Universidad de las Ciencias Informáticas

Facultad 3

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

Algoritmo para la estimación de información ausente en las trazas utilizadas en la minería de proceso.

Autor: Yaidel Guedes Beltrán

Tutor: MSc. Raykenler Yzquierdo Herrera

Ciudad de la Habana, Cuba

Mayo, 2012

DECLARACIÓN DE AUTORÍA

MINISTERIO DE EDUCACIÓN SUPERIOR

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Declaro ser autor de la presente tesis y reconozco 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 _____.

Yaidel Guedes Beltrán

MSc. Raykenler Yzquierdo Herrera

Firma del autor

Firma del tutor

AGRADECIMIENTOS

En primer lugar a mis padres por creer en mí, por su apoyo incondicional, por todo su sacrificio y su infinito amor, porque este es su sueño hecho realidad, pues lo que soy se lo debo a ellos.

A mi novia Marita por su amor y apoyo incondicional durante todos los años que llevamos juntos.

A Raykenler y al equipo de trabajo por su apoyo y ayuda en todo momento.

Quiero agradecer a todas las personas con las cuales he tenido la dicha de compartir estos maravillosos cinco años que nunca olvidaré.

A todos los que me ayudaron de una u otra forma a realizar este trabajo…

Muchas gracias.

RESUMEN

La mayoría de las empresas utilizan sistemas de información para soportar la ejecución de sus procesos, los cuales registran en forma de trazas las acciones que se van realizando cuando se ejecutan instancias o casos del proceso de negocio. Al descubrimiento del proceso a partir de la información contenida en las trazas se le denomina minería de proceso. La mayoría de los algoritmos existentes de descubrimiento parten del supuesto que las trazas son completas y están libres de ruido, lo cual en la realidad rara vez se cumple, dando como resultado que los modelos descubiertos se vean afectados tanto estructuralmente como en su comprensión. En este trabajo se propone un algoritmo para la estimación de la información ausente en las trazas utilizadas para la minería de proceso, basado en la alineación las mismas y su pre procesamiento. Al finalizar esta investigación se valida la propuesta de solución mediante un grupo de métricas y la aplicación de la solución informática desarrollada a un proceso real.

Palabras claves: Ausencia de información, registro de evento, minería de proceso, modelo, traza.

TABLA DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

Actualidad del tema

La optimización de los procesos de negocio ha sido una estrategia de las empresas para reducir costos y aumentar su expansión en el mercado, así lo refleja un estudio realizado por IBM (IBM 2010), brindando como resultado que las empresas con mayor rendimiento son las que se han centrado en aumentar la agilidad de adaptarse ante los cambios mediante prácticas de trabajo y procesos empresariales. Los sistemas de información que soportan la gestión de los procesos de negocio han aumentado, ejemplo de ello lo constituye los ingresos que está obteniendo IBM en su línea de negocio enfocada a la gestión de procesos empresariales y software de integración.

Los Sistemas para la Gestión de Procesos de Negocio (BPMS, por sus siglas en inglés) (AALST, W.M.P. VAN DER and HEE 2004; HILL *et al.* 2009) posibilitan que se realice el modelado, automatización y el análisis de los procesos de la empresa posibilitando una mejor comprensión y desempeño actual (DAA 2010).

Otros tipos de sistemas de información, tales como los sistemas de Planificación de Recursos Empresariales (ERP, por sus siglas en inglés), Gestión de Cadena de Suministros (SCM, por sus siglas en inglés) y Gestión de la Relación con los Clientes (CRM, por sus siglas en inglés), posibilitan también gestionar los procesos de negocio de la organización (CHANGA *et al.* 2008; HENDRICKSA *et al.* 2007; SAP 2010; TARANTILISA *et al.* 2008). Estos sistemas soportan las diferentes tareas que se realizan como parte de un proceso que puede o no estar explícitamente modelado. Los sistemas de información registran en forma de trazas las acciones que se van realizando cuando se ejecutan instancias o casos del proceso de negocio.

Las trazas o registro de evento contienen información de los casos como puede ser: tiempo en el que se ejecutó una tarea, sistema en el que se realizó, usuario que ejecutó una acción, entre otros datos. El descubrimiento del proceso a partir de la información contenida en las trazas se le denomina minería de proceso o de flujo de trabajo (AALST, W.M.P. VAN DER 2011; AGRAWAL *et al.* 1998; COOK 1996; COOK and WOLF. 1995). Esta es un área de investigación relativamente nueva que permite hacer un análisis objetivo de los procesos basado en sus ejecuciones actuales.

El descubrimiento del modelo de procesos que se ejecuta es uno de los beneficios, sin embargo, la minería de proceso permite también realizar análisis delta, mediante la comparación del modelo prescrito o teórico con el modelo descubierto, también es

posible apoyar el rediseño de los procesos en la empresa (AALST, WIL M. P. VAN DER *et al.* 2003; AALST, W.M.P. VAN DER and WEIJTERS 2004b; AALST, W.M.P. VAN DER *et al.* 2004).

Un proceso puede ser analizado considerando las siguientes tres perspectivas o dimensiones, la perspectiva de control de flujo, la de los recursos (también llamada organizacional) y la de casos. En dependencia de la información contenida en las trazas disponibles se selecciona la o las perspectivas a analizar.

La más abordada de las perspectivas es la de control de flujo (DONGEN, BOUDEWIJN VAN 2007; MEDEIROS 2006), la cual posibilita determinar el cómo se organizan las tareas que dan lugar al proceso, quedando claramente definidas las dependencias existentes. En esta dimensión se debe contar con información relacionada con la identificación del caso o instancia ejecutada, la identificación de la tarea y el tiempo en el que se produjo la tarea. Las diferentes técnicas desarrolladas buscan identificar a partir de las trazas un modelo representativo del proceso en el que queden reflejadas las dependencias entre las diferentes tareas que han sido registradas. Los algoritmos existentes manejan generalmente un conjunto de constructores de flujo (AALST, W.M.P. VAN DER *et al.* 2007; DONGEN, BOUDEWIJN VAN 2007; MEDEIROS 2006) que permiten reconocer una determinada estructura que aparece de manera común en el flujo de tareas.

Algunos de los constructores de flujo de trabajo más comunes son:

- Secuencia
- Paralelismo
- Selección
- Lazos
- Selección no libre
- Tareas invisibles
- Tareas duplicadas.

Aún cuando existen trabajos que cubren la totalidad de los constructores de flujo de trabajo enunciados, la mayoría de los algoritmos desarrollados dan un cubrimiento parcial a los constructores de flujo (AALST, WIL M. P. VAN DER *et al.* 2003; AALST, W.M.P. VAN DER and WEIJTERS 2004b).

Los algoritmos existentes pueden tener problemas para manejar determinados constructores de flujo de trabajo debido a que la notación que utilizan para representar el modelo de procesos descubierto no los soporta (MEDEIROS 2006).

También se puede señalar que en situaciones reales la minería de proceso se enfrenta a otros factores como es el caso del ruido presente en las trazas.

Los sistemas generadores de trazas pueden almacenar trazas de manera incorrecta o en las trazas se pueden registrar situaciones excepcionales que pueden afectar posteriormente el modelo resultante a partir de alguna de las técnicas de minería de proceso (AALST, W.M.P. VAN DER *et al.* 2007; DONGEN, BOUDEWIJN VAN 2007; MEDEIROS 2006).

Problemática

La mayoría de los algoritmos de minería de proceso parten del supuesto que las trazas son completas y están libres de ruido. En la realidad este supuesto rara vez se cumple (AALST, W. M. P. V. D. and GÜNTHER 2007). El término completo se refiere a que las trazas reflejan de manera exacta el comportamiento de cada una de las instancias ejecutadas del proceso. Si las trazas reflejan parcialmente el comportamiento de cada una de las instancias ejecutadas del proceso se considera que las mismas presentan ausencia de información.

La ausencia de información se refleja de dos formas:

1. En el hecho que determinadas instancias de un proceso pueden no estar registradas en las trazas debido a que no han ocurrido, aún cuando dichas instancias pudiesen ser soportadas por los sistemas informáticos generadores de trazas. Esta situación puede afectar el modelo descubierto (AALST, W.M.P. VAN DER 2011; AALST, W.M.P. VAN DER and WEIJTERS 2004b) por lo cual se han desarrollado trabajos orientados a resolver esta problemática.
2. Dada la ausencia de una o varias tareas del proceso en las trazas, debido a que las mismas no son registradas al no ser soportadas por los sistemas utilizados o porque las trazas fueron afectadas por el ruido que posibilitó la ausencia de determinadas actividades, a esta manifestación se le denomina ausencia de información.

Existen dos posibilidades por las cuales una actividad no queda registrada en el registro de evento: la primara causa es que la actividad no se ha informatizado y la segunda es que la actividad fue informatizada pero el sistema de información no deja constancia de su ocurrencia, o el registro de evento fue afectado por el ruido. A este tipo de actividad se le denomina actividad o tarea invisible (AALST, W.M.P. VAN DER and WEIJTERS 2004b; MEDEIROS 2006).

Los sistemas informáticos usados para informatizar un determinado procesos cubren un subconjunto de tareas, por lo cual, durante la ejecución del proceso pueden realizarse de manera intercalada tanto tareas soportadas por los sistemas informáticos

(generadores de trazas) como tareas que no son soportadas y registradas por ningún sistema. En consecuencia, los sistemas almacenan en las trazas una secuencia incompleta de las tareas que conforman las instancias del proceso.

Esta ausencia de información puede generar que los algoritmos dirigidos a descubrir el proceso ejecutado obtengan modelos en los que se reflejan incorrectas relaciones entre las tareas, siendo esto un factor determinante en la falta de estructuración de los modelos descubiertos. Es necesario señalar que no siempre la ausencia de información produce serias afectaciones en el modelo descubierto, pero por lo general, dificulta la comprensión de dicho modelo (GAMA and CARMONA 2010).

Para cubrir completamente el comportamiento reflejado en las trazas el modelo descubierto se apoya en un grupo de patrones de flujo de control, sin embargo, ante situaciones de ausencia de información los modelos obtenidos por los diferentes algoritmos de minería de proceso pueden diferenciarse significativamente, es decir, existe ambigüedad en la interpretación de las trazas. Además, para una correcta comprensión del modelo descubierto puede ser útil reflejar explícitamente, de ser posible, las actividades invisibles.

Hay que resaltar que cuando se está frente a las trazas generadas por BPMS es poco frecuente la ausencia de información debido a que las instancias del proceso se ejecutaron a partir de un modelo de proceso explícito en la herramienta utilizada.

Para evitar los efectos de lidiar con trazas que no cumplen con las condiciones previas requeridas se puede pre-procesar las trazas con anterioridad.

En este sentido, se realizan acciones de verificación, limpieza, agrupamiento y alineado de las trazas (BOSE, R.P. JAGADEESH CHANDRA and AALST 2009a; BOSE, R. P. JAGADEESH CHANDRA and AALST 2009b; BOSE, R.P. JAGADEESH CHANDRA and AALST 2010; GÜNTHER, CHRISTIAN W. *et al.* 2009; ROZINAT *et al.* 2008a; SONG *et al.* 2009). Otras soluciones se han centrado en el desarrollado de algoritmos que son robustos ante el ruido y ante algunas situaciones en las que existe ausencia de información (AGRAWAL *et al.* 1998; BERGENTHUM *et al.* 2007; MEDEIROS 2006; SCHIMM 2004; 2003).

Existen algoritmos que pueden identificar los aspectos más significativos reflejados en las trazas y reflejarlos en el modelo descubierto, permitiendo esto manejar el ruido y la ausencia de información en alguna medida. La ausencia de información ha sido abordada directamente desde la perspectiva del constructor de tareas invisibles o variables escondidas.

Sin embargo, existen situaciones de mayor complejidad en las que existe ausencia de información y no pueden ser manejadas correctamente por los algoritmos desarrollados hasta el momento (AALST, W.M.P. VAN DER and WEIJTERS 2004b; MEDEIROS 2006).

Dada la **problemática** existente se plantea el siguiente **problema a resolver**:

¿Cómo disminuir la afectación que provoca la ausencia de información en las trazas sobre la estructura y comprensión del modelo de proceso descubierto, a partir de técnicas de minería de proceso?

Constituyó **objeto de estudio** la minería de proceso y el **campo de acción** en el tratamiento de la ausencia de información en las trazas usadas en la minería de proceso.

Definiendo como **objetivo general:**

Desarrollar un algoritmo para la estimación de la información ausente en las trazas utilizadas en las técnicas de minería de proceso, basado en un árbol de bloques de construcción para el descubrimiento de modelos de proceso más estructurados y comprensibles.

Y los siguientes **objetivos específicos:**

- Fundamentar la investigación mediante la realización del Marco Teórico.
- Definir el diseño de la propuesta de solución.
- Implementar el algoritmo diseñado.
- Validar la propuesta.

Como **idea a defender** se plantea lo siguiente:

Si se desarrolla un algoritmo para la estimación de información ausente en las trazas utilizadas en la minería de proceso, deben descubrirse modelos de procesos más estructurados y comprensibles.

Para dar cumplimiento a los objetivos propuestos se trazaron las siguientes **tareas investigativas**:

- Análisis de los principales conceptos y trabajos relacionados con la minería de proceso y la ausencia de información en las trazas.
- Estudio del mecanismo de alineación de las trazas propuesto para la minería de proceso.

- Estudio del mecanismo de descomposición del proceso analizado a partir de la alineación de las trazas.
- Definición de los operadores que permitan reconocer los patrones de ausencia de información en las trazas utilizadas en las técnicas de minería de proceso.
- Implementación de los operadores definidos 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 en un entorno real.

Métodos teóricos

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

Se enfocan las problemáticas asociadas a la ausencia de información en las trazas usadas en técnicas de minería de 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 siguió además un método hipotético deductivo porque a partir del problema concreto se plantearon objetivos específicos e idea a defender que en el transcurso de la investigación fueron 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 de ausencia de información en las trazas. 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 planteó el problema como un todo, donde las trazas utilizadas, la propia dinámica de aplicación de técnicas de minería de proceso en el descubrimiento de procesos, las técnicas computacionales desarrolladas para la estimación de información ausente en las trazas se funden en un sistema sostenible e integral.

Métodos empíricos

- Experimentación
- Medición

Además de seguir métodos teóricos se siguen los métodos empíricos basando la investigación en la experimentación con datos provenientes de situaciones reales suministrados por sistemas utilizados durante el proceso de desarrollo de software.

Se aplican pruebas estadísticas para analizar la efectividad del modelo desarrollado y la calidad de las respuestas finales. Se establecen estadígrafos e indicadores adecuados que permiten realizar correctas mediciones de los resultados.

Estructura del documento

El presente trabajo está conformado por tres capítulos.

- Capítulo 1: Se da una introducción a los aspectos generales de la minería de proceso y a las estrategias utilizadas en el manejo de ausencia de información. Se hace una evaluación crítica de las ventajas y desventajas de diferentes enfoques. Se hace un análisis del estado del arte relacionado con diferentes estrategias para el manejo de la ausencia de información.
- Capítulo 2: Se describe el diseño e implementación del algoritmo para la estimación de información ausente en trazas usadas en las técnicas de minería de proceso.
- Capítulo 3: Se presenta la validación de la solución propuesta, mediante el uso de un grupo de métricas usadas en la minería de proceso y la validación de la herramienta informática desarrollada para realizar la estimación de información ausente en un proceso real.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se hace una revisión de las diferentes aristas desde las que se ha analizado el problema de ausencia de información en las trazas, considerando los trabajos de los autores más referenciados en el área de minería de proceso. También se demuestra cómo las soluciones dadas al problema hasta el momento son insuficientes.

Minería de proceso

Las técnicas de minería de proceso, permiten extraer información no trivial y útil de los registros de eventos registrados por sistemas de información. Un elemento fundamental en la minería de proceso es el descubrimiento del control de flujo, esto es la construcción automática del modelo de proceso el cual representa un proceso de negocio (Definición 1), haciendo uso de una red de Petri (Definición 2) u otra notación, en el que se describen las dependencias causales entre las actividades del procesos (AALST, W.M.P. VAN DER 2011).

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.■ (AALST, WIL M. P. VAN DER *et al.* 2003)

Definición 2 (Red de Petri): Una red de Petri es un grafo dirigido bipartito, con un estado inicial, llamado marcación inicial. Los dos componentes principales de la red de Petri son los sitios (también conocidos como estados) y las transiciones.■ (AALST, WIL M. P. VAN DER *et al.* 2003)

La minería de proceso provee un importante puente entre la minería de datos y el modelado y análisis de los procesos de negocio. Bajo el área de Inteligencia de Negocio (BI, por sus siglas en inglés) se ha difundido un grupo de términos que encierran diferentes tipos de análisis en este contexto, tales como, Monitoreo de Actividades de Negocio (BAM, por sus siglas en inglés) referido a las tecnologías que hacen posible un análisis en tiempo real de los procesos de negocio. Procesamiento de Eventos Complejos (CEP, por sus siglas en inglés) referido a las tecnologías que permiten procesar grandes cantidades de eventos para monitorear, guiar y optimizar el negocio en tiempo real.

