

Universidad de las Ciencias Informáticas
Facultad 3



*Algoritmo para la detección de subprocesos mediante la
alineación de las trazas usadas en la minería de procesos.*

*Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas*

Autores: Alkaid Cruz Llanes Hernández.

Maikel Pino Marrero.

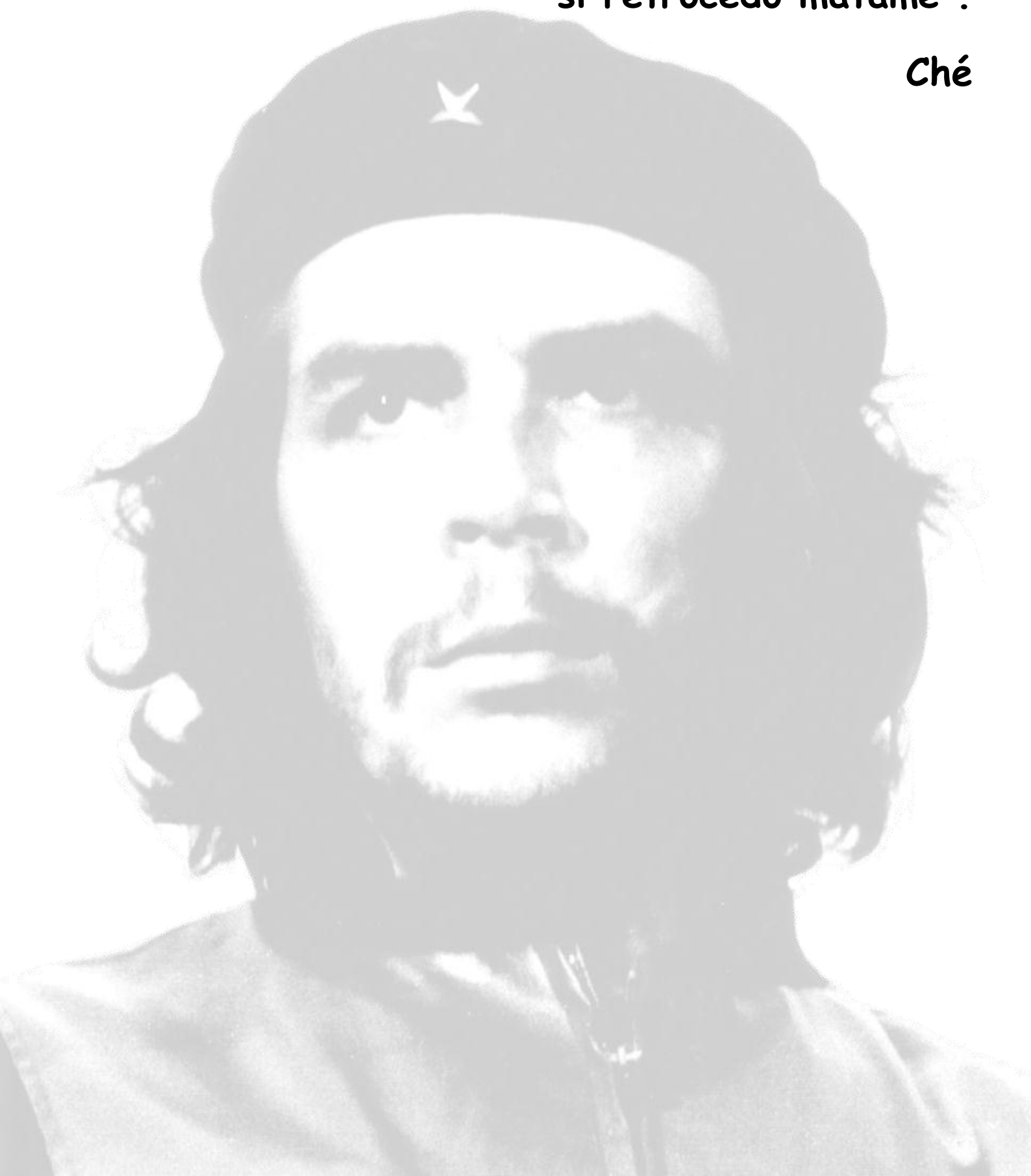
Tutor: MSc. Raykenler Yzquierdo Herrera.

La Habana, Cuba

Junio, 2012

**"Si avanzo sígueme, si me detengo empújame,
si retrocedo mátame".**

Ché



Declaración de Autoría

**MINISTERIO DE EDUCACIÓN SUPERIOR
UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

Declaramos ser autores del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Autorizamos a dicho centro para que haga el uso que estime pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Alkaid Cruz Llanes Hernández

Maikel Pino Marrero

Firma de los autores

Raykenler Yzquierdo Herrera

Firma del tutor

Datos de Contacto

Raykenler Yzquierdo Herrera. Especialidad de graduación Ingeniero en Ciencias Informáticas, categoría docente Instructor, categoría científica Máster en Gestión de Proyectos Informáticos, cuenta con 2 años de experiencia en el tema y 5 años de graduado de la Universidad de las Ciencias Informáticas. Correo Electrónico: ryzquierdo@uci.cu.

Agradecimientos

A mis padres que son los máximos responsables de mis resultados hasta ahora y porque siempre me han apoyado en todos los momentos y todas las decisiones que he tomado.

A mi tía Marlenis por ser como una segunda madre para mí, además de ser un ejemplo a seguir como persona y como profesional.

A mi difunto abuelo Melquiades por enseñarme tantas cosas de la vida y de la persona.

A mis primos Dixán y Kenny por creer en mí y por inspirarme a ser quien soy hoy.

A mi novia Yaillet por compartir conmigo estos años en las buenas y en las malas.

A mi tutor Raykenler por ser tan profesional y tan inteligente además de haberme brindado su amistad.

A Damián por haberme dado su apoyo durante la investigación, gracias por las críticas y recomendaciones.

A mis amigos de siempre Arturo, Leonardo y Alexander.

A mi amigo de la universidad Osvaldo por todos los momentos que hemos pasado juntos y todas las cosas que hemos vivido.

A mi compañero de tesis y amigo Alkaid por que me ha enseñado muchas cosas y por brindarme su amistad.

A Jorge, Jordan y Leiser, Félix.

A personas como Leisniel, Leonardo y Juan Ernesto que me han ayudado mucho en la universidad y por tener su amistad.

Maikel.

A mis padres por ser mi apoyo y mi sostén.

A mi hermana para enseñarme siempre y por confiar en mí.

A mi tío Pablo por ser un gran amigo.

A toda mi familia por creer en mí desde los primeros años.

A todos mis amigos, los del preuniversitario, preselección, universidad.

*A Adrián, a Manuel, al Vlady, porque en las buenas y malas han estado ahí,
por ser mis hermanos.*

A Marichal y Yandry por creer siempre.

A mis compañeros de Bankai, Olimpo, Vitamin AC y a Las Yucas.

A Luis y Maikel, los únicos compañeros de tesis que podría tener.

A Raykenler por ayudarme a creer en lo imposible.

Dedicatoria

A mis primos Dixon y Kenny

Maikel

A Yarlín

Alkaid

Resumen

En la actualidad las empresas se gestionan mediante un enfoque a procesos de negocio y lo hacen apoyándose en sistemas de información tales como los sistemas de planificación de recursos empresariales (ERP por sus siglas en inglés). Estos sistemas guardan en forma de traza cada una de las acciones que tienen lugar al ejecutarse las instancias o casos de los procesos de negocio. Las técnicas de minería de proceso se basan en la extracción de conocimiento no trivial y elementos interesantes del registro de evento. La minería de procesos provee un nuevo y bien recibido repertorio de técnicas de análisis de procesos de negocio y ha sido adoptada en varios sistemas comerciales. Desafortunadamente, los algoritmos tradicionales de descubrimiento del proceso tienen problema al lidiar con procesos desestructurados. Los modelos resultantes pueden resultar incomprensibles. El diagnóstico del proceso representa una vía para el entendimiento del proceso previo a la utilización de alguna técnica de minería, además permite análisis de rendimiento, detección de anomalías e inspección de patrones comunes presentes en las trazas. Las técnicas desarrolladas en esta área presentan problemas en el análisis de rendimiento, detección de anomalías e inspección de patrones interesantes a diferentes niveles de abstracción. En el presente trabajo se presenta una nueva técnica para el apoyo al diagnóstico manejando los diferentes niveles de abstracción.

Palabras Clave

Diagnóstico de proceso, minería de proceso, procesos de negocio, registro de eventos.

Contenido	
Declaración de Autoría	3
Datos de Contacto	4
Agradecimientos	5
Dedicatoria	7
Resumen	9
Introducción	12
Capítulo 1. Fundamentación Teórica	17
1.1. Minería de procesos	17
1.2. Diagnóstico de proceso	23
1.3. Trabajos relacionados con el diagnóstico del proceso	25
1.3.1. Análisis de Diagrama de Puntos	25
1.3.2. Stream Scope Visualization	27
1.3.3. Tandem Arrays.....	27
1.3.4. Descubrimiento de patrones usando árbol de sufijos	27
1.3.5. Alineación de trazas.....	28
1.3.6. Chequeo de conformidad.....	30
1.3.7. Minería Difusa	32
1.3.8. Minería basada en teoría de regiones	33
1.3.9. Agrupación Difusa.....	33
1.4. Conclusiones del capítulo	35
Capítulo 2. Propuesta de solución	36
2.1. Introducción	36
2.2. Algoritmo para el diagnóstico del proceso.....	36
2.3. Pasos generales que conforman el algoritmo propuesto.....	38
2.3.1. Alinear las trazas.....	38
2.3.2. Pre-procesar las trazas alineadas	38

2.3.3. Determinar el árbol de bloques de construcción	38
2.4. Conclusiones del capítulo	49
Capítulo 3. Validación de la solución	51
3.1. Introducción	51
3.2. Aplicación en un entorno real.....	51
3.2.1. Proceso Gestionar Roles	51
3.2.2. Proceso Gestionar Recursos.....	55
3.3. Conclusiones del capítulo	59
Conclusiones Generales	60
Recomendaciones	61
Glosario de términos	62
Bibliografía	64

Introducción

La mayoría de las empresas del mundo utilizan sistemas de información para la gestión de sus procesos de negocio. Ejemplo de estos tipos de sistemas son los utilizados para la gestión de procesos de negocio (BPM por sus siglas en inglés), sistemas de Planificación de los Recursos Empresariales (ERP por sus siglas en inglés), sistemas de Gestión de Cadenas de Suministros, sistemas de Gestión de la Relación con los Clientes (CRM por sus siglas en inglés), entre otros (HENDRICKS *et al.* 2007). Los sistemas de información registran en forma de trazas las acciones que se van realizando en la medida que se van ejecutando las instancias o casos de los procesos de negocio. Al descubrimiento de procesos a partir de la información contenida en las trazas se le denomina Minería de Procesos o Minería de Flujos de Trabajo (AGRAWAL *et al.* 1998; COOK and WOLF. 1995). El descubrimiento del modelo de procesos basado en las trazas permite hacer comparaciones del modelo prescrito o teórico y el modelo descubierto, además posibilita apoyar el diseño de los procesos de negocio. Investigaciones recientes describen sus aplicaciones en la “operacionalización” de los procesos empresariales. La idea de la minería de procesos es descubrir, monitorear y mejorar los procesos reales (procesos no ficticios) mediante la extracción del conocimiento de las trazas existentes en los sistemas de información de hoy en día (AALST, W.M.P. VAN DER 2011; AALST, W.M.P. VAN DER and WEIJTERS 2004).

La minería de proceso es un área joven en la investigación, pero a la vez poderosa, que permite hacer un análisis de los procesos basado en su funcionamiento actual. Tiene entre otros beneficios el descubrimiento de modelos representativos de la realidad, también la minería de proceso permite realizar un chequeo de conformidad para detectar desviaciones a partir de la comparación entre el registro de evento y el modelo representativo del proceso. Permite el descubrimiento de redes sociales y modelos organizacionales. Se puede extender el modelo de proceso teniendo en cuenta un análisis del funcionamiento del mismo y construir automáticamente modelos de simulación. Las técnicas de minería de proceso permiten, entre otras, hacer recomendaciones y predicciones teniendo en cuenta el análisis de los datos actuales y de los históricos.

Existen tres perspectivas a la hora de analizar un proceso, teniendo en cuenta el control de flujo, los recursos (organizacionales) y por último los casos. Se escoge una o varias perspectivas a analizar en dependencia de la información contenida en las trazas. La más común en cuanto a su uso es la de control de flujo la cual posibilita determinar cómo se organizan las tareas que dan lugar al proceso. Desde esta perspectiva se debe tener en cuenta la información relacionada con la identificación del

caso o instancia ejecutada, identificación de la tarea y el tiempo en que se produjo la tarea. Desde la perspectiva del recurso se debe tener en cuenta la información referente al usuario del sistema que ejecutó la tarea, y es posible determinar la estructura organizacional con respecto a la cual se realizan las diferentes tareas. Lo antes expuesto permite conocer las jerarquías que existen entre roles y usuarios durante el proceso. En la tercera perspectiva, que está relacionada con los casos, se debe tener en cuenta la información, en detalle, de las tareas como la edad y el sexo de los usuarios implicados. Desde esta perspectiva se puede descubrir casos excepcionales, como la probabilidad de que personas de determinado sexo realicen en tiempo determinada tarea.

Los procesos que presentan una estructura clara y en la mayoría de los casos presentan un comportamiento esperado son conocidos como procesos Lasaña. Normalmente, un proceso Lasaña es un proceso que está casi totalmente automatizado, donde las actividades tienen bien definidas las entradas y salidas. Los procesos Lasaña tienen relativamente pocas excepciones, y los actores del negocio tienen un entendimiento razonable de los flujos de trabajo. Es difícil definir un requisito formal que caracterice los procesos Lasaña, no obstante, como regla general de Van der Aalst (AALST, W.M.P. VAN DER 2011) usa el siguiente criterio no formal: un proceso es un proceso Lasaña si con limitados esfuerzos es posible descubrir un modelo de proceso que tenga como valor para la métrica fitness (idoneidad) al menos 0.8, es decir, más del 80 % de los eventos ocurren como están previstos y los interesados confirman la validez del modelo.

No obstante, la mayoría de los procesos en la vida real no son estrictamente regulados por los sistemas de información que los sostienen. Esto significa que aunque los actores tengan una noción del proceso, estos pueden ignorarlo parcial o totalmente. Estos ambientes flexibles se aprecian en áreas como la salud, el desarrollo de productos o la atención al cliente. En este contexto el descubrimiento de los procesos ejecutados se puede emplear para identificar posibles mejoras o para apoyar el proceso de control de la calidad. Cuando aplicamos las técnicas desarrolladas de minería de procesos surge un conjunto de problemas. Normalmente, aparece un amplio espectro de posibles comportamientos, en consecuencia, se obtienen modelos de procesos desestructurados. Los típicos problemas observados en estos modelos de procesos resultantes son la aparición de una gran cantidad de nodos de trabajo, y también un gran número de relaciones entre estos, resultando un modelo estereotipado como Espagueti. Es necesario señalar que estos problemas son inherentes a la técnica de minería de procesos que se use, pues el proceso es desestructurado por lo cual se descubrirá un modelo que refleja este comportamiento (BOSE, R.P. JAGADEESH CHANDRA and AALST 2010; DONGEN, B.F. VAN and ADRIANSYAH 2010b; SONG *et al.* 2009).

Uno de los retos actuales en la minería de procesos lo constituye el diagnóstico del proceso. El diagnóstico del proceso puede ser útil a la hora de tratar procesos desestructurados. Esta área del conocimiento abarca el análisis del rendimiento del proceso, la detección de anomalías, el diagnóstico e inspección de patrones interesantes. Sin embargo, las técnicas desarrolladas en esta área presentan problemas a la hora de identificar claramente los patrones más comunes de ejecución y las desviaciones en el registro de evento, considerando también, diferentes niveles de abstracción entre las tareas.

Problema:

¿Cómo se pueden detectar los patrones de flujo de trabajo más comunes, las desviaciones y los diferentes niveles de abstracción existentes entre las actividades durante la fase de diagnóstico del proceso?

Objetivo general:

Desarrollar un algoritmo para la detección de subprocessos mediante la alineación de las trazas de procesos que permita detectar los patrones más comunes de flujo de trabajo, las desviaciones y los diferentes niveles de abstracción existentes entre las actividades.

Objetivos específicos:

- Fundamentar la investigación mediante la evaluación del marco teórico.
- Diseñar algoritmo para la detección de subprocessos mediante la alineación de las trazas de procesos desestructurados que permita detectar los patrones más comunes de flujo de trabajo, las desviaciones y los diferentes niveles de abstracción existentes entre las actividades.
- Implementar el algoritmo diseñado.
- Validar la propuesta.

Objeto de estudio: Minería de proceso.

Campo de acción: El diagnóstico del proceso en la minería de proceso.

Idea a defender: Un algoritmo para la detección de subprocessos mediante la alineación de las trazas de procesos permite detectar los patrones más comunes de flujo de trabajo, las desviaciones y los diferentes niveles de abstracción existentes entre las actividades.

Posibles resultados: Un algoritmo para el tratamiento de procesos tanto estructurados como desestructurados en la minería de proceso.

Tareas a cumplir por estudiantes:

- Análisis de los principales conceptos y trabajos relacionados con la minería de proceso y el tratamiento de procesos desestructurados.
- Valoración de los mecanismos de alineación de las secuencias potencialmente útiles para la minería de proceso.
- Estudio de los patrones de flujo de trabajo Secuencia y XOR split-join y su tratamiento en la minería de proceso.
- Estudio de los patrones de flujo de trabajo AND split-join y su tratamiento en la minería de proceso.
- Diseño de un algoritmo que permita reconocer los patrones de flujo de trabajo Secuencia y XOR split-join a nivel de subprocesos considerando diferentes niveles de completitud de los datos.
- Diseño de un algoritmo que permita reconocer los patrones de flujo de trabajo AND split-join a nivel de subprocesos considerando diferentes niveles de completitud de los datos.
- Implementación de los algoritmos desarrollados como bibliotecas que permitan la reutilización del código y el desarrollo rápido de aplicaciones para la minería de proceso.
- Diseño del experimento que permitirá la evaluación de la propuesta de solución.
- Evaluación de los algoritmos y herramientas desarrolladas.

