Tratamiento de la ausencia de información en la minería de procesos

Treatment of lack of information in process mining

Raykenler Yzquierdo-Herrera[*]*, Rogelio Silverio-Castro, Manuel Lazo-Cortés*

Facultad No. 3, Universidad de las Ciencias Informáticas. Reparto Torrens, Carretera San Antonio Km 2 ½, Boyeros. C.P. 19370. La Habana, Cuba.

(Recibido el 12 de enero de 2012. Aceptado el 29 de junio de 2013)

Resumen

La minería de procesos es una disciplina que abarca el descubrimiento, monitoreo y mejora de los procesos reales, a través de la extracción de conocimiento de los registros de eventos ampliamente disponibles en los actuales sistemas de información. La mayoría de los algoritmos de minería de procesos parten del supuesto que las trazas son completas y están libres de ruido. En la realidad este supuesto rara vez se cumple. La ausencia de información afecta la comprensión del modelo descubierto. En el trabajo se propone un conjunto de pasos para la estimación de la información ausente en las trazas utilizadas para la minería de procesos. Como parte de la estimación se propone alinear las trazas durante la etapa de pre-procesamiento de estas para detectar las posibles situaciones en las que se manifiesta la ausencia de información. A partir de las situaciones detectadas se estima la información ausente y se genera en correspondencia un nuevo registro de evento, el cual puede ser utilizado por los diferentes algoritmos de descubrimiento de proceso existentes. Finalmente se discuten los resultados experimentales obtenidos al aplicar la propuesta.

Palabras clave: Ausencia de información, minería de procesos, registro de eventos, traza

Abstract

Process mining is a research discipline including discovery, monitoring and improvement of real processes by extracting knowledge from event logs readily available in today's information systems. Most process mining algorithms assume that the traces are complete and free of noise. In reality, this assumption is rarely met. The lack of information affects the structure and

* Autor de correspondencia: teléfono: + 53 + 7 + 8358038, fax: + 53 + 7 + 835 8196, correo electrónico: ryzquierdo@uci.cu (R. Yzquierdo)

understanding of the model discovered. The paper proposes a set of steps to estimate the lack of information in the traces used in processes mining. Align traces during the pre-processing stage is proposed as part of the estimation in order to detect possible situations in which lack of information is manifested. From detected situations, missing information is estimated and a new record is generated in correspondence to the event, which can be used by different process discovery algorithms. Finally we discuss the experimental results obtained from implementing the approach.

Keywords: Event log, lack of information, process mining, trace

Introducción

La mayoría de las empresas utilizan sistemas de información para gestionar la ejecución de sus procesos de negocio (en lo adelante solo proceso) [1]. Los sistemas de información utilizados registran en forma de trazas las acciones (actividades) que se van realizando cuando se ejecutan instancias o casos del proceso. Al descubrimiento del proceso a partir de la información contenida en las trazas se le denomina minería de procesos (MP) o de workflow [2, 3]. La minería de procesos permite también el monitoreo y la mejora de los procesos reales extraídos de las trazas almacenadas por estos sistemas [4, 5].

La mayoría de los algoritmos de minería de procesos parten del supuesto que las trazas son completas y están libres de ruido. En la realidad este supuesto rara vez se cumple [4, 6]. La completitud puede verse en dos sentidos. La primera manifestación se refleja en que determinadas instancias de un proceso pueden no estar registradas en las trazas debido a que no han ocurrido, aun cuando dichas instancias pudiesen ser soportadas por los sistemas de información empleados en la empresa. Esta situación puede afectar el modelo descubierto [4, 7]; por lo cual se han desarrollado trabajos orientados a resolver esta problemática [8]. La segunda manifestación se refleja en que una o varias actividades del proceso no se registran en las trazas. A esto se le denominará ausencia de información. Una actividad no queda reflejada en el registro de eventos si no fue informatizada, si fue informatizada pero el sistema de información no deja constancia de su ocurrencia en el registro de eventos o si la actividad fue eliminada del fichero por alguna razón. A este tipo de actividad se le denomina actividad o tarea invisible [9, 10]. Los sistemas de información usados para informatizar un determinado proceso de negocio cubren un subconjunto de actividades, por ello, durante la ejecución del proceso de negocio pueden realizarse de manera intercalada tanto actividades de las cuales queda evidencia en el registro de eventos como actividades de las cuales no se registra información. En consecuencia, los sistemas de información almacenan en las trazas una secuencia incompleta de las actividades que conforman las instancias del proceso.

