Trabajo de Fin de Grado

Inteligencia Empresarial basada en técnicas de minería de datos y de procesos

Fusión Minería de procesos evolutiva y patrones secuenciales frecuentes

Business Intelligence by mixing Process Mining and Data Mining techniques

Mining merger of evolutionary processes and frequent sequential patterns

Daniel Nicolás Fernández del Castillo Salazar

Departamento de Ingeniería Informática.

Escuela Superior de Ingeniería y Tecnología

La Laguna, 7 de julio de 2015

D. ***Pedro Antonio Toledo Delgado,*** *con N.I.F. 45.725.874-B profesor ayudante adscrito al Departamento de Ingeniería Informática de la Universidad de La Laguna, como tutor.*

D. ***Vanesa Muñoz Cruz****, con N.I.F. 78.698.687-R profesora ayudante Doctor adscrita al Departamento de Ingeniería Informática de la Universidad de La Laguna, como cotutor.*

C E R T I F I C A (N)

Que la presente memoria titulada:

"Inteligencia Empresarial basada en técnicas de minería de datos y de procesos.
Fusión Minería de procesos evolutiva y patrones secuenciales frecuentes"

ha sido realizada bajo su dirección por D. ***Daniel Nicolás Fernández del Castillo Salazar****, con N.I.F. 78.859.108-C.*

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 7 de julio de 2015.

Agradecimientos

Quiero agradecer a mi tutor Pedro, que me ha ayudado durante todo el proceso y desarrollo de este proyecto y a la gente que me ha apoyado y acompañado durante todo este tiempo.

Licencia

Resumen

En el proyecto se desarrollará una herramienta que integre técnicas de dos campos de la ciencia de la computación que están involucrados en la Inteligencia Empresarial (BI)[1], *estos son: la* Minería de Procesos*[2] y la* Minería de datos[3]. *Ambos campos están dedicados a la extracción de información, pero la primera suele actuar sobre registros tipo logs de eventos para construir modelos en forma de flujos de trabajo y la segunda frecuentemente utiliza fuentes de datos no estructuradas en forma de conjuntos de características y construye modelos diversos como: clasificadores, regresores, agrupaciones y patrones.*

Concretamente el objetivo del proyecto es mejorar y hacer útil los datos en el contexto de la inteligencia empresarial, mejorando los modelos de procesos extraídos con los resultados de la minerías de patrones frecuentes.

Para realizar esta tarea se utilizará un algoritmo genético, el cual a partir de un log obtendrá como resultado una población de posibles soluciones representadas en un workflow en forma de red petrinet.

Por otro lado se utiliza el modelo General Sequential Patterns[4](en adelante GSP) de Weka[5] *que descubre aquellas secuencias que se repiten con más frecuencia en la población de soluciones dada. Con el GSP obtendremos las secuencias frecuentes, que podrán ser utilizadas para realizar un score que permita realizar un ranking de esa población de soluciones.*

El resultado será una población de grafos del proceso ordenada según una función de score que valora el encaje de secuencias frecuentes en dichos modelos.

Este método mejora el ajuste de los workflows obtenidos a ciertas características de los datos originales, como es, el respeto de la secuenciación de eventos.

Palabras clave: *Minería de Datos, Minería de Procesos, Inteligencia de Negocios (BI), ProM, Weka, General Sequential Patterns, Genetic miner.*

Abstract

The project aims to develop a tool that integrates techniques from two different fields of computer science who are involved in business intelligence. These are: Process mining and data mining. Both fields are dedicated to extract information, but the first type usually acts on Event log records to build models as workflows and the second frequently uses unstructured data sources as feature sets and builds various models as classifiers, regressors, clusters and patterns.

Specifically, the aim of this thesis is to improve results and make the data useful in the context of business intelligence, enhancing process models extracted the results of the mining extraction of common frequencies.

To do this a genetic algorithm is used. It takes an event log as input and obtains a population of possible workflow solutions represented by PetriNets .

Aditionally General Sequential Patterns (GSP) are extracted using Weka, discovering those frequent sequences in the event log. With those results, a new score function for the genetic algorithm is defined. This score will allow a better ranking of the population of solutions.

This method improves the fit of the workflows obtained with certain features of the original data, such as respecting the sequencing of events..

Keywords: *Data Mining, Process Mining, Business Intelligence (BI), ProM, Weka, General Sequential Patterns (GSP), Genetic miner*

Índice General

Índice de figuras

Índice de tablas

Capítulo 1. Introducción

En el mundo actual tras el desarrollo de internet, las redes sociales, etc. se genera constantemente una cantidad enorme de datos, dando lugar a una fuente inmensa de información pero que por sí sola, sin un procesamiento adecuado, es difícil de utilizar para sacar alguna conclusión. Por esto se han desarrollado diversas técnicas que faciliten el procesamiento de datos de la forma más automatizada posible para que en las diversas organizaciones y empresas puedan extraer conclusiones y tomar decisiones a partir de esos datos.

La gran cantidad de datos existente en cualquier dispositivo electrónico, hace que de ser un producto, este pase a convertirse en una materia prima que necesita ser explotada para poder obtener un activo final, que es el conocimiento. Hoy en día el conocer la información da poder a cualquier organización o empresa, por ello los gobiernos e instituciones invierten gran cantidad de dinero en las TICs, y esto se encuentra a la vez relacionado con diversos campos como son: La minería de datos, la de procesos y la inteligencia empresarial.

La minería de datos es un proceso de estudio automático o semi-automático de grandes cantidades de datos. Para lograr esto utiliza análisis matemático, estadística y algoritmos de búsqueda basados o aproximados a la inteligencia artificial con el fin de deducir o descubrir patrones o tendencias ocultos en los datos y permitir con ello la toma de decisiones a partir del conocimiento obtenido.

A continuación se mencionan algunos modelos de minería de datos y se citan algunos ejemplos concretos para los que son habitualmente utilizados:

- *Pronóstico: cálculo de las ventas y predicción de las cargas del servidor o del tiempo de inactividad del servidor.*
- *Riesgo y probabilidad: elección de los mejores clientes para la*

distribución de correo directo, determinación del punto de equilibrio probable para los escenarios de riesgo, y asignación de probabilidades a diagnósticos y otros resultados.

· *Recomendaciones: determinación de los productos que se pueden vender juntos y generación de recomendaciones.*

· *Búsqueda de secuencias: análisis de los artículos que los clientes han introducido en el carrito de la compra y predicción de posibles eventos.*

· *Agrupación: distribución de clientes o eventos en grupos de elementos relacionados, y análisis y predicción de afinidades.*

La minería de procesos es una disciplina de investigación joven y surge a partir del modelado de datos en el entorno empresarial, con el fin facilitar la extracción de datos en forma de modelos de flujos de proceso.
El reciente interés en esta disciplina, se debe a dos razones fundamentales. Por un lado cada vez se registran mas eventos que proporciona información detallada sobre diferentes procesos y por otro lado surge la necesidad de mejorar y apoyar los procesos de negocios competitivos que forman parte de entornos dinámicos y cambiantes.

La minería de procesos incluye:

- *El descubrimiento automático de procesos. Por ejemplo: extraer modelos de procesos a partir de un registro de eventos.*
- *La verificación de conformidad, como monitorear desviaciones al comparar el modelo y el registro de eventos.*
- *La minería de redes sociales/organizacionales.*
- *La construcción automática de modelos de simulación, la extensión de modelos y la reparación de modelos.*
- *La predicción de casos, y las recomendaciones basadas en historia.*
  Resumiendo, la minería de procesos es una técnica de administración de procesos que permite analizar los procesos de negocio de acuerdo con un registro de eventos o log, para descubrir, monitorear y mejorar los mismos. [5-7] En la

actualidad, existen diferentes herramientas libres y comerciales para la minería de datos, como: Weka, Rapid Miner, Powerhouse y SAS. Otras tantas para la minería de procesos, entre las cuales, podemos destacar: ProM, Process Mining, Disco, ARIS Process Performance Manager. Sin embargo, no hay ninguna herramienta que explote ambas técnicas sobre los mismos datos y que ofrezcan por ejemplo: flujos de trabajos enriquecidos con la estructura de la información que está detrás de los procesos que intervienen en los workflows, o la utilización de los registros de eventos para compararlos con modelos probabilísticos.

En resumen, la ***minería de datos*** *y la* ***minería de procesos*** *no es más que una forma de extraer información y mediante algoritmos y estadísticas poder dotar de utilidad a datos que en soledad carecen de valor o significado. Ambos campos están dedicados a la extracción de información, pero la primera suele actuar sobre registros tipo logs de eventos para construir modelos en forma de flujos de trabajo y la segunda frecuentemente utiliza fuentes de datos no estructuradas en forma de conjuntos de características y construye modelos diversos como: clasificadores, regresores, agrupaciones y patrones.*

1.1 Antecedentes

En este proyecto, se partirá de un trabajo previo (Rendon 2014) *con el fin de aprovechar el trabajo ya realizado, y poder alcanzar el objetivo del proyecto.*

En el se desarrolló una herramienta que aproxima entre sí las técnicas, de los dos campos de la ciencia minería de datos y minería de procesos.
Para lograr esto se combinaron dos modelos de cada campo de forma que se complementasen el uno al otro. El modelo General Sequential Patterns (en adelante GSP) de Weka que descubre las secuencias más frecuentes de una fuente de datos de secuencias. Y el modelo de Petrinet de ProM que genera un workflow del proceso que es deducido de un fichero log. Ambos algoritmos tienen como fuentes de información el mismo fichero log, pero para que Weka pueda trabajar con él, se hace necesario una conversión previa del log por medio de una herramienta que facilita Weka.

El resultado de esto es un modelo compuesto por el grafo del proceso y una lista de caminos del grafo más frecuentes. Este modelo facilita la detección de cuellos de botella, el rediseño del modelo de trabajo para mejorar su eficiencia, o añadir información extra al flujo de proceso como los costes o las personas involucradas en las actividades más frecuentes.