Métodos teóricos:

- Histórico lógico
- Hipotético deductivo
- Analítico-Sintético
- Sistémico

Se enfocan las problemáticas asociadas al diagnóstico del proceso desde un enfoque histórico lógico, en la primera parte de la investigación se desarrolla un estudio del estado del arte de la problemática analizada; dando detalles de las bondades y deficiencias de cada uno de los métodos y las tendencias en la resolución de esta problemática.

La investigación sigue además un método hipotético deductivo porque a partir del problema concreto se plantean objetivos específicos e idea a defender que en el transcurso de la investigación son resueltas siguiendo métodos científicamente bien fundamentados.

El método Analítico-Sintético se sigue al descubrir los distintos elementos que componen la naturaleza o esencia asociada al fenómeno del diagnóstico del proceso. Definiéndose las causas y los efectos, para posteriormente integrar los elementos en una unidad nueva, en una comprensión total de la esencia de lo que ya se conoce en todos sus elementos y particularidades.

En cada caso se plantea el problema como un todo, donde las trazas utilizadas, la propia dinámica de aplicación de técnicas de minería de procesos en el descubrimiento de procesos, las técnicas computacionales desarrolladas para el diagnóstico del proceso se funden en un sistema sostenible e integral.

El presente documento está estructurado en: resumen, introducción y tres capítulos de los cuales a continuación se sintetiza su contenido.

Capítulo 1: Se discute las ideas básicas y las diferentes estrategias utilizadas en la minería de proceso y específicamente las relacionadas con el diagnóstico del proceso. Se hace una evaluación crítica de las ventajas y desventajas de los diferentes enfoques.

Capítulo 2: Se describe el algoritmo para la detección de subprocesos mediante la alineación de las trazas de procesos desestructurados que permita detectar los patrones de flujo de trabajo más comunes, las desviaciones y los diferentes niveles de abstracción existentes entre las actividades.

Capítulo 3: Se describe el mecanismo de validación de la propuesta y se presentan los resultados de su aplicación.

Capítulo 1. Fundamentación Teórica

1.1. Minería de procesos

Las técnicas de minería de proceso o de flujo de trabajo, permiten extraer información no trivial y útil de los registros de eventos almacenados por sistemas de información. Un elemento fundamental en la minería de proceso es el descubrimiento del flujo de trabajo, esto es la construcción automática del modelo de proceso (por ejemplo una red de Petri) en el que se describen las dependencias causales entre las actividades de procesos (AALST, W.M.P. VAN DER 2011).

La minería de proceso provee un importante puente entre la minería de dato y el modelado y análisis de los procesos de negocio. Bajo el área de BI (siglas de Business Intelligence) se han difundido un grupo de términos que encierran diferentes tipos de análisis en este contexto, tales como:

- BAM (siglas de Business Activity Monitoring) referido a las tecnologías que hacen posible un análisis en tiempo real de los procesos de negocio. (AALST, W.M.P. VAN DER *et al.* 2011a)
- CEP (siglas de Complex Event Processing) referido a las tecnologías que permiten procesar grandes cantidades de eventos para monitorear, guiar y optimizar el negocio en tiempo real. (AALST, W.M.P. VAN DER *et al.* 2011a)
- CPM (siglas de Corporate Performance Management) referido a la medición del funcionamiento del proceso o la organización. (AALST, W.M.P. VAN DER *et al.* 2011a)

Otros términos están vinculados con la gestión, como es el caso de:

- CPI (siglas de Continuous Process Improvement).(AALST, W.M.P. VAN DER *et al.* 2011a)
- BPI (siglas de Business Process Improvement).(AALST, W.M.P. VAN DER *et al.* 2011a)
- TQM (siglas de Total Quality Management).(AALST, W.M.P. VAN DER *et al.* 2011a)
- Six Sigma.(AALST, W.M.P. VAN DER *et al.* 2011a)

Las investigaciones en estas áreas tienen en común que los procesos son “empujados bajo el microscopio” con el objetivo de identificar posibles mejoras. La minería de proceso puede considerarse una tecnología que contribuye a cada una de las áreas antes mencionadas (AALST, W.M.P. VAN DER 2011; AALST, W.M.P. VAN DER *et al.* 2011a; AALST, W.M.P. VAN DER *et al.* 2007; GRIGORI *et al.* 2004).

En la última década se ha hecho común el tratamiento de las trazas o registro de evento y cada vez, más sistemas de información incorporan técnicas de minería de proceso. Algunos de los *software* que realizan minería de proceso son:

- ARIS Process Performance Manager (desarrollado por Software AG).(AALST, W.M.P. VAN DER *et al.* 2011a)
- Comprehend (desarrollado por Open Connect).(AALST, W.M.P. VAN DER *et al.* 2011a)
- Discovery Analyst (desarrollado por StereoLOGIC).(AALST, W.M.P. VAN DER *et al.* 2011a)
- Flow (desarrollado por Fourspark).(AALST, W.M.P. VAN DER *et al.* 2011a)
- Futura Reflect (desarrollado por Futura Process Intelligence).(AALST, W.M.P. VAN DER *et al.* 2011a)
- Interstage Automated Process Discovery (desarrollado por Fujitsu).(AALST, W.M.P. VAN DER *et al.* 2011a)
- OKT Process Mining suite (desarrollado por Exeura).(AALST, W.M.P. VAN DER *et al.* 2011a)
- Process Discovery Focus (desarrollado por Iontas/Verint).(AALST, W.M.P. VAN DER *et al.* 2011a)
- Process Analyzer (desarrollado por QPR).(AALST, W.M.P. VAN DER *et al.* 2011a)
- ProM (desarrollado por TU/e) (AALST, W. M. P. V. D. *et al.* 2009a).(AALST, W.M.P. VAN DER *et al.* 2011a)
- Rbminer/Dbminer (desarrollado por UPC).(AALST, W.M.P. VAN DER *et al.* 2011a)
- Reflect | one (desarrollado por Pallas Athena).(AALST, W.M.P. VAN DER *et al.* 2011a)

En las Figuras 1.1 y 1.2 respectivamente se muestran imágenes del ARIS Process Performance Manager desarrollado por Software AG y ProM desarrollado por TU/e. Estas son herramientas para el tratamiento de las trazas, o sea, herramientas que hacen minería de proceso.

Como punto de partida para la aplicación de técnicas de minería de proceso están las trazas (contenidas en un registro de evento), las cuales han sido generadas por uno o varios sistemas de

información usados en la empresa. Las trazas están constituidas por una secuencia de eventos de los cuales se almacena información como: el nombre del evento, el tipo de eventos, el usuario que ejecutó el evento, el tiempo en el que se produjo el evento.

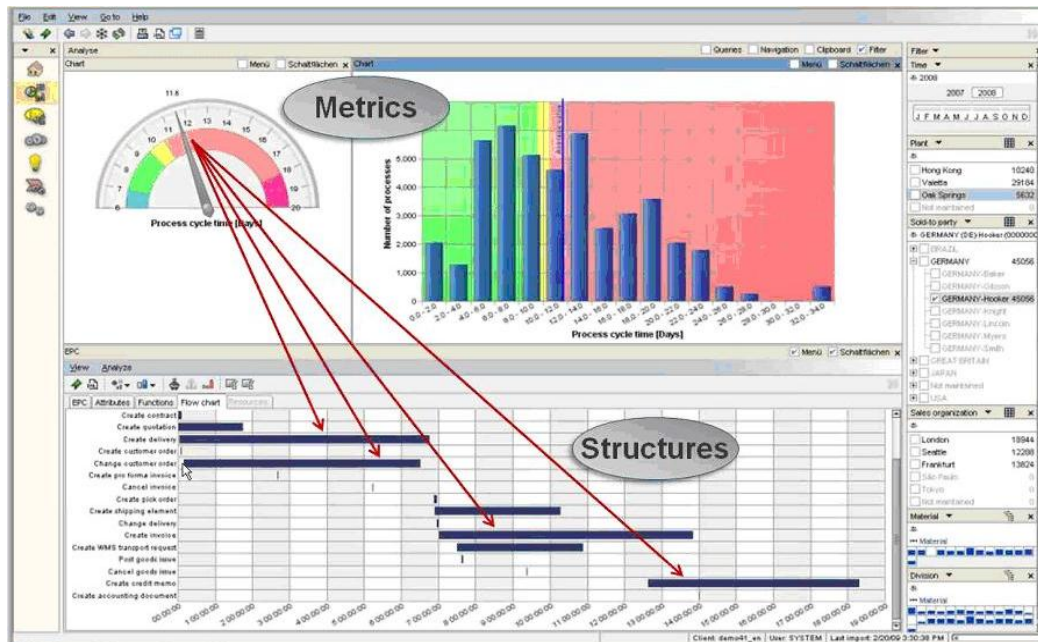


Figura 1.1 ARIS Process Performance Manager.



Figura 1.2. PROM.

A continuación se expone la definición de registro de evento y traza:

Definición 1 (Traza y registro de evento). Se denota por Σ el conjunto de todas las actividades. Σ^+ es el conjunto de todas las secuencias finitas de actividades no vacías sobre Σ . Cada $T \in \Sigma^+$ es una posible traza. Un registro de evento \mathcal{L} es un grupo de trazas (BOSE, R. P. JAGADEESH CHANDRA and AALST 2012).

Para la representación de un registro de evento se han definido dos estándares MXML y XES. Un ejemplo de un registro de evento se muestra en la Figura 1.3.

Caso 1			
Descripción	Evento	Usuario	yyyy/mm/dd hh:mm
	inicio	ryzquierdo	10/06/2011 10:50
Registrarse	procesando	ryzquierdo	10/06/2011 10:50
Registrarse	terminado	ryzquierdo	10/06/2011 10:50
Enviar cuestionario	procesando	ryzquierdo	10/06/2011 10:50
Evaluar	procesando	ryzquierdo	10/06/2011 10:50
Enviar cuestionario	terminado	ryzquierdo	10/06/2011 10:52
Recibir cuestionario	procesando	ryzquierdo	10/06/2011 10:52
Recibir cuestionario	terminado	ryzquierdo	10/06/2011 10:52
Evaluar	terminado	ryzquierdo	10/06/2011 10:52
Archivar	procesando	ryzquierdo	10/06/2011 10:52
Archivar	terminado	ryzquierdo	10/06/2011 10:52
	final		10/06/2011 10:52
Caso 2			
Descripción	Evento	Usuario	yyyy/mm/dd hh:mm
	inicio	ryzquierdo	10/06/2011 10:58
Registrarse	procesando	ryzquierdo	10/06/2011 10:58
Registrarse	terminado	ryzquierdo	10/06/2011 10:58
Enviar cuestionario	procesando	ryzquierdo	10/06/2011 10:58
Evaluar	procesando	ryzquierdo	10/06/2011 10:58
Enviar cuestionario	terminado	ryzquierdo	10/06/2011 10:58
Recibir cuestionario	procesando	ryzquierdo	10/06/2011 10:58
Recibir cuestionario	terminado	ryzquierdo	10/06/2011 10:58
Evaluar	terminado	ryzquierdo	10/06/2011 10:59
Proceso de queja	procesando	ryzquierdo	10/06/2011 10:59
Proceso de queja	terminado	ryzquierdo	10/06/2011 10:59
Proceso de chequeo	procesando	ryzquierdo	10/06/2011 11:00
Proceso de chequeo	terminado	ryzquierdo	10/06/2011 11:00
Archivar	procesando	ryzquierdo	10/06/2011 11:00
Archivar	terminado	ryzquierdo	10/06/2011 11:00
	final		10/06/2011 11:00

Figura 1.3. Registro de eventos.

Existen tres tipos de técnicas de minería de proceso como se muestra en la Figura 1.4. El primer tipo de técnica de minería de proceso es el descubrimiento. Este tipo de técnica permite descubrir un

modelo representativo del proceso ejecutado en la empresa a partir de un registro de evento. Existe una gran variedad de investigaciones desarrolladas en este sentido, entre los algoritmos desarrollados se puede mencionar: Alpha (AALST, W.M.P. VAN DER *et al.* 2004; GOEDERTIER *et al.* 2008; GOEDERTIER *et al.* 2009; MEDEIROS *et al.* 2003; ROZINAT *et al.* 2008), Genetic Miner (MEDEIROS 2006), Heuristic Miner (WEIJTERS and AALST 2003; WEIJTERS and RIBEIRO 2010), Transition System Miner (AALST, W.M.P. VAN DER *et al.* 2009b) y Fuzzy Miner (GÜNTHER and AALST 2007) entre otros.

Los modelos descubiertos a partir de la aplicación de técnicas de minería de proceso pueden ser representados utilizando diferentes notaciones, entre las más empleadas se encuentran las Redes de flujos de trabajo (basadas en las redes de Petri) (AALST, W.M.P. VAN DER 1996a; 1996b; AALST, WIL M. P. VAN DER *et al.* 2003), EPCs (siglas en inglés de *Event-driven Process Chains*) (DONGEN, B.F. VAN and AALST 2004), TS (siglas en inglés de *Transition System*) (RUBIN 2007), BPMN, diagrama de actividades UML (AALST, W.M.P. VAN DER *et al.* 2011a).

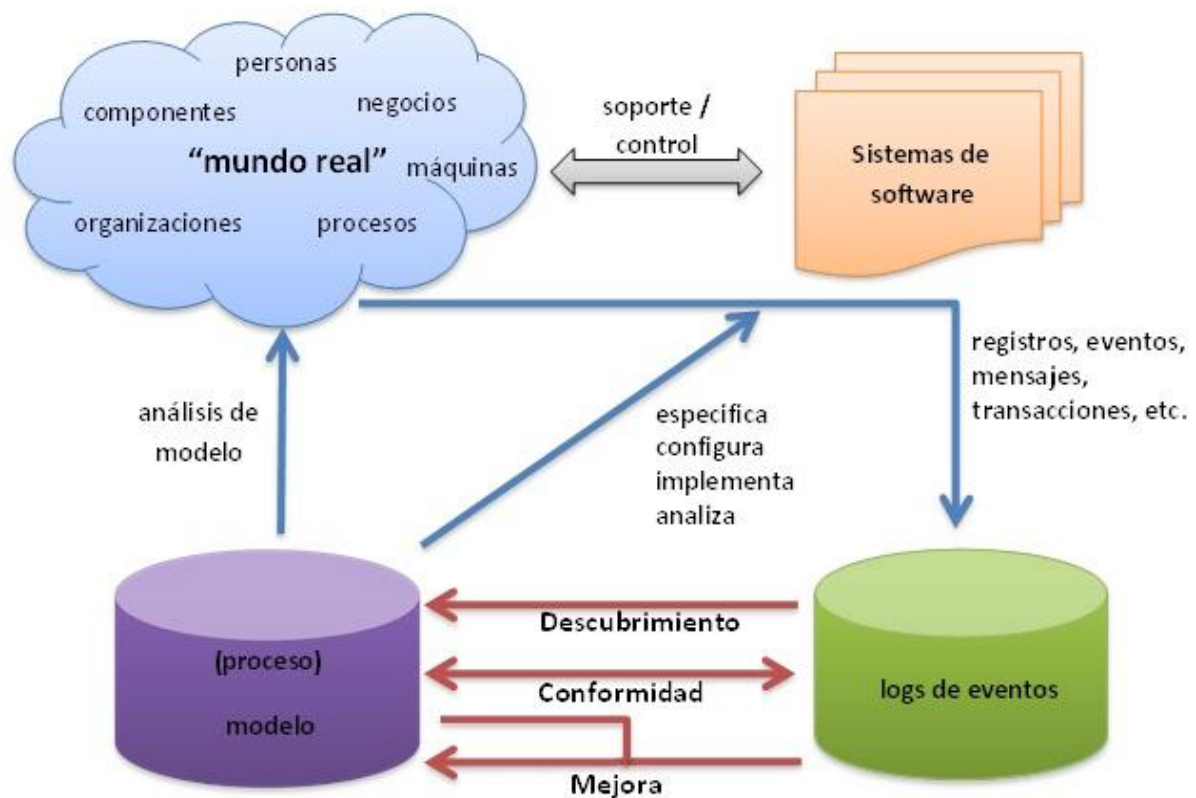


Figura 1.4. Representación de los tres tipos fundamentales de técnicas de minería de proceso: descubrimiento, conformidad y mejora.

El segundo tipo de técnica es la conformidad o chequeo de conformidad (GAMA and CARMONA 2010; MENDLING *et al.* 2007; ROZINAT and AALST 2008). Aquí, un modelo de procesos existente es comparado con un registro de eventos del mismo proceso. El chequeo de conformidad es un análisis que permite saber hasta qué punto el registro de eventos se corresponde con el modelo de procesos y viceversa. Diferentes tipos de modelos pueden ser usados en el chequeo de conformidad, tales como, modelos normativos o descriptivos, modelos organizacionales, reglas y políticas del negocio, leyes, etc.

Por último, se encuentra la mejora. Este tipo de técnica permite la extensión del conocimiento que se tiene del proceso de negocio o la mejora de este. A partir de un modelo de proceso existente y el registro de eventos correspondiente al mismo proceso se detectan aspectos como cuellos de botella, niveles de servicio, tiempos de espera y ejecución, frecuencia de algún evento, entre otros. Estos aspectos pueden reflejarse en un nuevo modelo de proceso (AALST, W.M.P. VAN DER 2010; 2011; AALST, W.M.P. VAN DER *et al.* 2011b; AALST, W.M.P. VAN DER *et al.* 2011c).

Como se había mencionado anteriormente un proceso puede ser analizado considerando las siguientes tres perspectivas o dimensiones, la perspectiva del flujo de trabajo, la de los recursos (también llamada organizacional) y la de casos. Esto ha permitido que las técnicas de minería de proceso desempeñen un papel importante en el ciclo de vida de BPM (siglas en inglés de *Business Process Management*). La Figura 1.5 muestra un esquema del ciclo de vida de BPM.