La ausencia de información puede generar que los algoritmos dirigidos a descubrir el proceso de negocio ejecutado, obtengan modelos en los que se reflejan incorrectas relaciones entre las actividades, esto no solo afecta la estructura del modelo obtenido sino que también se afecta la comprensión de este [11, 12]. Desde la arista del análisis de información, las actividades invisibles dificultan el análisis, impidiendo esto que se garantice niveles adecuados de confiabilidad y/o inferencias acertadas sobre el proceso analizado.

En este trabajo se propone un conjunto de pasos para realizar la estimación de información ausente en las trazas, para así garantizar la obtención de modelos más ajustados al comportamiento real del proceso analizado. La propuesta permite disminuir la afectación que provoca la ausencia de información sobre la estructura y la comprensión del modelo descubierto.

El artículo se estructura en las siguientes secciones: la segunda sección recoge un análisis de los diferentes trabajos asociados con el tratamiento de la ausencia de información. La tercera sección se describe un conjunto de definiciones que permiten el posterior entendimiento de la propuesta. En la cuarta sección se expone la propuesta para la detección y estimación de la información ausente en las trazas. Finalmente se hace un análisis de los resultados obtenidos en la aplicación y evaluación de la propuesta.

Trabajos relacionados

Seis situaciones de ausencia de información han sido descritas en la literatura, así como la interpretación que realizan los algoritmos de descubrimiento de las trazas afectadas por ellas. Las denominaciones de estas situaciones son: Situación de salto, Situación de división/unión, Actividades invisibles contra actividades duplicadas, Actividades invisibles contra lazos, Actividades invisibles contra sincronización y Lazos contra actividades invisibles junto a actividades duplicadas [9].

La ausencia de información en la minería de procesos ha sido tratada desde dos aristas fundamentales. Una está orientada a que las técnicas de descubrimiento sean robustas ante situaciones de ausencia de información, mientras que la segunda arista está dirigida a tratar con este tipo de problema en la etapa de pre-procesamiento del registro de evento, lo cual, es consecuencia de las deficiencias presentadas por las técnicas de descubrimiento en el tratamiento de situaciones asociadas fundamentalmente al ruido.

La capacidad de detección por un algoritmo de descubrimiento de las situaciones de ausencia de información está determinada en ocasiones, no solo por el algoritmo de descubrimiento, sino también por las posibilidades que tiene la notación utilizada para representar el modelo descubierto y específicamente las actividades invisibles [13]. Es necesario señalar que en los algoritmos desarrollados 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 [9].

Es necesario resaltar que algunos de los algoritmos desarrollados hasta el momento no manejan el constructor de actividades invisibles para la totalidad de las situaciones a las que se ha hecho referencia y la interpretación de las trazas varía de un algoritmo a otro [9].

Se han desarrollado un grupo de técnicas que durante la etapa de pre-procesamiento del registro de evento permiten detectar las afectaciones asociadas al ruido. En este sentido se puede resaltar algunas de las técnicas de visualización del registro de evento desarrolladas. Estas tienen como objetivo realizar un diagnóstico del proceso para poder identificar los aspectos generales del proceso, posibles anomalías, desviaciones y patrones interesantes. Ejemplos de este tipo de técnica lo constituyen Dotted chart analysis [14] y Trace alignment [15]. Aun cuando sea posible, preferentemente utilizando Trace alignment, identificar manualmente algunas de las situaciones más sencillas en las que se manifiesta la ausencia de información, esto solo sería posible en registros de eventos pequeños o medianos (registro de evento que contiene hasta cientos de casos).

Recientemente, se desarrolló un trabajo que realiza una transformación y limpieza semántica de los datos que forman el registro de evento [16]. Este trabajo está dirigido a tratar con las afectaciones asociadas específicamente con el ruido. Haciendo un análisis de esta técnica se determina que, aun cuando pudiesen definirse restricciones asociadas las situaciones en la que se manifiesta la ausencia de información, especialmente las más sencillas de reconocer (Situación de salto, Situación de división/unión, Actividades invisibles contra Actividades duplicadas); esto tendría que hacerse para cada proceso analizado y considerando los diferentes niveles de abstracción en los que se pueden manifestar la ausencia de información.

Además, la investigación no posibilita estimar las actividades invisibles en correspondencia con las situaciones posiblemente identificadas.

LTL Checker es una técnica que presenta las mismas limitantes que la técnica antes mencionada [17].

Ninguna de las técnicas analizadas tiene como propósito la detección de las situaciones en las que se manifiesta la ausencia de información abordada en este trabajo y tampoco realizan estimaciones de la información ausente. Pero, es significativo que estas técnicas proponen realizar un análisis en la etapa de pre-procesamiento de los registros de eventos, lo cual, puede ser útil para realizar la estimación de la información ausente. De esta forma se facilita la posterior aplicación de técnicas de descubrimiento.