1.2 Herramientas Utilizadas

1.2.1 ProM

*El prototipo de plugin se desarrolló en la herramienta **ProM** que es un framework extensible el cual soporta una variedad de técnicas de minería de procesos en forma de plugins. ProM 6 es distribuido en partes, pata ofrecer una mayor flexibilidad. Por una parte está el paquete ProM core con licencia GPL y por el otro están los paquetes ProM plugins que están distribuidos normalmente bajo licencia L-GPL.*
Actualmente están disponibles más de 120 paquetes que contienen más de 500 plugins.

Breve Historia:

En 2002, existían una gran variedad de herramientas básicas de minería de procesos. Todas ellas incapaces de procesar datasets de gran tamaño y sólo ejecutaban un determinado algoritmo. Evidentemente, no tiene mucho sentido desarrollar una herramienta, por algoritmo. Por este motivo, se crea el framework extensible ProM. El objetivo de la primera versión era proveer un soporte común de carga, y filtrado de los eventos de los logs y la visualización de resultados.

En 2004, se lanza ProM 1.1 la primera versión completa y funcional, con un total de 29 plug-ins; 6 de minería, 7 de análisis, 9 de exportación, 4 de importación y 3 de conversión.

En 2006, se libera la versión 4.0 con 142 plug-ins.

En 2009, ProM 5.2 con 286 plugins, Claramente ProM se convierte en un estándar de la minería de procesos, con grupos de

investigación repartidos por el mundo, contribuyendo a su desarrollo.

En 2010, nace ProM 6; con una interfaz de usuario reimplementada para lidiar con varios plugins, logs y modelos al mismo tiempo. Además, incluye un administrador de paquetes para poder añadir, eliminar y actualizar los plug-ins. De esta forma se evita tener instaladas las funcionalidades que no necesites.

1.2.2 WeKa

Para obterner las secuencias frecuentes se usó ***Weka*** *(Waikato Environment for Knowledge Analysis - Entorno para Análisis del Conocimiento de la Universidad de Waikato). Es una plataforma de software libre distribuido bajo licencia GNU-GPL, para aprendizaje automático y minería de datos escrito en Java y desarrollado en la Universidad de Waikato.*

Breve Historia:

En 1993, la Universidad de Waikato de Nueva Zelanda inició el desarrollo de la versión original de Weka (en TCL/TK y C).

En 1997, se decidió reescribir el código en Java incluyendo implementaciones de algoritmos.

En 2005, Weka recibe de SIGKDD (Special Interest Group on Knowledge Discovery and Data Mining) el galardón "Data Mining and Knowledge Discovery Service".

En 2006, Pentaho Corporation adquirió una licencia exclusiva para usar Weka para Inteligencia de negocio (Business Intelligence), dando lugar al componente de minería de datos y análisis predictivo del paquete de software Pentaho Business Intelligence.

1.3 Descripción del entorno de trabajo

En este apartado se describen las herramientas utilizadas durante todo el proyecto. Es importante buscar que las versiones de las herramientas utilizadas sean compatibles, porque su integración y su buen funcionamiento, depende utilizar la combinación versiones adecuadas. [12]

· *Prom 6.1* · *Java JDK 1.6_45* · *Subclipse 1.1.0* · *Trello*

· *Weka 3.6* · *Eclipse Luna v4.4.2* · *ObjectAid 1.1.6* · *Github*

1.4 Objetivo general

El objetivo que se persigue en este proyecto es llevar más allá la aproximación realizada en (Rendon 2014) *y combinar en un solo modelo resultante las técnicas seleccionadas de los campos de Minería de Procesos y Minería da Datos.*

Con la unión de los resultados obtenidos desde la perspectiva de ambos campos en un solo modelo permitirá obtener workflows válidos en términos de soporte a patrones frecuentes obtenidos de los mismos datos de partida y a partir de ahí sacar conclusiones sobre datos que pueden ayudar en el campo de la Inteligencia Empresarial.

1.5 Objetivos Específicos

- *Analizar, diseñar y buscar los algoritmos en forma de plugins que permitan ser complementados con información extraída mediante herramientas de otros campos y que permitan la interpretación y predicción de la información, utilizando resultados de minería de procesos como base.*
- *Crear un prototipo de plug-in.*
- *Implementar la aproximación analizada dentro del prototipo.*
- *Realizar un proceso de validación con datos reales públicos o sintéticos*

1.6 Alcance

Se pretende desarrollar un prototipo de plug-in de Weka para la plataforma ProM que fusione las técnicas de minería de datos y de proceso la cual permita obtener un único modelo resultante.

Para lograr esta fusión se deberá investigar que tipo de técnicas de minería de procesos y cuales de datos puede complementarse y fusionarse de forma que se justifique la unión de las mismas.

El resultado de este plugin debe ser una herramienta que permita obtener, visualizar y analizar, para un conjunto de datos dados información estructurada y enriquecida con resultados del análisis de datos de ambas disciplinas.

1.7 Metodología

La metodología que se ha utilizado para la elaboración de este proyecto es la metodología ágil conocida como ***Kanban*** *y a su vez combinando aspectos de la metodología* ***XP****.*

Para ello se ha usado la herramienta Trello que permite desarrollar con facilidad este tipo de metodologías.

Figura 2.1. *Herramienta Trello que permite llevar a cabo la metodología citada.*

Capítulo 2. El modelo

2.1 Finalidad y justificación del proyecto

Debido a la gran cantidad de datos, registros e información en general que se mueve en el mundo actual, se hace necesario modelar y procesar estas fuentes para dotar de un significado aprovechable por diferentes entidades, por esto existen diferentes disciplinas y ciencias que investigan como hacerlo.

En este proyecto se intenta aportar una vía más para el aprovechamiento de la información demostrando que ambos campos de la ciencia de minería de datos y de procesos son compatibles y que dicha unión devuelve un modelo más completo y adaptado que si por el contrario usasen ambas técnicas por separado.

Esto podría permitir deducir y extraer información más especifica sobre un conjunto de datos, de forma que ante una entrada de datos para unos patrones dados se obtenga un flujo de trabajo lo más cercano al resultado deseado e incluso con ello la posibilidad de prever sucesos.

La forma de lograr este objetivo sería consiguiendo un punto o nexo en común entre las dos disciplinas, que permitan integrarlas en un único modelo.

Tras una investigación sobre los diferentes algoritmos que podrían resultar útiles para este proyecto, se ha propuesto un algoritmo genético, debido a sus características evolutivas que permiten obtener nuevas poblaciones de soluciones a partir de una población inicial. Por otra parte se propuso un algoritmo capaz de detectar secuencias de tareas frecuentes conocido como el GSP de WeKa.

Se propone combinar ambos algoritmo entre sí mediante una función score que permita recalcular el original del algoritmo genético.

Esto permite obtener un modelo en el cual a partir de una fuente de entrada de datos, sea capaz de resolver un conjunto de posibles flujos de trabajo para unos patrones dados, que tendrán a su vez un score para establecer en un ranking cual es la solución más idónea para nuestra entrada de datos.

2.2 Petrinets

La red de Petri es uno de los modelados de procesos mas antiguos, usados para la representación de la teoría de autómatas. Un petrinet se compone de un grafo que contiene lugares, transiciones, arcos dirigidos y marcas o fichas que ocupan posiciones dentro de los lugares. En resumen su uso sirve para representar gráficamente sistemas de eventos discretos concurrentes de forma automatizada.

Su conjunto de reglas son las siguientes:

- *Los arcos conectan lugar con transición y una transición a un lugar.*
- *No puede haber arcos entre lugares ni entre las transiciones.*
- *Los lugares se componen de número finito o infinito contable de marcas.*
- *Las transiciones consumen marcas de una posición de inicio y producen marcas en una posición de llegada.*
- *Si una transición tiene marcas en todas sus posiciones de entrada se dice que está habilitada.*

Las redes de Petri se usan en diversas áreas, pero las más habituales son: el análisis de datos, el diseño de software, la evaluación de fiabilidad, elaboración de flujos de trabajo (workflow) y la programación concurrente.

En este proyecto la red Petri se usa para generar el workflow que describe el funcionamiento de una aplicación o empresa a partir de registros logs.

2.3 Algoritmo Genetic Miner

Los algoritmos genéticos son algoritmos que imitan los procesos biológicos de la evolución para encontrar modelos de comportamiento. Los algoritmos genéticos son procesos iterativos en los que sobre una población inicial van mutando, y combinándose con respecto a una medida de fitness que marcara si el modelo evolucionado ha mejorado o empeorado. Cada nueva hipótesis generada es llamada nueva generación. Las iteraciones se suceden hasta que

un criterio de parada se produce, que normalmente viene dado por un criterio de fitness alcanzado o numero de iteraciones máximo.

El Genetic Process Mining, toma como población inicial logs de registro que se combinan y mutan pseudoaleatoriamente para mejorar en cada iteración. Estas dos operaciones se efectúan sobre una generación para evolucionar hacia otra generación. La operación de combinación mezcla de una manera pseudoaleatoria dos candidatos para evolucionar a otro nuevo candidato. Por otro lado la operación de mutación añade, también de modo pseudoaleatoria, eventos a una población para generar una nueva generación. Las mejores evoluciones generadas serán aquellos candidatos que se mantendrán en la siguiente generación.

Para comprobar si el de nuevo candidato es mejor o peor que el anterior se utiliza como modelo de fitness la siguiente función:

$$Fitness(i) = Exactitud(i) - k * Precision(i)$$

La medida de fitness se basa en dos conceptos, la Exactitud y la Precisión.