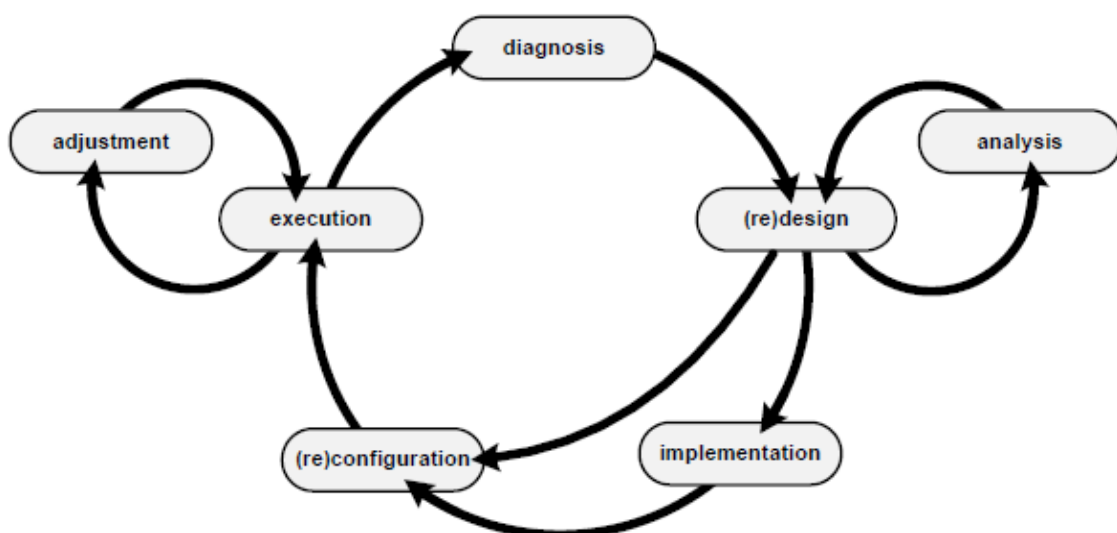


Figura 1.5. Ciclo de vida de BPM.

El ciclo de BPM muestra siete fases del proceso de negocio y el correspondiente sistema de información que lo soporta. En la fase de (re)diseño un nuevo modelo de proceso es creado o uno existente es adaptado. En la fase de análisis el modelo candidato y sus alternativas son analizados. Después del (re)diseño el modelo es implementado o en caso de existir una implementación puede ser (re)configurado considerando las modificaciones realizadas en el (re)diseño. En la etapa de ejecución el proceso informatizado se ejecuta. Durante la ejecución del proceso este es monitoreado, esto permite realizar pequeños ajustes (fase de ajustes) sin necesidad de realizar un rediseño del proceso. En la fase de diagnóstico el proceso ejecutado es analizado y los resultados pueden ser empleados en la fase de (re)diseño. Aunque la minería de proceso es utilizada fundamentalmente en la fase de diagnóstico puede utilizarse en otras fases. Por ejemplo en la fase de ejecución puede ser utilizada para el soporte operacional. En este sentido, se pueden hacer recomendaciones y predicciones a partir de los modelos descubiertos usando los datos históricos y los datos actuales de la ejecución. En la etapa de (re)configuración pueden emplearse también los análisis realizados usando la minería de proceso (AALST, W.M.P. VAN DER *et al.* 2011a).

1.2. Diagnóstico de proceso

La mayor parte de las investigaciones en la minería de proceso se ha dirigido al descubrimiento del flujo de trabajo. Abundantes son las experiencias en las cuales al aplicar las técnicas de minería de proceso, los procesos analizados se muestran menos estructurados que lo esperado. Por otra parte, los registros de evento tienden a ser incompletos, es decir, solo una fracción del posible comportamiento del proceso realmente ocurre. Por lo tanto, puede ser demasiado ambicioso esperar que el algoritmo de descubrimiento arroje como resultado un modelo que contemple lo que puede ocurrir en el futuro. Si los procesos son poco estructurados y los registros de eventos son incompletos, entonces resulta útil realizar un diagnóstico en etapas tempranas del análisis del proceso. El diagnóstico del proceso incluye análisis de rendimiento, detección de anomalías e identificación de patrones comunes (BOSE, R.P. JAGADEESH CHANDRA and AALST 2010). El diagnóstico ayuda a tener una visión general del proceso, de los aspectos más significativos del mismo y de las técnicas que pueden ser más útiles en el posterior análisis.

Las técnicas de diagnóstico de proceso permiten responder un conjunto de preguntas que surgen al analizar un proceso. Se enuncian a continuación algunas de estas.

1. ¿Cuál es el comportamiento más común que se ejecuta del proceso?

Dado un registro de evento sería interesante conocer qué componentes del proceso son críticos o esenciales. Esos componentes esenciales forman la estructura principal del proceso y deberían ser conservados. Los esfuerzos de mejoras del proceso deben enfocarse a dichos componentes.

2. ¿Dónde las instancias del proceso se desvían y que tienen estas en común?

En la práctica, a menudo existe una brecha significativa entre el modelo prescrito o lo que se supone debe suceder, y lo que realmente sucede. Existe una necesidad de aumentar el diagnóstico del proceso con técnicas que pueden ayudar a encontrar desviaciones analizando las trazas en el registro de evento. Hay muchas aplicaciones que tienen este requerimiento, por ejemplo, diagnóstico de fallas, detección de anomalías, diagnóstico de seguros fraudulentos, etc. Dado un registro de evento que contiene un conjunto de trazas que se generaron por funcionamiento normal del proceso y por mal funcionamiento del proceso, un análisis de estas trazas para encontrar desviaciones en trazas anómalas de trazas normales puede dar señales en el entendimiento de la anomalía o mal funcionamiento.

3. ¿Hay patrones comunes de ejecución en las trazas?

Un analista puede estar interesado en conocer cuando hay patrones de ejecución interesantes en el registro de eventos. La ausencia o presencia de estos patrones de ejecución puede ser el indicador de una anomalía o una violación de seguridad o un malfuncionamiento.

4. ¿En qué contexto se ejecuta una actividad o un conjunto de actividades en un registro de eventos?

Existen dependencias entre las actividades en un proceso, la ejecución de actividades se espera suceda en ciertos contextos. Puede haber dependencias de corto y largo rango entre actividades. Dependencias de largo rango son difíciles de descubrir. Un analista puede estar interesado en entender los contextos de ejecución de actividades y/o secuencias de actividades.

5. ¿Cuáles son las instancias del proceso que comparten/capturan un comportamiento deseado?

Comúnmente en el diagnóstico un analista puede estar interesado en encontrar instancias del proceso que comparten un comportamiento deseado. El comportamiento deseado puede ser expresado como una manifestación de un patrón de secuencia de actividades o alguna combinación compleja de estos patrones. Aunque los enfoques temporales lógicos pueden ayudar a abordar este problema en cierta medida, mediante el descubrimiento de casos de procesos que capturan el comportamiento deseado

exactamente, el interés podría también estar centrado en el descubrimiento de instancias de proceso que compartan el comportamiento deseado aproximadamente.

6. ¿Hay algunos patrones particulares en el proceso?

Los patrones de flujo de trabajo se refieren a las formas y estructuras recurrentes que abordan los requerimientos del negocio. Por ejemplo, los hitos indican especificaciones, puntos de ejecución en el modelo de proceso y proporcionan un mecanismo para apoyar la ejecución condicional de una tarea o sub-proceso. Un analista estaría interesado en descubrir la presencia y el análisis de los patrones de los hitos en el proceso de registro de eventos. De manera similar, el descubrimiento de los modelos de proceso con simultaneidad es uno de los problemas más difíciles en la minería de procesos. La presencia de actividades concurrentes crea permutaciones diferentes de actividades en el registro de eventos que se suma a la complejidad de los algoritmos de descubrimiento. La detección de la presencia de actividades simultáneas también podría ayudar en el pre-procesamiento de los registros.

1.3. Trabajos relacionados con el diagnóstico del proceso

Durante la investigación se hizo un estudio de los trabajos desarrollados en esta área, a continuación se hace un análisis con un subconjunto importante de ellos.

1.3.1. Análisis de Diagrama de Puntos

El primer enfoque, el Análisis de diagrama de puntos (DCA, por sus siglas en inglés), es una de las técnicas de visualización del registro de evento más usada. El DCA, análogo a los diagramas de Gantt, presenta una “vista helicóptero” del registro de evento y ayuda en el análisis de rendimiento del proceso representando los eventos del proceso de una forma gráfica, como puntos en un plano donde una dimensión hace referencia a los casos en el registro de eventos y la otra dimensión se refiere al tiempo en el que se ejecutaron los eventos. Se utiliza un color distinto para cada actividad que se encuentre almacenada en el registro de evento. Se puede notar que con esta forma de graficar los puntos o eventos no se encuentran alineados por lo tanto es muy difícil determinar patrones comunes entre casos distintos.

El DCA calcula algunos indicadores de desempeño, tales como el intervalo de mínimo, máximo y promedio entre eventos. Un analista de negocios necesita investigar manualmente el DCA para identificar los problemas potenciales de rendimiento.

Para un registro de evento con un gran número de actividades (del orden de unas pocas decenas hasta cientos), la inspección manual y la comprensión de la tabla de puntos se vuelven muy complicadas y con frecuencia es imposible de identificar patrones interesantes.

En la Figura 1.6 se muestra un Análisis de diagrama de puntos para un registro de evento, utilizando ProM.

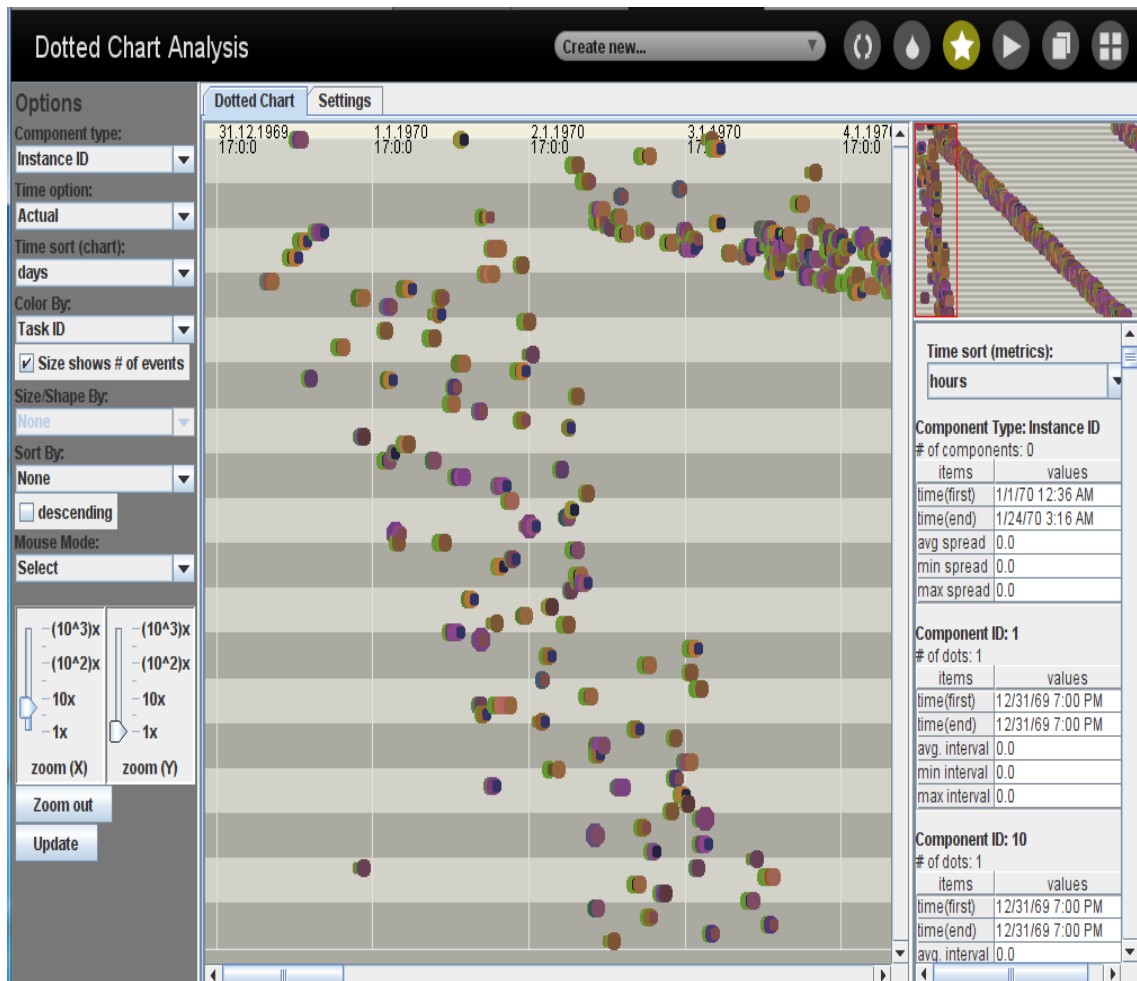


Figura 1.6. Análisis de Diagrama de Puntos en ProM.

Se puede concluir que la técnica de diagnóstico basada en el análisis de diagrama de puntos, tiene como ventaja que brinda un conjunto de variables de rendimiento al analista del negocio y da una idea general del proceso, pero es ineficiente a la hora de determinar patrones de flujo de trabajo así como el tratamiento de grandes registros de eventos, lo cual dificulta del diagnóstico del proceso.

1.3.2. Stream Scope Visualization

Stream Scope Visualization (GÜNTHER 2009) es una técnica de visualización de trazas que está basada en las correlaciones de clases de eventos. Usando Stream Scope Visualization los patrones de los eventos co-ocurrentes pueden ser detectados por su cercanía. No obstante, Stream Scope Visualization está limitado en el hecho de que visualiza cada traza por separado y no provee una visión general del registro de evento lo cual dificulta el diagnóstico.

1.3.3. Tandem Arrays

Una rama fundamental en el diagnóstico del proceso es el descubrimiento de patrones comunes de ejecución. Bose y Van Der Aalst (BOSE, R.P. JAGADEESH CHANDRA and AALST 2009) proponen el uso de Tandem Arrays y repeticiones máximas para capturar patrones recurrentes dentro y a través de las trazas aunque, al utilizar esta técnica, estos patrones se pueden determinar en tiempo lineal, ellos sufren dos limitaciones: primeramente el número de patrones descubiertos puede ser realmente grande, sin embargo, usando métricas así como las propuestas en (LI *et al.* 2010) se pueden filtrar patrones insignificantes. Por último, los patrones descubiertos son atómicos y las dependencias/correlaciones entre los patrones deben ser descubiertas por separado; en otras palabras, el contexto de sus manifestaciones se pierde y necesita ser restablecido por separado.

1.3.4. Descubrimiento de patrones usando árbol de sufijos

Las repeticiones que existen entre las trazas en el registro de evento podemos determinarlas aplicando los algoritmos de identificación de patrones en las secuencias obtenidas de concatenar las trazas en el registro de evento con un delimitador distinto de cualquier elemento en el alfabeto original, es decir, las actividades del registro de evento. Tal concatenación de trazas puede incurrir en una secuencia muy larga, en esta situación se puede adoptar una construcción de árbol de sufijos eficiente como (CHEUNG 2005) para manejar secuencias muy grandes. Repeticiones aproximadas pueden ser encontradas primeramente identificando repeticiones exactas y luego buscar todas las subsecuencias a una distancia determinada de las repeticiones exactas. Este tipo de técnica al igual que las dos anteriores tiende a influir positivamente en la determinación de patrones pero dificulta el diagnóstico por la falta de variables de rendimiento de las actividades presentes en el registro de eventos.

1.3.5. Alineación de trazas

Otra de las técnicas de diagnóstico es la Alineación de trazas propuesta por Bose y Van der Aalst (BOSE, R. P. JAGADEESH CHANDRA and AALST 2012). La alineación de trazas alivia el problema de determinar patrones interesantes en medianos y grandes log, determinándolos automáticamente y mostrándolos al usuario.

La alineación de trazas es una técnica que utiliza la programación dinámica para tabular las trazas de forma tal que se simplifiquen los problemas de entendimiento de las relaciones entre las actividades.

La técnica está inspirada en la alineación de múltiples secuencias (MSA, por sus siglas en inglés). La alineación de múltiples secuencias es una herramienta fundamental en el área de la bioinformática pues ayuda a desentrañar las estructuras secundarias y terciarias de las proteínas y moléculas, la evolución y funciones de estas, y a inferir las relaciones taxonómicas y filogenéticas entre los organismos y para el diagnóstico de enfermedades genéticas. No obstante, se consideró un desafío adaptar esta última técnica a la alineación de trazas. La minería de procesos también se relaciona con secuencias, estas son los casos relaciones en un registro de evento. MSA ha sido un sujeto fundamental de una búsqueda extensiva en la biología computacional por aproximadamente tres décadas. Todavía existen problemas por resolver: tratamiento de conjuntos de datos cada vez más grandes y complicados, mal alineamientos y errores de alineación, obtención de alineaciones precisas de secuencias no codificadas o no transcritas entre muchos otros. Ha sido todo un desafío la adaptación de la alineación de secuencias a la alineación de trazas.

La alineación de trazas es un tema muy poco tratado en estos días aunque es conocida su utilidad en el diagnóstico de procesos resolviendo muchas interrogantes y apoyando a otras herramientas que trabajan en este sentido.

La formalización de la alineación de traza se expone a continuación:

Definición 2: La alineación de las trazas sobre un conjunto de trazas $\mathbb{T} = \{T_1, T_2, \dots, T_n\}$ se define como el mapeo del conjunto de trazas de \mathbb{T} sobre otro conjunto de trazas $\bar{\mathbb{T}} = \{\bar{T}_1, \bar{T}_2, \dots, \bar{T}_n\}$ donde cada $\bar{T}_i \in (\Sigma \cup \{-\})^+$ para $1 \leq i \leq n$.

- Existe un $m \in \mathbb{N}$ tal que $|\bar{T}_1| = |\bar{T}_2| = \dots = |\bar{T}_n| = m$
- \bar{T}_i es igual a T_i después de eliminar todos los símbolos “-”

- No existe una $k \in \{1, \dots, m\}$ tal que $\forall 1 \leq i \leq n, \bar{T}_i(k) = \text{“-”}$.

En la definición dada, m representa la longitud de la alineación. Una alineación sobre un conjunto de trazas puede ser representada por una matriz rectangular $\mathcal{A} = \{a_{ij}\}$ ($1 \leq i \leq n, 1 \leq j \leq m$) sobre $\Sigma' = \Sigma \in \{-\}$ donde “-” denota un vacío. La tercera condición de la definición implica que no existe una columna de \mathcal{A} que contiene solo símbolos “-”. Es necesario resaltar que existen varias posibles alineaciones sobre un conjunto de trazas y que la longitud de la alineación, m , satisface la relación $l_{\max} \leq m \leq l_{\text{sum}}$ donde l_{\max} es la máxima longitud de una traza contenida en \mathbb{T}_Y y l_{sum} es la suma de las longitudes de todas las trazas contenidas en \mathbb{T} .