Gestión del Rendimiento Empresarial (CPM, por sus siglas en inglés) referido a la medición del funcionamiento del proceso o la organización. Otros términos están vinculados con la gestión, como es el caso de Proceso Continuo de Mejora (CPI, por sus siglas en inglés), Proceso de Mejora de Negocio (BPI, por sus siglas en inglés), y Gestión Total de la Calidad (TQM, por sus siglas en inglés), entre otros.

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 y cada vez más sistemas de información incorporan técnicas de minería de proceso.

Algunos de estos software son:

- ARIS Process Performance Manager (AG 2011)
- Comprehend (INCORPORATED 2011)
- Discovery Analyst (STEREOLOGIC)
- Flow (FOURSPARK 2011)
- Perceptive Reflect (SOFTWARE 2012)
- Interstage Automated Process Discovery (FUJITSU 2012)
- Process Discovery Focus (IMPROVEMENT 2009)
- Process Analyzer (OYJ 2011)
- ProM (AALST, W. M. P. V. D. *et al.* 2009a)

Como punto de partida para la aplicación de técnicas de minería de proceso están las trazas, las cuales han sido generadas por uno o varios sistemas de información usados en una empresa. Las trazas están constituidas por una secuencia de eventos de los cuales se almacena información como: el nombre del evento, el tipo de evento, el usuario que ejecutó el evento, el tiempo en el que se produjo el evento.

Para la representación de un registro de evento se han definido dos estándares:

1. Lenguaje de Marcado Extensible de Minería (MXML, por sus siglas en inglés) (AALST, WIL M. P. VAN DER *et al.* 2003)
2. Secuencia de Eventos Extensible (XES, por sus siglas en inglés) (GÜNTHER, CHRISTIAN W. *et al.* 2008).

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

Definición 3 (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.■

Un ejemplo de un registro de evento se muestra en la Tabla 1.

Tabla 1 Registro de evento

Caso 1			
Descripción	Evento	Usuario	Fecha
	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 10:51
Recibir cuestionario	procesando	ryzquierdo	10/06/2011 10:51
Archivar	procesando	ryzquierdo	10/06/2011 10:52
Archivar	terminado	ryzquierdo	10/06/2011 10:52
	final	ryzquierdo	10/06/2011 10:52

Existen tres tipos de técnicas de minería de proceso como se muestra en la Figura 1.

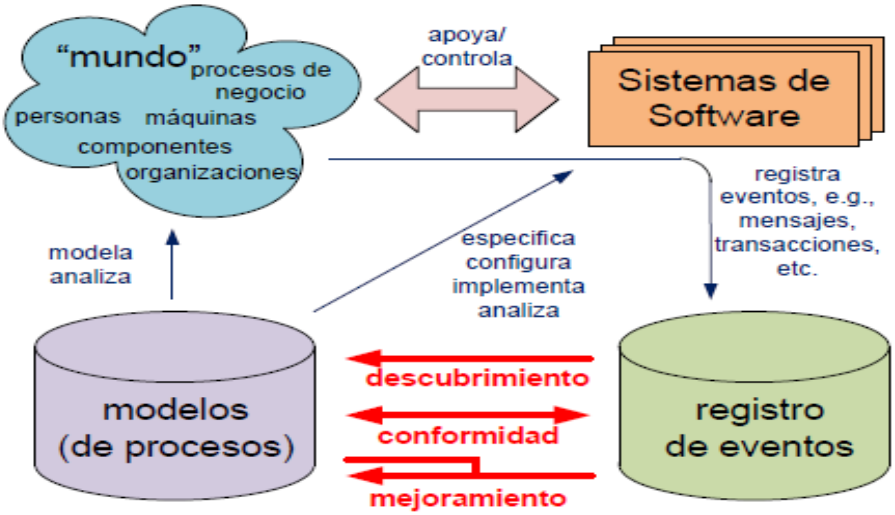


Figura 1 Tipos fundamentales de técnicas de minería de proceso: descubrimiento, conformidad y mejora.

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.

Existen 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.* 2008b)
- 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)
- Fuzzy Miner (GÜNTHER, C.W. and AALST 2007)

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:

- Redes de flujo de trabajo (basadas en redes de Petri) (AALST, W.M.P. VAN DER 1996a; 1996b; AALST, WIL M. P. VAN DER *et al.* 2003)
- Cadena de Procesos Guiado por Eventos (EPCs, por sus siglas en inglés) (DONGEN, B.F. VAN and AALST 2004)
- Sistema de Transición (TS, por sus siglas en inglés) (RUBIN 2007)
- Notación de Modelos de Procesos de Negocio (BPMN, por sus siglas en inglés) (OBJECT MANAGEMENT GROUP 2012)
- Diagrama de actividades del Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) (AALST, W.M.P. VAN DER *et al.* 2011a)

El segundo tipo de técnica es la conformidad o chequeo de conformidad (GAMA and CARMONA 2010; MENDLING *et al.* 2007; ROZINAT and AALST 2008). La idea fundamental de estas técnicas es comparar un modelo de procesos existente con un registro de evento del mismo proceso. El chequeo de conformidad es un análisis que permite saber hasta qué punto el registro de evento 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 entre otros.

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 evento 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). Se puede observar en la Figura 2, un enfoque orientado a las entradas y salidas de las técnicas de minería de proceso. Tomado de (AALST, W.M.P. VAN DER *et al.* 2011a).

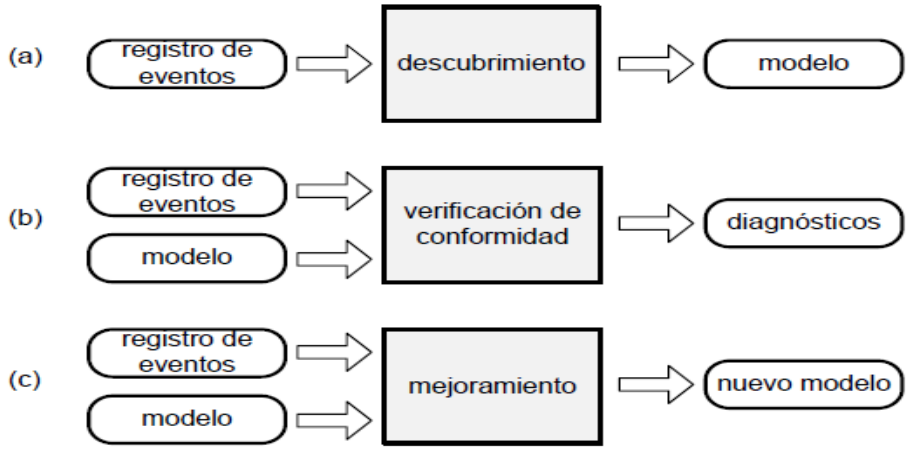


Figura 2 Tipos fundamentales de técnicas de minería de proceso explicada en forma de entradas y salidas: (a) descubrimiento, (b) conformidad y (c) mejora.

Un proceso puede ser analizado considerando las siguientes tres perspectivas o dimensiones, la perspectiva del control de flujo, la de los recursos (también llamada organizacional) y la de casos. Esto ha permitido que las técnicas de minería de proceso jueguen un papel importante en el ciclo de vida de la Gestión de Procesos de Negocio (BPM, por sus siglas en inglés). En la Figura 3, podemos observar las fases del ciclo de vida de BPM.

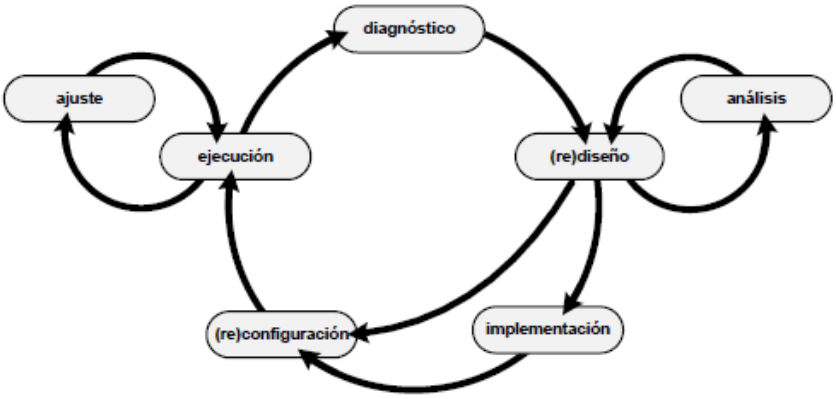


Figura 3 Ciclo de vida de BPM.

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, lo cual 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).

Ausencia de información

Descubrir un modelo de procesos requiere interpretar el comportamiento reflejado en las trazas y mostrar en dicho modelo un grupo de patrones de control de flujo. No todos los algoritmos interpretan de la misma forma las trazas y no manejan todos los patrones de flujo de trabajo (MEDEIROS 2006).

Para que un modelo descubierto a partir del uso de técnica de minería de proceso sea útil es necesario que las trazas analizadas reflejen las instancias del proceso de la

manera más exacta posible, de lo contrario se estaría infiriendo un comportamiento que no es representativo de la realidad.

Para entender en toda su dimensión las manifestaciones de la ausencia de información en las trazas usadas para la minería de proceso se exponen un grupo de situaciones. Es necesario señalar que se abordará la ausencia de información como la ausencia en las trazas de una o varias tareas ejecutadas en las instancias del proceso, debido a que las mismas no pueden ser registradas por los sistemas informáticos usados. A este tipo de tarea se le denomina actividad invisible.

Las actividades o tareas invisibles han sido abordadas en investigaciones anteriores ya sean asociadas al descubrimiento del modelo de proceso (AALST, W.M.P. VAN DER 2011; AALST, W.M.P. VAN DER and WEIJTERS 2004b; MEDEIROS 2006) o asociadas al chequeo de conformidad (ADRIANSYAH *et al.* 2011; GAMA and CARMONA 2010).

Para entender mejor las manifestaciones de la ausencia de información en la minería de proceso se expone un grupo de situaciones que las ejemplifican. Dichas situaciones han sido descritas en su mayoría con anterioridad por autores como Medeiros (MEDEIROS 2006) y permiten apreciar la interpretación que se realiza (por los algoritmos de descubrimiento) de las trazas afectadas por la ausencia de información. Los modelos de procesos que se muestran han sido descritos usando redes de flujo de trabajo (AALST, W.M.P. VAN DER *et al.* 2011b).

Es importante plasmar la definición de subproceso, la cual es usada por las diferentes situaciones de ausencia de información.

Definición 4 (Subproceso): Un subproceso es una agrupación 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. ■ (RAYKENLER YZQUIERDO HERRERA 2012)

Un proceso puede descomponerse en varios subprocesos mediante los patrones de flujo de control 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 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.

Manifestación de la ausencia de información

Situación de salto

Una tarea invisible se puede manifestar cuando se produce un salto de una o varias tareas en una situación de selección. La Figura 4 refleja el comportamiento registrado en la secuencia de tareas ABD, ACD, AD. El recuadro gris representa una actividad invisible.

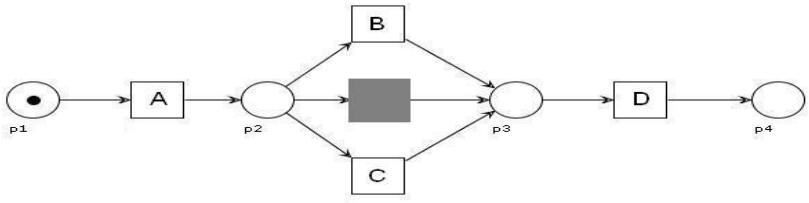


Figura 4 Situación de salto.

Situación de división/unión

Una actividad invisible puede manifestarse en una situación en la que es necesario dividir la ejecución o unirla producto de un punto de selección, después de esta selección aparece una situación de paralelismo entre las tareas. La Figura 5 muestra un ejemplo de esta situación y refleja el comportamiento registrado en la secuencia de actividades ADE, ACBE, ABCE. En este caso, primero se produjo una selección y luego un paralelismo, pero en otros casos se pueden manifestar si el evento de inicio y el de fin de un subproceso no se refleja en las trazas.

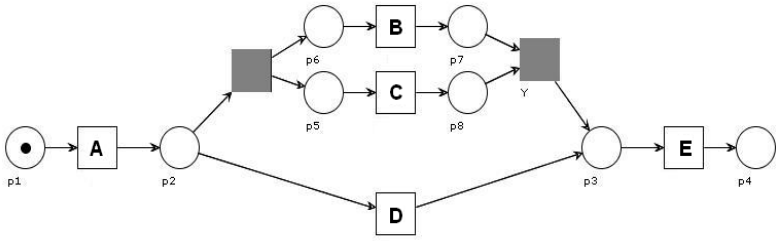


Figura 5 Situación de división/unión.

La capacidad de detección por un algoritmo de descubrimiento de estos patrones está determinada en ocasiones, no solo por el algoritmo de descubrimiento, sino también por las posibilidades que tiene la notación utilizada para la representación del modelo descubierto y específicamente las actividades invisibles. Es necesario señalar que en los algoritmos analizados el tratamiento de las actividades invisibles no se realiza de manera explícita en todos los casos, es decir, al detectar una posible actividad invisible no se adiciona al modelo una nueva actividad. En ocasiones se obtiene un modelo que implícitamente puede reflejar la ausencia de información.

Algoritmos como el Alpha no manejan el constructor de tareas invisibles debido a que asume como precondición que las trazas son completas y están libres de ruido. Esto provoca que de aplicarse dichos algoritmos ignorando las precondiciones mencionadas, el modelo descubierto represente incorrectamente la realidad, aún cuando se cubra completamente el comportamiento descrito en las trazas.

Los siguientes ejemplos ilustran cómo la ambigüedad en las interpretaciones de las trazas puede manifestarse en situaciones en las que existe ausencia de información.

Actividades invisibles contra actividades duplicadas

Una misma tarea puede aparecer más de una vez en una misma traza, la mayoría de los algoritmos desarrollados consideran que cada tarea solo aparece una sola vez en cada traza, por lo cual ignoran las tareas duplicadas o las manejan a partir del constructor de lazos, como se ilustra en la Figura 6 a). Esta situación puede interpretarse también como ausencia de información.

La siguiente secuencia de tareas ABBC descrita en una traza, puede reflejarse por los dos modelos representados en la Figura 6. En el modelo representado en a) se emplea el constructor de actividades duplicadas mientras que en b) se usa el constructor de actividades invisibles.

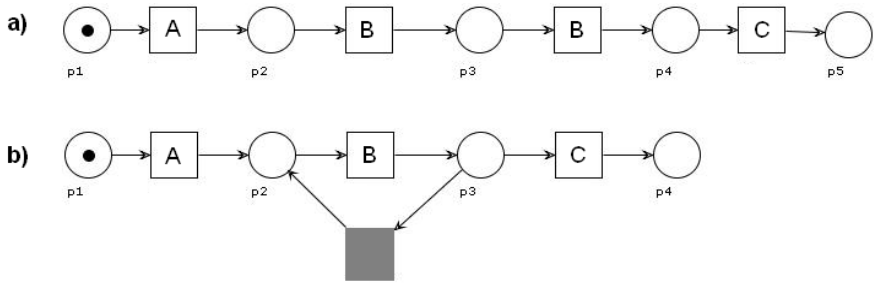


Figura 6 a) Uso del constructor de tareas duplicadas b) Constructor de tareas invisibles.

Actividades invisibles contra lazos

Esta es una situación que se puede considerar como una unión de la situación de actividades invisibles contra actividades duplicadas con la situación de salto.

Las siguientes secuencias de tareas AC, ABC y ABBC pueden reflejarse por los dos modelos representados en la Figura 7 a) y b).

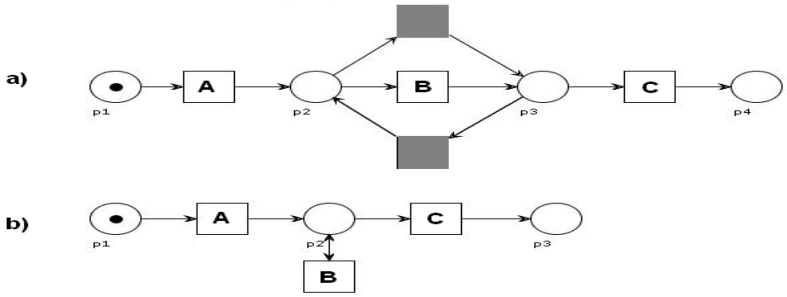


Figura 7 a) Constructor de tareas invisibles b) Constructor de lazos.