Definiciones preliminares

Se abordará la ausencia de información como la falta 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 denominará *tarea invisible*.

Otros términos como traza, registro de eventos y estimación de ausente son definidos para un correcto entendimiento de la propuesta.

Definición 1: (Traza y registro de eventos). 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 eventos L es un grupo de trazas [15].

Definición 2: (Estimación de información ausente): La estimación de información ausente es el proceso mediante el cual se transforma el conjunto de trazas $T = \{T_1, T_2, ..., T_n\}$ en otro conjunto de trazas $\check{T} = \{\check{T}_1, \check{T}_2, ..., \check{T}_n\}$ donde, $\check{T}_i \in (\Sigma \cup \Lambda)^+$ para $1 \leq i \leq n$ y $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_w\}$ es el conjunto de actividades invisibles estimadas. $| T_i | \leq | \check{T}_i |$.

Tratamiento de la ausencia de información

Para el tratamiento de la ausencia de información en las trazas se proponen un conjunto de cuatro pasos que se aplican durante la etapa de pre-procesamiento del registro de eventos. Esto permitirá obtener un nuevo registro de eventos con la información estimada, el cual podrá ser utilizado en el descubrimiento sin importar el algoritmo empleado en este proceso.

Para identificar las actividades invisibles en un registro de eventos es necesario reconocer cada una de las manifestaciones de ausencia de información a las que se ha hecho referencia para posteriormente, poder estimar la información ausente.

El primero de los pasos consiste en la alineación de las trazas y tiene como objetivo preparar el registro de eventos para la posterior identificación de las manifestaciones de ausencia de información. La alineación de las trazas es una técnica propuesta por Bose y Van der Aalst (2012) [15] y que permite obtener una matriz *A* que representa un orden relativo entre las actividades y los casos. Para la implementación de este paso se usa un plug-in de la herramienta ProM 6 llamado Trace Aligment [18, 19].

Para identificar las manifestaciones de ausencia de información es necesario identificar los subprocesos que componen el proceso analizado. Esto posibilita que se puedan detectar manifestaciones en diferentes niveles de abstracción en el proceso. En correspondencia para la implementación del segundo paso se emplea una propuesta realizada por los autores del presente trabajo [20]. Como resultado de este paso se obtiene un árbol de bloques de construcción que representa la descomposición jerárquica en subprocesos del proceso analizado. Como se explica en [20] cada bloque de construcción representa un posible subproceso. La figura 1 muestra un ejemplo de un árbol de bloques de construcción.

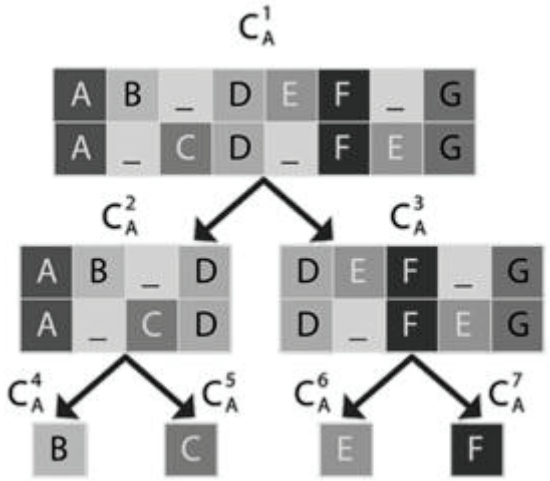

Figura 1 Árbol de bloques de construcción

Posteriormente y considerando el árbol de bloques de construcción obtenido anteriormente se aplican un conjunto de operadores que identifican cada una de las manifestaciones de ausencia de información y estiman las actividades correspondientes. Como resultado se obtiene un árbol de bloques de construcción con la información estimada. Por su importancia este paso se describe en el siguiente epígrafe.

En el cuarto paso a partir del árbol con la información estimada y el registro de eventos original se construye un nuevo registro de eventos con la información estimada.

Aplicación de los operadores de ausencia de información

Para cada bloque de construcción identificado se aplican un conjunto de operadores definidos para la estimación de la información ausente. Un operador permite estimar la información ausente en correspondencia con una determinada situación. Estas situaciones se abordan junto con la descripción de cada operador.