La Exactitud es una medida de como el candidato acepta las muestras de entrada. cuanto mayor sea la Exactitud, mejor será el fitness del modelo. La Exactitud se calcula utilizando la siguiente formula:

$$Exactitud = \frac{E_C - \frac{E_F}{T_T - T_F + 1} - \frac{E_E}{T_T - T_F + 1}}{E_T}$$

Siendo EC el número de eventos aceptados por el candidato, EF el número de eventos no aceptados por el candidato, EE el número de eventos extra de nodos para el candidatos e inexistentes en la muestra, TT el número de muestras totales y TF el número de trazas que han fallado.

Por otro lado, la Precisión mide la información extra que se encuentra para el candidato. Cuanto mayor sea la Precisión peor será el fitness del modelo. La precisión se calcula utilizando la siguiente formula:

$$Precision(i) = \frac{EventosAccedidos_i}{max(EventosAccedidos_{1..n})}$$

Donde Eventos Accedidos$_i$ son el numero de nodos que han sido accedidos durante el proceso de verificación de la instancia i de la muestra.

El algoritmo de Genetic Process Mining utilizado (Process Tree) genera un grafo de dependencias que representa el WF, lo que facilita le creación de WF legibles.

Aquí podemos ver un pseudocódigo del funcionamiento del algoritmo genético.

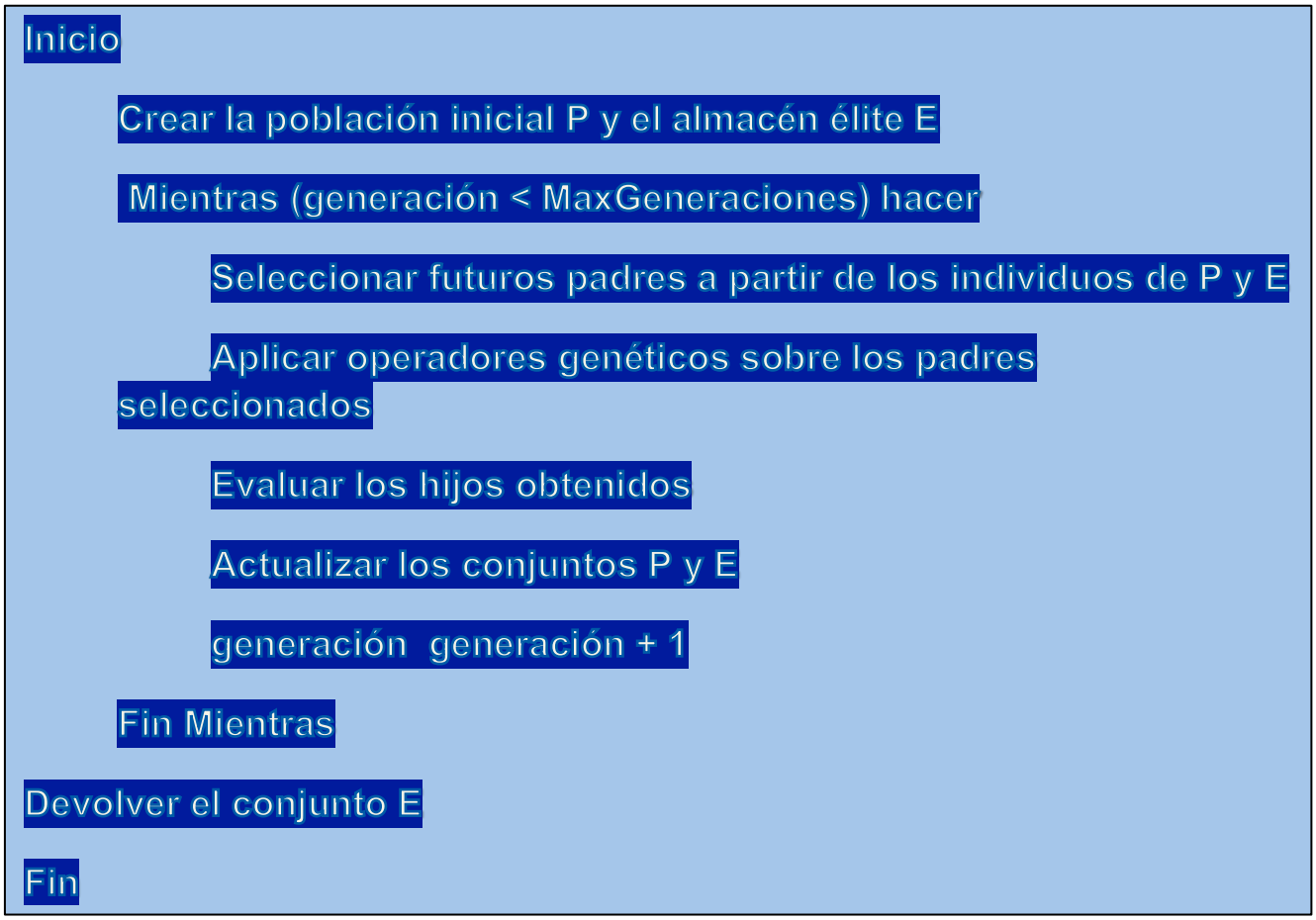

Figura 4.1. *Pseudocódigo del algoritmo genético*

2.4 Algoritmo GSP

El algoritmos GSP (Generalized Sequential Patterns) hace uso de la estrategia apriori para obtener el conjunto de las secuencias frecuentes. Una secuencia es una sucesión de acciones o tareas compuesta por lo que el algoritmo denomina itemsets, el proceso para hallar las secuencias candidatas, está dividido en dos fases:

Fase de enlace: *es la encargada de generar el conjunto de secuencias candidatas. La generación de este conjunto es mediante la intersección del conjunto de secuencias frecuentes de la iteración anterior, con él mismo.*

Fase de poda: *es la encargada de determinar cuáles de las secuencias candidatas puede ser eliminada antes de realizar el recorrido de la base de datos para el conteo de las mismas.*

El proceso explicado anteriormente también es usado por varios de los algoritmos de detección de secuencias frecuentes.

Los pasos usados por el algoritmo GSP para la obtención del conjunto de secuencias candidatas:

Fase de enlace: *las secuencias candidatas se generan al hacer una unión del conjunto L_{k-1} (conjunto de secuencias frecuentes) con él mismo. Una secuencia S_1 se une con una secuencia S_2, si luego de eliminar el primer ítem de S1 es igual a S_2 después de haberle eliminado el último ítem. La secuencia candidata es la formada por la secuencia S_1 adicionándole el último ítem de S_2. Este ítem se adiciona como un nuevo elemento si está como un elemento separado de S_2 y como parte del último elemento en caso contrario.*

Fase de poda: *eliminar del conjunto de secuencias candidatas aquellas secuencias que tengan una subsecuencia continua que no aparezca entre las secuencias frecuentes del conjunto L_{k-1}.*

3-secuencias frecuentes	Secuencias, después de la fase de enlace	Secuencias, después de la fase de poda
$\langle(b)(d,e)\rangle$	$\langle(b,c)(d,e)\rangle$	$\langle(b,c)(d,e)\rangle$
$\langle(c)(d,e)\rangle$	$\langle(b,c)(d)(f)\rangle$	
$\langle(b,c)(d)\rangle$		
$\langle(b,c)(e)\rangle$		
$\langle(b)(d)(e)\rangle$		
$\langle(b)(d)(f)\rangle$		

Tabla 2.1. *La Tabla muestra las secuencias frecuentes y las obtenidas después de la fase de enlace y de la fase de poda*

Se puede observar que en la fase de poda se elimina la secuencia ⟨(b, c)(d)(f)⟩ porque no cumple con la condición que se plantea, ya que la sub-secuencia ⟨(c)(d)(f)⟩ no aparece dentro del conjunto de secuencias frecuentes.

2.5 Función Score

La función score se mueve en un intervalo de entre '0' y '1', el valor más cercano a '0' será el óptimo y el más cercano al 1 el peor.

Cada vez que obtenemos una coincidencia, el valor del score se usa como valor de fitness en detrimento del propio del algoritmo genético.

Esto es ventajoso ya que con esto conseguimos que la población resultante sea menos general y se concrete más sobre la información que estamos trabajando.

Para calcular el fitness se usan los siguientes elementos

1. *Precision: se calcula de dividiendo el número de nodos usados entre los totales, en el caso ideal de que todos los nodos sean usados será 0.*
2. *Simplicity: discrimina negativamente a los candidatos con nodos repetidos.*
3. *Generalization: Dependiendo del tipo de nodos OR,XOR,AND o LOOP*

se asignan unos pesos según se recorran o no sus diferentes hojas.

4. *GPS: Cuanto más nodos que estén en el GPS se usen mejor será la puntuación.*

Con todo esto se pondera cada uno de los valores con el peso que se le asigna y se normaliza dividiendo la suma entre el tamaño de la secuencia.

2.6 Integración de los modelos

Los modelos antes explicados generan los siguientes resultados: Población de soluciones provenientes del algoritmo genético ProcessTree y las secuencias de tareas frecuentes mediante el General Sequential Patterns (GSP)[4].

Durante la ejecución del algoritmo se va realizando un comprobación de coincidencias entre las secuencias frecuentes y cada población de soluciones que es generada por el algoritmo genético. Si hay una coincidencia se usa el peso del nodo correspondiente en ese momento y se suma al score.

Para comprobar las coincidencias se aprovecha la estructura del resultado devuelto por el algoritmo genético. Se trata de una estructura en bloque y esto permite con cierta facilidad realizar esa comparativa.

Para realizar la comparación se va analizando cada bloque moviéndose por aquellos nodos que le suceden, comprobando en cada transición antes que nada cual o cuales serán el nodo hijo o los nodos hijos que le preceden, esta comprobación estará definida por los operadores (and,xor,Loop) que van a indicar si los nodos hijos que preceden se cumplen a la ves, se cumple uno u otro, se pueden cumplir todos, etc

2.7 Modelo resultante

El modelo que se obtiene será la combinación de ambas minerías complementadas la uno con la otra, de forma que la minería de datos mediante el algoritmo GSP es capaz de obtener las secuencias que más se repiten y aprovechar estas para obtener un valor de un score y a su vez filtrar en ProcessTree las soluciones más cercanas al flujo de trabajo que queremos.