Actividades invisibles contra sincronización

Esta situación puede considerarse como una generalización de la situación de división/unión. Los dos modelos de la Figura 8 pueden reflejar las mismas secuencias de tareas (ABDG, ADBG, ACEFG, ACFEG) reflejadas en las trazas. En el modelo representado en a) se emplea el constructor de XOR-división/unión mientras que en b) se usa el constructor de actividades invisibles.

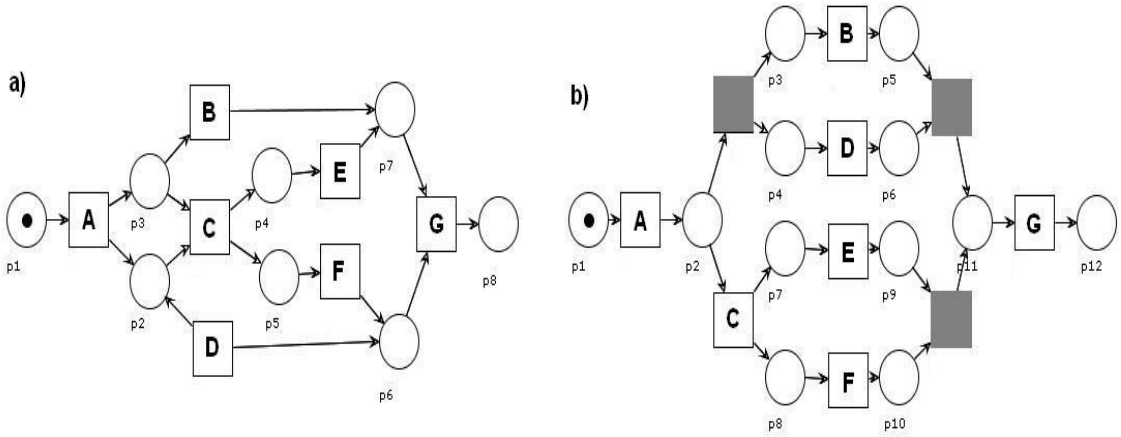


Figura 8 a) Constructor de XOR-división/unión b) Constructor de tareas invisibles.

Lazos contra actividades invisibles junto a actividades duplicadas

Esta situación se puede interpretar como una unión de la situación de salto con la situación de división/unión.

Los dos modelos de la Figura 9 pueden representar las mismas secuencias de tareas (ABCD, ACBD, AD, ABD, ACD) reflejadas en las trazas. En la Figura 9 a) se emplea el constructor de actividades invisibles junto al de actividades duplicadas mientras que en b) se usa el constructor de lazos.

El modelo descubierto usando el constructor de lazos Figura 9 b) es más expresivo que el modelo de la Figura 9 a), es decir, sobrepasa el comportamiento reflejado en la secuencia de actividades expuestas.

Un ejemplo de lo que pudiese denominarse generalización del modelo Figura 9 b) se aprecia si se considera que el mismo puede cubrir la secuencia de actividades ABBBD, secuencia que no es cubierta por el modelo de la Figura 9 a).

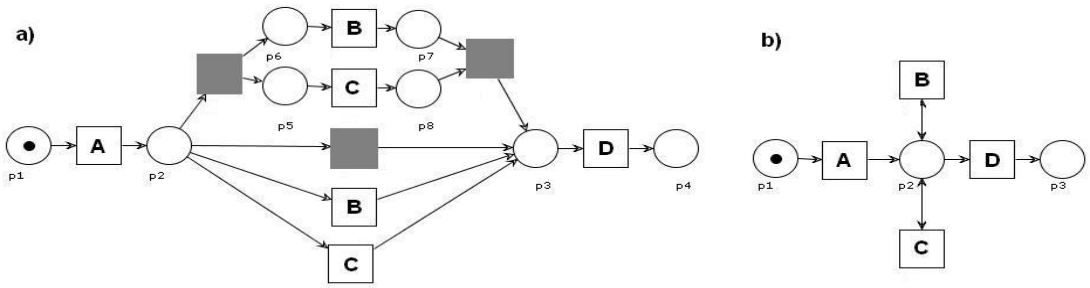


Figura 9 a) Constructor de actividades invisibles unido al de actividades duplicadas b) Constructor de lazos.

Es necesario resaltar que algunos de los algoritmos desarrollados hasta el momento no manejan la totalidad de los constructores a los que se ha hecho referencia.

Hay otras circunstancias en las que pudiese existir ausencia de información, sin embargo, no ser "significativas" al no afectarse la estructura del modelo descubierto. Por ejemplo, cuando en un proceso existe una secuencia de tareas y alguna de ellas no queda reflejada en las trazas. Existen otras circunstancias en las que se evidencia ausencia de información y la afectación en la estructura del modelo descubierto es "significativa", a pesar de ello, estas manifestaciones no han sido tomadas en consideración por los algoritmos desarrollados hasta el momento.

Situación de opciones equiprobables

Considérese un proceso en el que las opciones asociadas a una situación de selección son equiprobables. Un ejemplo de esta situación es el proceso (P0) representado en la Figura 10. Las tareas B y F no se encuentran informatizadas, y por tanto, no parecen en las trazas almacenadas.

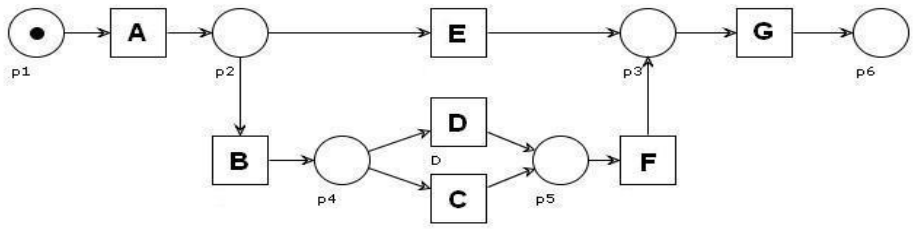


Figura 10 Representación mediante una red de Petri del proceso P0.

Una muestra representativa de las trazas almacenadas se representa en la Tabla 2. La columna correspondiente a la Clase representa los diferentes tipos de clases que se corresponden con cada secuencia de tareas, a iguales secuencias de tareas les corresponde una única clase. El caso 1 y 2 tienen igual secuencia de tareas (AEG) por lo que tienen la misma clase (C1).

Tabla 2 Representación de los casos.

Caso	Secuencia de tareas	Clase
1	AEG	C1
2	AEG	C1
3	ADG	C2
4	ACG	C3

A partir de las trazas reflejadas los algoritmos desarrollados descubren un modelo de procesos (P_1) equivalente al reflejado en la Figura 11, P_1 difiere significativamente de P_0, afectándose por la ausencia de información la estructura del modelo de procesos descubierto.

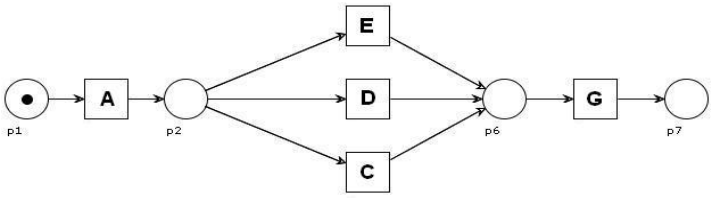


Figura 11 Representación mediante una red de Petri del proceso P1.

Sea $|C_i|$ la cardinalidad (cantidad de casos) de la clase C_i. Si se analiza la cardinalidad de las clases se puede percibir que $|C_1|$ representa el 50 % del total de casos, mientras que $|C_2|$ y $|C_3|$ representan cada una el 25%. Esta distribución indica la ausencia de las actividades B y F.

De esta misma forma pudiese ilustrarse otro grupo de ejemplos en los que bajo las mismas condiciones puede encontrarse evidencia de la ausencia de información.

El análisis desde esta arista es desechado habitualmente, debido a que, por la propia naturaleza de las instancias del proceso no se cumplen las condiciones antes expresadas. Sin embargo, para determinados procesos ante una situación de selección es conocida la frecuencia relativa de ocurrencia de cada una de las opciones y se puede partir del supuesto teórico expresado anteriormente, lo cual, puede ser empleado también para detectar y estimar posible información ausente. El objetivo de este análisis es ilustrar, cómo de suceder una situación como esta, no se toma en cuenta como evidencia de la ausencia de información.

Situación de secuencia oculta de subprocesos

Las secuencias de actividades BCXYDLI, BOXYLDI, BWXYDLI se corresponden de manera correcta con el modelo que se muestra en la Figura 12. En el modelo se pueden identificar dos subprocesos ordenados secuencialmente. El primer subproceso se inicia en la actividad B y termina en la actividad X. El segundo subproceso se inicia en la actividad Y y termina en la actividad I. Si X e Y son actividades invisibles el modelo descubierto utilizando el algoritmo Alpha (AALST, W.M.P. VAN DER *et al.* 2004) quedaría como se muestra en la Figura 13. El modelo refleja incorrectas relaciones de causalidad entre las actividades. El modelo que se muestra en la Figura 13 presenta problemas:

La actividad I no se puede habilitar debido a que requiere dos identificadores de entrada y estos no es posible producirlos simultáneamente.

Se establecen relaciones de causalidad incorrectas entre las actividades W y D, C y D, O y D.

Estos problemas se manifestaron como consecuencia de la ausencia de información. Específicamente las causas están en la ausencia de actividades que permitan definir el fin del subproceso (en este caso X) y el inicio del que le sigue secuencialmente (en este caso Y).

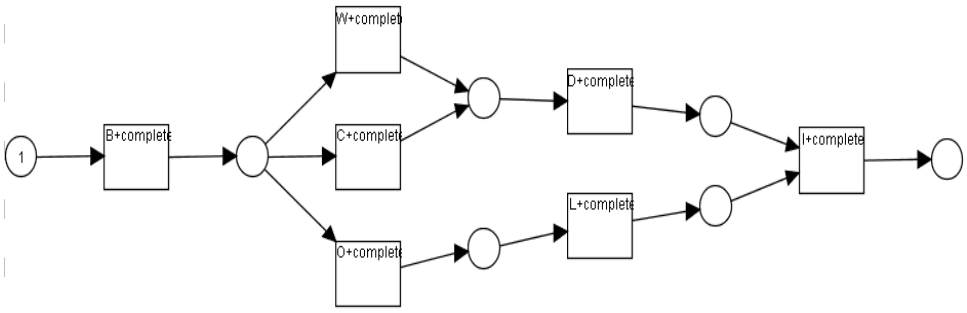


Figura 12 Representación mediante una red de Petri del proceso P2 a.

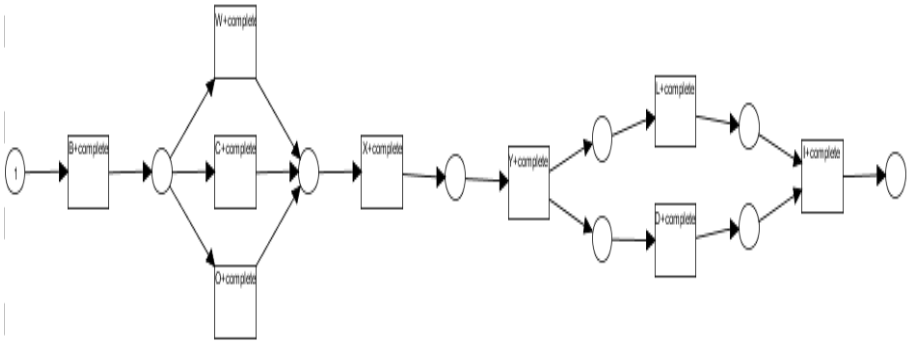


Figura 13 Representación mediante una red de Petri del proceso P2 b.

Diferentes enfoques en el tratamiento de la ausencia de información

En esta sección se hace un análisis del tratamiento de las actividades invisibles desde el enfoque de los principales autores en el área de minería de proceso. Algunas de las investigaciones analizadas se implementaron como algoritmos que forman parte de la herramienta ProM (AALST, W. M. P. V. D. *et al.* 2009a), dichos algoritmos se sometieron a un grupo de pruebas asociadas a cada una de las manifestaciones de ausencia de información antes expuestas.

Es necesario antes de analizar las técnicas de descubrimiento desarrolladas en la minería de proceso, conocer en qué consiste el problema asociado al descubrimiento de un modelo de procesos a partir de un registro de evento. Varios han sido los autores que de una forma u otra han hecho referencia a este problema (BERGENTHUM *et al.* 2007; COOK 1996; COOK *et al.* 2004; MEDEIROS 2006; WEN *et al.* 2006), para su formalización se considera la definición dada por Van der Aalst (AALST, W.M.P. VAN DER 2011), debido a que sintetiza lo abordado en los trabajos relacionados.

Definición 5 (Problema de descubrimiento de proceso): Sea *L* un registro de evento según la especificación del estándar XES. Un algoritmo de descubrimiento de proceso es una función que mapea *L* en un modelo de proceso tal que el modelo es "representativo" del comportamiento reflejado en el registro de evento. El reto es encontrar este tipo de algoritmo. ■

Algunos de los primeros trabajos en la minería de proceso fueron realizados por Jonathan Cook en el área del desarrollo de software. Un análisis de los algoritmos desarrollados por este autor y su equipo de trabajo demuestra que los modelos descubiertos no eran del todo correctos y completos, sin embargo, se lograban identificar los principales patrones de control de flujo presentes en las trazas. Para la representación de los modelos se empleaban Máquinas de Estado Finito.

Cook participó en el desarrollo de tres algoritmos RNet, KTail y Markov, siendo este último el que mejor reflejaba las trazas analizadas. Markov es un algoritmo que tiene una base estadística y permite identificar los predecesores y sucesores de una tarea, para ello se apoya en una tabla de frecuencias en la que se refleja la probabilidad de que un evento ocurriese. Este algoritmo fue extendido posteriormente para manejar procesos concurrentes. Se agregaron cuatro métricas que permiten identificar puntos en los que aparecen relaciones XOR/AND-División/unión. Este algoritmo es robusto ante el ruido debido a su basamento probabilístico (COOK 1996; COOK *et al.* 2004; COOK and WOLF 1998a; 1998b; COOK and WOLF. 1995).

La implementación de los algoritmos desarrollados se realizó en la herramienta DaGama, la cual forma parte del marco de trabajo Balboa.

En los algoritmos analizados no se hace un tratamiento de las tareas invisibles de manera explícita. Las investigaciones realizadas por estos autores permiten manejar las tareas invisibles en los casos de *Situación de salto* y *Situación de división/unión* (MEDEIROS 2006).

Uno de los aspectos más importante de los trabajos de Herbst es que sus algoritmos manejaban las tareas duplicadas. Este autor participó en el desarrollo de tres algoritmos: MergeSeq, SplitSeq y SplitPar. Los algoritmos ejecutan dos pasos, el primero permite capturar las dependencias entre las tareas apoyándose en el uso de la métrica LLH (Log-Likelihood, por sus siglas en inglés) desarrollada.

Dicha métrica indicaba cuán bien el modelo expresaba el comportamiento recogido en las trazas. El segundo paso permite transformar el SAG a ADL (Adonis Definition Language, por sus siglas en inglés).

El ADL es un lenguaje de estructuras en bloques que permite especificar modelos de flujos de trabajo. La conversión de SAG a ADL permite la creación de un modelo bien definido. MergeSeq y SplitSeq son útiles para manejar procesos secuenciales, mientras que SplitPar puede manejar también procesos concurrentes.

La implementación de los algoritmos desarrollados se realizó en la herramienta InWoLve (HERBST 2001; HERBST ; HERBST and KARAGIANNIS 2000; 2004).

Los algoritmos desarrollados pueden comportarse robustamente ante el ruido y manejan los constructores principales, secuencia, selección, paralelismo, lazos y tareas duplicadas, sin embargo, no se hace un tratamiento de las tareas invisibles de manera explícita. Se manejan las tareas invisibles en la Situación de salto y la Situación de selección (MEDEIROS 2006).

Lo más significativo de los trabajos de Schimm es que permiten descubrir un modelo de procesos completo y mínimo. El modelo descubierto no generaliza un comportamiento más allá de lo observado en las trazas. Normalmente se detecta una relación de paralelismo entre dos tareas cuando estas aparecen intercambiadas en el orden de aparición en las trazas, sin embargo, Schimm identifica dos posibilidades para esta situación. Solo si el tiempo de inicio y fin de las tareas involucradas coinciden se identifica una situación de paralelismo.

Schimm analizó el proceso mediante un grupo de reglas de rescritura que aplica sobre el álgebra de flujo de trabajo. Utilizó un modelo de estructura en bloque que permite formar correctos modelos. El autor realizó extensiones al álgebra de flujo de trabajo para poder representar en los modelos una sincronización parcial. Se concibió una fase de pre-procesamiento en la que se detectan aspectos como las tareas duplicadas y se trata el ruido (de estos aspectos no se dan detalles de cómo se solucionan en los trabajos analizados).