Una de las aplicaciones de la alineación de trazas es el descubrimiento de las desviaciones entre las trazas anómalas y las trazas normales. Esta técnica también permite la visualización de múltiples trazas a la vez y es capaz de descubrir los patrones comunes de ejecución dentro y a través del registro de evento.

Un ejemplo de la alineación de trazas se muestra en la Figura 1.7. Al obtener una alineación de trazas se pueden determinar patrones interesantes como secuencias entre bloques de eventos en el ejemplo de la Figura 1.7, las actividades j, g, c y b entre otras representan columnas que dividen secuencias bien conservadas de bloques de actividades.

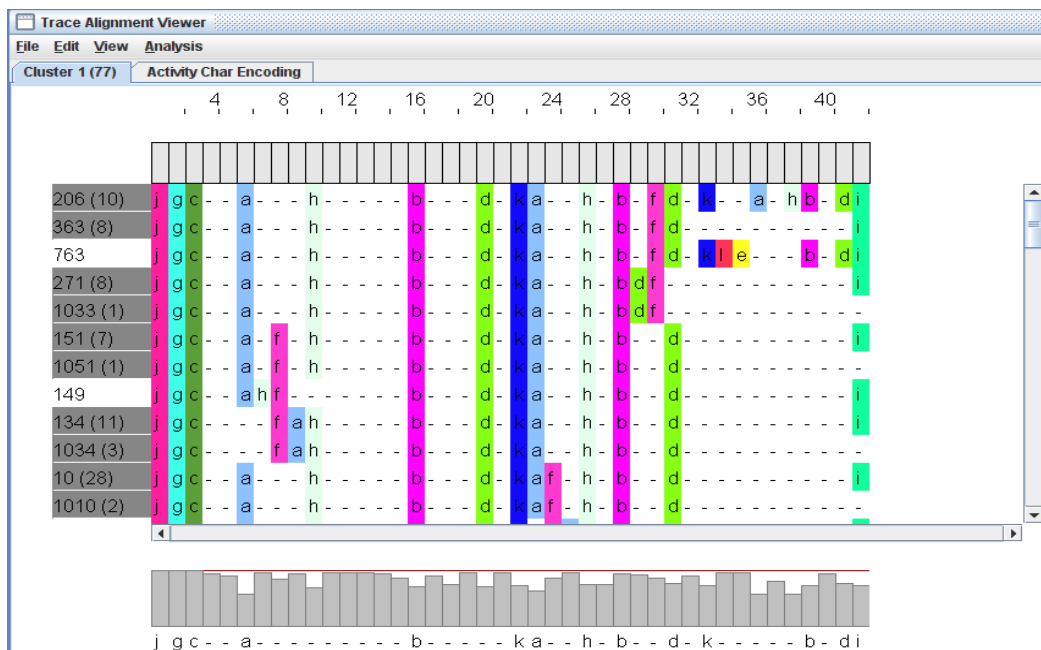


Figura 1.7. Alineación de trazas en ProM.

El paralelismo también puede ser fácil de determinar en algunas alineaciones (Figura 1.8 actividades E y F), pero existen otros patrones como la selección (Figura 1.8 actividades B y C) para los cuales, en caso de grandes alineaciones existe una gran dificultad a la hora de determinarlos.



Figura 1.8. Ejemplo de selección y paralelismo en trazas alineadas.

1.3.6. Chequeo de conformidad

Mirando desde otro punto de vista, sería interesante detectar desviaciones que ocurren en la ejecución de un proceso respecto al modelo. El Chequeo de conformidad (AALST, W.M.P. VAN DER 2011) compara un modelo teórico inicial con el comportamiento observado en las trazas y apunta a la detección de inconsistencias y/o desviaciones entre el modelo del proceso y su registro de evento.

En las figuras 1.9 y 1.10 se encuentran reflejados chequeos de conformidad utilizando ProM para métricas Fitness y ETConformance.

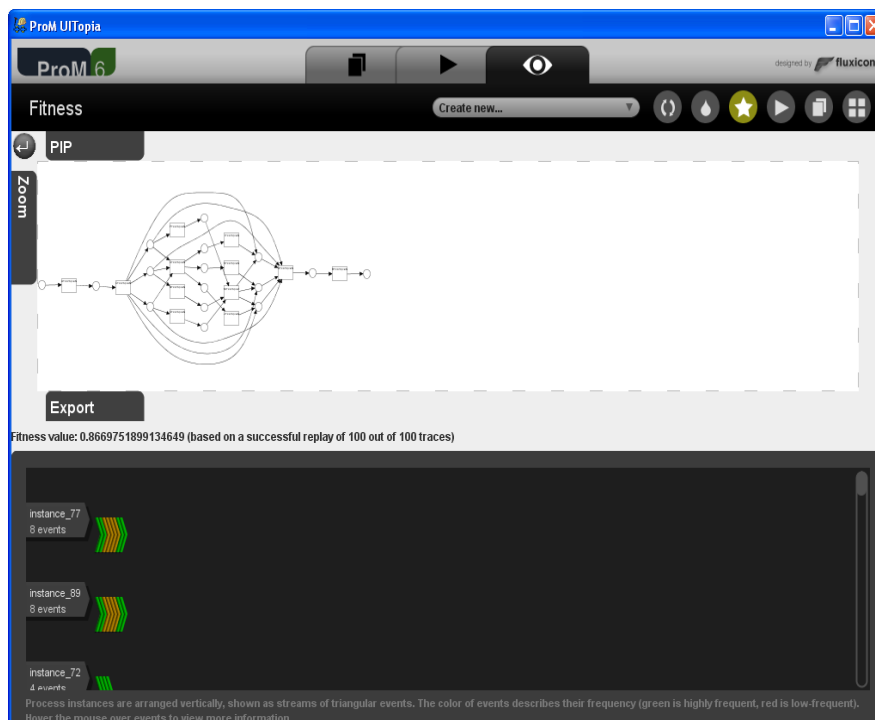
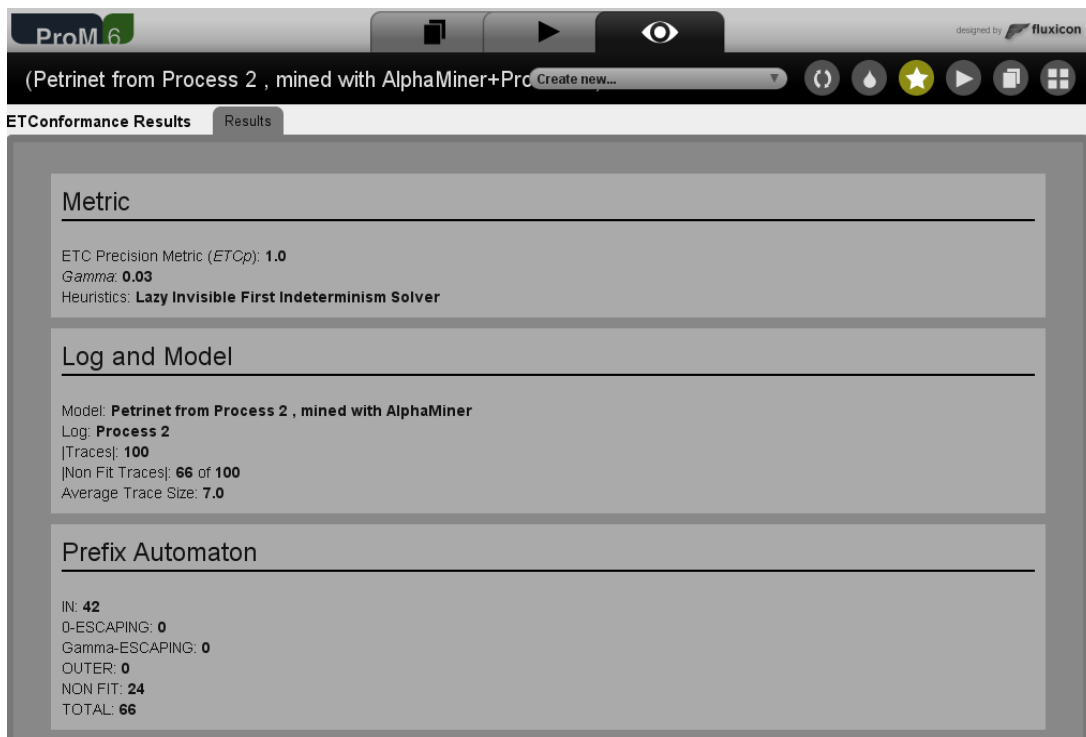


Figura 1.9. Cálculo de la métrica Fitness.

El Chequeo de conformidad tiene inherentes limitaciones en su aplicación especialmente en el diagnóstico de procesos. Primeramente asume que hay un modelo del proceso, cuando en la realidad sabemos que el modelo del proceso puede no estar presente o puede contar con problemas. Se puede argumentar que el modelo del proceso puede ser descubierto a partir del registro de evento y con este se podría hacer el Chequeo de conformidad. Sin embargo, este enfoque no es adecuado para el análisis de los procesos altamente complejos y/o flexibles, la clase de modelos que representan la mayoría de los registros de eventos de la vida real y que culminan en el descubrimiento de modelos estereotipados como "espagueti".

Incluso en los casos donde el modelo del proceso está disponible como red de Petri, es muy difícil mirar adentro del proceso para identificar y localizar problemas especialmente en el caso de modelos que sean muy grandes.

Es bueno aclarar que las técnicas anteriores derivan modelos de proceso de un registro de evento suponiendo que todos los eventos en el log se refieren a una actividad y que estas actividades se producen en el mismo nivel de abstracción. Las siguientes técnicas apuntan a resolver este problema.



The screenshot displays the ProM 6 software interface. The title bar shows 'ProM 6' and 'fluxicon'. The main window title is '(Petri net from Process 2 , mined with AlphaMiner+Pro... Create new...'. Below the title bar, there is a toolbar with icons for home, play, eye, refresh, star, and window management. The main content area is titled 'ETCConformance Results' and has a 'Results' tab selected. It contains three sections:

- Metric**
 - ETC Precision Metric ($ETCp$): **1.0**
 - Gamma: **0.03**
 - Heuristics: **Lazy Invisible First Indeterminism Solver**
- Log and Model**
 - Model: **Petri net from Process 2 , mined with AlphaMiner**
 - Log: **Process 2**
 - [Traces]: **100**
 - [Non Fit Traces]: **66 of 100**
 - Average Trace Size: **7.0**
- Prefix Automaton**
 - IN: **42**
 - 0-ESCAPING: **0**
 - Gamma-ESCAPING: **0**
 - OUTER: **0**
 - NON FIT: **24**
 - TOTAL: **66**

Figura 1.10. Cálculo de la métrica ETCConformance.

1.3.7. Minería Difusa

La Minería Difusa es el primer intento de dejar a un lado la suposición de que las actividades se producen en el mismo nivel de abstracción.

La Minería Difusa produce un modelo basado en grafos donde se utilizan dos tipos de nodos, los nodos que se refieren a una actividad, y los nodos que se refieren a un conjunto de actividades, o clústeres. Por lo tanto, el modelo es capaz de proporcionar una vista de alto nivel de un proceso abstrayendo detalles no deseados.

Sin embargo, la Minería Difusa todavía supone que cada evento en el log pertenece a uno de estos nodos, es decir, hay una relación uno a muchos ya que cada nodo puede representar muchas actividades, pero cada actividad está representada por un nodo exactamente. A continuación se muestra en la (figura 1.11), un ejemplo de diagrama de minería difusa.

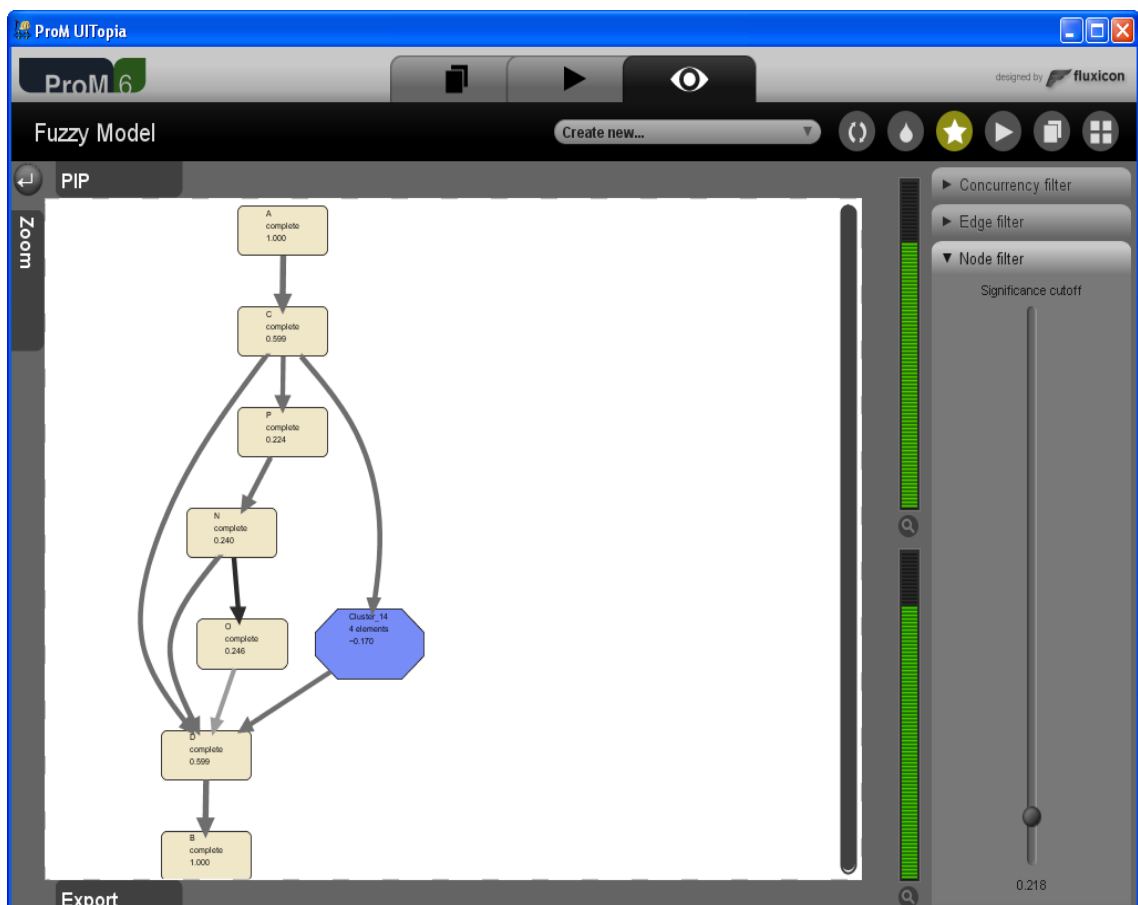


Figura 1.11. Diagrama de Minería Difusa obtenido de la herramienta ProM.

1.3.8. Minería basada en teoría de regiones

En (AALST, W.M.P. VAN DER *et al.* 2009b) se presenta una técnica donde las relaciones entre nodos y actividades es realmente de muchos a muchos.

Inicialmente esta teoría solo podía ser aplicada a un reducido conjunto de sistemas de transición. En cambio, al transcurrir el tiempo, este enfoque fue extendido para permitir la síntesis de cualquier sistema de transición finito.

Es necesario tener en cuenta, que solo un grupo reducido de personas han aplicado las regiones basadas en estados en la minería de procesos aunque recientemente algunos trabajos acerca de teoría de regiones basadas en lenguaje han aparecido (BERGENTHUM *et al.* 2007) y en este mismo trabajo se muestra como esta teoría se puede aplicar a la minería de procesos. Estos enfoques son muy interesantes y construyen directamente una red de Petri y no construyen un sistema de transición intermedio, esto tiene ventajas, por ejemplo en función de eficiencia, pero tiene desventajas porque este enfoque es poco configurable.

Resumiendo, la técnica primeramente construye un espacio utilizando el registro de evento y luego usa teoría de regiones para construir una red de Petri a partir del espacio creado anteriormente. Este enfoque está limitado en dos sentidos, primeramente solo se consideran redes de Petri como modelos del proceso, aunque se sabe que hay muchos otros modelos tales como redes de flujos de trabajo o redes elementales, por otra parte, se encuentra el cuello de botella que se puede generar al aplicar esta teoría.

1.3.9. Agrupación Difusa

La agrupación difusa, otro de los enfoques en el diagnóstico de procesos, es un algoritmo para agrupar los eventos automáticamente a un nivel deseado de la abstracción (DONGEN, B. F. and ADRIANSYAH 2010a). Los diagramas obtenidos en este trabajo no permiten conocer las relaciones entre las actividades que conforman el clúster. Las figuras 1.12 y 1.13 muestran diagramas de Agrupación Difusa.

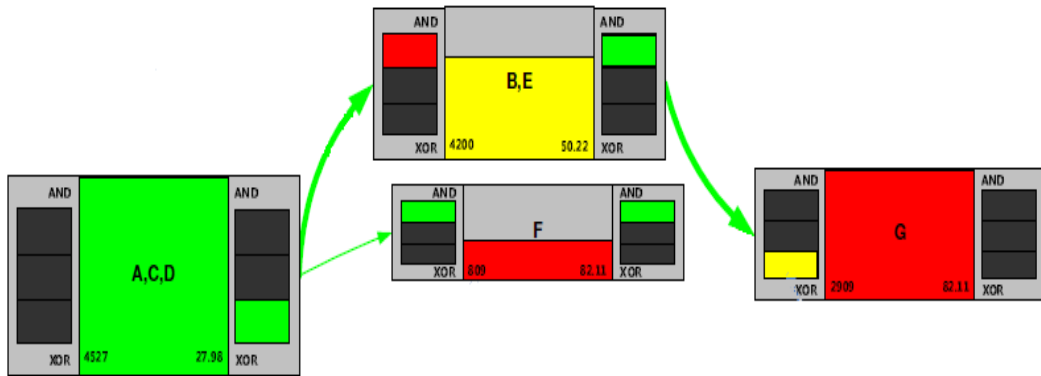


Figura 1.12. Diagrama Agrupación Difusa tomado de (DONGEN, B. F. and ADRIANSYAH 2010a).