Nuestro modelo resultante será una red Petrinet que representará un workflow proveniente de la fusión del Algoritmo Genético (Process Tree) con GSP. Mediante una función Score conseguiremos conectar las secuencias frecuentes con el procesamiento de los logs permitiendo percibir información y patrones que por sí solos podrían perderse.

2.8 Beneficios del modelo resultante

2.8.1 Modelo Normal

Al poder obtener una representación de poblaciones de soluciones con un ranking y score asociados provenientes de un conjunto de datos o patrones en una red Petri, se facilita la comprensión de los resultados, su análisis y el descubrimiento de información oculta.

En este modelo se obtiene el ranking con aquellas secuencias que más repeticiones tiene permitiendo beneficios como:

- *Obtener un workflow compatible a partir de un conjunto de datos*
- *Detectar acciones más frecuentes que se realizan en un proceso permitiendo ahorrar costes o tiempo.*
- *Anticiparse o prevenir consecuencias de un evento o suceso.*
- *Prever caídas de un sistema.*
- *Rediseñar un flujo de trabajo para lograr un objetivo.*

2.8.2 Modelo Inverso

Al igual que en el modelo normal el modelo inverso se configura siguiendo los mismos criterios, pero en este caso por el contrario el score puntuara cada vez que no haya una coincidencia.

Con esto obtendremos un ranking de población de soluciones que representará aquellos workflows en los que menos posibilidad de encontrarnos secuencias tenemos.

Cuando se habla de conceptos abstractos a veces resulta más útil plantear un problema al revés por esto este modelo permite obtener beneficios como:

- *Obtener un workflow en los que no se encontraran un conjunto de datos*
- *Detectar acciones menos frecuentes permitiendo ahorrar costes o tiempo.*
- *Crear workflows a medida para que no se generen determinados datos o patrones.*
- *Generar workflows que permitan evitar la caída de un sistema.*
- *Rediseñar un flujo de trabajo para lograr un objetivo.*

Capítulo 3. Desarrollo

Para el desarrollo del prototipo de plug-in para la plataforma ProM se seleccionaron diferentes librerías y algoritmos.

Por un lado se sustenta sobre un algoritmo genético concretamente el (Process Street Genetic miner) *donde se calculan las posibles poblaciones de soluciones correspondientes a un log. La elección de este algoritmo viene justificada ya que este devuelve una estructura en bloque que hace mucho más sencillo la comparación con las secuencias frecuentes.*

Por otro lado se utiliza las librerías de asociaciones de Weka como es el GSP (Generalized Sequential Patterns) para obtener las secuencias frecuentes y la red Petrinet de ProM para su visualización.

Capítulo 4. Implementación del plugin

4.1 Descripción de la interfaz

Cuando se inicia Prom 6 se observan 3 pestañas, la primera es la del Worskpace (espacio de trabajo). Si es la primera vez que lo usamos, este se encontrará vacío, aquí se almacenan todos los datos con los que trabaja la herramienta tanto para entradas y salidas. Ejemplo en la Figura 2.2

La segunda pestaña es la de Action (acción), aquí se listan todos los plugins instalados, incluido el desarrollado para este proyecto. A continuación accedemos al área de búsqueda y escribimos Genetic Miner GSP Plugin para encontrar el plugin que se ha desarrollado. Una vez encontrado y seleccionado, se observa que en los datos de entrada se reconoce los archivos importados para tal fin como se puede observar en la Figura 4.6.

En caso de no tener ningún fichero importado se debe proceder a su importación conociendo el tipo de entrada necesario para el funcionamiento de plugin. Para ejecutar el plugin necesitamos importar un fichero xes(xlog) que será el que usar el algoritmo genético. También necesitaremos un objeto instances, que se obtiene de transformar el mismo fichero log utilizado a un fichero arff (tipo de ficheros con los que trabaja Weka). Este fichero se usará para obtener las secuencias frecuentes y aplicarla para obtener el valor de la función Score.

Ahora sólo hay que importar el fichero generado pulsando el botón import de la Figura 4.2.

Por último, ejecutar el plugin presionando el botón Start de la venta de Acción que se encuentra en la Figura 4.7.

4.1.1 Botón import

Este botón que se muestra en la Figura 4.2, se debe utilizar para importar los

objetos descritos, anteriormente. Estos se obtienen con la importación de un fichero con extensión arff para las secuencias frecuentes y otro fichero normalmente con extensión xes para ejecutar el algoritmo genético.

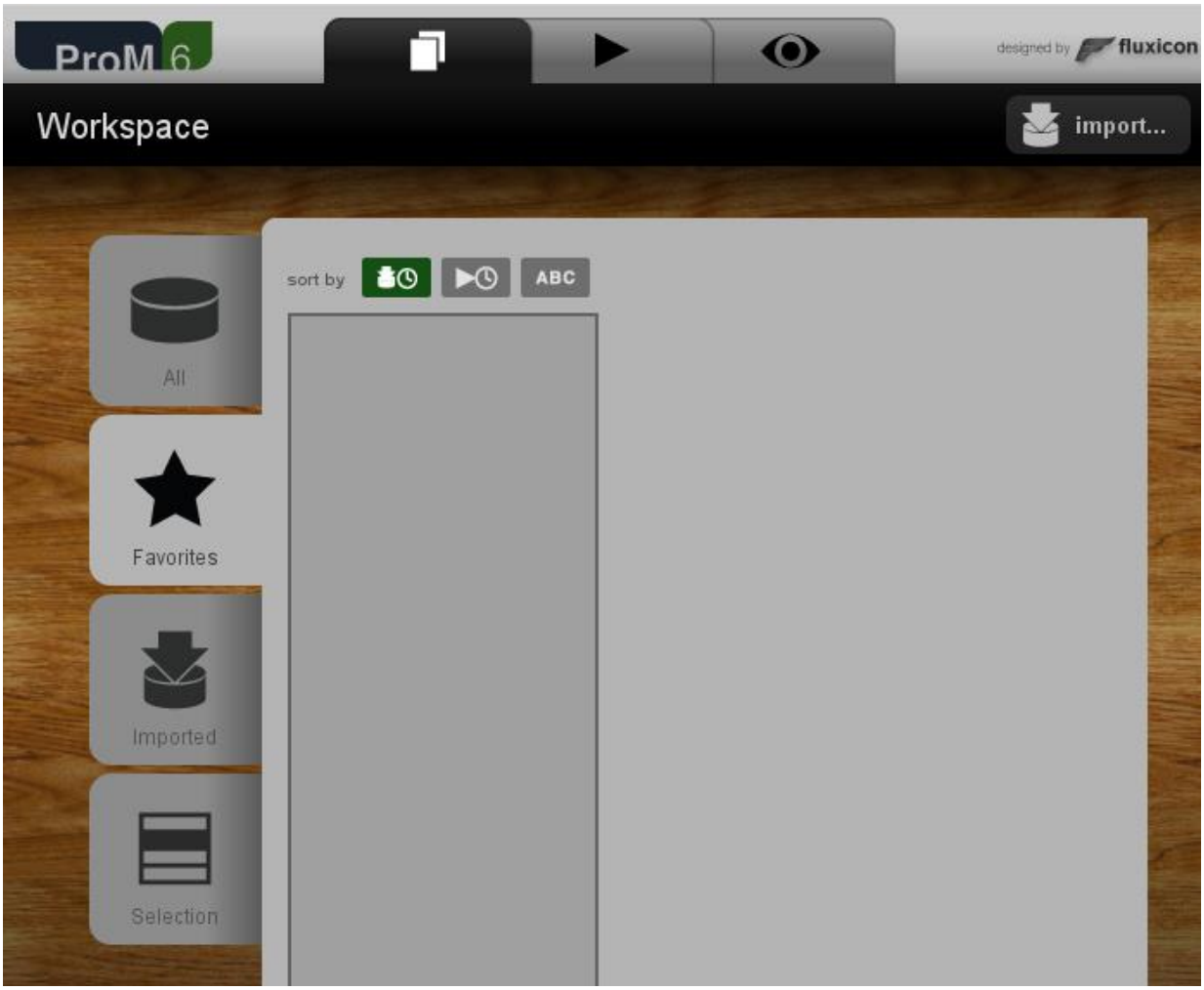

Figura 4.1. *Interfaz del Prom para importar archivos.*

4.1.2 Botón View

Una vez importado el objeto podremos ver su contenido presionando el botón view como se muestra en la Figura 4.3.