En dependencia de las características del bloque de construcción puede o no aplicarse la totalidad de los operadores propuestos. Estos son: Operador de salto, Operador de división/unión, Operador de lazo y Operador probabilístico. Cada operador se aplica sobre el 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 todos los operadores. 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 (skip): Una tarea invisible se puede manifestar cuando se produce un salto de una o varias tareas como consecuencia de la ausencia de una opción de selección. La figura 2 refleja el comportamiento registrado en la secuencia de actividades ABD, ACD, AD.

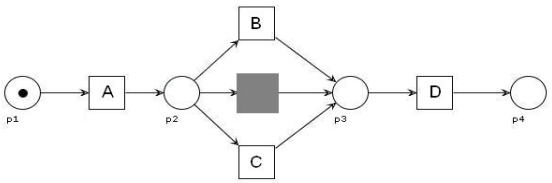

Figura 2 Red de workflow que muestra una tarea invisible en gris según una situación de salto

Esta situación quedaría reflejada en un bloque de construcción de la siguiente forma: C^i_A = [A B - D, A - C D, A - - D].

Al aplicar el segundo paso en la descomposición de C^i_A se obtiene como opciones de una selección los bloques de construcción C^{i+1}_A = [B], C^{i+2}_A = [C], C^{i+3}_A = [-].

Para aplicar el *Operador de salto* se verifica que una matriz con una fila, sea una opción de selección y cumpla que $\forall_{0 \le k < n}\ C^i_A\ [0, k]$ = "-", tal que *n* es la cantidad de columnas. Estas condiciones se cumplen en el bloque de construcción C^{i+3}_A y el bloque de construcción estimado sería, $\hat{C}^{i+3}_A$ = [λ_1], tal que $\lambda_1 \in H$, donde *H* es el conjunto de tareas estimadas.

Operador de división/unión (Split/Join): Una tarea invisible puede manifestarse en una situación en la que hay ausencia del evento de inicio o fin de una opción de selección que contiene una situación de paralelismo. También una tarea invisible puede manifestarse en la situación inversa. La figura 3 refleja un ejemplo de la situación enunciada. El modelo representa el comportamiento registrado en la secuencia de tareas ADE, ACBE, ABCE.

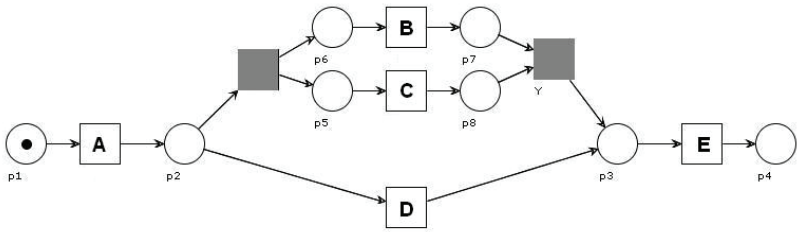

Figura 3 Red de workflow que muestra una tarea invisible en gris según una situación de división/unión

Para aplicar el *Operador de división/unión* se identifican bloques de construcción que se generaron mediante relaciones de selección o paralelismo. Posteriormente se verifica que estos bloques de construcción tengan la primera y la última columna completa, es decir, la columna no tiene símbolos vacíos ("–"). Si alguna de las columnas no es completa entonces se crea una nueva columna con tareas invisibles en la primera o en la última posición según corresponda.

Operador de lazo: Una tarea invisible se manifiesta cuando una tarea o secuencia de tareas se repiten dos o más veces en una traza. La siguiente secuencia de tareas ABBC descrita en una traza, puede reflejarse por los dos modelos representados en la figura 4.

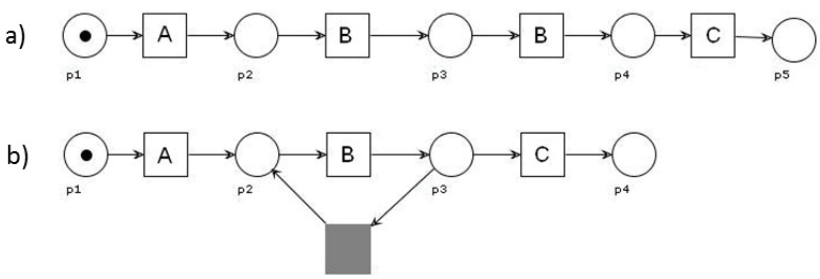

Figure 4 Red de workflow, muestra en a) el uso del constructor de tareas duplicadas y en b) el constructor de tareas invisibles

En la figura 4 a) se emplea el constructor de tareas duplicadas mientras que en b) se usa el constructor de tareas invisibles.