El algoritmo consta de seis pasos y puede manejar los principales operadores de flujo de control (AALST, WIL M. P. VAN DER *et al.* 2003; SCHIMM 2000; 2004; 2003; 2002).

La implementación del algoritmo desarrollado se realizó en la herramienta Process Miner (SCHIMM).

No se hace un tratamiento de las tareas invisibles de manera explícita. Se manejan las actividades invisibles solo en la Situación de salto (MEDEIROS 2006).

El aspecto que más resalta en los trabajos de Grecco es que se obtiene como resultado de la minería de proceso un árbol jerárquico que representa los diferentes niveles de abstracción del modelo de procesos. La raíz del árbol representa el modelo más general que encierra los aspectos comunes en las instancias del proceso reflejadas en las trazas. Los nodos que se encuentran entre la raíz y las hojas representan características comunes presentes en los nodos hijos.

El algoritmo se divide en dos pasos, el primero consiste en obtener un modelo que cubra completamente las trazas analizadas, a partir de este modelo base se hace una selección de atributos para agrupar las instancias del proceso y formar particiones. Este proceso se realiza por cada nivel hasta que se cumplen las condiciones de parada. El segundo de los pasos permite construir las abstracciones correspondientes a cada nivel, para ello agrupa en un nodo las tareas comunes en sus hijos y agrupa en subprocesos las tareas particulares. Se emplea una red heurística para representar el modelo de procesos descubierto. El algoritmo no es robusto ante el ruido (GRECO *et al.* 2005; GRECO *et al.* 2004).

El primero de los pasos fue implementado en el complemento Disjunctive Workow Schema Mining (DWS, por sus siglas en inglés) e incorporado a la herramienta ProM (AALST, W. M. P. V. D. *et al.* 2009a).

Las pruebas realizadas tanto a este algoritmo (usando la herramienta ProM) como a los que se mencionan a continuación arrojaron que, no se hace un tratamiento de las tareas invisibles de manera explícita y no se toma en cuenta evidencia que puede indicar una posible ausencia de información para enfrentar situaciones en las que existe ambigüedad en la interpretación de las trazas. Para cada uno de los algoritmos analizados, se muestran en la Tabla 3 al final de la sección los casos en los que el modelo resultante se distanció más de una correcta interpretación. Entiéndase como una correcta interpretación los casos en los que el modelo resultante aún cuando no refleja explícitamente ninguna tarea invisible es posible inferir su ocurrencia.

Para un mejor entendimiento considérese que una correcta interpretación de la situación de tareas invisibles contra lazos sería el modelo representado en Figura 7 y así respectivamente sucede con cada uno de los casos analizados.

Wil van der Aalst y su equipo de trabajo fueron los desarrolladores del conocido algoritmo Alpha. El principal aporte fue que garantizaron una clase de modelo que permitía de manera efectiva el trabajo en el área. Esa clase de modelo se formalizó como Redes de Flujo de Trabajo Estructurada (SWF-nets, por sus siglas en inglés).

El algoritmo Alpha se basa en cuatro tipos de relaciones binarias: *secuencial*, *causal*, *paralelismo* y *sin relación*.

La relación de secuencia es la básica a partir de la cual se infieren las demás. Existe una secuencia cuando al menos en una instancia del proceso la actividad A es seguida directamente por B. La relación causal surge cuando A es seguida directamente por B y B no antecede a A directamente.

Si A es seguida por B y B es seguida por A entonces se establece una relación de paralelismo. Si A y B no están envueltos en una relación de secuencia entonces la relación es: sin relación.

El algoritmo Alpha no maneja actividades duplicadas y extensiones realizadas al mismo permitieron manejar correctamente lazos cortos, considerar tareas no atómicas, y de manera general mejoraron las relaciones binarias y las nociones de completitud de las trazas. Hay que resaltar que el algoritmo mina las trazas a partir de la concepción de que las trazas son completas y libres de ruido (AALST, W.M.P. VAN DER 1996a; 1996b; AALST, W.M.P. VAN DER and HEE 2004; AALST, WIL M. P. VAN DER *et al.* 2003; AALST, W.M.P. VAN DER *et al.* 2004).

Van der Aalst ha trabajado también en otras investigaciones referenciadas en este trabajo. Entre sus últimas investigaciones se debe resaltar el trabajo (AALST, W.M.P. VAN DER *et al.* 2009b). En este trabajo se propone un algoritmo para el descubrimiento de un modelo de proceso que presente un balance adecuado entre generalización y especificación. Es posible cubrir algunos aspectos relacionados con la completitud de los casos pero no es posible resolver todos los problemas asociados con la ausencia de información. Se manejan las actividades invisibles en la *Situación de salto* y la *Situación de división/unión*.

Las investigaciones de Ton Weijters y su equipo de trabajo pueden verse como una extensión del algoritmo Alpha. Para establecer las relaciones de secuencia se emplea

la frecuencia de aparición de dichas relaciones en las trazas analizadas. Así se favorecen las relaciones entre las actividades que se manifiestan con mayor frecuencia y se desechan las menos frecuentes. Para identificar los patrones de control de flujo XOR y AND se realiza una reproducción de las trazas sobre el modelo descubierto.

El modelo es representado mediante la notación Redes Causales (C-Net, por sus siglas en inglés). La implementación de dicho algoritmo se realizó en la herramienta Little Thumb y también en ProM como el complemento Heuristics Miner. El algoritmo es robusto ante el ruido (AALST, W.M.P. VAN DER and WEIJTERS 2005; 2004a; WEIJTERS and AALST 2003).

Boudewijn van Dongen introduce como resultado de la minería de las trazas el modelo EPCs. El algoritmo desarrollado se estructura en dos pasos y en el primero de estos se obtiene un modelo de procesos por cada una de las trazas analizadas. En este paso se asegura que cada actividad aparezca solo una vez en la traza, a cada instancia de una tarea en la traza se le asigna un identificador único.

Ya en el segundo paso se mezclan los modelos obtenidos en el paso anterior y se definen relaciones de división y unión en sus diferentes variantes: AND, OR y XOR (DONGEN, B.F. VAN and AALST 2005; 2004; DONGEN, B.F. VAN *et al.* 2005). El algoritmo no es robusto ante el ruido.

Los trabajos de Lijie Wen estuvieron dirigidos a la extensión del algoritmo Alpha. Las primeras mejoras se realizaron en el algoritmo Beta, el cual fue implementado como el complemento Tsinghuaalpha en la herramienta ProM. Lo distinguía el tratamiento de las actividades no atómicas y el uso del tiempo de ejecución de las tareas para determinar las relaciones de paralelismo y los lazos cortos. Una posterior mejora se implementó como el algoritmo Alpha++, este incluía otras mejoras como el tratamiento del constructor de selección-no-libre. El algoritmo no es robusto ante el ruido (WEN *et al.* 2004; WEN *et al.* 2006).

Los principales aportes de Ana Karla Alves de Medeiros y su equipo de trabajo estuvieron en la aplicación de algoritmos genéticos en el área de la minería de proceso. El resultado de la aplicación de este tipo de técnica garantizó el manejo de todos los constructores estructurales en el caso del algoritmo GA, exceptuando las tareas duplicadas. El algoritmo DGA es una extensión de GA que permite manejar las tareas duplicadas. El principal inconveniente de estos algoritmos es el tiempo de ejecución. Los algoritmos son robustos ante el ruido (MEDEIROS 2006).

Las pruebas realizadas utilizando la implementación realizada sobre ProM arrojaron en cada uno de los ejemplos antes expuestos que el modelo obtenido puede interpretarse satisfactoriamente.

Se puede inferir que existe ausencia de información, aunque esto no se hace en ningún caso de manera explícita aún cuando existe evidencia que la denota.

Las investigaciones de Bergenthum y su equipo de trabajo se centraron en el uso de métodos de síntesis basados en región para construir redes de Petri. Para ello se consideran las trazas como un lenguaje finito y el modelo obtenido es una red que recoge el comportamiento del lenguaje dado.

Los métodos existentes para la síntesis basada en región se adaptaron para satisfacer las necesidades del área de la minería de proceso.

El algoritmo trata de evitar el comportamiento adicional, es decir, trata de reflejar de la manera más precisa el comportamiento reflejado en las trazas. En consecuencia el algoritmo no es robusto ante el ruido (BERGENTHUM *et al.* 2007).

Rubin es otro de los autores que aplica técnicas de minería de proceso en el área de desarrollo de software. La fuente de información en este caso no fueron las trazas registradas por los sistemas, sino los documentos e información de manera general almacenados durante el proceso de desarrollo de software en los repositorios de información. Los autores introducen como resultado de la minería de las trazas el modelo TS.

El algoritmo desarrollado se estructura en dos pasos. En el primer paso se genera el modelo en TS. Para obtener el modelo se transita por una fase de pre-procesamiento en la que se estructuran las trazas necesarias para el proceso de minería. Posteriormente se definen los posibles estados y transiciones derivados de los eventos almacenados en las trazas y se construye el modelo. Dicho modelo se le aplican un grupo de estrategias de modificación que permiten obtener un modelo que cubre el comportamiento expresado en las trazas y elimina los lazos. Un segundo paso permite generar una red de Petri a partir del modelo en TS ya descubierto, este paso se basa en la teoría de la región. El algoritmo es capaz de manejar diferentes niveles de abstracción lo cual facilita la adaptabilidad del mismo. El algoritmo no es robusto ante el ruido (RUBIN 2007).

Las investigaciones de Christian Günther y su equipo de trabajo se dirigieron al minado de trazas que refleja el comportamiento de procesos poco estructurados.

En consideración refieren que se deben resaltar en estos casos los aspectos importantes del comportamiento analizado y ocultar lo que puede no ser significativo. Para ello se desarrollaron dos métricas que guían el proceso de descubrimiento del modelo de procesos.

Esta especifica el nivel de interés que se puede tener sobre un evento o la ocurrencia de este después de otro. Un aspecto a considerar puede ser la frecuencia de aparición de un evento, mientras mayor es la frecuencia mayor es la significación del mismo.

La otra métrica es la correlación, determinada por la medida en la que una relación de precedencia entre los eventos puede ser relevante. Permite tener una medida de cuán estrechamente relacionados están dos eventos.

Basado en estas dos métricas el algoritmo estructura el modelo de procesos. El algoritmo es robusto ante el ruido (AALST, W. M. P. V. D. and GÜNTHER 2007).

El algoritmo fue implementado como el complemento Fuzzy Miner e incorporado a la herramienta ProM.

Tabla 3 Resultados de los algoritmos analizados.

Algoritmos	Situación de salto	Situación de división/unión	Actividades invisibles contra actividades duplicadas	Actividades invisibles contra lazos	Actividades invisibles contra sincronización	Lazos contra actividades invisibles junto a actividades duplicadas	Secuencia oculta de subprocesos	Opciones equiprobables
Heuristics miner (Weijters)		*		*	*	*	*	*
Multi-phase (Dongen)					*	*	*	*
Alpha (Van der Aalst)	*	*	*		*	*	*	*
Alpha ++ (Wen)	*	*			*	*	*	*
DWS mining (Grecco)		*			*	*	*	*

Genetic algorithm (Medeiros)							*	*
Transition System (Rubin)			*	*			*	*
Region Miner (Bergenthum)			*	*	*	*	*	*
FuzzyMiner (Günther)						*	*	*

Definiciones preliminares

El algoritmo propuesto está basado en la alineación de las trazas realizado por Bose y Van der Aalst y teniendo como premisa un conjunto de definiciones, las cuales se encuentran detalladas en el trabajo (RAYKENLER YZQUIERDO HERRERA 2012).

Definición 6 (Alineación de trazas): La alineación de las trazas sobre un conjunto de trazas $\mathbb{T}= \{T_1, T_2, \ldots, T_n\}$ se define como el mapeo del conjunto de trazas de $\mathbb{T}$ sobre otro conjunto de trazas $\overline{\mathbb{T}}= \{\overline{T}_1, \overline{T}_2, \ldots, \overline{T}_n\}$ donde cada $\overline{T}_i \in (\Sigma \cup \{-\})^+$ para $1 \leq i \leq n$ y existe un $m \in \mathbb{N}$ tal que $|\overline{T}_1| = |\overline{T}_2| = \ldots = |\overline{T}_n| = m$, $\overline{T}_i$ es igual a T_i después de eliminar todos los símbolos "–", No existe una $k \in \{1, \ldots, m\}$ tal que $\forall_{1 \leq i \leq n}, \overline{T}_i(k) = -$.

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 $\Sigma' = \Sigma \cup \{-\}$ 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 $l_{max} \leq m \leq l_{sum}$ donde l_{max} es la máxima longitud de una traza contenida en $\mathbb{T}$ y l_{sum} es la suma de las longitudes de todas las trazas contenidas en $\mathbb{T}$. ■.

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

Definición 7 (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 $Q_{\mathcal{A}}$ el conjunto de todas las sub-matrices de $\mathcal{A}$. ■

Se denota por $Q'_{\mathcal{A}}$ el conjunto de sub-matrices que representan a los subprocesos de S, tal que $Q'_{\mathcal{A}} \subseteq Q_{\mathcal{A}}$. Sean $C^{j}_{\mathcal{A}}$ y $C^{j+1}_{\mathcal{A}}$ sub-matrices de $Q'_{\mathcal{A}}$, la relación de secuencia entre dos subprocesos representados por $C^{j}_{\mathcal{A}}$ y $C^{j+1}_{\mathcal{A}}$ se denota por $C^{j}_{\mathcal{A}} >'_{\mathcal{L}} C^{j+1}_{\mathcal{A}}$.

De forma análoga se denota la relación de selección (XOR específicamente) por $C^{j}_{\mathcal{A}} \#'_{\mathcal{L}} C^{j+1}_{\mathcal{A}}$ y la relación de paralelismo por $C^{j}_{\mathcal{A}} ||'_{\mathcal{L}} C^{j+1}_{\mathcal{A}}$.

En el caso de manifestarse un lazo la descomposición de $C^{i}_{\mathcal{A}}$ se realiza en un único sub-proceso que se repite múltiples veces y se denota por $(C^{j}_{\mathcal{A}})^*$.

Sea $s_i \in S$ un subproceso representado por la matriz $C^{i}_{\mathcal{A}} \in Q'_{\mathcal{A}}$ y que está compuesto por una secuencia de subprocesos representados por $C^{j}_{\mathcal{A}}$, …, $C^{j+k}_{\mathcal{A}}$, entonces tanto la matriz $C^{i}_{\mathcal{A}}$ como el conjunto $\{C^{j}_{\mathcal{A}}, \ldots, C^{j+k}_{\mathcal{A}}\}$ se le denominan *bloques de construcción* y los sub-procesos representados por $\{C^{j}_{\mathcal{A}}, \ldots, C^{j+k}_{\mathcal{A}}\}$ se relacionan de una única forma (secuencia, paralelismo, XOR o lazo).■ (RAYKENLER YZQUIERDO HERRERA 2012)

La propuesta tiene como objetivo construir un árbol de bloques de construcción que representa la descomposición del proceso analizado. La Figura 14 muestra un ejemplo de un árbol de bloques de construcción. Los bloques de construcción $C^{2}_{\mathcal{A}}$ y $C^{3}_{\mathcal{A}}$ representan subprocesos ordenados secuencialmente, $C^{4}_{\mathcal{A}}$ y $C^{5}_{\mathcal{A}}$ representan subprocesos ordenados como opciones de una selección. $C^{6}_{\mathcal{A}}$ y $C^{7}_{\mathcal{A}}$ representan subprocesos en paralelo.

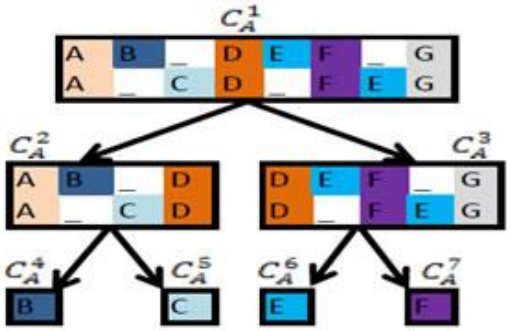


Figura 14 Árbol de bloques de construcción

Métricas utilizadas

Para la evaluación de las afectaciones que causa la ausencia de información en los modelos de proceso descubiertos, se han reportado un conjunto de métricas que son útiles para medir su impacto. La ausencia de información en el registro de evento provoca que en el modelo descubierto se establezcan incorrectas relaciones entre las actividades incorporadas al mismo. Por lo cual es necesario medir el grado en el que el modelo descubierto representa el comportamiento observado en el registro de evento. Este tipo de verificaciones se enmarcan en el área de chequeo de conformidad (AALST, W.M.P. VAN DER 2011; ADRIANSYAH *et al.* 2011) (ROZINAT and AALST 2008), para la cual han sido desarrolladas un conjunto de métricas.

