.A., R.S. Mans, and R.A. van der Toorn, *Improved model management with aggregated business process models.* Data & Knowledge Engineering, 2009. **68**(2): p. 221-243.
3. Rajnoha, R., A. Sujová, and J. Dobrovič, *Management and Economics of Business Processes Added Value.* Procedia - Social and Behavioral Sciences, 2012. **62**: p. 1292-1296.
4. Vukšić, V.B., M.P. Bach, and A. Popovič, *Supporting performance management with business process management and business intelligence: A case analysis of integration and orchestration.* International Journal of Information Management, 2013. **33**(4): p. 613-619.
5. Schlegel, T., et al., *Management of interactive business processes in decentralized service infrastructures through event processing.* Journal of King Saud University - Computer and Information Sciences, 2012: p. 137-144.
6. La Rosa, M., et al., *APROMORE: An advanced process model repository.* Expert Systems with Applications, 2011. **38**: p. 7029-7040.
7. Yan, Z., R. Dijkman, and P. Grefen, *Business process model repositories – Framework and survey.* Information and Software Technology, 2012. **54**: p. 380-395.
8. Goedertier, S., De Weerdt, J., Martens, D., Vanthienen, J., & Baesens, *Process discovery in event logs: An application in the telecom industry.* Applied Soft Computing, 2010.
9. Dijkman, R., M.L. Rosa, and H.a. Reijers, *Managing large collections of business process models—Current techniques and challenges.* Computers in Industry, 2012. **63**: p. 91-97.
10. Rembert, A.J. and C.S. Ellis, *An Initial Approach to Mining Multiple Perspectives of a Business Process.* Business: p. 35-40.
11. Ehrig, M., *Measuring Similarity between Semantic Business Process Models.* Reproduction, 2008. **5**: p. 10.
12. Rosso-Pelayo, D.a., et al., *Business Process Mining and Rules Detection for Unstructured Information.* 2010 Ninth Mexican International Conference on Artificial Intelligence, 2010: p. 81-85.
13. Weidlich, M., J. Mendling, and M. Weske, *on Behavioral Profiles of Process Models.* 2011. **37**: p. 410-429.
14. Cristhian Figueroa, J.C.C., *Business Process Retrieval based on Behavioral Semantics.* Revista EIA, 2012.
15. Kunze, M., A. Meyer, and M. Weske, *A Platform for Research on Process Model Collections.* Business Process Model and Notation - Lecture Notes in Business Information Processing 2012. **125**: p. 8-22.
16. Mahmod, N.M., *Structural similarity of business process variants.* 2010 IEEE Conference on Open Systems (ICOS 2010), 2010: p. 17-22.
17. Aiolli, F., A. Burattin, and A. Sperduti, *Metric for Clustering Business Processes Based on Alpha Algorithm Relations.* Business, 2011: p. 17.
18. Qiao, M., R. Akkiraju, and A.J. Rembert, *Towards Efficient Business Process Clustering and Retrieval: Combining Language Modeling and Structure Matching.* 2011. **57**: p. 199-214.
19. Melcher, J., D. Seese, and I. Aifb, *Visualization and Clustering of Business Process Collections Based on Process Metric Values.* Measurement, 2008. **8**: p. 9.
20. Ordoñez, H., J.C. Corrales, and C. Cobos, *Business Processes Retrieval based on Multimodal Search and Lingo Clustering Algorithm.* IEEE Latin America Transactions, 2015. **13**(9): p. 40-48.
21. Ordonez, H., et al., *Collaborative grouping of business process models.* EATIS 2014, Valparaiso -Chile, 2014: p. 1-2.
22. Gong, Y. and M. Janssen, *From policy implementation to business process management: Principles for creating flexibility and agility.* Government Information Quarterly, 2012. **29**: p. S61-S71.
23. Dijkman, R., et al., *Identifying refactoring opportunities in process model repositories.* Information and Software Technology, 2011. **53**: p. 937-948.
24. Chinosi, M. and A. Trombetta, *Computer Standards & Interfaces BPMN : An introduction to the standard.* Computer Standards & Interfaces, 2012. **34**: p. 124-134.
25. Carlos Cobos, M.M., Elizabeth León, Milos Manic, and Enrique Herrera-Viedma, *TopicSearch—Personalized Web Clustering Engine Using Semantic Query Expansion, Memetic Algorithms and Intelligent Agents.* Polibits, 3012. **47**.
26. WIVES, L.K., *Um estudo sobre Agrupamento de Documentos Textuais em Processamento de Informações não Estruturadas Usando Técnicas de "Clustering"*. Instituto de Informática da UFRGS, 1999. **Dissertação de mestrado**.
27. J.A. Aslam, E.P., and D. Rus, *The Star Clustering Algorithm for Information Organization.* Recent advances in clustering, Springer 2006.
28. Kowalski, G., *Information Retrieval Systems: Theory and Implementation.* Kluwer Academic Publishers, 1997. **1**: p. 290.
29. MARIA HALKIDI, Y.B., MICHALIS VAZIRGIANNIS, *On Clustering Validation Techniques.* Journal of Intelligent Information Systems, 2001.
30. Brun, M., et al., *Model-based evaluation of clustering validation measures.* Pattern Recognition, 2007. **40**(3): p. 807-824.
31. Szyma´nski, K.D.a.J., *External Validation Measures for Nested Clustering of Text Documents.* Springer-Verlag Berlin Heidelberg - Emerging Intelligent Technologies in Industry,, 2011.
32. Baeza-Yates, R., A. and B. Ribeiro-Neto, *Modern Information Retrieval.* Addison-Wesley Longman Publishing, 1999.
33. Manning, C.D., P. Raghavan, and H. Schutze, *Introduction to Information Retrieval.* 2008.
34. Koschmider, A., T. Hornung, and A. Oberweis, *Recommendation-based editor for business process modeling.* Data & Knowledge Engineering, 2011. **70**(6): p. 483-503.

35. Reijers, H.A., et al., *Syntax highlighting in business process models.* Decision Support Systems, 2011. **51**(3): p. 339-349.
36. Huang, Z., et al., *Business Process Decomposition Based on Service Relevance Mining.* 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2010: p. 573-580.
37. Chris J. Turner, A.T., Jorn Mehnen, *A Genetic Programming Approach to Business Process Mining.* 2010.
38. Li, C., M. Reichert, and A. Wombacher, *Mining business process variants: Challenges, scenarios, algorithms.* Data & Knowledge Engineering, 2011. **70**(5): p. 409-434.
39. Kunze, C.R.a.M., *An Extensible Platform for Process Model Search and Evaluation.* Business Process Management 2013. **Demos 2013: Beijing, China**.
40. Yan, Z., R. Dijkman, and P. Grefen, *Business process model repositories – Framework and survey.* Information and Software Technology, 2012. **54**(4): p. 380-395.
41. Engineering, I.S. and U. States, *Hierarchical clustering of business process models.* Computer, 2009. **5**: p. 613-616.
42. Diamantini, C., D. Potena, and E. Storti, *Clustering of Process Schemas by Graph Mining Techniques (Extended Abstract).* Methodology, 2011. **4**: p. 7.
43. Ferreira, D.R., *Applied Sequence Clustering Techniques for Process Mining.* Science, 2009: p. 492-513.
44. Ferreira, D., et al., *Approaching Process Mining with Sequence Clustering: Experiments and Findings.* Engineering, 2008. **7**: p. 1-15.