Es necesario resaltar que la mayoría de los algoritmos analizados al enfrentarse a situaciones en las que aparecen tareas duplicadas (lazos de tamaño uno) obtienen un modelo como el de la figura 5 b), el cual, generaliza el comportamiento reflejado en la secuencia de tareas descrita, lo cual resta utilidad al modelo en cuestión. Para aplicar el *Operador de lazo* a partir de un bloque de construcción C_A^i que representa un vector fila, se determina cual es la tarea o secuencia de tareas que aparecen duplicadas. Además, se verifica que las repeticiones no se encuentren separadas por una secuencia de una o más tareas diferentes.

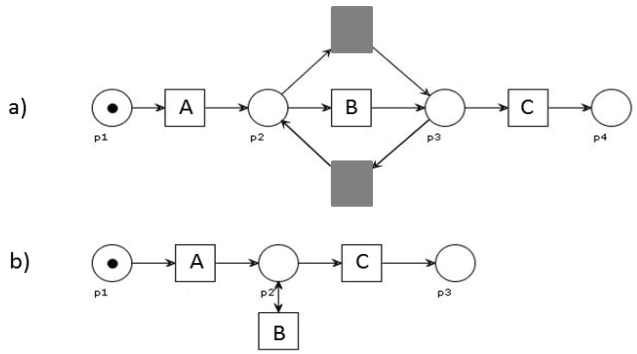

Figura 5 Red de workflow, muestra en a) el uso del constructor de tareas invisibles y en b) el constructor de lazos

El bloque de construcción estimado $\hat{C}^i_A$ es un vector fila que está compuesto por las tareas que conforman a C^i_A y la tarea invisible $\lambda_1 \in H$ que se inserta entre cada una de las secuencias que se repiten. Para el ejemplo anterior quedaría $\hat{C}^i_A$ = [A B λ_1 B C].

El operador que se describe a continuacion permite el tratamiento de una manifestación que no habia sido previamente reportadas en la literatura.

Operador probabilístico: Considérese que existe un proceso en el que las opciones asociadas a una situación de selección son equiprobables. Un ejemplo de esta situación es el proceso P_0 representado en la figura 6a). Las tareas B y F no parecen en las trazas almacenadas. Una muestra representativa de las trazas almacenadas se representa en la tabla 1. 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 (C_1).

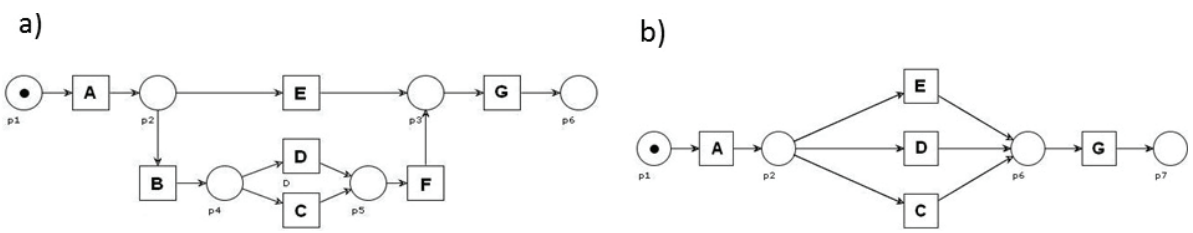

Figura 6 a) Modelo del proceso P_0, b) Modelo del proceso P_1

Tabla 1 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 descubrirían un modelo de procesos (P_1) equivalente al reflejado en la figura 6 b). P_1 difiere significativamente de P_0. La estructura del modelo de procesos descubierto se afectó por la ausencia de información.

Sea M(C_i) la multiplicidad (cantidad de casos) de la clase C_i. Si se analiza la multiplicidad de las clases se puede percibir que M(C_1) representa el 50 % del total de casos, mientras que M(C_2) y M(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 que las opciones poseen la misma probabilidad de ocurrencia. 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. Teniendo en cuenta la situación enunciada se ha desarrollado un operador (Probabilístico) que hace uso de las frecuencias relativas de aparición de las opciones de una selección para estimar la información ausente.

El objetivo del *Operador probabilístico* es encontrar un árbol que se adecue a las frecuencias relativas asociadas a las opciones analizadas y que contenga en cada nodo hoja una de las opciones y como nodos intermedios las actividades invisibles. A partir del árbol obtenido se modifican los bloques de construcción representativos de las opciones.

Después de aplicarle los operadores deseados a cada bloque de construcción las actividades invisibles estimadas se propagan desde los nodos hojas hasta la raíz del árbol, considerando, las referencias establecidas durante su construcción.