Las métricas se han agrupado considerando varias dimensiones (ARENDONK 2011; ROZINAT et al. 2007; WEERDT et al. 2010), por ejemplo:

- Fitness
- Precisión
- Generalidad
- Estructura

La mayoría de las métricas desarrolladas consideran que el modelo de proceso está representado por una red de Petri, aún cuando la métrica puede generalizarse para notaciones con semejante expresividad. En este trabajo las métricas seleccionadas utilizan modelos representados por redes de Petri en todos los casos.

Para medir las afectaciones que provoca la ausencia de información sobre la estructura del modelo descubierto se emplean dos métricas asociadas al Fitness, específicamente Fitness Unsatisfied y Fitness Unhandled (ARENDONK 2011).

Ambas métricas permiten medir la afectación que provocan las incorrectas relaciones que se establecen en el modelo descubierto a partir del registro de evento con ausencia de información. Es necesario señalar que en la mayoría de los trabajos relacionados con el chequeo de conformidad se hace referencia a la métrica Fitness

que incorpora las dos antes mencionadas. Sin embargo, en este trabajo se utiliza de manera fragmentada para ganar en precisión en la medición.

Fitness Unsatisfied: Indica cuán bien un modelo se corresponde con un registro de evento, penalizándose el comportamiento que no se refleja en el modelo y que aparece en el registro de evento.

Fitness Unhandled: Se penaliza en correspondencia con la cantidad de identificadores (considerando que el modelo está representado por una red de Petri) que quedan a la izquierda del modelo al completar la reproducción de una traza. Indicando esto el comportamiento que no se pudo manejar en el modelo.

Para medir las afectaciones que provoca la ausencia de información sobre la comprensión del modelo descubierto se emplean de manera tradicional la métrica asociada al Fitness y se propone además dos métricas asociadas a la medición de la Precisión, específicamente: Precision y Non Fit Traces (por sus respectivos nombres en inglés) (MUÑOZ-GAMA and CARMONA 2010). La propuesta de medir la precisión del modelo se realiza considerando que un modelo con un alto grado de precisión facilita la comprensión del modelo analizado.

Precision: permite evaluar la precisión de un modelo en correspondencia con el comportamiento observado en un registro de evento. Penaliza el comportamiento extra reflejado en el modelo. En el presente trabajo se utilizó el término ETCPrecision para evitar confusión.

Non Fit Traces: Cuantifica la trazas que no pudieron ser representadas correctamente por el modelo.

Se verifica también que en el modelo descubierto no existan problemas asociados a la dimensión de estructura. En este sentido se penaliza, mediante la métrica Improved Structural Appropriateness (ROZINAT and AALST 2005), que el modelo no presente actividades invisibles redundantes y actividades duplicadas alternativamente.

Esta evaluación contribuye a la medición de las afectaciones estructurales y de compresión. Cada una de las métricas propuestas para este trabajo se seleccionaron considerando recientes estudios realizados (ARENDONK 2011; MUÑOZ-GAMA and CARMONA 2010).

Conclusiones del capítulo

La mayoría de los algoritmos analizados parten del principio que las trazas son completas y están libres de ruido. Ante situaciones de ausencia de información en las trazas el modelo descubierto no siempre puede interpretarse correctamente, debido a que los algoritmos analizados no reflejan las tareas invisibles de manera explícita.

Durante las pruebas realizadas a los algoritmos implementados en la herramienta ProM se evidenció que ante situaciones similares de ausencia de información los modelos descubiertos por los algoritmos eran diferentes.

Esta ambigüedad en la interpretación de las trazas está determinada por los constructores de flujo de trabajo que puede manejar cada algoritmo y la forma en que lo hace.

Se hace necesario que como parte de la etapa de pre-procesamiento de las trazas, la cual actualmente se le ha dado notable importancia, se considere la estimación de información ausente en las trazas para así garantizar la obtención de modelos mucho más ajustados al comportamiento real del proceso.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

En el presente capítulo se presenta un algoritmo para la estimación de información ausente en las trazas usadas en la minería de proceso.

El algoritmo que se propone tiene como objetivo la estimación de información ausente en las trazas usadas en la minería de proceso por lo cual es necesario definir qué se entiende por *Estimación de información ausente*.

Definición 8 (Estimación de información ausente): La estimación de información ausente es es una función que transforma de trazas $\mathbb{T} = \{T_1, T_2, \ldots, T_n\}$ en un conjunto de trazas $\check{\mathbf{T}} = \{\check{T}_1, \check{T}_2, \ldots, \check{T}_n\}$ donde, $\check{T}_i \in (\Sigma \cup \Lambda)^+$ para $1 \leq i \leq n$ y $\Lambda = \{\emptyset, \lambda_1, \lambda_2, \ldots, \lambda_w\}$ es el conjunto de actividades invisibles estimadas. $|T_i| \leq |\check{T}_i|$ ■ (RAYKENLER YZQUIERDO HERRERA 2012)

Constructores de información estimada

Se ha definido un grupo de operadores que representan cada una de las situaciones identificadas de ausencia de información.

Estos son:

- Operador de salto
- Operador de lazo
- Operador de división/unión
- Operador de secuencia oculta
- Operador probabilístico.

En dependencia de las características del bloque de construcción puede o no aplicarse la totalidad de los operadores propuestos. Cada uno de los operadores se aborda en detalles en la siguiente sección.

Cada operador se aplica sobre un bloque de construcción que puede o no contener actividades invisibles estimadas por operadores antes empleados. El operador aplicado modifica el bloque de construcción solo si detecta alguna actividad invisible.

Para analizar cada bloque de construcción se recorre el árbol de bloques de construcción E in-orden, al visitar un nodo del árbol (contiene un bloque de construcción) se aplican los operadores seleccionados. Para reflejar la información estimada se crea un nuevo árbol de bloques de construcción estimados Ĕ.

A continuación se describen cada uno de los operadores de estimación de información ausente propuestos.

Operador de salto

Este operador se aplica para identificar la *Situación de salto* antes descrita y estimar la información ausente. El objetivo de este operador es detectar cuando un bloque que es producto de una selección posee en su matriz una sola columna y esta solo contiene Graps ("–") o caracteres de relleno creado en el proceso de alineación de las trazas, en caso de cumplirse estas condiciones se crea una actividad invisible y se adiciona en la primera columna de la matriz del bloque de construcción. Los pasos que se siguen están descritos en el Algoritmo 1.

Algoritmo 1 Operador de salto

Entrada: Bloque de construcción: *Ce*
Salida: Bloque de construcción

1: **Si** Ce es resultado de una descomposición según la relación OR o XOR **Entonces**
2: **Si** Ce contiene una sola fila y esta solo tiene símbolos vacíos ("–") **Entonces**
3: Se genera una actividad invisible λ*i* y se adiciona a la primera columna de la matriz contenida en Ce. *i* se calcula en función de la cantidad de actividades invisibles estimadas para el proceso analizado.
4: Se devuelve Ce modificado
FinSi
FinSi

La complejidad temporal del algoritmo *Operador de salto* es O(1).

Operador de lazo

Este operador se aplica para identificar situaciones en las que existen tareas duplicadas, específicamente formando un lazo de longitud uno y estimar la información ausente. El objetivo del operador lazo es detectar cuando un bloque es producto de un lazo y además posee en su matriz una sola fila y columna, en caso de cumplirse estas condiciones se crea una actividad invisible y se adiciona al bloque de construcción.

Los pasos que se siguen están descritos en el Algoritmo 2.

Algoritmo 2 Operador de lazo

Entrada: Bloque de construcción: *Ce*
Salida: Bloque de construcción

1: **Si** *Ce* es resultado de una descomposición según la relación Lazo **Entonces**
2: **Si** *Ce* contiene una sola actividad **Entonces**
3: Se genera una actividad invisible λ*i* y se adiciona en una nueva columna que se coloca detrás de la actividad existente en *Ce*.
4: Se devuelve *Ce* modificado
FinSi
FinSi

La complejidad temporal del algoritmo *Operador de salto* es O(1).

Operador de división/unión

Este operador se aplica para identificar la *Situación de división/unión* antes descrita y estimar la información ausente. El objetivo del operador división/unión es determinar si el bloque analizado es producto de una selección, un paralelismo o un XOR, en caso de tener un hijo este tiene que cumplir con la condición anterior, además de determinar si la columna inicial de la matriz del bloque analizado es completa, una columna se considera completa cuando todos los caracteres de la misma son iguales y ocupan todas las posiciones de la misma excepto por los caracteres de relleno ("–") o Grap introducido en el proceso de alineación de las trazas, la misma comprobación para la última columna de la matriz, en caso de no ser completa se crea una nueva actividad invisible y se adiciona al inicio o fin de la matriz o ambos.

Los pasos que se siguen están descritos en el Algoritmo 3.

Algoritmo 3 Operador de división/unión

Entrada: Bloque de construcción: *Ce*,
Lista de bloques de construcción obtenidos al descomponer a *Ce*: *Lista*
Salida: Bloque de construcción

1: **Si** *Ce* es resultado de una descomposición según la relación OR, XOR o paralelismo **Entonces**
2: **Si** |*Lista*| > 0 y *Lista*[1] se descompuso según la relación OR, XOR o paralelismo **Entonces**
3: **Si** la primera columna de *Ce* contiene símbolos vacíos **Entonces**
4: Se inserta una nueva columna en la primera posición de *Ce* y en cada fila de la columna insertada se pone la misma actividad invisible λi.
FinSi
5: **Si** la última columna de *Ce* contiene símbolos vacíos **Entonces**
6: Se inserta una nueva columna en la última posición de *Ce* y en cada fila de la columna insertada se pone la misma actividad invisible λj, tal que $j \neq i$
FinSi
7: Se devuelve *Ce* modificado
FinSi
FinSi

La complejidad temporal del algoritmo *Operador de lazo* es O(*n*).

Operador de secuencia oculta

Este operador se aplica para identificar la *Situación de secuencia oculta* antes descrita y estimar la información ausente. El objetivo del operador secuencia oculta es detectar cuando un bloque tiene hijos y este es producto de una secuencia oculta, para cada

hijo contenido en el árbol, si es el primero se genera una actividad invisible y se adiciona a la matriz del mismo en la primera columna, en caso de ser el último hijo, se aplica el mismo procedimiento pero la actividad se inserta en la última columna de la matriz. En caso de ser otro hijo se crea una segunda actividad invisible y se adicionan tanto la primera y la última en las columnas primera y última respectivamente, posteriormente se le asigna a la primera actividad invisible el valor de la segunda y se repite el proceso para cada hijo.

Los pasos que se siguen están descritos en el Algoritmo 4.

Algoritmo 4 Operador de secuencia oculta

Entrada: Bloque de construcción: *Ce*,
Lista de bloques de construcción obtenidos al descomponer a *Ce*: *Lista*
Salida: Lista de bloques de construcción

1: **Si** |*Lista*| > 0 y *Lista*[1] se descompuso según la relación *Secuencia oculta* **Entonces**
2: Se crean *k* actividades invisibles.
k = (*cantidad bloques de construcción hijos Ce*) -1
3: **Para** *i* = 1 **Hasta** *k* **Con Paso** 1 **Hacer**
4: Se inserta una nueva columna en la última posición del bloque de construcción *Lista*[*i*] y en cada fila de la columna insertada se pone la misma actividad invisible λ*i*
5: Se inserta una nueva columna en la primera posición del bloque de construcción *Lista*[*i*+1] y en cada fila de la columna insertada se pone la misma actividad invisible λ*i*
FinPara
6: Se devuelve *Lista* modificada
FinSi

La complejidad temporal del algoritmo Operador de secuencia oculta es O(k*n). Donde k = (cantidad bloques de construcción hijos Ce) -1.

Operador probabilístico

Este operador se aplica para identificar la situación de *Opciones equiprobables* antes descrita y estimar la información ausente.

Los pasos que se siguen están descritos en el Algoritmo 5.

Algoritmo 5 Operador probabilístico

Entrada: Bloque de construcción: *Ce*,
Lista de bloques de construcción obtenidos al descomponer a *Ce*: *Lista*,
Árbol de bloques de construcción *E*
Salida: Nuevo árbol de bloques de construcción Ĕ

1: **Si** |*Lista*| > 0 y *Lista*[1] se descompuso según la relación OR o XOR **Entonces**
2: **Para** cada bloque de construcción contenido en *Lista* **Hacer**
3: Se calcula la cantidad τ de trazas que forman el bloque de construcción *Ce* según la Ecuación 1

$\tau = \sum_{j=1}^{n} \gamma_j$, γ es la multiplicidad de la traza j y n la cantidad de filas del bloque de construcción analizado **(1)**

4: Se calcula la frecuencia relativa *fr* asociada al bloque de construcción según la Ecuación 2

$fr = \frac{\tau}{\sum_{i=1}^{|Lista|} \tau_i}$, donde 0 > *fr* < 1 **(2)**

FinPara
5: Se almacenan en *Lf* ordenadas de mayor a menor, todas las frecuencias relativas calculadas.
6: Se crea una lista que se denominará *Menores* en la que se guarda el árbol que representa el caso base (caso en el que no se consideran actividades invisibles) y el correspondiente error. El error asociado a un árbol de manera general se calcula según la Ecuación 3

$E = \frac{\sum_{i=1}^{|Lista|} e_i}{|Lista|}$, e_i es el error asociado a cada opción de selección ubicada en un nodo. $e_i = |PN_j - fr_i|$, donde PN_j es la probabilidad de cada nodo hijo del nodo *j* y se calcula según las Ecuación 4 **(3)**

$$PN_i = \frac{\textit{Probabilidad de ocurrencia del nodo padre (j)}}{\textit{cantidad de hijos del nodo j}} \quad \textbf{(4)}$$

7: **Para** *i* = 2 **Hasta** |*Lista*| **Con Paso** 1 **Hacer**

8: Comb = *SubArboles(Lf*, *i*, *probpadre*, *Ep*, *ε). Siendo probpadre* la probabilidad de ocurrencia del nodo padre y se inicializa en 1, *Ep* el error parcial que se va incrementando en la medida que se determina e_r para cada opción ubicada como nodo hoja en el árbol y se inicializa en 0, *ε es el e*rror asociado a *Menores*[1].

El procedimiento *SubArboles* permite determinar los subárboles en los cuales se consideran como nodos intermedios bloques de construcción que contienen actividades invisibles y como nodos hojas se tienen los bloques de construcción contenidos en *Lista*. La variable *i* condiciona la cantidad de subárboles que se generan. En la medida en que a cada árbol se le adiciona un nodo hoja se incrementa el *Ep* y se verifica que este no rebase el valor de *ε*.

9: **Si** |*Comb*| > 0 **Entonces**.

10: Se adiciona a *Menores* el resultado de concatenar la raíz con las combinaciones contenidas en *Comb*.

FinSi

FinPara

11: Se determina la mejor solución considerando el menor *Ep* acumulado en cada elemento de *Menores*

12: **Si** la mejor solución es diferente de *Menores*[1] **Entonces**

13: Se crea un nuevo árbol Ĕ según la estructura de la mejor solución encontrada, este nuevo árbol contiene la información estimada.

FinSi

14: Se devuelve Ĕ

FinSi

La complejidad temporal del algoritmo Operador probabilístico es $O(2^{w(w-1)/2})$, donde w es la cantidad de opciones en una selección. Hay que señalar que la cantidad de opciones de una selección en entornos reales difícilmente alcanzará valores superiores a diez, por lo cual el algoritmo propuesto es factible. Además, para evitar que se generen la totalidad de las soluciones se utiliza *Ep*, lo que permite desechar en cada momento las soluciones parciales que presentan un error superior a ε.

Propagación de la información estimada

A partir del árbol de bloques de construcción estimado Ĕ se propaga la información estimada en cada nodo hijo. Cada bloque de contrucción contenido en el árbol Ĕ puede tener actividades invisibles originadas por la aplicación de los operadores antes expuestos. La propagación de las actividades invisibles se realiza desde las hojas del árbol hasta la raíz, considerando en cada caso la posición relativa de cada actividad en el bloque de construcción padre y tipo de descomposición que dio lugar al bloque de construcción analizado.

Es necesario reflejar en el bloque de construcción contenido en la raíz del árbol todas las actividades invisibles generadas en los bloques de construcción contenidos en los nodos restantes de Ĕ. Finalmente se corrigen las actividades invisibles se han propagado de manera incorrecta.

Esta corrección está orientada a lograr que las actividades invisibles ocupen la menor cantidad de columnas posibles en el bloque de construcción contenido en la raíz del árbol Ĕ.

La complejidad temporal del algoritmo implementado para la propagación de la información estimada es $O(n*m*(m+n)^2)$.