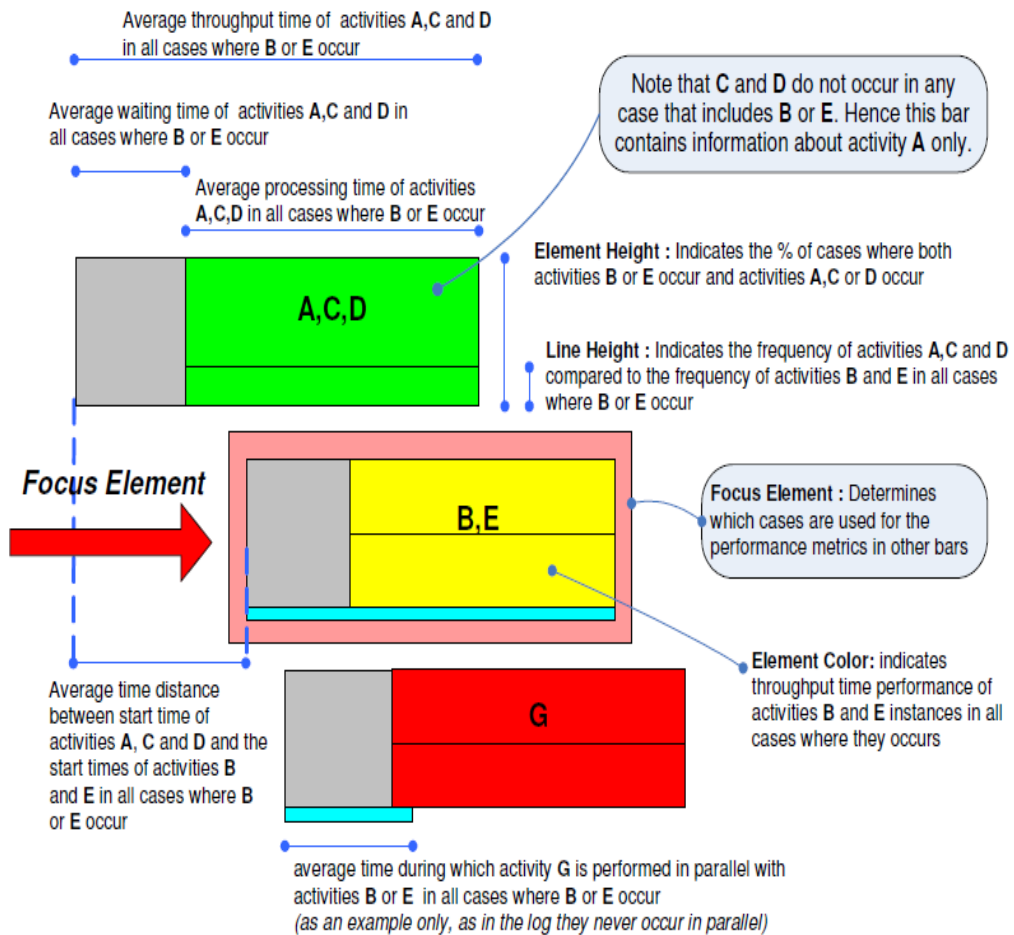


Figura 1.13. Diagrama de Agrupación Difusa AAPD tomado de (DONGEN, B. F. and ADRIANSYAH 2010a).

Los diagramas obtenidos en este trabajo no permiten conocer las relaciones entre las actividades que conforman el clúster. Además, la similitud entre las actividades y la probabilidad de que sean miembros de un clúster están definidas a partir de ver cómo a menudo los eventos relacionados con estas actividades se encuentran consecutivos dentro del registro de evento. El enfoque no considera otra relación entre las actividades para formar el grupo o clúster. No se puede obtener una visión general que tenga en cuenta el tiempo relativo considerando la perspectiva lógica del registro de evento.

1.4. Conclusiones del capítulo

En el capítulo se hizo un análisis de la Minería de procesos así como el Diagnóstico del proceso, además se trataron las principales técnicas que se han desarrollado en este ámbito, entre las que podemos contar como principales a la Alineación de trazas y la Agrupación Difusa así dando cumplimiento al objetivo número uno planteado en la investigación: Fundamentar la investigación mediante la evaluación del marco teórico.

Se puede concluir que las técnicas desarrolladas hasta el momento en el diagnóstico del proceso presentan dificultades al obtener los patrones de flujo de trabajo de forma clara, las desviaciones del registro de eventos o en el manejo de los subprocesos a diferentes niveles de abstracción así como el trabajo con grandes registros y la determinación de variables de rendimiento que puede brindar gran apoyo al analista del negocio a la hora de diagnosticar el proceso.

Se seleccionó como base para una nueva técnica de diagnóstico a la alineación de trazas, ya que apoya de forma eficiente la determinación de flujos de trabajo al mantener las actividades organizadas mediante un orden relativo.

Capítulo 2. Propuesta de solución

2.1. Introducción

En este trabajo se presenta un nuevo algoritmo para el diagnóstico que permite construir un árbol de matrices representativas de los subprocesos que componen al proceso analizado. El algoritmo propuesto permite agrupar las tareas automáticamente y resalta los aspectos más significativos del proceso en cada momento. El descubrimiento de los subprocesos que componen al proceso analizado, sus dependencias y correlaciones permiten una mayor precisión en el diagnóstico realizado.

2.2. Algoritmo para el diagnóstico del proceso

Inicialmente se exponen un conjunto de definiciones que son necesarias para el correcto entendimiento de la propuesta.

Definición 1 (Proceso de negocio): Un proceso de negocio es una colección de actividades que son realizadas coordinadamente en un ambiente técnico y organizacional. La conjunción de estas actividades logra un objetivo del negocio. Cada proceso de negocio es ejecutado por una simple organización, pero con él pueden interactuar procesos de negocios de otras organizaciones. ■

Definición 2 (Subproceso): Un subproceso es una encapsulación de actividades del negocio que representan una compleja y lógica unidad de trabajo. Los subprocesos tienen sus propios atributos y metas, pero contribuyen a la meta del proceso que los contiene. Un subproceso es también un proceso y su mínima expresión es una actividad. ■

Un proceso puede descomponerse en varios subprocesos mediante los patrones de flujo de trabajo siguientes:

- Secuencia: dos subprocesos se encuentran ordenados secuencialmente si inmediatamente después de que ocurra el primer subproceso ocurre el segundo.
- Selección (XOR u OR): dos subprocesos se encuentran ordenados como opciones de una selección si en cada caso o instancia del proceso solo ocurre uno de ellos (XOR) u ocurren los dos en cualquier orden (OR).
- Paralelismo: dos subprocesos se encuentran ordenados en paralelo si ocurren los dos simultáneamente.

- Lazo: un lazo se manifiesta cuando un subproceso se repite en múltiples ocasiones.

Los subprocesos pueden descomponerse en otros subprocesos hasta el nivel de actividad. Esto permite construir un árbol en el que cada nivel tiene menor grado de abstracción.

En la siguiente definición se formalizan los aspectos asociados a la descomposición de un proceso y su representación mediante bloques de construcción.

Definición 3 (Bloque de construcción y descomposición en bloques de construcción): Sea S el conjunto de todos los subprocesos que componen a un proceso P , \mathcal{L} el registro de evento que representa a las instancias del proceso P ejecutadas, \mathcal{A} la matriz obtenida a partir de la alineación de las trazas contenidas en \mathcal{L} y Q_A el conjunto de todas las sub-matrices de \mathcal{A} .

Se denota por Q'_A el conjunto de sub-matrices que representan a los subprocesos de S , tal que $Q'_A \subseteq Q_A$. Sean C^j_A y C^{j+1}_A sub-matrices de Q'_A , la relación de secuencia entre dos subprocesos representados por C^j_A y C^{j+1}_A se denota por $C^j_A >_L C^{j+1}_A$. De forma análoga se denota la relación de selección (XOR específicamente) por $C^j_A \#_L C^{j+1}_A$ y la relación de paralelismo por $C^j_A ||_L C^{j+1}_A$. En el caso de manifestarse un lazo la descomposición de C^j_A se realiza en un único sub-proceso que se repite múltiples veces y se denota por $(C^j_A)^*$.

Sea $s_i \in S$ un subproceso representado por la matriz $C^j_A \in Q'_A$ y que está compuesto por una secuencia de subprocesos representados por C^j_A, \dots, C^{j+k}_A , entonces tanto la matriz C^j_A como el conjunto $\{C^j_A, \dots, C^{j+k}_A\}$ se le denominan *bloques de construcción* y los sub-procesos representados por $\{C^j_A, \dots, C^{j+k}_A\}$ se relacionan de una única forma (secuencia, paralelismo, XOR o lazo).■

La propuesta tiene como objetivo construir un árbol de bloques de construcción que representa la descomposición del proceso analizado. La Figura 2.1 muestra un ejemplo de un árbol de bloques de construcción. Los bloques de construcción C^2_A y C^3_A representan subprocesos ordenados secuencialmente, C^4_A y C^5_A representan subprocesos ordenados como opciones de una selección. C^6_A y C^7_A representan subprocesos en paralelo.

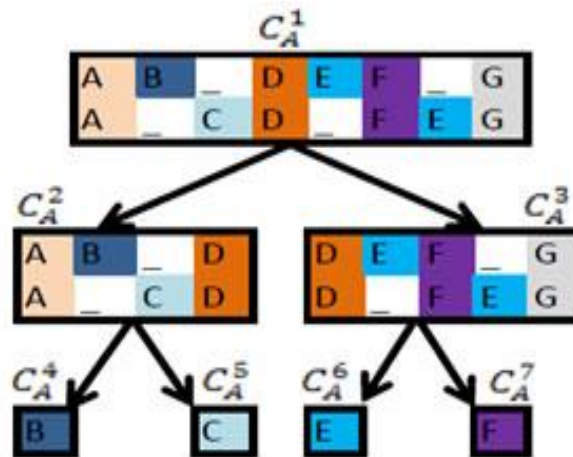


Figura 2.1. Árbol de bloques de construcción.

2.3. Pasos generales que conforman el algoritmo propuesto

Considerando los aspectos ya mencionados se presentan los pasos generales que conforman el algoritmo propuesto.

2.3.1. Alinear las trazas

El primer paso es el agrupamiento de las trazas y la alineación de las mismas. Las trazas alineadas constituyen una representación de las actividades de acuerdo con un orden relativo y una estructuración de las mismas en casos. El orden establecido entre las actividades permite identificar los patrones de flujo de trabajo que se manifiestan entre los subprocesos. Como resultado de la alineación de un grupo de trazas se obtiene una matriz \mathcal{A} .

2.3.2. Pre-procesar las trazas alineadas

A partir de las trazas alineadas se pueden determinar los casos incompletos (casos que no terminan con las actividades finales identificadas para el proceso). Estos casos pueden ser tratados o eliminados según se considere, así se puede volver a alinear las trazas.

2.3.3. Determinar el árbol de bloques de construcción

En Algoritmo 1 se describen los pasos para determinar el árbol de bloques de construcción. Se emplea un árbol para la representación porque muestra la forma en la que el proceso se compone en otros subprocesos en cada nivel.

Algoritmo 1 Algoritmo para determinar el árbol de bloques de construcción

Entrada: Matriz \mathcal{A}'

Salida: Árbol de bloques de construcción

1: Crear un árbol vacío.

2: Crear un bloque de construcción $C_{\mathcal{A}}^1$ y asociarlo al nodo raíz del árbol la matriz \mathcal{A}' , tal que $C_{\mathcal{A}}^1 = \mathcal{A}'$.

3: **Si** $C_{\mathcal{A}}^1$ no es una matriz con una sola fila **Entonces**

4: $LH = \text{Buscar secuencia}(C_{\mathcal{A}}^1)$. LH es la lista que almacena los bloques de construcción obtenidos producto a la descomposición realizada.

5: **Si** $|LH| < 1$ **Entonces**

6: $LH = \text{Buscar lazo}(C_{\mathcal{A}}^1)$

7: **Si** $|LH| < 1$ **Entonces**

8: $LH = \text{Buscar XOR-OR}(C_{\mathcal{A}}^1)$

9: **Si** $|LH| < 1$ **Entonces**

10: $LH = \text{Buscar paralelismo}(C_{\mathcal{A}}^1)$

11: **Si** $|LH| < 1$ **Entonces**

12: $LH = \text{Buscar secuencia oculta}(C_{\mathcal{A}}^1)$

FinSi

FinSi

FinSi

FinSi

13: **Para** cada bloque de construcción i contenido en la lista LH **Hacer**

14: Modificar i . Elimínandose de ser necesario, las filas repetidas y las columnas que solo contienen símbolos vacíos (“-”)

15: Adicionar i como hijo del nodo que contiene a $C'_{\mathcal{A}}$

16: Aplicar el *Algoritmo para determinación del árbol de bloques de construcción* al bloque de construcción modificado comenzando por el paso 3

17: **Si** el árbol obtenido en el paso anterior $\neq \emptyset$ **Entonces**

18: Adicionar los nodos hijos de la raíz del árbol obtenido como hijos del nodo que contiene el bloque de construcción i

FinSi

FinPara

Sino

19: Devolver un árbol vacío

FinSi

20: Devolver el árbol de bloques de construcción construido

En el algoritmo para determinar el árbol de bloques de construcción la operación que mayor complejidad temporal tiene es *Buscar secuencia oculta* en un bloque de construcción. Dicha complejidad es $O(m^4 \cdot n)$ y se explica posteriormente cuando se describe el procedimiento. Si se considera que la mayor cantidad de niveles del árbol de bloques de construcción que se pueden construir utilizando esta operación es $m-1$, entonces la complejidad temporal de este algoritmo es $O(m^5 \cdot n)$.

A continuación se describen los algoritmos para Buscar secuencia, Buscar lazo, Buscar XOR-OR, Buscar paralelismo y Buscar secuencia oculta.

El *Algoritmo 2 Buscar secuencia* tiene como objetivo determinar si el proceso analizado se puede descomponer en otros subprocesos ordenados secuencialmente. En caso de ser posible la

descomposición, se devuelve la lista de los bloques de construcción detectados, en caso contrario, se devuelve la lista vacía. Los subprocesos ordenados secuencialmente pueden ser claramente identificados debido a que estos están separados por una o varias actividades consecutivas que aparecen ocupando una columna completa en cada caso. En ocasiones este tipo de actividades no se puede identificar debido a que no se almacenaron en el registro de eventos.

Algoritmo 2 Buscar secuencia

Entrada: Bloque de construcción $C_{\mathcal{A}}^i$

Salida: Lista de bloques de construcción

1: Crear dos listas vacías *ListaSeparador* y *ListaHijos*. *ListaSeparador* tiene como objetivo almacenar el identificador de las columnas que delimitan cada bloque de construcción. *ListaHijos* almacena los bloques de construcción obtenidos en una posible descomposición

2: **Para** cada columna de $C_{\mathcal{A}}^i$ **Hacer**

3: Verificar que la columna sea completa. Una columna es completa si no tiene símbolos vacíos (“-”)

4: **Si** la columna es completa **Entonces**

5: Verificar que no existan actividades que estén antes y también detrás de la

columna, de cumplirse esto *separador* = verdadero. Esto permite desechar las columnas correspondientes a actividades que están en paralelo o que forman un lazo.

6: **Si** *separador* = verdadero **Entonces**

7: Adicionar el valor de la columna a la lista *ListaSeparador*

FinSi

FinSi

FinPara

8: **Si** $|ListaSeparador| > 0$ **Entonces**

9: Formar los bloques de construcción a partir de la lista *ListaSeparador* y almacenarlos en *ListaHijos*

FinSi

10: Devolver *ListaHijos*

El algoritmo *Buscar secuencia* se compone de dos grandes partes. La primera asociada con la búsqueda de las columnas que permiten separar dos bloques de construcción ordenados secuencialmente y la segunda parte está asociada a la conformación de los nuevos bloques de construcción. Esta segunda parte tiene asociada una complejidad temporal $O(m.n)$, dado que es necesario recorrer todo el bloque de construcción analizado. Mientras que para la primera parte es necesario recorrer todo el bloque de construcción y para cada columna que sea completa se verifica que sea un separador. Esta verificación se realiza usando una lista con las actividades que contiene cada columna del bloque de construcción, cuya dimensión es m . En correspondencia con lo antes analizado la complejidad temporal del algoritmo *Buscar secuencia* es $O(m^2.n)$.

El *Algoritmo 3 Buscar lazo* tiene como objetivo determinar si el proceso analizado se puede descomponer en un subproceso que se realiza múltiples veces. En caso de ser posible la descomposición, se devuelve una lista con un solo bloque de construcción, el cual, representa al subproceso que se repite; en el caso contrario, se devuelve la lista vacía. Para determinar un bloque de construcción que representa un subproceso que se repite en múltiples ocasiones es necesario identificar la actividad inicial de ese subproceso. Esta actividad inicial permitirá separar secuencias de actividades que posteriormente conforman las filas del nuevo bloque de construcción. Las secuencias repetidas se desechan.

Algoritmo 3 Buscar lazo

Entrada: Bloque de construcción $C_{\mathcal{A}}^i$

Salida: Lista de bloques de construcción

1: Crear una lista vacía *ListaHijos*. *ListaHijos* almacena los bloques de construcción obtenidos en una posible descomposición

2: Definir inicio y fin del análisis. Es posible que el bloque de construcción tenga toda la primera columna ocupada por una actividad (no hay símbolos “-”) que representa el evento de inicio del subproceso. Esta actividad debe ser eliminada del análisis (el análisis comenzaría en la segunda columna), de manera análoga se procede con la última columna, la cual puede corresponderse con el evento de fin del subproceso. La actividad contenida en la primera columna no puede estar duplicada de lo contrario se considera esta columna como parte del análisis

3: Buscar la primera actividad que aparece en cada fila del bloque de construcción

4: **Si** la primera actividad es la misma en cada fila **Entonces**

5: Definir las secuencias delimitadas por la primera actividad o el fin del bloque de construcción.

6: Alinear las secuencias de actividades definidas. La alineación se realiza con los mismos parámetros que definidos en el *Módulo para la alineación de las trazas*

7: Construir un bloque de construcción (*BC*) con la matriz obtenida en la alineación

8: Adicionar *BC* a *ListaHijos*

FinSi

9: Devolver *ListaHijos*

En el algoritmo *Buscar lazo* las operaciones que tienen mayor complejidad temporal son las correspondientes al paso cinco y seis. La primera de estas operaciones requiere que se recorra todo el bloque de construcción por lo cual su complejidad es $O(m.n)$. Mientras que la alineación de las secuencias definidas (paso seis) tiene una complejidad temporal de $O(l.n + l^2)$, siendo l el promedio de las longitudes de las secuencias a alinear. Para este cálculo se toma como referencia el trabajo de *Alineación de trazas* de Bose y Van der Aalst. Considerando que no es posible establecer ninguna relación de igualdad o desigualdad entre las variables m , n y l se determina que la complejidad temporal del algoritmo *Buscar lazo* es $O(m.n + l^2)$.