Resultados experimentales y discusión

Se desarrolló una aplicación informática que permite realizar la estimación de información ausente a partir de las trazas usadas en la minería de procesos. La herramienta permite utilizar un registro de eventos en el formato XES o MXML como entrada y como salida se genera un fichero con la información estimada y en formato XES o MXML según corresponda.

Para evaluar la propuesta es necesario mostrar que la estimación realizada permite descubrir modelos de procesos más estructurados y comprensibles. En consecuencia, es necesario determinar la forma en la que se debe medir la estructura y comprensión de los modelos descubiertos. Para ello se seleccionan un conjunto de métricas que permiten evaluar las afectaciones de la ausencia de información sobre la estructura y comprensión del modelo descubierto [21-23]. Se seleccionan dos métricas para medir las afectaciones que provoca la ausencia de información sobre la estructura del modelo descubierto, Fitness Unsatisfied y Fitness Unhandled [22], estas se agrupan originalmente como parte de la métrica Fitness.

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. Ambas métricas están contenidas en la métrica ETCPrecision [24]. 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. Se verifica también que en el modelo descubierto no existan problemas asociados a la dimensión de Estructura. En este sentido se utiliza Improved Structural Appropriateness (ISA) [25].

Para el experimento se definen cuatro grupos de procesos y tres momentos. Cada grupo

esta 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 eventos en su estado original, con ausencia de información y con información estimada. La tabla 2 muestra el diseño experimental realizado.

Tabla 2 Diseño experimental propuesto

Grupos	*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}

G: Grupo de participantes. R: Asignación al alzar. X: Tratamiento o estímulo. O: Observación.

En este caso cada grupo está compuesto por 45 procesos, 15 asociados a cada momento (Original, Ausencia de Información e Información estimada). X_1 se corresponde con la extracción de información del registro de eventos analizado en cada grupo durante el primer momento. X_2 se corresponde con la aplicación del modelo propuesto para la estimación de información ausente.

Se han empleado cuatro procesos diferentes (reflejan diferentes patrones de flujo de trabajo, diferentes cantidades de casos, eventos y clases de eventos), uno para cada grupo. Se aplicaron dos algoritmos de descubrimiento, Alpha++ y ILP [26, 27]. A partir de su aplicación se evaluaron las cinco métricas enunciadas, 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 eventos 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 eventos 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 eventos con ausencia de información se transformó usando la aplicación informática desarrollada y se obtuvo un nuevo registro de eventos con la información estimada. En consecuencia, cada observación asociada a este momento contiene 15 registros de eventos.

Se realizan un conjunto de pruebas con el objetivo de comparar las observaciones en los diferentes momentos. Se esperaba detectar afectaciones en los datos asociados al segundo de los momentos teniendo como referencia los datos asociados al primer momento. También se esperaba percibir una mejora en el tercer momento teniendo como referencia el segundo. La primera de las pruebas 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 [28]. Esta evaluación detectó diferencias

significativas (significación 0.000 < 0.01) entre las observaciones (por ejemplo O_1, O_2 y O_3) en cada evaluación asociada a las métricas utilizadas. Los valores observados decrecen para el segundo de los momentos (Ausencia de información) y aumentan en el último (Información estimada).

Luego se realizaron comparaciones por pares en un grupo utilizando el test no paramétrico de signos con rangos de Wilcoxon, con el objetivo de entender las diferencias detectadas en la primera prueba [28]. Se analizan los datos correspondientes al primer y segundo momento, por ejemplo O_1 y O_2, y se detectó de manera significativa (significación 0.000 < 0.01) 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, y se detectó de manera significativa (significación 0.000 < 0.01) 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, y se detectó un empate entre los valores (significación 1.000), lo cual no revela diferencias significativas.

Este resultado se ajusta a lo esperado y evidencia la efectividad de la propuesta.

La figura 7 muestra los valores de los rangos medios para cada grupo al aplicar ambos algoritmos y medir las métricas Fitness Unsatisfied y Fitness Unhandled. Esta figura permite apreciar los resultados obtenidos. En el caso de las métricas Precision y Non Fit Traces se obtuvo un resultado semejante al reflejado en la figura 7. Para la métrica ISA se obtuvo en todos los casos el mejor resultado posible.