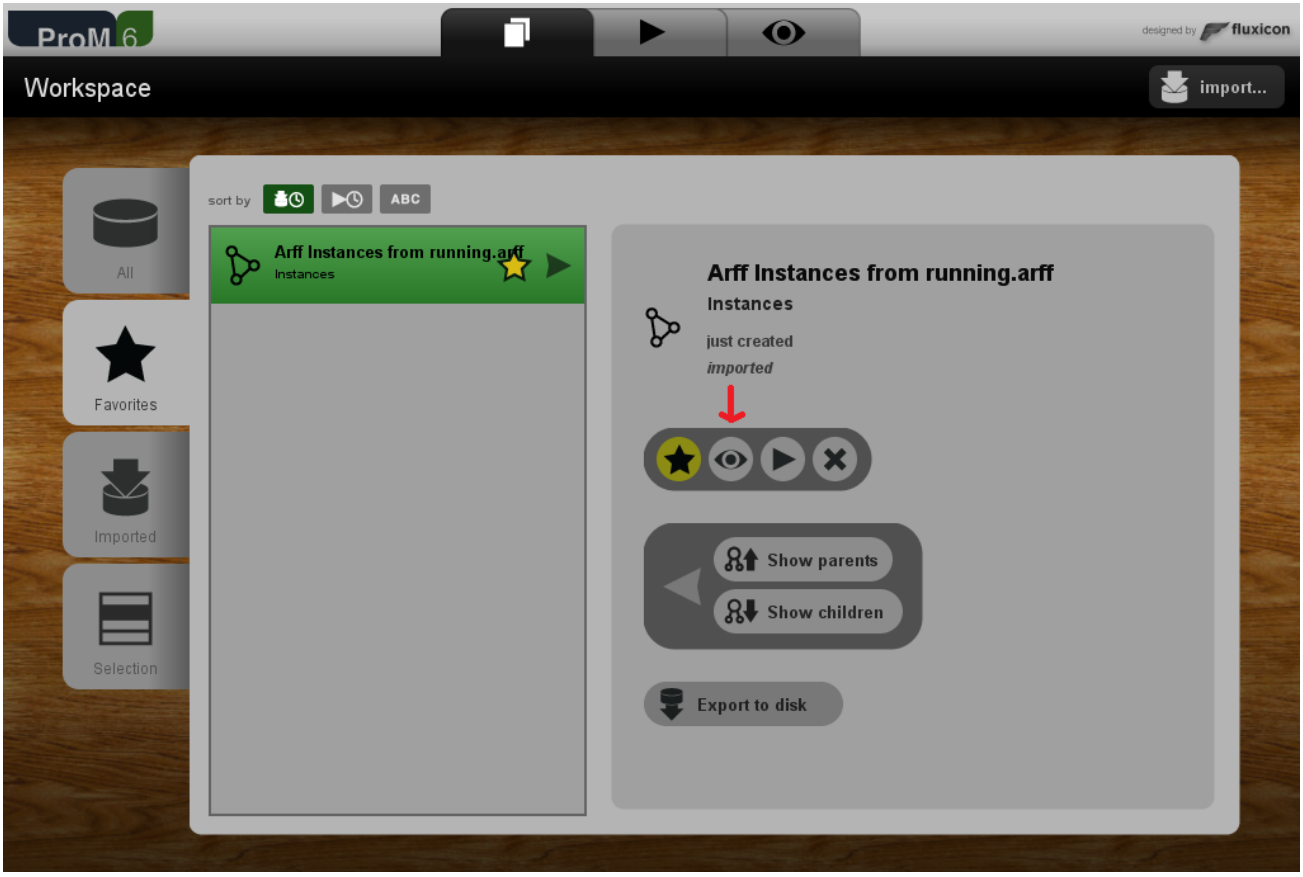

Figura 4.2. *Botón view. Para visualizar los objetos.*

Al presionar el botón view sobre un objeto, ProM cambiará a la pestaña de visualización y nos mostrará el contenido del objeto. En la Figura 4.4 se aprecia el visualizer diseñado para los objetos instances.

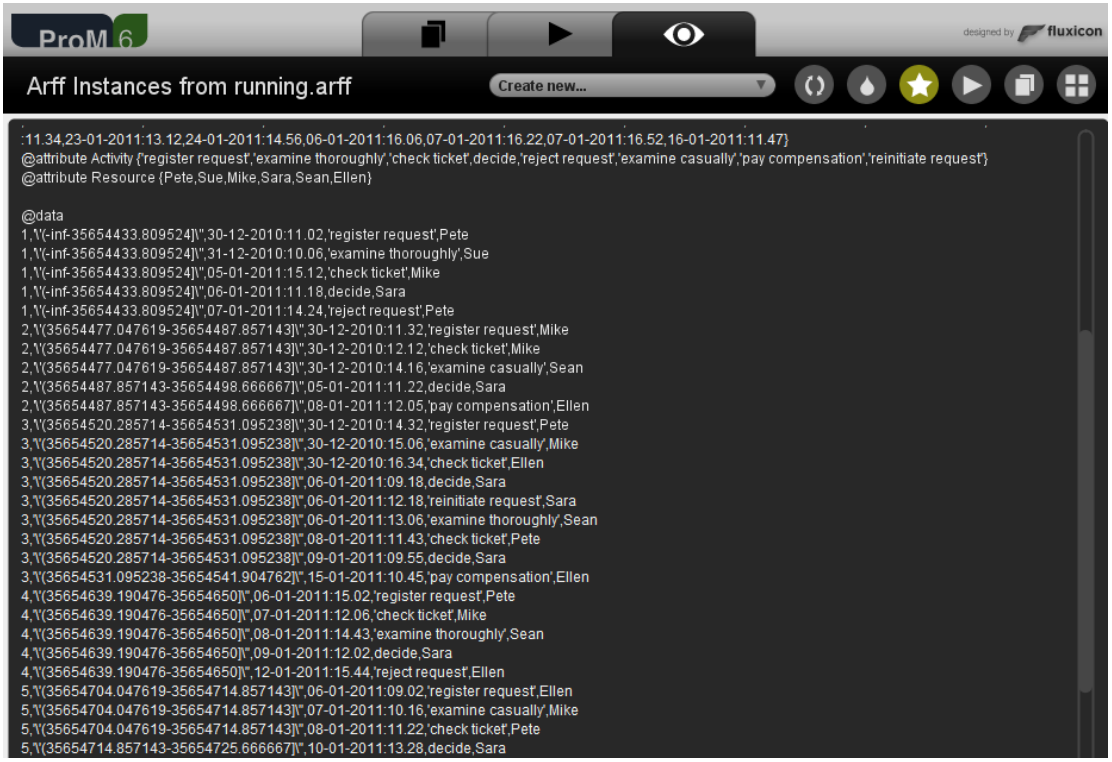

Figura 4.3. *Ventana de visualización de objetos.*

4.1.3 Botón Action

Este botón cambia el ProM a modo de acción mostrando los plugins que se pueden ejecutar sobre el objeto seleccionado.

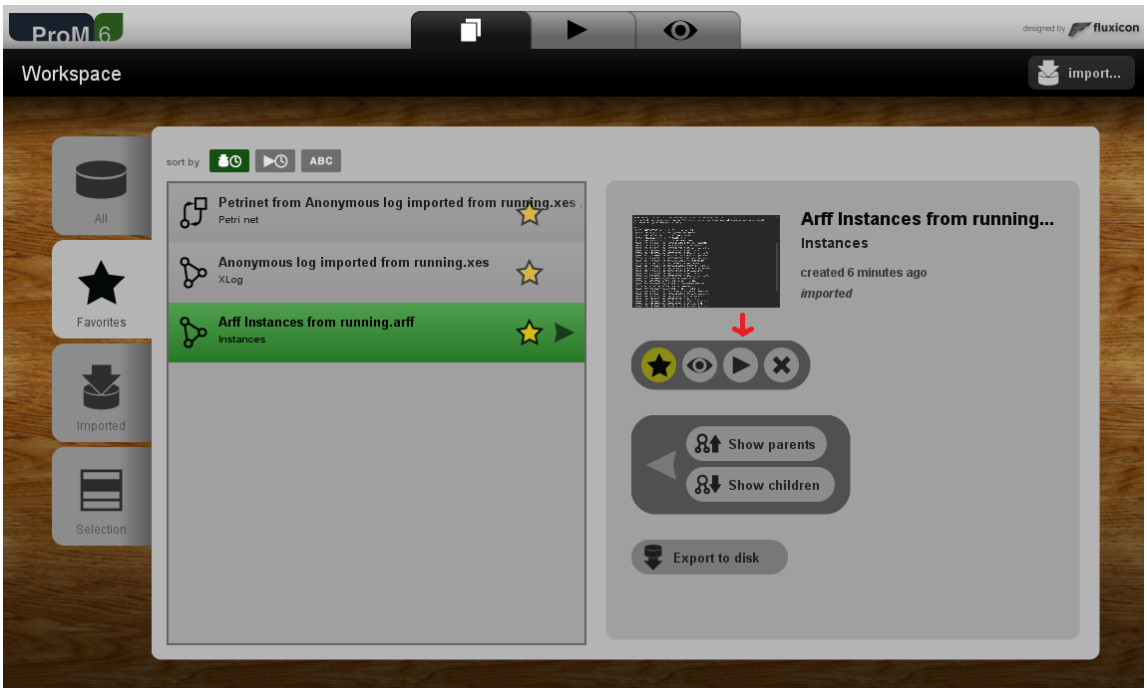

Figura 4.4. *Botón action de un objeto.*

En la siguiente foto se muestra la ventana Actions, donde se listan todos los plugins instalados y que se pueden ejecutar en la herramienta. Mediante la barra de búsqueda es posible filtrar la lista para encontrar el plugin deseado. Al hacer clic sobre un plugin, se muestra los objetos requeridos de entrada y los de salida.

En este proyecto aparecen dos plugins. El "normal" y el inverso, la diferencia entre ambos, es que el normal devuelve aquellas soluciones donde más se repiten determinados patrones y el inverso por el contrario devuelve las que menos.