Construcción del registro de evento con información estimada

Se construye el registro de evento a partir del bloque de construcción contenido en la raíz del árbol Ĕ. Se eliminan del bloque de construcción contenido en la raíz del árbol Ĕ todos los símbolos vacíos (“–”) y sin considerar las actividades invisibles insertadas se buscan las coincidencias de cada caso contenido en el bloque de construcción (Cada caso ocupa una fila de la matriz que constituye el bloque de construcción analizado) con los casos contenidos en el registro de evento original. Según las coincidencias se insertan las actividades invisibles en el registro de evento. El registro de evento modificado constituye la salida del modelo propuesto.

La complejidad temporal del algoritmo para la construcción del registro de evento a partir del bloque de construcción contenido en la raíz del árbol Ĕ es $O(k*(n*\log m + m))$, donde k es la cantidad de casos en el registro de evento original y $O(n*\log m)$ es la complejidad temporal de la búsqueda de un caso sobre una estructura de datos balanceada.

Conclusiones del capítulo

En este capítulo se detallaron cada uno de los operadores que componen la propuesta de solución, representados en forma de seudocódigo para facilitar la implementación de los mismos, así como el cálculo de la complejidad temporal de cada algoritmo, permitiendo saber de antemano los posibles tiempos de respuesta y su factibilidad. Además de brindar una explicación detallada de cada una de las partes que componen el algoritmo, como parte de un flujo de ejecución. Mediante las descripciones de los operadores se realizó la implementación de los mismos en la aplicación desarrollada para la validación de la propuesta de solución. Dando así cumplimiento a los objetivos específicos planteados.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN

En este capítulo se presenta la aplicación desarrollada a partir del algoritmo propuesto. Se realizan pruebas considerando cuatro procesos diferentes y se utiliza la aplicación desarrollada para estimar la información ausente en cada uno de los procesos de prueba.

Aplicación informática para la estimación de información ausente.

Se desarrolló una aplicación informática que permite realizar la estimación de información ausente a partir de las trazas usadas en la minería de proceso. La herramienta posibilita el uso de un registro de evento en el formato XES o MXML como entrada. Como salida se genera un fichero con la información estimada y en formato XES o MXML según corresponda. El desarrollo se realizó utilizando Java como lenguaje de codificación.

La aplicación desarrollada posibilita transitar por tres momentos importantes en la estimación de información ausente.

- En un primer momento la aplicación permite leer un registro de evento y configurar los parámetros necesarios para la alineación de las trazas.
- Posterior a la alineación de las trazas se realiza la descomposición del proceso.
- En un último momento se realiza la estimación de información ausente a partir del árbol de bloques de construcción obtenido anteriormente.

La Figura 15 muestra la estimación realizada a un proceso analizado, donde se muestran tres paneles verticales. El primero (de izquierda a derecha) muestra en la sección superior el identificador del árbol asociado a cada grupo obtenido en la alineación. En la sección inferior aparecen los operadores que pueden aplicarse al bloque de construcción selecionado o todos los bloques de construcción que conforman el árbol obtenido anteriormente.

En el panel del centro se muestra el árbol, de bloques de construcción obtenido, así como los detalles del bloque de construcción selecionado y la leyenda.

En el panel de la derecha se muestra en su sección superior el árbol de bloques de construcción estimado. En la segunda sección aparecen los detalles del bloque de construcción seleccionado en la sección anterior y en la última sección se muestra un resumen de las actividades invisibles que fueron insertadas y la causa por la cual se adicionó.

En la parte superior derecha de la Figura 15 aparece un menú que permite salvar la solución obtenida a partir de la estimación de información ausente.

Figura 15 Estimación de información ausente.

Resultados experimentales

Se definió un experimento para probar la efectividad del algoritmo desarrollado y para ello se emplea la aplicación informática expuesta anteriormente, junto a las métricas definidas en el Capítulo 1.

Diseño experimental

Para el experimento se definen cuatro grupos de procesos y tres momentos. Cada grupo está asociado a un proceso de negocio. Los momentos están asociados a las etapas por las que transita un proceso, es decir, la evaluación a partir de un registro de evento en su estado original, con ausencia de información y con información estimada.

La Tabla 4 muestra el diseño experimental realizado y la simbología utilizada es la siguiente:

- G: Grupo de participantes. En este caso cada grupo está compuesto por 45 procesos, 15 asociados a cada momento (Original, Ausencia de Información e Información estimada).
- R: Asignación al alzar.
- X: Tratamiento o estímulo. En este caso X_1 se corresponde con la extracción de información del registro de evento analizado en cada grupo durante el primer momento. X_2 se corresponde con la aplicación del modelo propuesto para la estimación de información ausente.
- O: Observación.

Tabla 4 Diseño experimental propuesto.

	Original		**Ausencia de información**		**Información estimada**
R G_1	O_1	X_1	O_2	X_2	O_3
R G_2	O_4	X_1	O_5	X_2	O_6
R G_3	O_7	X_1	O_8	X_2	O_9
R G_4	O_{10}	X_1	O_{11}	X_2	O_{12}

Se han empleado 4 procesos diferentes, uno para cada grupo. Se aplicaron dos algoritmos de descubrimiento, Alpha++ e ILP. A partir de su aplicación se evaluaron las cinco métricas enunciadas anteriormente, por lo cual se emplea el mismo diseño experimental propuesto para cada evaluación realizada y en correspondencia con la métrica y el algoritmo utilizado.

En el primer momento (Original) las observaciones O_1, O_4, O_7 y O_{10} representan la evaluación de la métrica analizada para el registro de evento en su estado original. En el segundo momento (Ausencia de información) las observaciones O_2, O_5, O_8 y O_{11} están asociadas a la evaluación de la métrica analizada después de extraer información del registro de evento original (X_1) y aplicar el algoritmo de descubrimiento. En este punto para cada observación se hacen 15 mediciones. Al registro de evento original se le extrajo de manera aleatoria el 3, 5 y 10 % de la información.

Se formaron tres grupos de cinco procesos cada uno en correspondencia con los porcientos de ausencia de información. Los registros de eventos que conforman un grupo son diferentes.

En el último momento (Información estimada) las observaciones O_3, O_6, O_9 y O_{12} están asociadas a la evaluación de la métrica analizada después de aplicar el modelo propuesto (X_2) y el algoritmo de descubrimiento. Cada registro de evento con ausencia de información se transformó usando la aplicación informática desarrollada y se obtuvo un nuevo registro de evento con la información estimada.

En consecuencia, cada observación asociada a este momento contiene 15 registros de eventos.

Luego se realizaron comparaciones por pares en un grupo utilizando el test no paramétrico de signos con rangos de Wilcoxon (JOHN E. FREUND 2006). Se analizan los datos correspondientes al primer y segundo momento, por ejemplo O_1 y O_2, buscando detectar de manera significativa un predominio del decremento en la segunda observación. También se analizan los datos correspondientes al segundo y tercer momento, por ejemplo O_2 y O_3, buscando detectar de manera significativa un predominio del incremento en la tercera observación. Por último, se analizan los datos correspondientes al primer y tercer momento, por ejemplo O_1 y O_3, buscando detectar un empate entre los valores, lo cual no revelaría diferencias significativas.

Se realizan análisis transversales que permiten comparar las observaciones de los diferentes grupos en un mismo momento, por ejemplo O_1, O_4, O_7 y O_{10}. En estos casos, se espera detectar diferencias significativas al aplicar el test de Kruskal-Wallis (JOHN E. FREUND 2006). La diferencia entre los valores estaría determinada por el hecho de que los registros de eventos analizados presentan diferentes características, lo que influye en la complejidad del proceso de descubrimiento y en la evaluación de las métricas utilizadas.

Para entender las diferencias detectadas se realizan las comparaciones por pares utilizando el test de Mann-Whitney (JOHN E. FREUND 2006). Se espera detectar diferencias significativas entre las observaciones asociadas a los registros de eventos con diferentes características.

Características de los procesos analizados

Se utilizaron para realizar las pruebas cuatro procesos que fueron generados de utilizando la aplicación informática Process Log Generator en la versión 1.4 beta

(PROCESS-MINING-GROUP 2011). Se decidió utilizar una herramienta que generara de manera aleatoria un registro de evento artificial, debido a que, en el área de minería de proceso no se detectó una base de datos que contenga procesos que puedan ser utilizados para la validación de las técnicas desarrolladas. Aún cuando existen algunos registros de eventos asociados a procesos reales estos no cuentan con las características adecuadas para realizar una correcta validación de la propuesta. Los principales problemas están relacionados con el reflejo parcial de los patrones de flujo de trabajo y la cantidad de casos.

Hay que señalar además que de estos registros de eventos no se conoce el proceso original a partir del cual tienen lugar, en consecuencia, no es posible saber que actividades pueden estar faltando o cuanto ruido pudo manifestarse.

Un registro de evento generado de manera artificial puede reflejar los patrones de flujo de control conocidos, tener diferentes niveles de patrones anidados y la cantidad de casos puede variar en correspondencia con los elementos enunciados.

Las características de los registros de eventos generados son las siguientes:

Tabla 5 Descripción de los registros de eventos.

	Proceso 1	Proceso 2	Proceso 3	Proceso 4
Cantidad de casos	100	1000	2000	500
Eventos	762	10400	23145	6655
Clases de eventos	18	34	28	40
Cantidad de patrones anidados	2	3	3	3
Refleja patrones de flujo de trabajo: Secuencia, AND Split/join, XOR Split/join	Sí	Sí	Sí	Sí
Refleja el patrón de flujo de trabajo Lazo	No	No	No	Sí

Para la conformación de los registros de eventos se fue incrementando las variables, cantidad de casos, cantidad de patrones anidados y la probabilidad de reflejar los patrones de flujo de trabajo. La complejidad del descubrimiento aumenta en la medida en la que se manifiestan, en el registro de evento, mayor cantidad de variables y estas reflejan un aumento.

Es necesario señalar que no es objetivo de este trabajo establecer una relación entre las características de un registro de evento y la complejidad en el descubrimiento del modelo de procesos representativo del mismo.

Hay que considerar que los algoritmos de descubrimiento de procesos se diferencian en la forma en la que manejan aspectos como los patrones de flujo de trabajo, por lo cual, variables como la cantidad de casos influyen en el resultado pero no necesariamente determina una mejor solución.

De los registros de eventos generados el que se considera de menor complejidad (para el descubrimiento) es el asociado al Proceso 1. El de mayor complejidad es el asociado al Proceso 4, debido a que, aún cuando no tiene la mayor cantidad de casos, si posee la mayor cantidad de clases de eventos y refleja todos los patrones de flujo de trabajo que se enuncian.

Algoritmos de descubrimiento utilizados

En la experimentación se utilizaron dos técnicas de descubrimiento de procesos, el Alpha++ (WEN et al. 2004; WEN et al. 2006) y el ILP (WERF et al. 2008; WIEL 2010). Ambas técnicas tienen como resultado una red de flujo de trabajo que representa el proceso descubierto. Esto da la posibilidad de que se muestren como recuadros grises las actividades invisibles que sean detectadas.

Según el análisis realizado, el algoritmo Alpha++ cubre tres de las ocho situaciones de ausencia de información enunciadas en el Capítulo 1 (Actividades invisibles contra Actividades duplicadas, Actividades invisibles contra lazos y Actividades invisibles contra sincronización). Para la aplicación de esta técnica se utiliza la implementación realizada en la herramienta ProM en su versión 6.1 (EINDHOVEN-UNIVERSITY-OF-TECHNOLOGY 2012).

El algoritmo ILP cubre cuatro de las ocho situaciones de ausencia de información enunciadas en el Capítulo 1 (Actividades invisibles contra Actividades duplicadas, Actividades invisibles contra lazos, Actividades invisibles contra sincronización y Lazos contra actividades invisibles junto a actividades duplicadas). Para la aplicación de esta técnica se utiliza la implementación realizada en la herramienta ProM en su versión 6.1 (EINDHOVEN-UNIVERSITY-OF-TECHNOLOGY 2012).

No se seleccionan otras técnicas analizadas que tienen mejor desempeño que los seleccionados, por ejemplo Genetic Miner, debido a que estas no satisfacen las

situaciones de ausencia de información agregando al modelo descubierto una actividad invisible.

El modelo propuesto estima la información ausente a partir del registro de evento por lo cual se elimina la ambigüedad en la interpretación de este tipo de soluciones, independientemente del algoritmo de descubrimiento que se emplee.

Análisis de los resultados

Considerando el diseño experimental expuesto se realizan un conjunto de pruebas. La primera estuvo dirigida a comparar mediante un análisis de varianza de segunda vía no paramétrico de Friedman, los datos asociados a cada uno de los momentos enunciados para un grupo específico. Esta evaluación detectó diferencias altamente significativas (significación 0.000 < 0.01) entre las observaciones. Los valores observados disminuyeron para el segundo de los momentos (Ausencia de información) y aumentaron en el último (Información estimada).

La Figura 16 muestra los valores de los rangos medios para cada grupo al aplicar ambos algoritmos y medir las métricas Fitness Unsatisfied y Fitness Unhandled. Para ambas métricas obtienen los mismos valores, por lo cual, se representan en una misma figura.

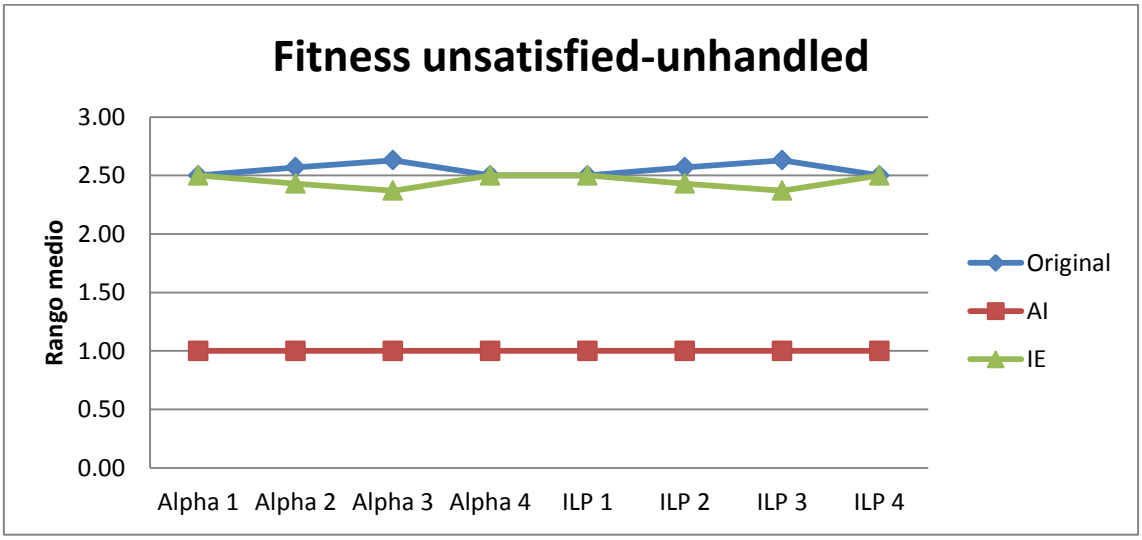


Figura 16 Evaluación de los rangos medios para los diferentes grupos, métrica Fitness Unsatisfied y Fitness Unhandled.

Los grupos se reflejan en el eje de las abscisas en las figuras 16, 17 y 18. Distinguiéndose las mediciones asociadas a la aplicación de cada algoritmo de descubrimiento usado.

Las comparaciones por pares en un grupo se realizaron utilizando el test no paramétrico de signos con rangos de Wilcoxon.

Se analizaron los datos correspondientes al primer y segundo momento y se detectó de manera altamente significativa (significación 0.000 < 0.01) un predominio del decremento en la segunda observación (AI, siglas de Ausencia de información). También se analizaron los datos correspondientes al segundo y tercer momento y se detecta de manera altamente significativa (significación 0.000 < 0.01) un predominio del incremento en la tercera observación (IE, siglas de Información estimada).

Por último, se analizaron los datos correspondientes al primer y tercer momento y se detecta un empate entre los valores, lo cual no revela diferencias significativas (significación 1.000).

Queda demostrado que la ausencia de información produce una reducción en esta medida que luego se recupera al estimar la información ausente utilizando el modelo propuesto.

Análogamente se realiza la evaluación para la métrica ETCPrecision. En la Figura 17 se muestra los valores de los rangos medios para cada grupo al aplicar ambos algoritmos y medir la métrica. El resultado de la evaluación es semejante a los obtenidos para las métricas Fitness Unsatisfied y Fitness Unhandled, excepto para la evaluación realizada al grupo 4 al aplicar el algoritmo ILP. Al hacer las comparaciones por pares en el grupo utilizando el test no paramétrico de signos con rangos de Wilcoxon no se aprecian diferencias significativas.