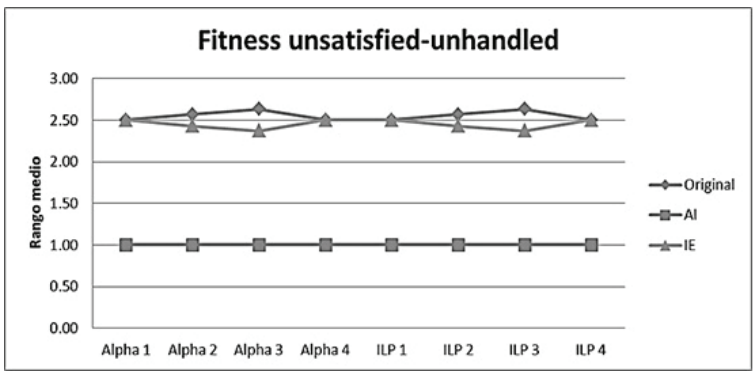

Figura 7 Evaluación de las métricas Fitness Unsatisfied y Fitness Unhandled

Además se realizan análisis transversales que permitieron comparar las observaciones de los diferentes grupos en un mismo momento, por ejemplo O_1, O_4, O_7 y O_{10}. El objetivo de estas pruebas es evidenciar que con independencia de las características del registro de evento empleado la aplicación del modelo propuesto permite recuperar la información ausente y en consecuencia disminuir las afectaciones que se originan. En estos casos, se detectó diferencias significativas (significación 0.000 < 0.01) al aplicar el test de Kruskal-Wallis [28]. La diferencia entre los valores está 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 realizaron las comparaciones por pares utilizando el test de Mann-Whitney [28].

En este caso se consideran los diferentes grupos en todos los momentos y utilizando los dos algoritmos de descubrimiento.

A partir de los resultados obtenidos se demuestra que para las medidas analizadas la propuesta

soluciona las afectaciones provocadas por la ausencia de información sobre el registro de eventos original, sin importar el algoritmo de descubrimiento utilizado. Al solucionarse las afectaciones de la ausencia de información sobre el registro de eventos se elimina la ambigüedad en la interpretación de las trazas hechas por cada algoritmo de descubrimiento.

La evaluación de la métrica ISA para todos los modelos obtenidos es 1.0. Esto revela la propuesta, independientemente del momento, no introduce en el modelo descubierto problemas estructurales y facilita la comprensión de este.

Conclusiones

Los diferentes algoritmos de descubrimiento desarrollados hasta el momento presentan problemas al manejar diferentes situaciones en la que existe ausencia de información. Estos problemas provocan afectaciones en la estructura del modelo descubierto y dificultan la comprensión del proceso analizado. La propuesta realizada permite hacer la estimación de información ausente y facilita el entendimiento del contexto en el que se manifiesta dicha ausencia. Para la estimación se propusieron un conjunto de pasos que permitieron la descomposición del proceso analizado, la detección de las diferentes manifestaciones de ausencia de información, la estimación de la información ausente y la generación en correspondencia de un nuevo registro de evento. Considerando los pasos descritos se desarrolló una herramienta informática que permitió la aplicación y evaluación de la propuesta. Los resultados obtenidos son satisfactorios y constituyen la evidencia de la efectividad de la propuesta en la reducción de las afectaciones que genera la ausencia de información sobre el registro de eventos.

Referencias

1. C. Hentrich, Z. Uwe. *Service Integration Patterns for Invoking Services from Business Processes*. In Proceedings of 12th European Conference on Pattern Languages of Programs (EuroPLoP 2007). Irsee, Germany. 2007. pp. 1-45.
2. R. Agrawal, D. Gunopulos, F. Leymann. *Mining Process Models from Workflow Logs*. In 6th International Conference on Extending Database Technology. Ed. Springer-Verlag. London, UK. 1998. pp. 469-483.
3. J. Cook, A. Wolf, "Discovering Models of Software Processes from Event-Based Data," *ACM Transactions on Software Engineering and Methodology*. Vol. 7. 1998. pp. 215-249.
4. W. Van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Ed. Springer-Verlag. Heidelberg, Germany. 2011. pp. 352.
5. W. Van der Aalst, A. Adriansyah, A. de Medeiros, F. Arcieri, T. Baier, T. Blickle, J. Chandra. "Business Process Management Workshops". *Lecture Notes in Business Information Processing*. Vol. 99. 2011. pp. 167-168.
6. C. Günther, W. van der Aalst. "Fuzzy Mining: Adaptive Process Simplification Based on Multi-Perspective Metrics". *Lecture Notes in Computer Science*. Vol. 4714. 2007. pp. 328-343.
7. W. van der Aalst, A. Weijters, "Process Mining: A Research Agenda." *Special Issue of Computers in Industry*. Vol. 53. 2004. pp. 231-244.
8. R. Farkhady, S. Aali. "A Probabilistic Approach for Process Mining in Incomplete and Noisy Logs". *Lecture Notes in Engineering and Computer Science.* Vol. 2188. 2011. pp. 415-420.
9. A. de Medeiros. *Genetic Process Mining*. PhD. Thesis. Technische Universiteit Eindhoven. Eindhoven, Netherlands. 2006. pp. 384.
10. A. Tiwari, C. Turner, B. Majeed. "A review of business process mining: state-of-the-art and future trends." *Business Process Management*. Vol. 14. 2008. pp. 5-22.