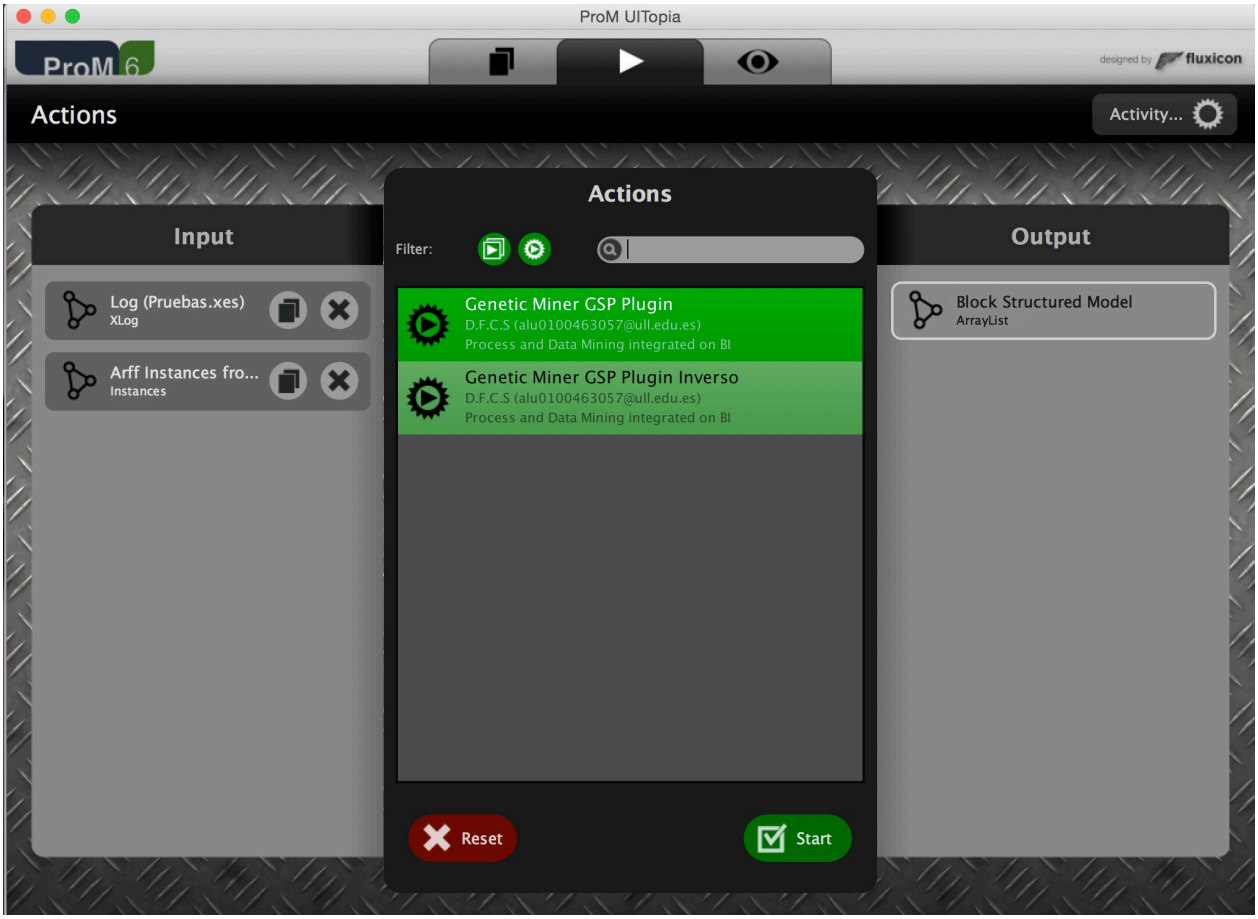

Figura 4.5. *Ventana de Acciones filtradas por "Genetic Miner GSP" con los objetos de entrada seleccionados para ejecutar el plugin, pulsando el botón Start.*

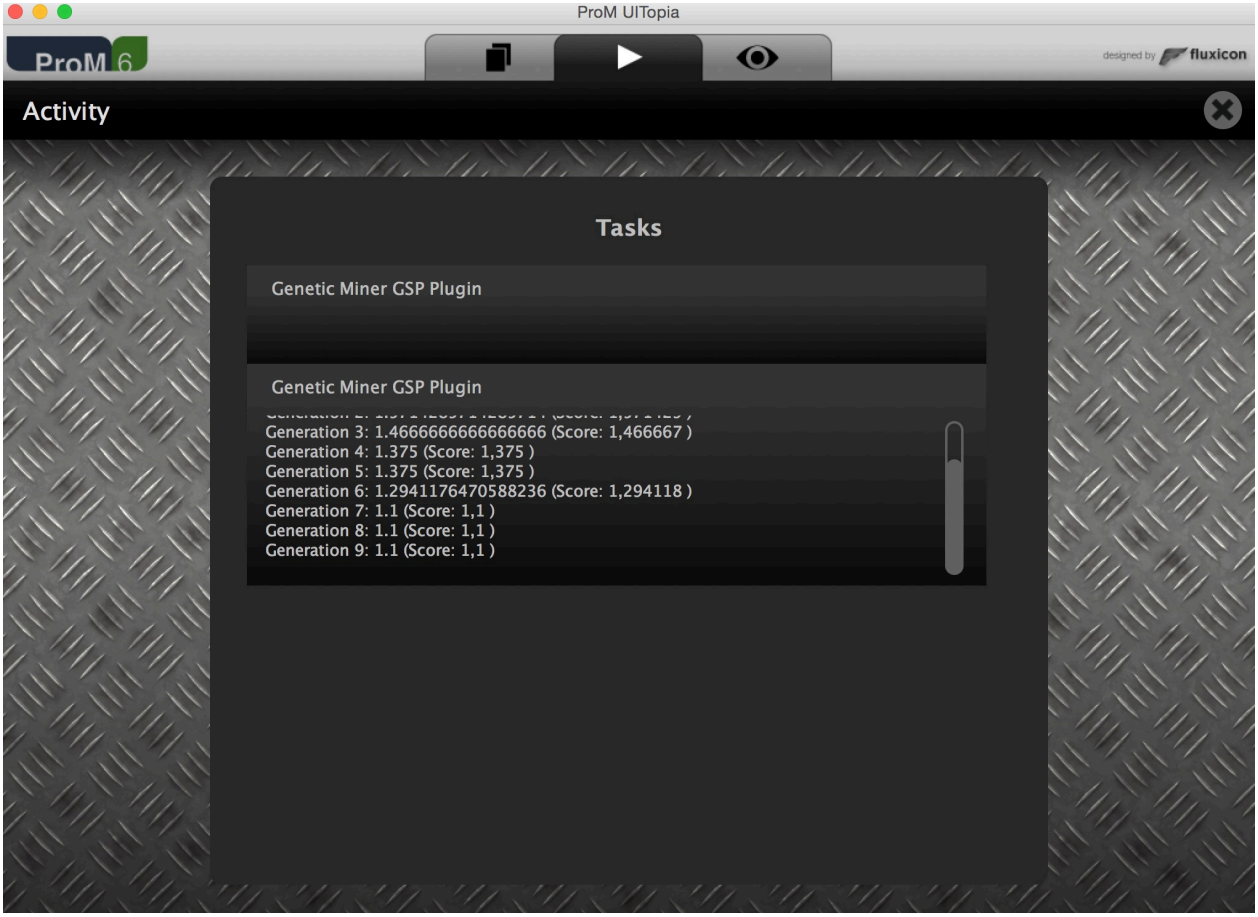

Figura 4.6. *Ventana ejecución del plugin Genetic Miner con los objetos seleccionados.*

4.1.4 Visualización del resultado

Una vez configurados los parámetros, se ejecutará el algoritmo, generará el objeto resultante y automáticamente abrirá el visualizador del mismo.

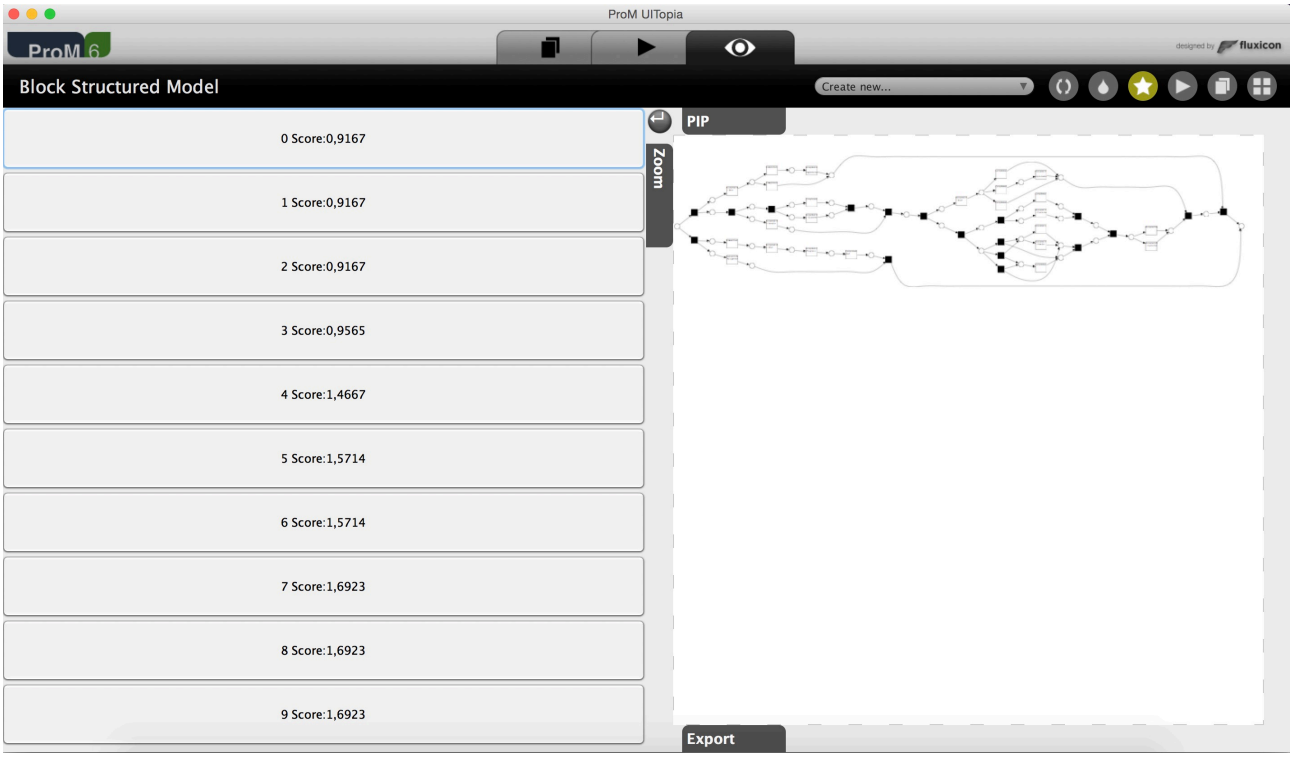

Figura 4.7. *Ventana de Visualización del resultado final del GSP plug-in.*

En la imagen superior se observa el grafo (red de Petri) con la selección de una de las soluciones candidatas y en el lado derecho se observa el ranking de las mejores soluciones posibles obtenidas de la población de soluciones dadas por el Genetic Miner

4.1.1 Funcionamiento

Con la combinación de la función score y la población de soluciones obtenida se invoca al mecanismo de visualización que facilita ProM para las redes de Petri y se le anexa un nuevo componente gráfico que estará formado por un ranking con una población de soluciones, de mejor a peor y un porcentaje de fiabilidad asociado (score) que dará una idea de cuál es el mejor flujo de trabajo para un conjunto de datos.