El *Algoritmo 4 Buscar XOR-OR* tiene como objetivo determinar si el proceso analizado se puede descomponer en otros subprocesos ordenados como opciones de una selección. En caso de ser posible la descomposición, se devuelve la lista de los bloques de construcción detectados, en caso contrario, se devuelve la lista vacía. Inicialmente se busca identificar una descomposición según XOR. Para determinar los bloques de construcción que representan opciones de una selección (XOR), se construyen conjuntos disjuntos con las actividades que componen el bloque de construcción analizado. Inicialmente existe un conjunto que contiene las actividades que comparten una fila del bloque de construcción analizado, posteriormente se unen los conjuntos que se interceptan mediante alguna actividad. Si al finalizar este proceso queda más de un conjunto, entonces se construyen los bloques de construcción que representan cada una de las opciones.

Si solo queda un conjunto entonces se busca identificar una descomposición según OR. Para ello se determinan las secuencias base. Una secuencia base es una fila del bloque de construcción que no está compuesta completamente por la unión de otras filas. Las secuencias que contienen actividades comunes se unen en un mismo conjunto. Si al finalizar este proceso queda más de un conjunto, entonces se construyen los bloques de construcción que representan cada una de las opciones.

Algoritmo 4 Buscar XOR-OR

Entrada: Bloque de construcción $C_{\mathcal{A}}^i$

Salida: Lista de bloques de construcción

- 1: Crear una lista vacía *ListaHijos*. *ListaHijos* almacena los bloques de construcción obtenidos en una posible descomposición
- 2: Definir inicio y fin del análisis. Este paso es igual al que se realiza en el algoritmo *Buscar lazo*, paso 2.
- 3: Construir n conjuntos. Donde cada conjunto contiene una fila diferente del bloque de construcción y esta fila representa el elemento raíz del conjunto. La raíz de un conjunto es el número de una fila del bloque de construcción, dicha fila debe estar contenida en el conjunto y lo representa unívocamente. El procedimiento *raíz(x)* determina el elemento raíz en el conjunto a donde x pertenece, siendo x el número de una fila en el bloque de construcción.

4: **Para** $i = 1$ Hasta n Con Paso 1 Hacer

5: **Para** $j = 1$ Hasta n Con Paso 1 Hacer

6: **Si** $raíz(i) \neq raíz(j)$ **Entonces**

7: **Si** fila i y fila j tienen actividades en común **Entonces**

8: $Unir(raíz(i), raíz(j))$. El procedimiento $Unir(x,y)$ une los conjuntos que contienen a x y y como raíces, respectivamente.

FinSi

FinSi

FinPara

FinPara

9: **Si** la cantidad de conjuntos > 1 **Entonces**

10: **Para** cada conjunto **Hacer**

11: Crear un bloque de construcción que contiene como filas los elementos del conjunto. En este caso se ha descompuesto el proceso analizado mediante un XOR.

12: Adicionar el bloque de construcción creado a *ListaHijos*

FinPara

Sino

13: **Para** $i = 1$ Hasta n Con Paso 1 Hacer

14: Verificar que la fila i es base del OR. Una fila es base del OR si no se puede descomponer completamente a partir de la unión de otras filas

15: **Si** la fila i es base del OR **Entonces**

16: Crear un conjunto con la fila i contenida. La raíz del conjunto es i

FinSi

FinPara

17: **Para** cada base del OR **xHacer**

18: **Para** cada base del OR **yHacer**

19: **Si** tiene actividades en común con **yEntonces**

20: *Unir(raíz(x), raíz(y)).*

FinSi

FinPara

FinPara

21: **Si** la cantidad de conjuntos > 1 **Entonces**

22: **Para** cada conjunto **Hacer**

23: Crear un bloque de construcción que contiene como filas los elementos del conjunto. En este caso se ha descompuesto el proceso analizado mediante un OR.

24: Adicionar el bloque de construcción creado a *ListaHijos*

FinPara

FinSi

25: Devolver *ListaHijos*

El algoritmo *Buscar XOR-OR* tiene dos partes fundamentales, una asociada a la detección de una selección XOR y otra asociada a la detección de una selección OR. La primera de estas operaciones tiene una complejidad $O(n^2)$ y está determinada por un recorrido por todo par de filas del bloque de construcción. Las operaciones que se realizan durante el recorrido están asociadas a la estructura de datos Conjuntos Disjuntos, en la cual la operación unión tiene una complejidad temporal de $O(\log^* n)$. En la segunda parte las operaciones de mayor complejidad temporal están asociadas con la búsqueda de las filas base. Para determinar que una fila es base es necesario recorrer los m elementos de las

restantes $n-1$ filas del bloque de construcción y determinar las coincidencias. La complejidad temporal de estas operaciones es $O(n^2.m)$. El resto de las operaciones siguientes tiene menor complejidad temporal dado que consiste en un recorrido por cada par de filas base (como máximo n) y la posible unión de estas (usando Conjuntos Disjuntos). A partir del análisis realizado se determina que la complejidad temporal del algoritmo *Buscar XOR-OR* es $O(n^2.m)$.

El *Algoritmo 5 Buscar paralelismo* tiene como objetivo determinar si el proceso analizado se puede descomponer en otros subprocesos ordenados en paralelo. En caso de ser posible la descomposición, se devuelve la lista de los bloques de construcción detectados, en caso contrario, se devuelve la lista vacía. Para determinar los bloques de construcción que representan subprocesos en paralelo se construyen conjuntos disjuntos con las actividades que componen el bloque de construcción analizado. Las actividades que pertenecen a conjuntos diferentes se encuentran en paralelo, mientras que las actividades que forman un mismo conjunto se relacionan mediante otro tipo de patrón de flujo de trabajo. Si como resultado se obtiene más de un conjunto, entonces los bloques de construcción que se forman a partir de estos representan subprocesos en paralelo.

Algoritmo 5 Buscar paralelismo

Entrada: Bloque de construcción C_A^i

Salida: Lista de bloques de construcción

- 1: Crear una lista vacía *ListaHijos*. *ListaHijos* almacena los bloques de construcción obtenidos en una posible descomposición
- 2: Definir inicio y fin del análisis. Este paso es igual al que se realiza en el algoritmo *Buscar lazo*, paso 2.
- 3: Formar k conjuntos que cumplen las condiciones siguientes. Entre las actividades que conforman un conjunto no existe una relación de paralelismo. Una relación de paralelismo se establece entre dos actividades cuando en el bloque de construcción existe una fila en la que la actividad a esta antes que la b y existe otra fila en la que la actividad b esta antes que la a . Además, se cumple que para todo a, b ; tal que a pertenece al conjunto A y b pertenece al conjunto B , siendo A y B conjuntos disjuntos de actividades contenidas en el bloque de construcción analizado; a está en paralelo con b .

4: Si $k > 1$ Entonces

5: **Para** cada conjunto **Hacer**

6: Crear un bloque de construcción formado a partir de las columnas del bloque analizado que contienen las actividades incluidas en el conjunto.

7: Adicionar el bloque de construcción creado a *ListaHijos*

FinPara

FinSi

8: Devolver *ListaHijos*

En el algoritmo *Buscar paralelismo* el paso de mayor complejidad temporal es el encargado de formar k grupos siguiendo un grupo de condiciones. Para formar los grupos de actividades es necesario que se tome como referencia cada actividad que conforma el bloque de construcción, lo que significa recorrer una lista de m elementos. Para cada actividad se verifica la forma en la que se relaciona con el resto de las actividades y para esto se recorre todo el bloque de construcción. En correspondencia con el análisis realizado la complejidad temporal del algoritmo *Buscar paralelismo* es $O(m^2 \cdot n)$.

El *Algoritmo 6 Buscar secuencia oculta* tiene como objetivo determinar si el proceso analizado se puede descomponer en otros subprocesos ordenados secuencialmente. La diferencia de este algoritmo con el *Algoritmo 2 Buscar secuencia* es que en este caso no existe una columna que no contiene símbolos vacíos (“-”) y que permite delimitar el final de un bloque de construcción e inicio del bloque de construcción que le sigue. En consecuencia, se determinan las posibles soluciones (variantes de descomposición que representan subprocesos ordenados secuencialmente) considerando los aspectos que se enuncian a continuación.

- Cada bloque de construcción que conforma una solución se puede descomponer mediante XOR, OR, paralelismo o lazo.
- Las soluciones se evalúan y se seleccionan las mejores, considerando en la evaluación que los bloques de construcción formados disminuyen la cantidad de lazos y paralelismos rotos. Un lazo roto se evidencia cuando una actividad aparece múltiples veces en una misma fila en el

bloque de construcción analizado y en la solución propuesta no aparece como parte de un mismo bloque de construcción. De manera análoga se determina la cantidad de paralelismos rotos.

En el algoritmo *Buscar secuencia oculta* las operaciones de mayor complejidad temporal son las relacionadas con la formación y evaluación de cada posible bloque de construcción hijos (Pasos del 6 al 11 en el pseudocódigo). Para cada intervalo de columnas del bloque de construcción original $[i, j]$ se evalúa la mejor descomposición mediante el cálculo de μ . Este cálculo requiere de dos operaciones. La primera de estas está dirigida a contar la cantidad de tareas duplicadas en todo el bloque de construcción y dentro del bloque de construcción formado desde i hasta j . Esta operación tiene una complejidad temporal de $O(m^2 \cdot n)$, debido a que para cada actividad contenida en el bloque de construcción se recorre todo el bloque de construcción buscando repeticiones de dicha actividad. La segunda operación está dirigida a contabilizar la cantidad de pares de actividades que se encuentran en paralelo en todo el bloque de construcción y dentro del bloque de construcción formado desde i hasta j . La complejidad temporal de esta operación es $O(m^2 \cdot n)$, debido a que, para cada actividad se determinan los pares de actividades en paralelo que él conforma. Considerando el análisis realizado la complejidad temporal del algoritmo *Buscar secuencia oculta* es $O(m^4 \cdot n)$.

2.4. Conclusiones del capítulo

En este capítulo se obtuvo el diseño de un algoritmo para la detección de subprocesos mediante la alineación de las trazas de procesos desestructurados que permite detectar los patrones más comunes de flujo de trabajo, las desviaciones y los diferentes niveles de abstracción existentes entre las actividades.

La alineación de las trazas se hace utilizando la técnica de diagnóstico *Trace Alignment* descrita en la fundamentación teórica de la investigación. La alineación permite establecer un orden relativo entre las actividades contenidas en el registro de eventos. La selección de esta técnica de representación del registro de evento constituye un paso importante para poder realizar posteriormente la descomposición del proceso. El orden establecido entre las tareas facilita la identificación de los patrones de flujo de trabajo básicos.

Pre-procesar la alineación permite la corrección de determinados aspectos de la alineación que pudieran influir posteriormente en la descomposición del proceso. En este punto se da la posibilidad de que se modifique el registro de eventos y se vuelva a alinear las trazas.

Determinar el árbol de bloques de construcción permite la descomposición del proceso analizado. La descomposición se realizó considerando los patrones básicos de flujo de trabajo, Secuencia, Lazo, Selección (OR o XOR) y Paralelismo. Estos patrones se identificaron a nivel de subprocesos lo cual constituye un aporte de este trabajo. La descomposición permite representar un determinado proceso como un bloque de construcción.

Capítulo 3. Validación de la solución

3.1. Introducción

En este capítulo se realizará la validación del algoritmo propuesto. En este sentido, el algoritmo tiene que ser capaz de responder algunas preguntas que puede confrontar un analista del negocio, tales como, ¿Cuál es el comportamiento más común del proceso que se ejecuta?, ¿Dónde se desvían las instancias del proceso?, ¿Qué tienen estas en común?, ¿Hay patrones comunes de ejecución en las trazas?, ¿En qué contexto se ejecuta una actividad o conjunto de actividades en un proceso?, ¿Hay algunos patrones interesantes en el proceso?

Con el fin de mostrar la utilidad del algoritmo a la hora de resolver estas preguntas se analizó el registro de eventos correspondiente a dos procesos provenientes de un entorno real, el Sistema Único de Identificación Nacional.

3.2. Aplicación en un entorno real

El Sistema Único de Identificación Nacional (SUIN), es un sistema informático desarrollado por el Ministerio del Interior de Cuba (MININT) en conjunto con la Universidad de las Ciencias Informáticas, fue creado a partir del rediseño de los procesos con un enfoque de simplificación de trámites, permite la integración de las múltiples bases de datos y sistemas existentes en la actualidad y se ajusta a las legislaciones y normativas tanto nacionales como extranjeras suscritas por el país.

Para la validación del algoritmo propuesto se procesaron las trazas del Sistema Único de Identificación Nacional (SUIN), específicamente de los módulos Gestionar Roles y Gestionar Recursos.

3.2.1. Proceso Gestionar Roles

El registro de evento correspondiente al proceso Gestionar Roles, aunque contenía solo 31 casos, 804 eventos, 52 clases de eventos y 3 tipos de eventos, permitió determinar algunas anomalías que se presentaban en dicho proceso.

La realización del diagnóstico del proceso consistió en un conjunto de pasos enunciados a continuación.

El primer paso consistió en aplicar el algoritmo de alineación de trazas desarrollado por Bose y Van der Aalst (BOSE, R. P. JAGADEESH CHANDRA and AALST 2012). La Figura 3.1 muestra la alineación obtenida a partir del registro de evento. Se procedió a analizar los resultados de la alineación en busca

de patrones significativos o anomalías visibles, los resultados no fueron significativos pues la alineación solo arrojó la existencia de algunos patrones como secuencia entre bloque de actividades y selecciones exclusivas.

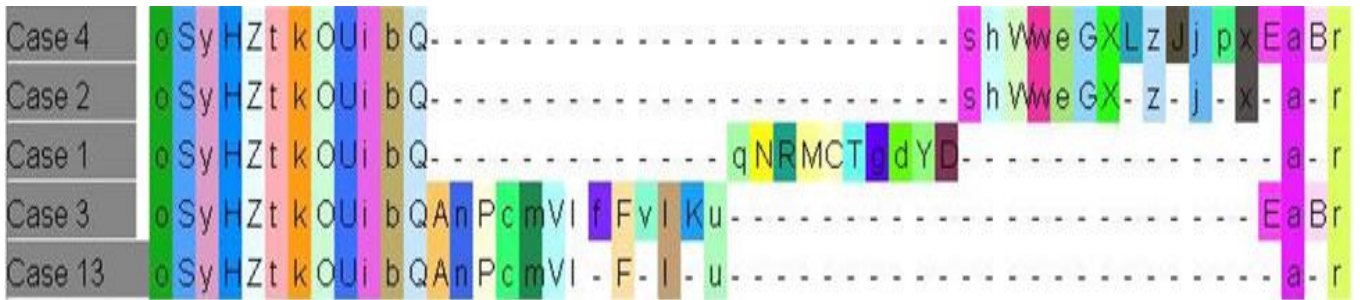


Figura 3.1. Alineación de las trazas

A la matriz obtenida de la alineación (Figura 3.1) se le aplica el algoritmo propuesto y se obtiene el árbol de bloques de construcción que se muestra en la Figura 3.2 (panel izquierdo). En la Figura 3.2 aparece seleccionado el bloque de construcción BB_2_4 (encerrado en el círculo) el cual corresponde al último subprocesso producto de la descomposición mediante una secuencia del bloque de construcción BB_1_1. Se escoge este bloque de construcción porque posibilita saber cómo termina el proceso. BB_2_4 contiene solo dos casos, el primero con multiplicidad 12 y el segundo con multiplicidad 19. El primero de los casos está asociado a la actividad B la cual representa al evento fallo de la actividad Gestionar Roles como se muestra en la Figura 3.3. Es notorio que este proceso falló 12 de las 31 veces que se ejecutó, lo que representa el 38,7 %. En consecuencia, se buscaron las causas de los fallos del proceso analizado. Se busca el origen de las causas en el bloque de construcción BB_2_3, que representa el subprocesso Condicionar Operación (incluye las posibles acciones relacionadas con agregar, editar o eliminar un rol). De la descomposición de BB_2_3 se obtienen dos bloques de construcción, el BB_3_5 y BB_3_6, los cuales representan opciones de una selección. BB_3_6 representa al subprocesso de Crear Rol y el mismo no ejercía ninguna influencia sobre el fallo del proceso Gestionar Roles.

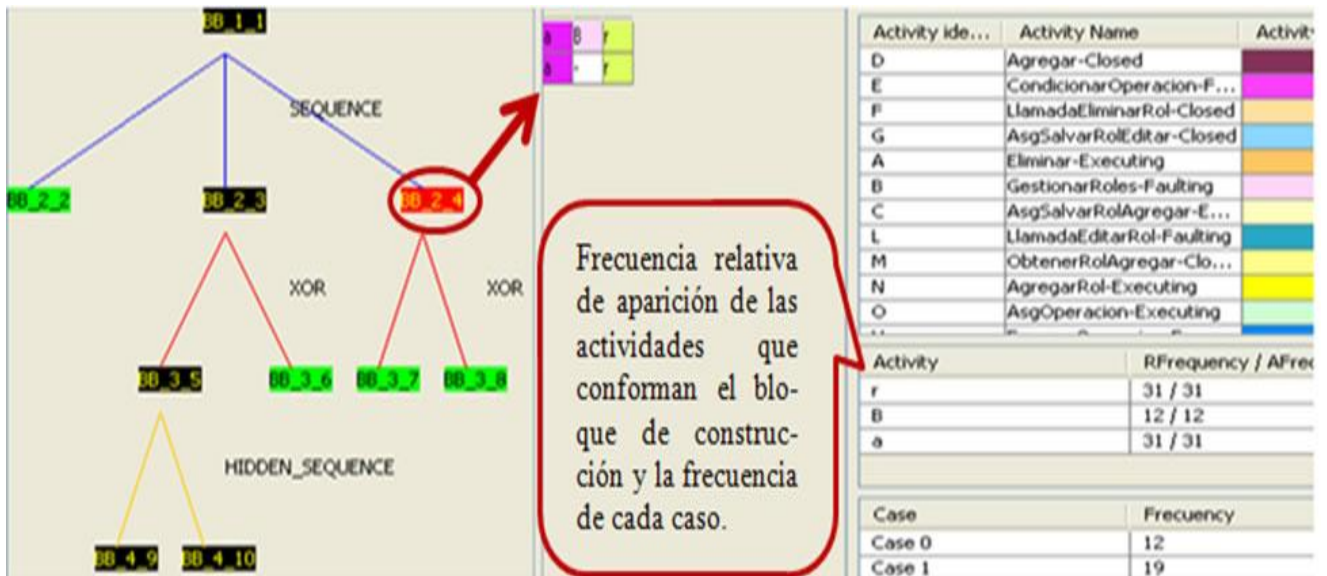


Figura 3.2. Árbol de bloques de construcción.