11. A. Adriansyah, B. van Dongen, W. van der Aalst. *Conformance Checking Using Cost-Based Fitness Analysis*. In EDOC '11 Proceedings of the 2011 IEEE 15th International Enterprise Distributed Object Computing Conference, Helsinki, Finland. Ed. IEEE Computer Society. Washington DC, USA. 2011. pp. 55-64.

12. J. Muñoz, J. Carmona. *A fresh look at precision in process conformance*. Ed. Springer-Verlag. Hoboken, NJ, USA. 2010. pp. 211-226.

13. W. Van der Aalst. *On the Representational Bias in Process Mining*. In Proceedings of the 2011 IEEE 20th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. Paris, France. 2011. pp. 2-7.

14. M. Song, W. Van der Aalst. *Supporting process mining by showing events at a glance.* In K. Chari & A. Kumar (Eds.). Proceeding of the Seventeenth Annual Workshop on Information Technologies and Systems (WITS'07). Montreal, Canada. December 8-9. 2007. pp. 139-145.

15. R. Bose, W. Van der Aalst. "Process diagnostics using trace alignment: Opportunities, issues and challenges." *Inf. Syst*. Vol. 37. 2012. pp. 117-141.

16. L. Ly, C. Indiono, J. Mangler, S. Rinderle. "Data Transformation and Semantic Log Purging for Process Mining". *Lecture Notes in Computer Science*. Vol. 7328. 2012. pp. 238-253.

17. W. Van der Aalst, H. Beer, B. Dongen. "Process mining and verification of properties: An approach based on temporal logic." *Lecture Notes in Computer Science*. Vol. 3761. 2005. pp. 130-147.

18. W. Van. der Aalst, B. Van Dongen, C. Günther, A. Rozinat, E. Verbeek, A. Weijters. *ProM: the process mining toolkit*. In A.K. Alves de Medeiros & B. Weber (Eds.), Proceedings of the BPM 2009 Demonstration Track. Ulm, Germany. 2009. pp. 1-4.

19. H.. Verbeek, J. Buijs, B. van Dongen, W. van der Aalst. *ProM6: The Process Mining Toolkit*. Proceeding of the Proceedings of the Business Process Management 2010 Demonstration Track. Hoboken NJ, USA. 2010. Vol. 615. pp. 34-39.

20. R. Yzquierdo, R. Silverio, M. Lazo, A. Torres. "Diagnóstico de proceso basado en el descubrimiento de subprocesos." *Revista Ingeniería Industrial*. Vol. 33. 2012. pp. 133-141.

21. A. Rozinat, W. Van der Aalst. "Conformance checking of processes based on monitoring real behavior." *Inf. Syst*. Vol. 33. 2008. pp. 64-95.

22. R. Van Arendonk. *A Benchmark Set for Process Discovery Algorithms*. Master Thesis. Eindhoven University of Technology. Eindhoven, Netherlands. 2011. pp. 69.

23. J. Weerdt, M. Backer, J. Vanthienen, B. Baesens. "A critical evaluation study of model-log metrics in process discovery". *Lecture Notes in Business Information Processing*. Vol. 66. 2011. pp. 158-169.

24. A. Adriansyah, B. Van Dongen, W. Van der Aalst. "Towards Robust Conformance Checking". *Lecture Notes in Business Information Processing*. Vol. 66. 2011. pp. 122-133.

25. A. Rozinat, W. Van der Aalst. "Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models." *Lecture Notes in Computer Science*. Vol. 3812. 2006. pp. 163-176.

26. L. Wen, J. Wang, W. Van der Aalst, Z. Wang, J. Sun. "A Novel Approach for Process Mining Based on Event Types." *Journal of Intelligent Information Systems*. Vol. 32. 2009. pp. 163-190.

27. J. Werf, B. Dongen, C. Hurkens, A. Serebrenik. *Process Discovery Using Integer Linear Programming*. In Proceedings of the 29th international conference on Applications and Theory of Petri Nets. Xi'an, China. 2008. pp. 368-387.

28. W. Conover. *Practical nonparametric statistics*. 2nd ed. Ed. John Wiley & Sons. New York, US. 1998. pp. 332-467.