Para el funcionamiento del modelo se requiere un log de registro y una instancia de datos. El ProM permite con diversos plugins obtener estos requerimientos a partir de un fichero csv o incluso un xls.

El log de registro será un archivo del tipo xes y el instances generalmente será un arff (aunque podrá procesarse directamente desde un csv).

Capítulo 5.
Pruebas y resultados

5.1 Comprobación de funcionamiento

En esta primera prueba se utilizó un ejemplo con datos sintéticos pequeños, para la construcción del plugin y comprobar su funcionamiento.

A continuación se establece la salida del mismo en su ejecución por consola:

```
GeneralizedSequentialPatterns
=============================

Number of cycles performed: 4
Total number of frequent sequences: 26

Frequent Sequences Details (filtered):

- 1-sequences

[1] <{register request}> (6)
[2] <{check ticket}> (6)
[3] <{decide}> (6)
[4] <{A}> (6)
[5] <{B}> (6)
[6] <{C}> (5)

- 2-sequences

[1] <{register request}{check ticket}> (6)
[2] <{register request}{decide}> (6)
[3] <{register request}{A}> (6)
[4] <{register request}{B}> (6)
[5] <{register request}{C}> (5)
[6] <{check ticket}{decide}> (6)
[7] <{check ticket}{B}> (6)
[8] <{A}{decide}> (6)
[9] <{A}{B}> (6)
[10] <{A}{C}> (5)
[11] <{decide,B}> (6)
```

- 3-sequences

[1] <{register request}{check ticket}{decide}> (6)
[2] <{register request}{check ticket}{B}> (6)
[3] <{register request}{decide,B}> (6)
[4] <{register request}{A}{decide}> (6)
[5] <{register request}{A}{B}> (6)
[6] <{check ticket}{decide,B}> (6)
[7] <{A}{decide,B}> (6)

- 4-sequences

[1] <{register request}{check ticket}{decide,B}> (6)
[2] <{register request}{A}{decide,B}> (6)

Generaciones	Score original	Score del GSP
0	Fitness:8.028241977689392	GSP:6.489260248818181
1	Fitness:0.0	GSP:6.489260248818181
2	Fitness:6.508313154123814	GSP:5.3066505232990515
3	Fitness:6.508313154123814	GSP:5.3066505232990515
4	Fitness:5.752358669820789	GSP:4.701886935856631
5	Fitness:4.289683181978578	GSP:3.5317465455828625
6	Fitness:4.289683181978578	GSP:3.5317465455828625
7	Fitness:4.289683181978578	GSP:3.5317465455828625
8	Fitness:3.511124521488227	GSP:2.908899617190582
9	Fitness:2.334478895717286	GSP:1.9675831165738287

Leyenda:
[1] Secuencia Número 1
{register request}
{check ticket}
{decide}
{A, B , C}

Vemos la salida del GSP junto al fitness original del Process Tree (Genetic Miner) y el nuevo score obtenido. Vemos que las puntuaciones varían, siendo la del score mas afinada a los criterios de la fuente de datos de entrada.

En la generación 1 el score original es 0 sin embargo en el GSP da un valor alto sobre el workflow de la solución que sin la combinación de la técnica se

perdería.

5.2 Probando algoritmo ProcessTree

Para esta prueba se han utilizado datos reales del BPI Challenge 2013 concretamente se ha usado un fichero que contiene la información sobre el registro de incidencias en un sistema.

Este es el resultado que podemos obtener utilizando sólo el plugin con el algoritmo ProcessTree, a partir de un log de registro (xes)

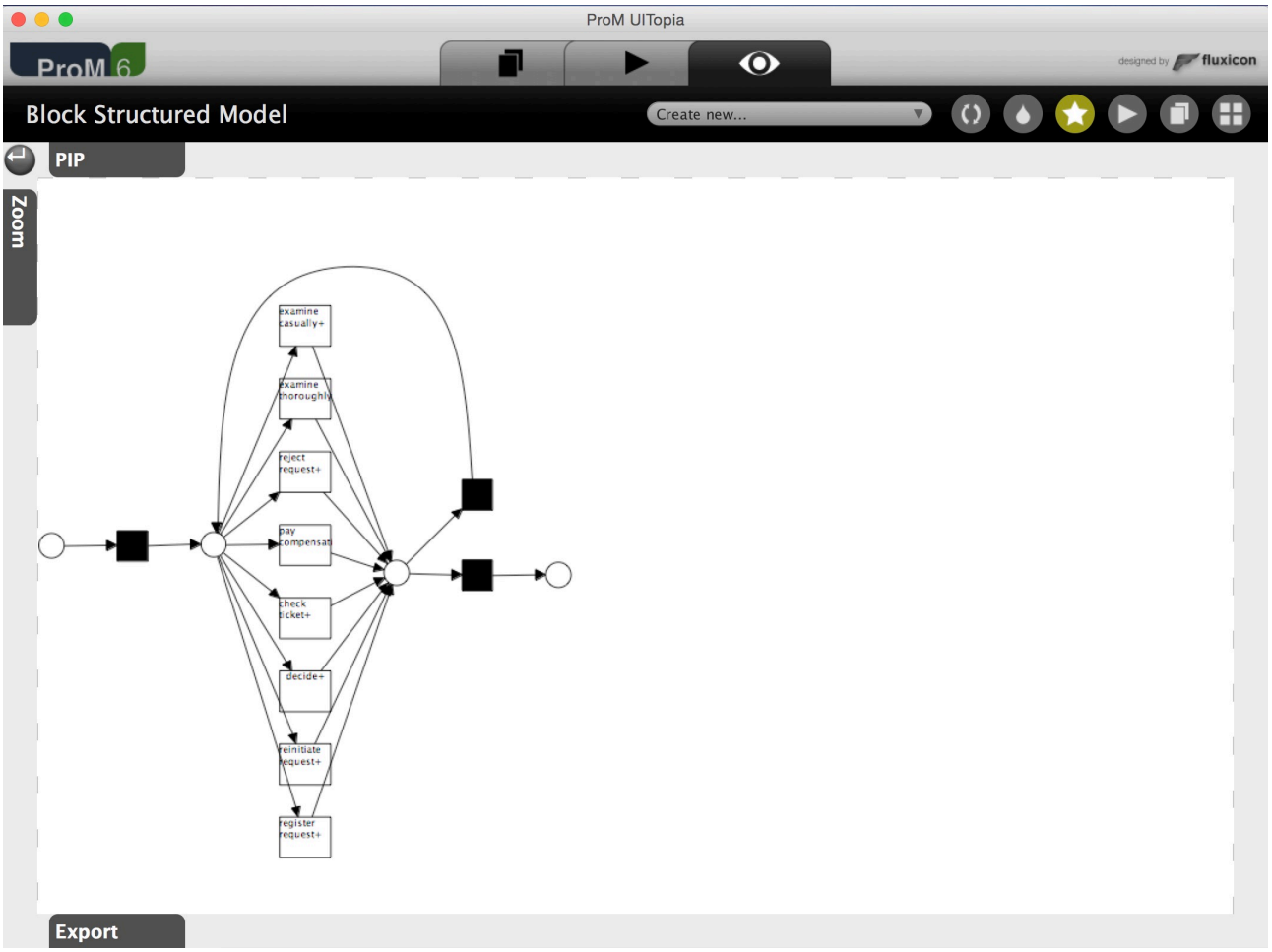

Figura 6.1. *Ejecución del Plugin ProcessTree con su fitness original.*

Los resultados tras aplicarse un algoritmo genético, muestran el worklflow de una posible solución obtenida de forma aleatoria de una población inicial de soluciones.

Pero aunque usar un algoritmo genético a priori puede acercarnos a una

posible solución, esta no tiene porque ser una solución óptima. Al estar trabajando con fuentes de información es importante cualquier dato o detalle que pueda aportarnos utilidad y significado a la información, para poder comprender y sacar conclusiones.

Por ello en las siguientes pruebas se usará el plugin completo con el modelo resultante que se describe como objetivo de este proyecto, esto permitirá combinar minería de datos y de procesos, permitiendo obtener una mejor información.

5.3 Combinando Minería de datos y de procesos

Para esta prueba también se han utilizado los datos reales del BPI Challenge 2013.