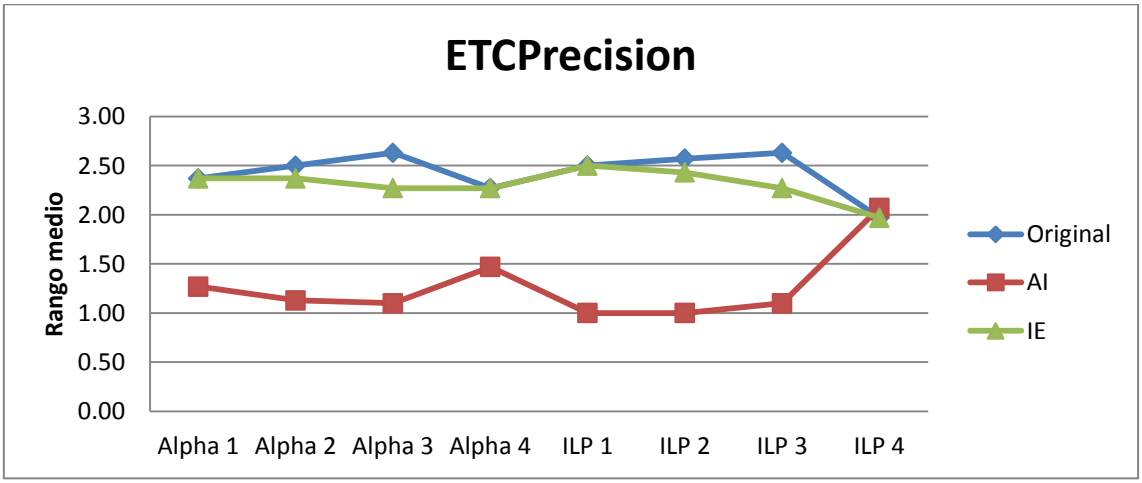


Figura 17 Evaluación de los rangos medios para los diferentes grupos, métrica ETCPrecision.

El grupo 4 se evalúa a partir del Proceso 4 generado. El registro de evento correspondiente presenta mayor cantidad de clases de eventos y refleja el patrón de flujo de trabajo Lazo. Esto propicia que los algoritmos de descubrimiento utilizados descubran modelos menos precisos.

El valor de la métrica aumenta en el momento asociado a la ausencia de información porque algunas de las actividades que se eliminaron del registro de evento original determinaban un lazo, por lo cual se restringe el comportamiento reflejado en el modelo descubierto.

Mientras que el modelo descubierto utilizando el registro de evento original y el que posee la información estimada genera modelos que generalizan el comportamiento asociado a cada lazo y por tanto disminuye la precisión.

Aún considerando los aspectos antes señalados para la comparación de las observaciones del primer y último momento predominan los empates, por lo cual, no hay diferencias significativas (significación 1.000).

Esto demuestra que para la medida analizada la aplicación del modelo propuesto soluciona las afectaciones provocadas por la ausencia de información sobre el registro de evento original.

Se realiza la evaluación para la métrica Non Fit Traces de manera análoga a como se realizó para las métricas Fitness Unsatisfied y Fitness Unhandled. En la Figura 18 se muestra los valores de los rangos medios para cada grupo al aplicar ambos algoritmos y medir la métrica. El resultado de la evaluación es semejante a los obtenidos para las métricas Fitness Unsatisfied y Fitness Unhandled, excepto para la evaluación realizada al grupo 4 aplicando el algoritmo de descubrimiento Alpha.

Este resultado se debe a que al aplicar el algoritmo Alpha sobre un registro de evento que refleja el patrón de flujo de trabajo lazo se obtiene un modelo que generaliza el comportamiento observado, esto provoca que las trazas contenidas en el registro de evento no puedan reproducirse completamente sobre el modelo.

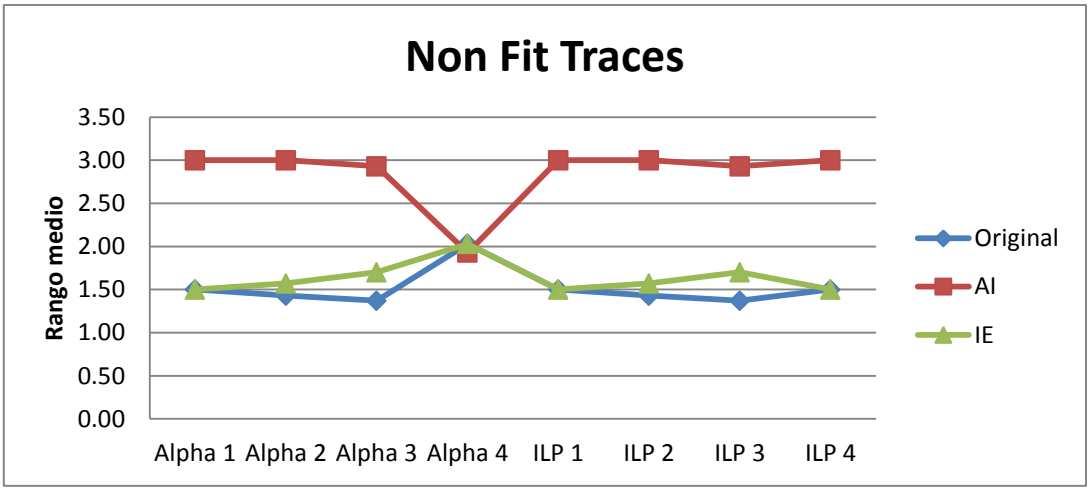


Figura 18 Evaluación de los rangos medios para los diferentes grupos, métrica Non Fit Traces.

La disminución de los valores asociados a la evaluación en el segundo momento se debe a que el modelo descubierto, a partir del registro de evento con ausencia de información, restringe el comportamiento observado en el registro de evento original, lo que propicia que mayor cantidad de trazas puedan reproducirse sobre el modelo.

Esta situación no se manifiesta en la evaluación de los modelos obtenidos utilizando el algoritmo ILP, debido a que, este busca en cada momento cubrir con el modelo obtenido el comportamiento observado en cada una de las trazas analizadas.

Aún considerando los aspectos antes señalados para la comparación de las observaciones del primer y último momento predominan los empates, por lo cual, no hay diferencias significativas (significación 1.000). Esto demuestra que para la medida analizada la aplicación del modelo propuesto soluciona las afectaciones provocadas por la ausencia de información sobre el registro de evento original.

La evaluación de la métrica Improved Structural Appropriateness (ISA, por sus siglas en inglés) para todos los modelos obtenidos es 1.0. Esto revela que los algoritmos aplicados, independientemente del momento, no introducen en el modelo descubierto problemas estructurales (presencia de actividades invisibles redundantes y actividades duplicadas alternativamente), lo cual es positivo dado que los problemas estructurales dificultan el entendimiento del proceso analizado.

Aplicación de la propuesta en un entorno real

Para la aplicación de la propuesta en un entorno real se analizó un análisis de un registro de evento almacenado por el Sistema Único de Identificación Nacional (SUIN),

específicamente del módulo Gestionar Recursos. El SUIN fue desarrollado por el Ministerio del Interior de Cuba en conjunto con la Universidad de las Ciencias Informáticas.

El proceso Gestionar Recursos a partir del cual se generó el registro de evento utilizado tiene como objetivo gestionar las operaciones que se realizan sobre cualquier recurso del entorno informatizado. Entiéndase que un recurso puede ser desde un dispositivo de hardware hasta una interface del SUIN. Los recursos se asignan a los roles creados en una determinada entidad.

El SUIN fue desarrollado utilizando la tecnología .Net, específicamente Windows Workflow Foundation. Como base para la implementación del flujo de trabajo correspondiente al proceso Gestionar Recursos se utilizó el modelo que se muestra en el Anexo 1.

El registro de evento utilizado presenta 51 casos, 11011 eventos, 90 clases de eventos, 3 tipos de eventos (Execute, Closed y Faulting) y 47 participantes.

Inicialmente haciendo uso de la herramienta informática desarrollada se alinean las trazas en solo grupo. Posteriormente se realiza un análisis de las trazas alineadas y se determina que solo siete casos son completos. A pesar de esto, se decide trabajar con los 51 casos dado que de extraer los casos incompletos se perdería representatividad, es decir, el registro de evento representaría solo una parte del comportamiento registrado del proceso. Si se dejan solo los siete casos completos no se aprecian eventos relacionados con el subproceso Agregar recurso. Después de este análisis se descompone el proceso analizado.

A partir de este árbol se realiza la estimación de información ausente y se obtienen ocho actividades invisibles.

La Figura 19 muestra en el panel izquierdo, en la sección de configuración de los operadores, los operadores seleccionados para la estimación (*Operador de salto*, *Operador de lazo*, *Operador de división/unión* y *Operador de secuencia oculta*). En el panel derecho, en la sección donde se describen las actividades invisibles, se muestran las actividades estimadas λ_1, λ_{11}, λ_{16}, λ_{17}, λ_{18}, λ_{19}, λ_{20} y λ_{21}. Siete de las ocho actividades estimadas se originaron por el *Operador de salto*, mientras que la restante se creó por el *Operador de secuencia oculta*. Las actividades invisibles no aparecen numeradas de manera consecutiva porque en el proceso de estimación se originan 22 actividades invisibles, pero se eliminan del resultado final las actividades invisibles que

solo tienen sentido localmente, es decir solo las asociadas a un determinado subproceso.

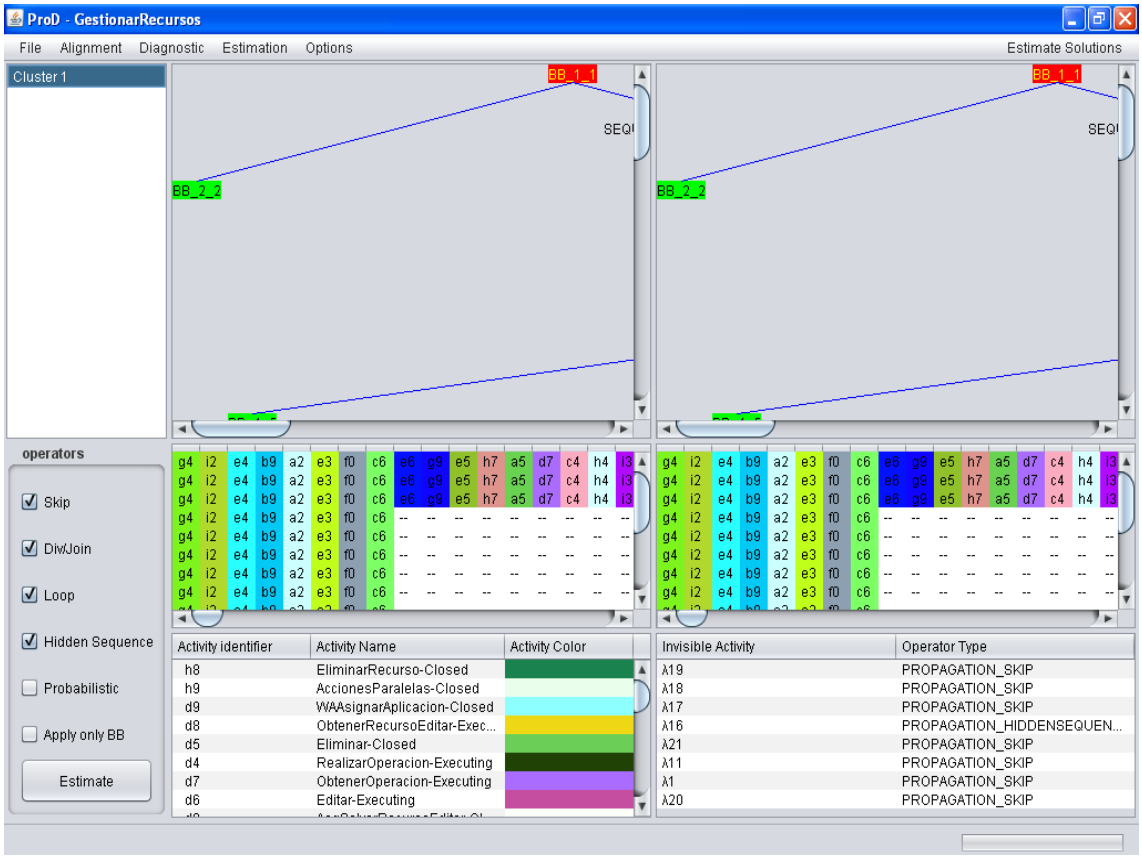


Figura 19 Estimación de información ausente para el proceso Gestionar Recursos

A partir de la estimación realizada se hace un análisis en conjunto con un especialista del negocio para garantizar que las actividades estimadas se manifiestan en el proceso ejecutado. De este análisis se concluye lo siguiente:

- Las actividades λ_{11} y λ_{21} indican casos incompletos (ausencia de actividades finales del proceso). Es decir, durante el pre-procesamiento de la alineación no se eliminaron del registro de evento los posibles casos incompletos, para así evaluar la respuesta del sistema desarrollado en estos casos. Cada una de las actividades estimadas se corresponde con la ausencia de información en casos diferentes.
- Las actividades λ_1, λ_{17}, λ_{18}, λ_{19}, y λ_{20} indican el existo en los cinco subprocesos Agregar, Editar, Eliminar recursos, Condicionar Operación y Flujo de Negocio. En cada subproceso se almacenó un evento de fallo pero no aparece un evento que indique el éxito. Las actividades estimadas permitieron identificar fácilmente los casos en los que el proceso funcionó correctamente.

- La actividad λ_{16} permite delimitar el subproceso Condicionar Operación. Después del evento *Condicionar Operación* en uno de los casos no se registró ningún otro evento (caso incompleto), por lo cual, no es posible en el registro de evento analizado identificar adecuadamente el subproceso Condicionar Operación. El subproceso Condicionar Operación es el que incluye los subprocesos Agregar, Editar y Eliminar recursos. La actividad invisible en este caso permitió delimitar acertadamente el subproceso Condicionar Operación lo cual facilita el análisis de la información contenida en el registro de evento.

Como se puede apreciar cada una las actividades estimadas se corresponden con actividades existentes en el proceso de negocio ejecutado, aún cuando existen casos incompletos que pudiesen dificultar el análisis.

Las actividades estimadas facilitan la comprensión del proceso analizado al reflejar actividades ausentes e impedir que se establezcan incorrectas relaciones entre las actividades existentes. Además, las actividades λ_1, λ_{17}, λ_{18}, λ_{19}, y λ_{20} resultan útiles para la posterior mejora del sistema implementado.

Conclusiones del capítulo

Para la validación de la solución propuesta se han realizado un grupo de experimentos con varios procesos generados, a partir de los análisis realizados se evidenció que el algoritmo propuesto identifica y maneja correctamente las diferentes situaciones de ausencia de información. Haciendo uso de la aplicación desarrollada se aplicaron correctamente los operadores definidos en la investigación y se pudieron obtener los valores correspondientes a los modelos con información estimada, demostrando la efectividad de la estimación de información ausente.

Además la aplicación desarrollada se aplicó en un entorno real (SUIN), específicamente en el módulo Gestionar Recursos Las actividades estimadas se validaron con la ayuda de un especialista del negocio y se comprobó la efectividad del algoritmo propuesto. La estimación realizada facilitó la comprensión del proceso analizado y sirvió como base para futuras mejoras del SUIN.

CONCLUSIONES

En la investigación se plantearon un conjunto de objetivos que se fueron cumpliendo progresivamente y permitieron arribar a las siguientes conclusiones:

- En el tratamiento de la ausencia de información se pueden identificar dos aristas, una se desarrolla en la etapa de pre-procesamiento del registro de evento y la segunda durante el descubrimiento del modelo de procesos. Ninguna de las técnicas analizadas para ambas aristas permite el tratamiento de todas las situaciones asociadas a la ausencia de información que fueron enunciadas.
- El algoritmo propuesto posibilita que se cubran todas las situaciones de ausencia de información identificadas, mediante un conjunto de operadores desarrollados, los cuales permiten detectar y estimar la información ausente en el registro de evento, haciendo uso de la alineación de las trazas y la descomposición en un árbol de bloques de construcción.
- Las pruebas realizadas haciendo uso de la aplicación informática desarrollada arrojaron que, aun cuando existen registros de eventos diferentes, la ausencia de información produce generalmente una reducción en las medidas de las métricas utilizadas, que luego se recupera al estimar la información ausente utilizando el algoritmo propuesto.

RECOMENDACIONES

Después de haberse realizado esta investigación se plantean un grupo de recomendaciones a tener en cuenta:

- Perfeccionar la herramienta desarrollada, para permitir una mejor representación visual de los elementos que componen la misma y el análisis de varios procesos simultáneamente.
- Generalizar el uso de la herramienta desarrollada en los sistemas que lo permitan, para facilitar el análisis del funcionamiento de los mismos.
- Realizar la implementación de la aplicación propuesta como complemento para la herramienta ProM.