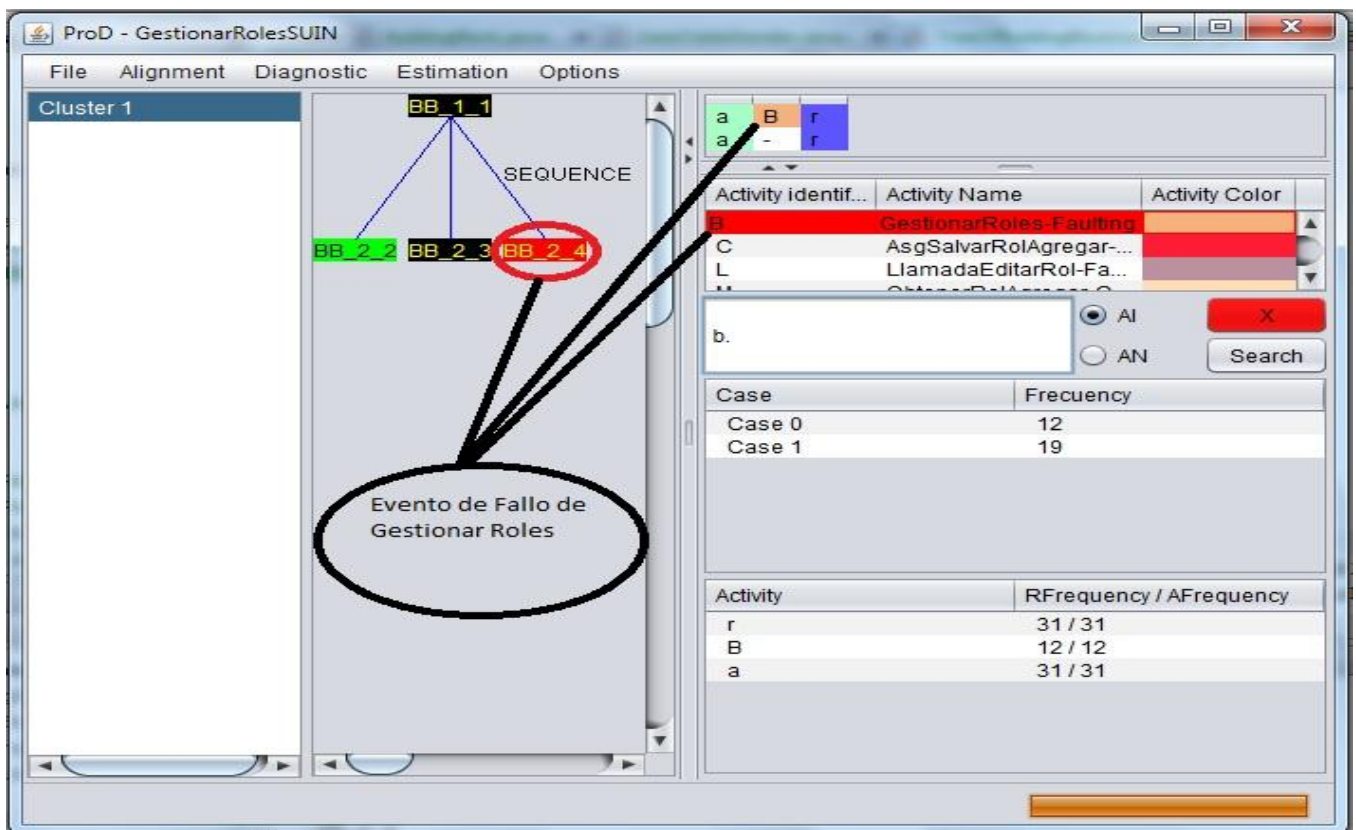


Figura 3.3. Ocurrencia de fallo en el proceso gestionar roles (actividad B).

Por otra parte, la descomposición de BB_3_5 separó los bloques representativos de los subprocesos de Editar Rol y Eliminar Rol, los cuales mostraban el origen de los fallos. Manifestándose en los bloques de construcción BB_4_9 y BB_4_10 eventos de fallo que evidenciaban el incorrecto uso de la aplicación como se muestra en la Figura 3.4.

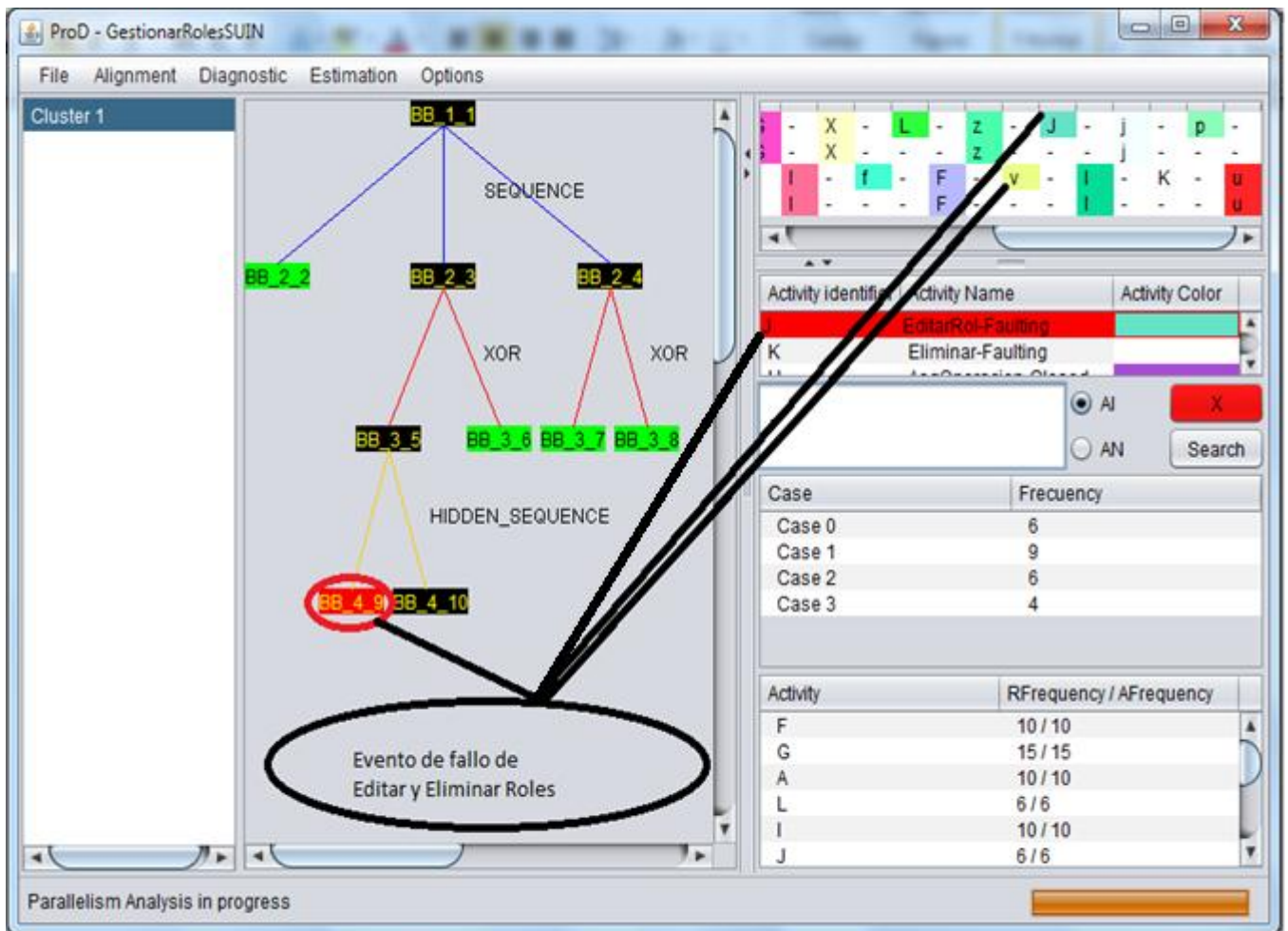


Figura 3.4. Eventos de fallo de editar y eliminar roles respectivamente.

Como conclusión del análisis se determina que uno de los administradores del sistema cerraba la ventana del navegador dejando incompleto el proceso en curso en las actividades de Editar Rol o Eliminar Rol.

El algoritmo desarrollado e implementado permitió detectar las anomalías o deficiencias que se produjeron en el proceso, definiéndose con claridad el origen de las mismas. Como se hace uso de la alineación de las trazas para representar el contenido de cada bloque de construcción es posible

aprovechar en cada momento las ventajas de esta técnica antes desarrollada. El entendimiento del proceso se facilita al establecer una descomposición jerárquica del mismo y contar con la relación que se establece entre los bloques de construcción, además de los valores de frecuencia relativa de actividades y casos en el registro de eventos.

3.2.2. Proceso Gestionar Recursos

El registro de eventos del módulo Gestionar Recursos del Sistema Único de Identificación Nacional contenía 51 casos, 11011 eventos, 90 clases de eventos y 3 tipos de eventos.

El primer paso consistió en aplicar el algoritmo de alineación de trazas desarrollado por Bose y Van der Aalst (BOSE, R. P. JAGADEESH CHANDRA and AALST 2012). La Figura 3.5 muestra la alineación obtenida a partir del registro de evento.

9b5853c-5d6d-4fb0-be9f-48a5dd2edffd	g4	i2	e4	b9	a2	e3	f0	c6	e6	g9	e5	h7	a5
c419c7ba-59d0-45b6-8e93-dc1c98684641	g4	i2	e4	b9	a2	e3	f0	c6	e6	g9	e5	h7	a5
5686082d-b9f6-429b-a991-df85a451a65c	g4	i2	e4	b9	a2	e3	f0	c6	e6	g9	e5	h7	a5
119168f8-250e-4d6a-ba8b-12ebb4cb8496	g4	i2	e4	b9	a2	e3	f0	c6	--	--	--	--	--
312fe41c-a5f4-4efc-8eb5-f70fb5e28b3b	g4	i2	e4	b9	a2	e3	f0	c6	--	--	--	--	--
2120d44c-4dba-44a9-b989-9f4b2b743a13	g4	i2	e4	b9	a2	e3	f0	c6	--	--	--	--	--
b1823be1-ae8b-4538-863f-99548f114776	g4	i2	e4	b9	a2	e3	f0	c6	--	--	--	--	--
bca6b180-b7bb-4df3-8d55-e5aa7c8203fe	g4	i2	e4	b9	a2	e3	f0	c6	--	--	--	--	--
cb9848ce-8c47-45b0-a8f0-f08de8712912	g4	i2	e4	b9	a2	e3	f0	c6	--	--	--	--	--
4c1b27d7-b75c-40ad-bf30-deaa36b83a8d	g4	i2	e4	b9	a2	e3	f0	c6	--	--	--	--	--
pac15d4b-8f0c-4460-b71c-5d9ea8274b1b	g4	i2	e4	b9	a2	e3	f0	c6	--	--	--	--	--
1ab2869c-5a60-4017-8396-37cf028fea88	g4	i2	e4	b9	a2	e3	f0	c6	--	--	--	--	--
d0f8eaf6-8b1c-4c67-a51f-1f833ad01cd6	g4	i2	e4	b9	a2	e3	f0	c6	--	--	--	--	--
ad243d9e-ebaa-400e-a3f2-e35e6a0f4691	g4	i2	e4	b9	a2	e3	f0	c6	--	--	--	--	--

Figura 3.5. Alineación de trazas.

La matriz obtenida de la alineación resultó de tamaño significativo, tales circunstancias dieron por nulas las posibilidades de determinar resultados relevantes en cuanto a búsqueda de patrones y anomalías.

A la matriz obtenida de la alineación (Figura 3.5) se le aplica el algoritmo propuesto y se obtiene el árbol de bloques de construcción que se muestra en la Figura 3.6 (panel central).

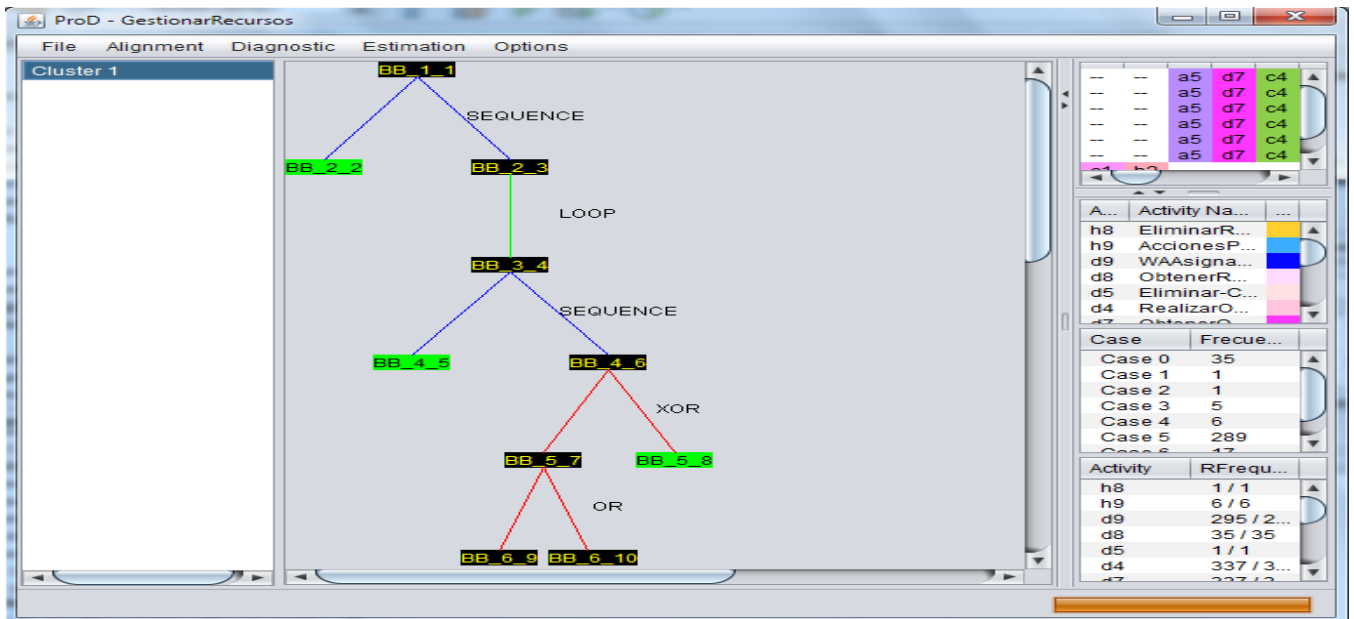


Figura 3.6. Detalle del árbol de bloques de construcción.

A partir de un análisis de los tipos de eventos se determinaron un conjunto de 11 actividades de fallo que se enuncian a continuación: fallo de *GestionarRecurso*, fallo de *AccionesParalelas*, fallo de *FlujoNegocio*, fallo de *CicloRepetición*, fallo de *SecuenciaCiclo*, fallo de *RealizarOperación*, fallo de *CondicionarOperación*, fallo de *Agregar*, fallo de *AgregarRecurso*, fallo de *WAAsignarAplicación*, fallo de *LlamadaAsignarRecurso*. Haciendo un análisis de rendimiento se determina que cada una de las actividades de fallo se ejecutó un total de 6 veces, la figura 3.7 es una muestra de la actividad de fallo de *GestionarRecursos*. Al determinarse que todas las actividades de fallo ocurrían la misma cantidad de veces, además encontrándose en este conjunto la actividad de fallo del proceso general (*GestionarRecurso*), se deduce que entre todas las actividades fallidas debía existir un orden jerárquico y que la ocurrencia de estas no era más que el resultado de una propagación de los fallos desde la actividad de menor jerarquía. El estudio fue centrado en la determinación del orden jerárquico

entre las actividades con fallo incluyendo la determinación de la actividad de menor orden, la cual sería la causa origen de los fallos.

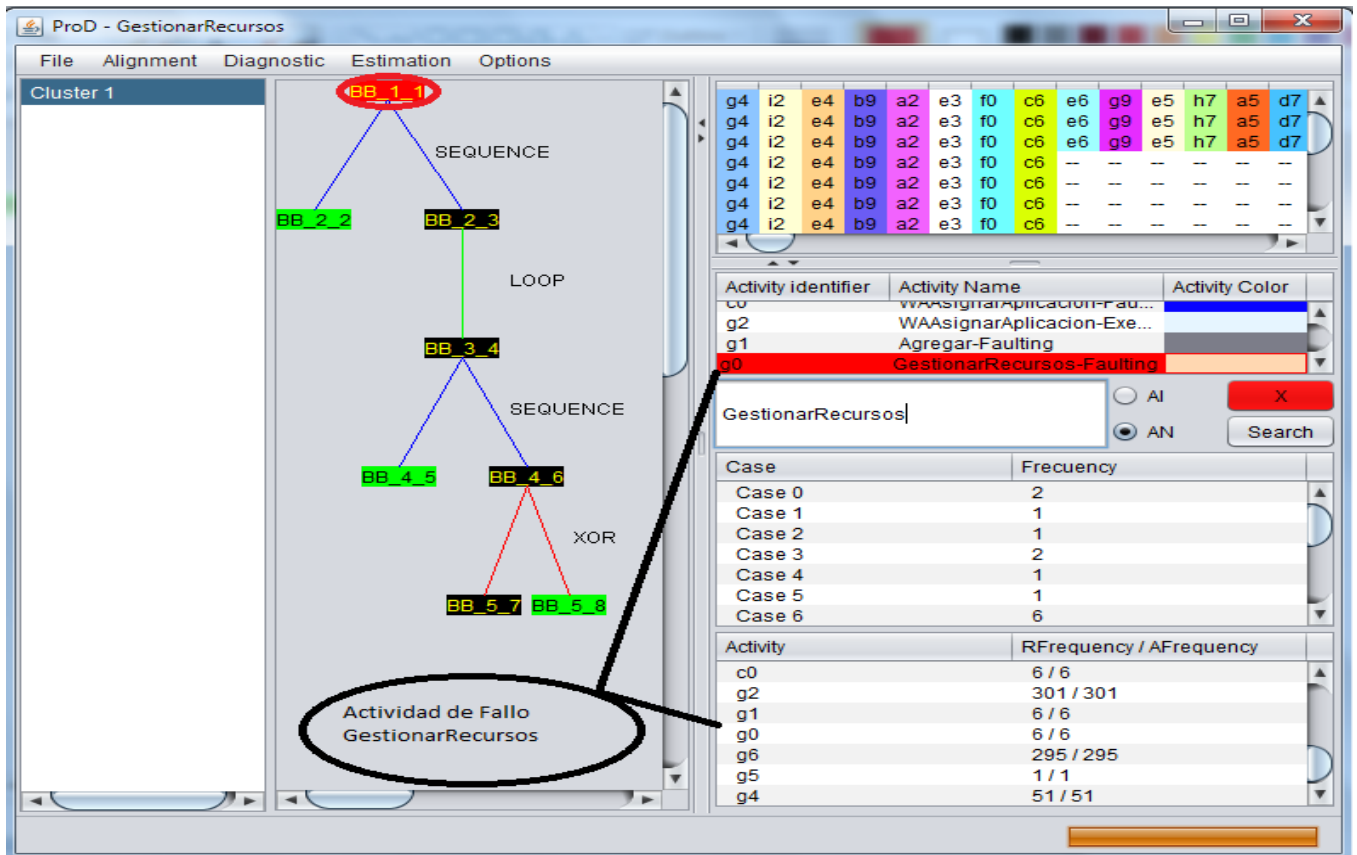


Figura 3.7. Actividad de fallo GestionarRecursos (g0).