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.
- **Generalización:** es una medida para determinar cuán bien el modelo es capaz de describir un comportamiento desconocido
- **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.
- **Precisión:** es una medida para determinar si el modelo prohíbe un comportamiento muy diferente al comportamiento observado en el registro de evento.
- **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).
- **Simplicidad:** el modelo más simple que pueda explicar el comportamiento observado en el registro de evento, es el mejor modelo. La simplicidad se puede cuantificar de distintas maneras, por ejemplo, la cantidad de nodos y arcos en el modelo.
- **Soporte Operacional:** es un análisis en línea de los datos de eventos con el objetivo de supervisar e influir en las instancias del proceso en ejecución. Se pueden identificar tres actividades de soporte operacional: detectar (generar una alerta si el comportamiento observado se desvía del comportamiento modelado), predecir (predecir el comportamiento futuro basado en el comportamiento pasado, por ejemplo, predecir el tiempo de procesamiento restante), y recomendar (sugerir las medidas adecuadas para alcanzar un objetivo concreto, por ejemplo, minimizar costos).
- **Verificación 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.
- **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. (www.xes-standard.org).

REFERENCIAS

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. p. 1–14

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* p. *96/23*, 1996.

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), 1996. p.179-201.

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*, 2011, p. 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, 2009.

AALST, W. M. P. V. D. and C. W. GÜNTHER. *Finding Structure in Unstructured Processes: The Case for Process Mining*. ACSD '07 Proceedings of the Seventh International Conference on Application of Concurrency to System Design, IEEE Computer Society, p. 7-10. 2007

AALST, W. M. P. V. D. and K. M. V. HEE. *Workflow Management: Models, Methods, and Systems*. USA, 2004. p. 0-262-01189-1

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.

AALST, W. M. P. V. D.; H. A. REIJERS*, et al.* Business process mining: An industrial application *Information Systems*, p. 32, 2007.

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*, p. 9(1): 87-111, 2009.

AALST, W. M. P. V. D.; M. H. SCHONENBERG*, et al.* Time Prediction Based on Process Mining *Information Systems*, 2011c, p. 36(2): 450–475.

AALST, W. M. P. V. D. and A. J. M. M. WEIJTERS Chapter 10: Process Mining *Process-Aware Information Systems: Bridging People and Software Through Process Technology. John Wiley & Sons Inc*, 2005.

AALST, W. M. P. V. D. and A. J. M. M. WEIJTERS Process Mining *Special Issue of Computers in Industry. Elsevier Science Publishers, Amsterdam*, 2004, p. 53.

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, p. 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, p. 16(9): 1128-1142.

ADRIANSYAH, A.; B. F. V. DONGEN*, et al. Towards Robust Conformance Checking*. BPM 2010 Workshops, Proceedings of the 6th Workshop on Business Process Intelligence (BPI2010), Lecture Notes in Business Information Processing. Springer, Berlin, 2011.

AG, S. *ARIS Process Performance Manager*, 2011. [Disponible en: http://www.softwareag.com/corporate/products/aris_platform/aris_controlling/aris_process_performance/overview/default.asp

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

ARENDONK, R. P. J. M. V. *A Benchmark Set for Process Discovery Algorithms*. Mathematics & Computer Science. Eindhoven, Eindhoven University of Technology, 2011.

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,: p.159-175.

BOSE, R. P. J. C. and W. M. P. V. D. AALST. *Context Aware Trace Clustering: Towards Improving Process Mining Results*. International Conference on Data Mining, is.ieis.tue.nl, 2009. p. 401-412

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

COOK, J. E. *Process Discovery and Validation Through Event-Data Analysis*, 1996. p.

COOK, J. E.; Z. DU*, et al.* Discovering Models of Behavior for Concurrent Workflows *Computers in Industry*, 2004: 297-319.

COOK, J. E. and A. L. WOLF Discovering Models of Software Processes from Event-Based Data *ACM Transactions on Software Engineering and Methodology*, 1998: p. 215-249.

COOK, J. E. and A. L. WOLF. *Event-Based Detection of Concurrency*. Proceedings of the Sixth International Symposium on the Foundations of Software Engineering (FSE-6), New York, NY, USA, 1998. p. 35-45

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

CHANGA, M.-K.; W. CHEUNGB*, et al.* Understanding ERP system adoption from the user's perspective *International Journal of Production Economics*, 2008, 113(2): 928-942.

DAA. *INFORME ESPECIAL DE IT-LATINO.NET*. Barcelona, España, DAA CONTENIDOS DIGITALES, S.L., 2010.

DONGEN, B. F. V. and W. M. P. V. D. AALST. *Multi-phase Process mining: Aggregating Instance Graphs into EPCs and Petri Nets*. Second International Workshop on Applications of Petri Nets to Coordination, Workflow and Business Process Management(PNCWB), 2005.

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: p. 362-376.

DONGEN, B. F. V.; W. M. P. V. D. AALST*, et al.* Verication of EPCs: Using Reduction Rules and Petri Nets *CAiSE, Lecture Notes in Computer Science.Springer*, 2005, 3520: p. 372-386.

DONGEN, B. V. *Process Mining and Verification*, Technische Universiteit Eindhoven, 2007.

EINDHOVEN-UNIVERSITY-OF-TECHNOLOGY. *ProM*, 2012.

FOURSPARK. *Flow Overview*, 2011. [Disponible en: http://fourspark.no/?page_id=11

FUJITSU. *Automated Process Discovery Service*, 2012. [Disponible en: http://www.fujitsu.com/global/services/software/interstage/solutions/bpmgt/bpm-services/apd/

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. p. 42–53

GOEDERTIER, S.; D. MARTENS*, et al.* Robust Process Discovery with Artificial Negative Events *Journal of Machine Learning Research*, 2009.

GRECO, G.; A. GUZZO*, et al.* Mining Hierarchies of Models: From Abstract Views to Concrete Specications *Business Process Management*, 2005, 3649: p. 32-47.

GRECO, G.; A. GUZZO*, et al.* Mining Expressive Process Models by Clustering Workflow Traces *PAKDD. Lecture Notes in Computer Science. Springer*, 2004, 3056: p. 52-62.

GRIGORI, D.; F. CASATI*, et al.* Business Process Intelligence *Computers in Industry*, 2004, 53(3): p. 321-343.

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

GÜNTHER, C. W.; S. RINDERLE-MA*, et al.* Using Process Mining to Learn from Process Changes in Evolutionary Systems *Inderscience*, 2008.

GÜNTHER, C. W.; A. ROZINAT*, et al.* Activity Mining by Global Trace Segmentation. en: *Business Process Management Workshops*. Lecture Notes in Business Information Processing, 2009. p. 43.

HENDRICKSA, K. B.; V. R. SINGHALB*, et al.* The impact of enterprise systems on corporate performance: A study of ERP, SCM, and CRM system implementations *Operations Management*, 2007, 25(1): p. 65-82.

HERBST, J. *Ein induktiver Ansatz zur Akquisition und Adaption von Workflow-Modellen*, University Ulm, 2001.

HERBST, J. *A Machine Learning Approach to Workflow Management*. 11th European Conference on Machine Learning.

HERBST, J. and D. KARAGIANNIS Integrating Machine Learning and Work-flow Management to Support Acquisition and Adaptation of Workflow Models *International Journal of Intelligent Systems in Accounting, Finance and Management*, 2000: p. 67-92.

HERBST, J. and D. KARAGIANNIS Workflow Mining with InWoLvE. *Computers in Industry*, 2004, 53(3): p. 245-264.

HILL, J. B.; M. CANTARA*, et al. Magic Quadrant for Business Process Management Suites*, Gartner, Inc., 2009.

IBM. *A New Way of Working: Insights from Global Leaders*. United States of America, IBM Institute for Business Value, 2010.

IMPROVEMENT, I. P. P. *Process Discovery Focus*, 2009. [Disponible en: http://www.iontas.com/pages/products/pdf.php

INCORPORATED, O. S. *Comprehend Overview*, 2011. [Disponible en: http://www.oc.com/technology/

JOHN E. FREUND, I. R. M., RICHARD JOHNSON. *Probabilidad y Estadística para Ingenieros*. La Habana, Editorial Félix Varela, 2006. p. *Probabilidad y Estadística para Ingenieros.*

MEDEIROS, A. K. A. D. *Genetic Process Mining*, Technische Universiteit Eindhoven, 2006.

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. p. 389–406.

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. p. 113-130.

MUÑOZ-GAMA, J. and J. CARMONA. *A fresh look at precision in process conformance*. 8th international conference on Business process management, Hoboken, NJ, USA, Springer-Verlag, 2010. p. 211-226 p. 3-642-15617-7

OBJECT MANAGEMENT GROUP, I. *Object Management Group*

Business Process Model and Notation, 2012. [Disponible en: http://www.bpmn.org/

OYJ, Q. *Automated Business Process Discovery Software*

QPR ProcessAnalyzer, 2011. [Disponible en: http://www.qpr.com/products/qpr-processanalyzer.htm

PROCESS-MINING-GROUP. *Process Log Generator*. *Process Mining group*. Padua, Italy, 2011.

RAYKENLER YZQUIERDO HERRERA, R. S. C., MANUEL LAZO CORTÉS. *OPERADORES PARA EL TRATAMIENTO DE AUSENCIA DE INFORMACIÓN EN LA MINERÍA DE PROCESOS*, 2012. Evento UCIENCIA. [Disponible en: http://uciencia.uci.cu/es/node/1334]

ROZINAT, A. and W. M. P. V. D. AALST Conformance checking of processes based on monitoring real behavior *Inf. Syst.*, 2008, 33(1): p. 64-95.

ROZINAT, A. and W. M. P. V. D. AALST. *Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models*. Third International Conference on Business Process Management(BPM 2005), France, 2005.

ROZINAT, A.; A. K. A. D. MEDEIROS*, et al.* The Need for a Process Mining Evaluation Framework in Research and Practice *Springer*, 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. p. 84-89

ROZINAT, A.; A. K. A. D. MEDEIROS*, et al.* Towards an Evaluation Framework for Process Mining Algorithms *BPM Center Report*, 2007.

RUBIN, V. *A Workflow Mining Approach for Deriving Software Process Models*, University of Paderborn, 2007.

SAP. 2012. [Disponible en: http://www.sap.com

SCHIMM, G. *Generic Linear Business Process Modeling*. the ER 2000 Workshop on Conceptual Approaches for E-Business and The World Wide Web and Conceptual Modeling, Lecture Notes in Computer Science. Springer-Verlag, Berlin., 2000. p. 31-39

SCHIMM, G. Mining Exact Models of Concurrent Workflows *Computers in Industry*, 2004, 53(3): p. 265-281.

SCHIMM, G. *Mining Most Specic Workflow Models from Event-Based Data.* International Conference on Business Process Management (BPM 2003), Lecture Notes in Computer Science, 2003. p. 25-40.

SCHIMM, G. *Process Miner - A Tool for Mining Process Schemes from Event-based Data*. 8th European Conference on Articial Intelligence (JELIA), Berlin, Lecture Notes in Computer Science, 2002. p. 525-528.

SCHIMM, G. *Process Mining* 2012. Disponible en: http://www.processmining.de/

SOFTWARE, P. *Perceptive Reflect Overview*, 2012. [Disponible en: http://www.perceptivesoftware.com/products/product-explorer/business-process/perceptive-reflect.psi

SONG, M.; C. W. GÜNTHER*, et al. Trace Clustering in Process Mining*. Business Process Management Workshops (2009), Milano, Italy Lecture Notes, 2009. p. 109-120

STEREOLOGIC. *StereoLOGIC Discovery Analyst™ (*2012*)*. Disponible en: http://www.stereologic.com/stereologic_software.htm

TARANTILISA, C. D.; C. T. KIRANOUDISB*, et al.* A Web-based ERP system for business services and supply chain management: Application to real-world process scheduling *European Journal of Operational Research*, 2008, p. 187 (3): 1310-1326.

WEERDT, J. D.; M. D. BACKER*, et al. A critical evaluation study of model-log metrics in process discovery*. 6th International Workshop on Business Process Intelligence, Springer, 2010. p. 158-169 p. 978-3-642-20510-1

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: p. 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.

WEN, L.; J. WANG*, et al.* A Novel Approach for Process Mining Based on Event Types *BETA Working Paper Series, WP 118, Eindhoven University of Technology, Eindhoven*, 2004.

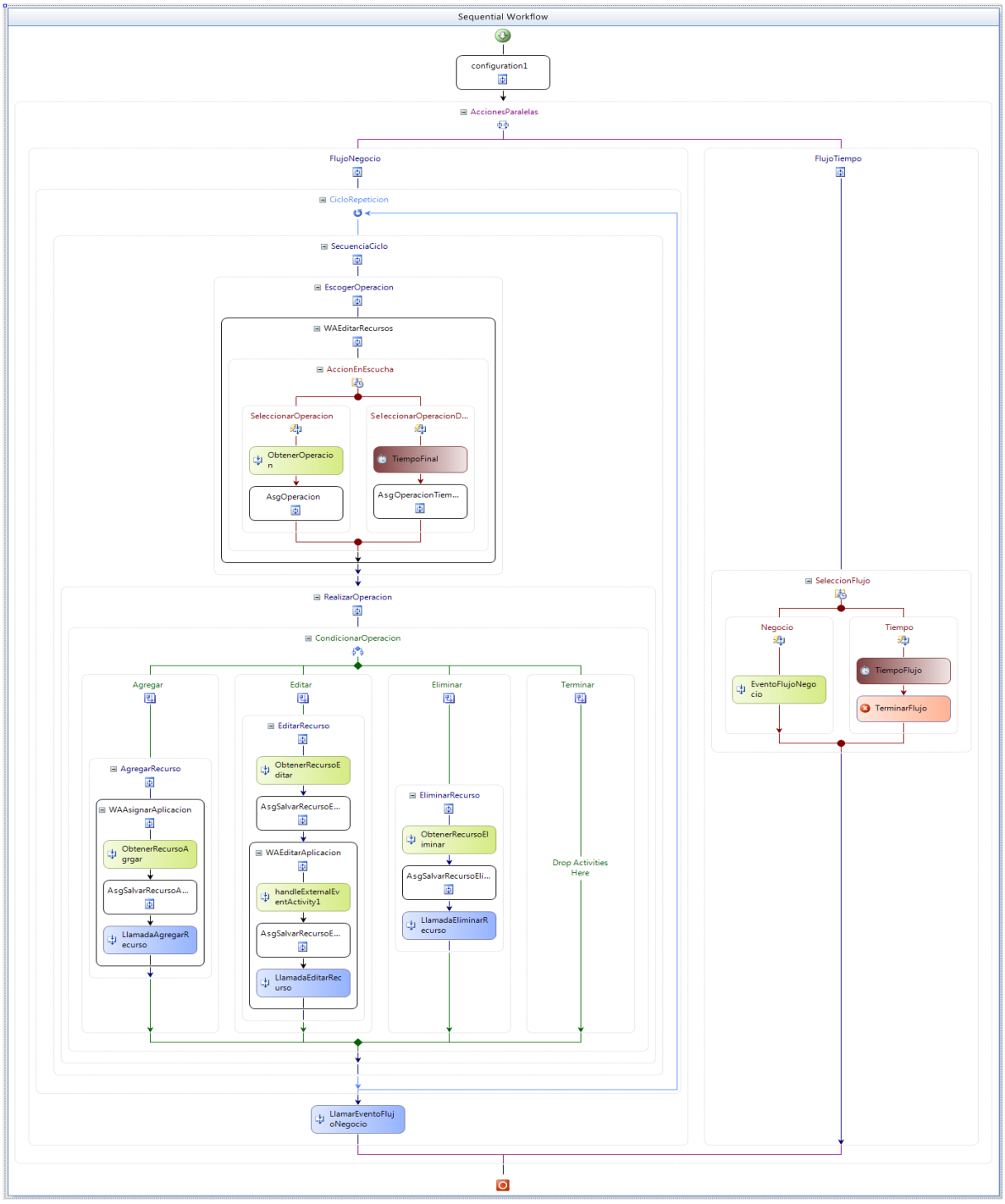
WEN, L.; J. WANG*, et al.* Detecting Implicit Dependencies Between Tasks from Event Logs *APWeb, Lecture Notes in Computer Science. Springer*, 2006, 3841: p. 591-603.

WERF, J. M.; B. F. DONGEN*, et al. Process Discovery Using Integer Linear Programming*. *Proceedings of the 29th international conference on Applications and Theory of Petri Nets*. Xi'an, China, Springer-Verlag, 2008. p. 368-387.

WIEL, T. V. D. *Process Mining using Integer Linear Programming*. Department of Mathematics and Computer Science. Eindhoven, Eindhoven University of Technology, 2010.

ANEXOS

Anexo 1



Modelo representativo del proceso Gestionar Recursos.