Un análisis riguroso de los niveles de abstracción entre las actividades antes mencionadas, utilizando el árbol de bloques de construcción que dio como salida el algoritmo propuesto, arrojó la jerarquía entre actividades que se muestra en la figura 3.8.

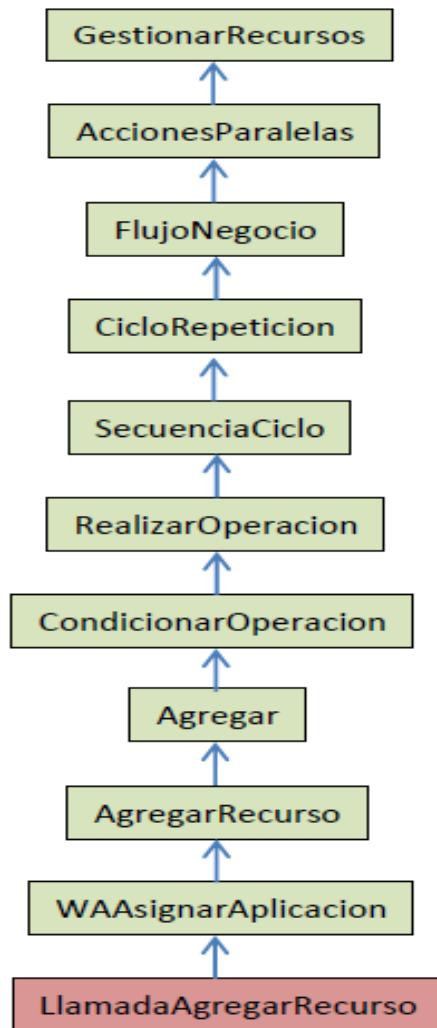


Figura 3.8. Jerarquía entre las actividades de fallo.

Este análisis dio la posibilidad de determinar que la actividad *LlamadaAgregarRecurso* era la actividad de fallo de menor jerarquía en el registro de eventos y por tanto la causa de estos. La actividad detectada tenía como objetivo almacenar en bases de datos el recurso creado, al producirse un fallo en la ejecución de esta actividad se evidencia un error en la implementación del sistema, el análisis realizado favoreció la posterior mejora de la solución informática Sistema Único de Identificación Nacional.

3.3. Conclusiones del capítulo

En este capítulo se demuestra que el algoritmo es útil al ser utilizado por un analista del negocio, ya que es capaz de responder un conjunto de interrogantes cruciales en el momento de hacer diagnóstico del proceso.

El diagnóstico de los registros de eventos seleccionados permitió la solución de problemas que no habían sido detectados en el Sistema Único de Identificación Nacional durante la fase de desarrollo.

Con la utilización de la alineación de trazas, primer paso propuesto en el algoritmo, se pueden determinar un conjunto de patrones a simple vista, que aunque en grandes registros de eventos no brindan mucha utilidad, son una buena base para comenzar el entendimiento del proceso. El árbol de bloques de construcción es una herramienta esencial especialmente a la hora de determinar los patrones interesantes en el proceso, las desviaciones del proceso original, los elementos comunes, el contexto en el que se ejecutan las actividades y además es una herramienta esencial a la hora de entender el proceso fundamentalmente en casos en los que no se cuenta con un modelo. Es útil también tener en cuenta que las variables de análisis de rendimiento que se pueden obtener del árbol de bloque de construcción representan una utilidad a la hora de determinar anomalías en el log de eventos, como se mostró en los casos de estudio Gestionar Roles y Gestionar Recursos del Sistema Único de Identificación Nacional.

Conclusiones Generales

El diagnóstico del proceso representa una ventaja a la hora de entender el proceso en etapas tempranas de su análisis, además es útil a la hora de tratar procesos tanto estructurados como desestructurados, también constituye una solución a un conjunto de interrogantes que puede confrontar un analista del negocio. En cambio, las técnicas desarrolladas en el área del diagnóstico del proceso presentan problemas a la hora de resolver algunas de las interrogantes antes mencionadas.

Por otra parte, la técnica propuesta permite detectar los patrones más comunes de ejecución, las desviaciones en el registro de eventos considerando también diferentes niveles de abstracción entre las actividades. Permite combinar la frecuencia de ocurrencia de los casos y actividades con la descomposición del proceso lo cual facilita el diagnóstico.

Una vez finalizado el desarrollo de la técnica se utilizó el registro de eventos del Sistema Único de Identificación Nacional (SUIN), específicamente de los módulos Gestionar roles y Gestionar recursos para validar el algoritmo propuesto, el algoritmo fue capaz de responder un conjunto de interrogantes que definen el diagnóstico de un proceso, además de determinar algunas anomalías o eventos que no se esperaba sucedieran durante la ejecución, lo que facilitó la posterior mejora del sistema, quedando así demostrada la efectividad del algoritmo propuesto para el diagnóstico del proceso.

Recomendaciones

El diagnóstico del proceso en la Minería de proceso es un aspecto fundamental ya que es útil a la hora de tratar procesos tanto estructurados como desestructurados, por otra parte, es muy importante tener una idea inicial del proceso analizado, así como los principales aspectos que caracterizan al proceso, tales como Análisis de rendimiento del proceso, inspección y detección de los principales patrones de flujo de trabajo y detección de anomalías. Por estas razones se recomienda:

- Aplicar el algoritmo propuesto para la detección de subprocessos mediante la alineación de las trazas de procesos desestructurados en empresas que lo requieran para el tratamiento de sus procesos de negocio.
- Identificar otros patrones de flujo de trabajo como es el caso de: *Non Free Choice*.

Glosario de términos

- **Actividad:** es un paso bien definido en el proceso. Los eventos pueden referirse al inicio, conclusión, cancelación, etc., de una actividad para una instancia específica del proceso.
- **Fitness:** es una medida para determinar cuán bien un modelo dado se ajusta al comportamiento observado en el registro de evento. Un modelo tiene un ajuste perfecto si todas las trazas en el registro de evento pueden ser reproducidas por el modelo de principio a fin.
- **Caso:** véase **Instancia de un Proceso**.
- **Minería de Dato:** análisis de conjuntos de datos (a menudo grandes) para encontrar relaciones inesperadas y para resumir los datos, de manera que proporcionen nuevos entendimientos.
- **Descubrimiento de Procesos:** es uno de los tres tipos básicos de minería de proceso. Basado en un registro de evento, se crea un modelo de proceso. Por ejemplo, el algoritmo *Alpha* es capaz de descubrir una red de Petri mediante la identificación de patrones de procesos en colecciones de eventos.
- **Evento:** es una acción almacenada en el registro, por ejemplo, el inicio, conclusión o cancelación de una actividad para una instancia particular de un proceso.
- **Gestión de Procesos de Negocio (BPM):** es la disciplina que combina conocimiento sobre tecnología de información y conocimiento sobre las ciencias de gestión y lo aplica en conjunto a los procesos de negocio operacionales.
- **Inteligencia de Negocios (BI):** es una amplia colección de herramientas y métodos que utilizan datos para apoyar la toma de decisiones.
- **Instancia de un Proceso:** es la entidad siendo ejecutada por el proceso que es analizado. Los eventos se refieren a instancias del proceso. Ejemplos de instancias de un proceso son: pedidos de los clientes, reclamos de seguros, solicitudes de préstamos, etc.
- **Minería de Proceso:** son técnicas, herramientas y métodos para descubrir, monitorear y mejorar los procesos reales (es decir, no los procesos supuestos) a través de la extracción de conocimiento de los registros de eventos, ampliamente disponibles en los actuales sistemas de información.
- **MXML:** es un formato basado en XML para el intercambio de registros de eventos. XES reemplaza a MXML como el nuevo formato para minería de proceso no dependiente de la herramienta.

- **XES:** es un estándar XML para los registros de eventos. El estándar ha sido adoptado por la *IEEE Task Force on Process Mining* como el formato de intercambio de registros de eventos por defecto.
- **Registro de Evento:** es la colección de eventos utilizados como entrada para la minería de proceso. Los eventos no necesitan ser almacenados en un archivo de registro por separado (por ejemplo, los eventos pueden estar dispersos en diferentes tablas de bases de datos).
- **Verificación o chequeo de Conformidad:** analiza si la realidad, según consta en un registro de evento, se ajusta al modelo y viceversa. El objetivo es detectar las discrepancias y medir su gravedad. La verificación de conformidad es uno de los tres tipos básicos de minería de proceso.

Bibliografía

- AALST, W. M. P. V. D. *Business Process Simulation Revisited*. Enterprise and Organizational Modeling and Simulation, Lecture Notes in Business Information Processing. Springer, Berlin., 2010. 1–14 p.
- AALST, W. M. P. V. D. *Process Mining. Discovery, Conformance and Enhancement of Business Processes*. Springer Heidelberg Dordrecht London New York, 2011. p. 978-3-642-19344-6
- AALST, W. M. P. V. D. Structural Characterizations of Sound Workflow Nets *Computing Science Reports 96/23*, 1996a.
- AALST, W. M. P. V. D. *Three Good Reasons for Using a Petri-net-based Workflow Management System*. The International Working Conference on Information and Process Integration in Enterprises (IPIC'96), 1996b. 179-201 p.
- AALST, W. M. P. V. D.; A. ADRIANSYAH, *et al.* Process Mining Manifesto *Business Process Management Workshops 2011, Lecture Notes in Business Information Processing*. Springer-Verlag, 2011a, 99.
- AALST, W. M. P. V. D.; B. F. V. DONGEN, *et al.* *ProM: The Process Mining Toolkit*. Proceedings of BPM (Demos)'2009, Ulm, Germany, 2009a. p.
- AALST, W. M. P. V. D.; K. M. V. HEE, *et al.* Soundness of Workflow Nets: Classification, Decidability and Analysis *Formal Aspects of Computing*, 2011b.
- AALST, W. M. P. V. D.; A. H. M. T. HOFSTEDE, *et al.* Business Process Management: A Survey *Lecture Notes in Computer Science*, 2003, 2678.
- AALST, W. M. P. V. D.; H. A. REIJERS, *et al.* Business process mining: An industrial application *Information Systems*, 2007, 32(5).
- AALST, W. M. P. V. D.; V. RUBIN, *et al.* ProcessMining: A Two-Step Approach to Balance Between Underfitting and Overfitting *Software and Systems Modeling*, 2009b, 9(1): 87-111.
- AALST, W. M. P. V. D.; M. H. SCHONENBERG, *et al.* Time Prediction Based on Process Mining *Information Systems*, 2011c, 36(2): 450–475.
- AALST, W. M. P. V. D. and A. J. M. M. WEIJTERS Process Mining: A Research Agenda *Special Issue of Computers in Industry, Elsevier Science Publishers, Amsterdam*, 2004, 53(3).
- AALST, W. M. P. V. D.; A. J. M. M. WEIJTERS, *et al.* Workflow Mining: Discovering process models from event logs *IEEE Transactions on Knowledge and Data Engineering*, 2004, 16(9): 1128-1142.
- AGRAWAL, R.; D. GUNOPULOS, *et al.* *Mining Process Models from Workflow Logs*. EDBT '98 Proceedings of the 6th International Conference on Extending Database Technology: Advances in Database Technology, Springer-Verlag London, UK, 1998. 1-15 p. 3-540-64264-1
- BERGENTHUM, R.; J. DESEL, *et al.* Process Mining Based on Regions of Languages *Lecture Notes in Computer Science*, 2007, 4714: 375-383.

- BOSE, R. P. J. C. and W. M. P. V. D. AALST. Abstractions in Process Mining: A Taxonomy of Patterns. en. DAYAL, U.;EDER, J.*et al*, Springer Berlin / Heidelberg, 2009. 5701: 159-175.p.
- BOSE, R. P. J. C. and W. M. P. V. D. AALST Process diagnostics using trace alignment: Opportunities, issues, and challenges *Inf. Syst.*, 2012, 37(2): 117-141.
- BOSE, R. P. J. C. and W. M. P. V. D. AALST. *Trace Alignment in Process Mining: Opportunities for Process Diagnostics*. International Conference on Business Process Management (BPM'2010), Springer-Verlag Berlin, Heidelberg, 2010. 227-242 p. 3-642-15617-7 978-3-642-15617-5
- CHEUNG, C. F., YU, J.X., LU, H. Constructing Suffix Tree for Gigabyte Sequences with Megabyte Memory *IEEE Trans. Knowl. Data Eng.*, 2005, 17(1).
- COOK, J. E. and A. L. WOLF. *Automating Process Discovery Through Event-Data Analysis*. ICSE '95: Proceedings of the 17th international conference on Software engineering, New York, USA, ACM Press, 1995. 1-10 p. 0-89791-708-1
- DONGEN, B. F. and A. ADRIANSYAH. Process Mining: Fuzzy Clustering and Performance Visualization Business Process Management Workshops. en. RINDERLE-MA, S.;SADIQ, S.*et al*, Springer Berlin Heidelberg, 2010a. 43: 158-169.p.
- DONGEN, B. F. V. and W. M. P. V. D. AALST Multi-phase Process Mining: Building Instance Graphs *Lecture Notes in Computer Science*, 2004, 3288: 362-376.
- DONGEN, B. F. V. and A. ADRIANSYAH. *Process Mining: Fuzzy Clustering and Performance Visualization*. Business Process Management Workshops, Lecture Notes in Business Information Processing, 2010b. 158-169 p.
- GAMA, J. M. and J. CARMONA A fresh look at Precision in Process Conformance, 2010.
- GOEDERTIER, S.; D. MARTENS, *et al*. *Process Mining as First-Order Classification Learning on Logs with Negative Events*. BPM 2007 International Workshops (BPI, BPD, CBP, ProHealth, RefMod, Semantics4ws), Lecture Notes in Computer Science. Springer, Berlin., 2008. 42–53 p.
- GOEDERTIER, S.; D. MARTENS, *et al*. Robust Process Discovery with Artificial Negative Events *Journal of Machine Learning Research*, 2009.
- GRIGORI, D.; F. CASATI, *et al*. Business Process Intelligence *Computers in Industry*, 2004, 53(3): 321-343.
- GÜNTHER, C. W. *Process Mining in Flexible Environments*, Eindhoven University of Technology, 2009. p.
- GÜNTHER, C. W. and W. M. P. V. D. AALST. *Fuzzy Mining: Adaptive Process Simplification Based on Multi-Perspective Metrics*. International Conference on Business Process Management (BPM 2007), Lecture Notes in Computer Science. Springer, Berlin., 2007. 328-343 p. 3-540-75182-3, 978-3-540-75182-3

- HENDRICKS, K. B.; V. R. SINGHAL, *et al.* The impact of enterprise systems on corporate performance: A study of ERP, SCM, and CRM system implementations *Operations Management*, 2007, 25(1): 65-82.
- LI, J.; R. P. J. C. BOSE, *et al.* *Mining Context-Dependent and Interactive Business Process Maps using Execution Patterns*. BPM 2010 Workshops, Springer-Verlag., 2010. 109 -121 p.
- MEDEIROS, A. K. A. D. *Genetic Process Mining*, Technische Universiteit Eindhoven, 2006. p.
- MEDEIROS, A. K. A. D.; W. M. P. V. D. AALST, *et al.* *Workflow Mining: Current Status and Future Directions*. The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, Lecture Notes in Computer Science. Springer, Berlin., 2003. 389–406 p.
- MENDLING, J.; G. NEUMANN, *et al.* *Understanding the Occurrence of Errors in Process Models Based on Metrics*. Proceedings of the OTM Conference on Cooperative Information Systems (CoopIS 2007), Lecture Notes in Computer Science. Springer, Berlin., 2007. 113-130 p.
- ROZINAT, A. and W. M. P. V. D. AALST Conformance Checking of Processes Based on Monitoring Real Behavior, 2008.
- ROZINAT, A.; A. K. A. D. MEDEIROS, *et al.* *The Need for a Process Mining Evaluation Framework in Research and Practice*. . BPM 2007 International Workshops (BPI, BPD, CBP, ProHealth, RefMod, Semantics4ws), Lecture Notes in Computer Science. Springer, Berlin., 2008. 84-89 p.
- RUBIN, V. *A Workflow Mining Approach for Deriving Software Process Models*, University of Paderborn, 2007. p.
- SONG, M.; C. W. GÜNTHER, *et al.* *Trace Clustering in Process Mining*. Business Process Management Workshops (2009), Milano, Italy Lecture Notes, 2009. 109-120 p.
- WEIJTERS, A. J. M. M. and W. M. P. V. D. AALST Rediscovering Workflow Models from Event-Based Data using Little Thumb *Integrated Computer-Aided Engineering*, 2003: 151-162.
- WEIJTERS, A. J. M. M. and J. T. S. RIBEIRO Flexible Heuristics Miner (FHM). *BETA Working Paper Series, WP 334, Eindhoven University of Technology, Eindhoven.*, 2010.