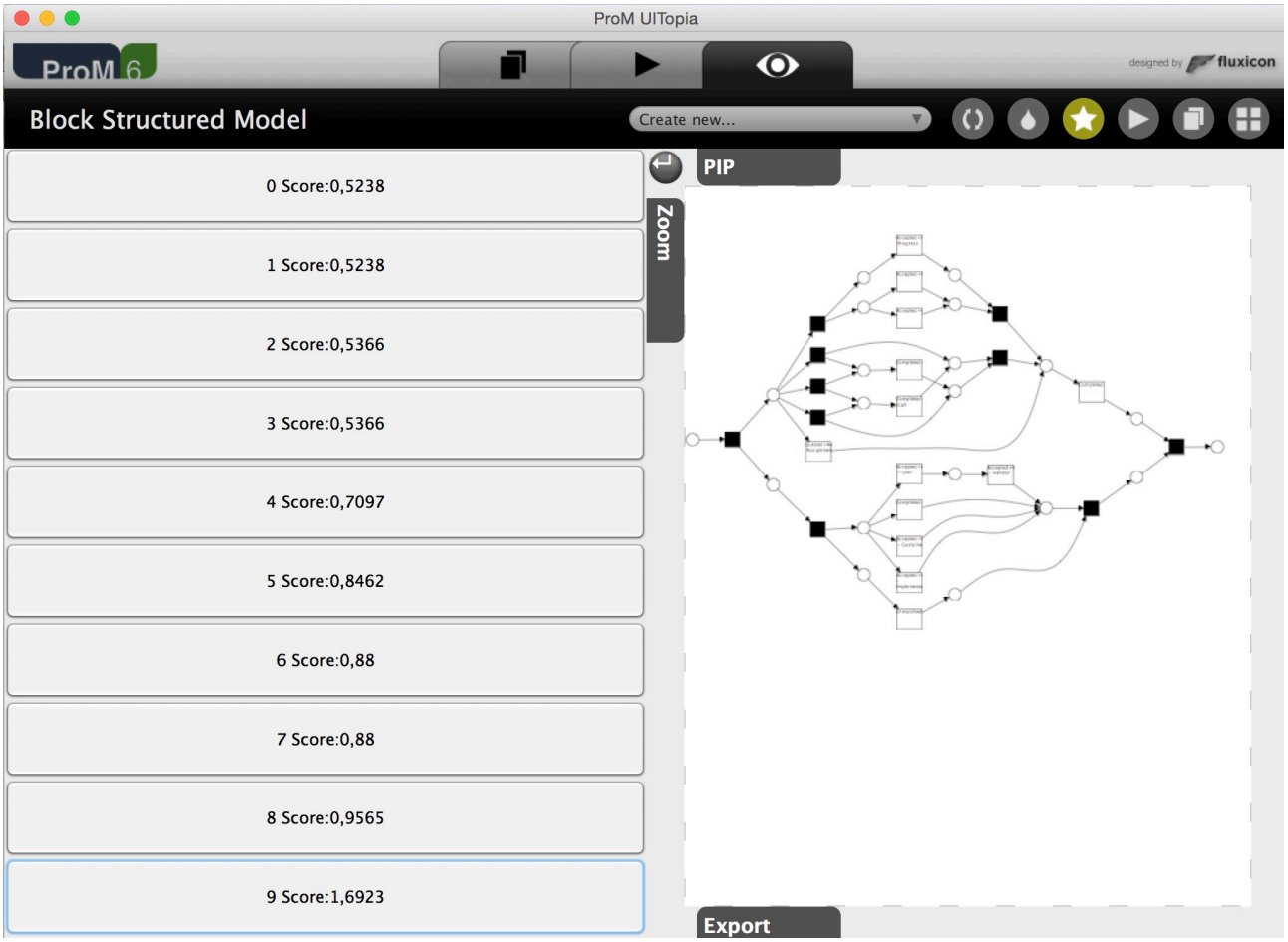

Figura 6.2. *Ejecución del Plugin combinando minería de datos y procesos.*

Tras la ejecución del algoritmo, se obtiene un ranking de soluciones ponderado por un score, siendo el valor 0 el objetivo deseado (donde más se

repiten las secuencias) y 1 el que se quiere desechar (donde menos se repiten las secuencias).

Hay que dejar constancia de que al ser un algoritmo genético siempre trabajará sobre una población de soluciones inicial la cual en cada ejecución genera de forma aleatoria, una nueva población de soluciones

En este caso hemos obtenido 4 soluciones candidatas en las que se repiten secuencias frecuentes.

Por lo tanto gracias a este resultado, podemos usar el plugin para intentar obtener workflows en los que se repitan patrones y prever resultados.

En este caso estamos hablando de un registro de incidencias, el algoritmo genético genera workflows posibles de incidencias generadas y combinado con el algoritmo GSP, se obtiene aquellos workflows en los que más posibilidad habrá de generarse una incidencia.

Es decir con esto se puede concluir que gracias a la combinación de las técnicas esta herramienta es capaz de predecir y resolver flujos de trabajo para un conjunto de patrones dado.

5.4 Invirtiendo el Proceso.

Para esta prueba también se han utilizado los datos reales del BPI Challenge 2013 y se ha usado el plugin inverse.

Es evidente que si el plugin es capaz de predecir aquellos workflows más cercanos a los patrones dados, también será capaz de resolver aquellos workflows para los que no ocurrirían determinados patrones.

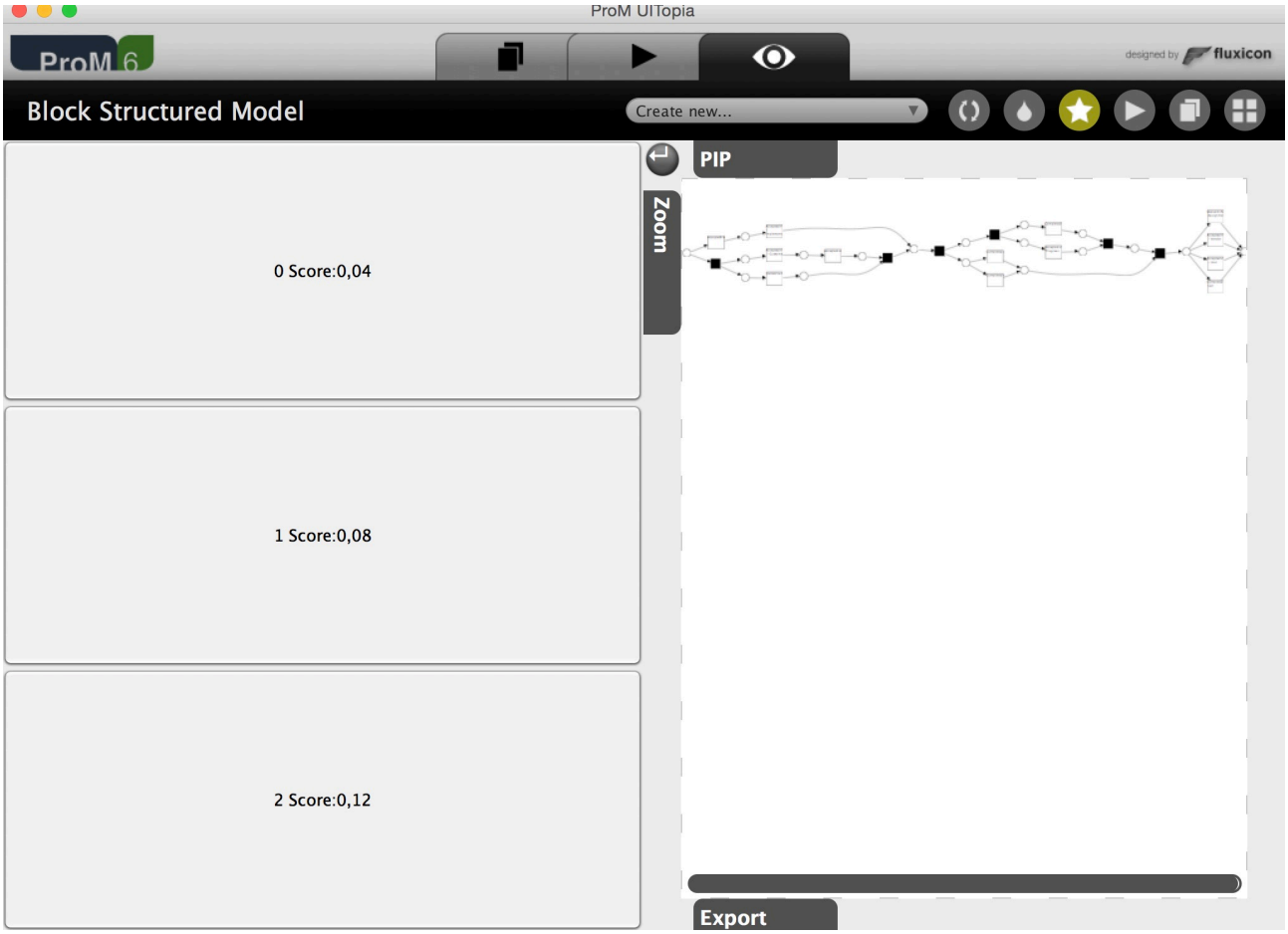

Figura 6.3. *Ejecución del Plugin combinando invirtiendo el proceso.*

En el caso del registro de incidencias, esto podría ser útil para predecir los casos ideales en los que se producirían pocas o ninguna incidencias.

5.5 Validando con un ejemplo de casos reales.

Usando un ejemplo donde se registran logs del funcionamiento de una tienda en cuanto a la atención de reclamaciones se refiere, introducimos la siguiente fuente de entrada como patrones:

1,30-12-2010:11.02,'register request',Pete

1,31-12-2010:10.06,'register request',Sue

1,05-01-2011:15.12,'register request',Mike

1,06-01-2011:11.18,'register request',Sara

1,07-01-2011:14.24,'register request',Pete

2,30-12-2010:11.32,'register request',Mike

2,30-12-2010:12.12,'register request',Mike

2,30-12-2010:14.16,'register request',Sean

2,05-01-2011:11.22,'register request',Sara

2,08-01-2011:12.05,'register request',Ellen

3,30-12-2010:14.32,'register request',Pete

3,30-12-2010:15.06,'register request',Mike

Estos patrones son el objetivo a conseguir. Queremos obtener un workflow donde se den los menos casos posibles de 'register request', para saber como debe gestionarse un negocio y tener el menor numero de reclamaciones posibles.

El log se ha generado desde una fuente de información CSV y en base a esa hemos añadido esos patrones para lograr el objetivo.

Ejecutando el plugin inverso obtenemos como resultado un workflow, con un score prácticamente con valor 0, que permite como se esperaba siguiendo ese flujo obtener el menor numero de 'register request'.

A continuación se ve el grafo que lo valida.

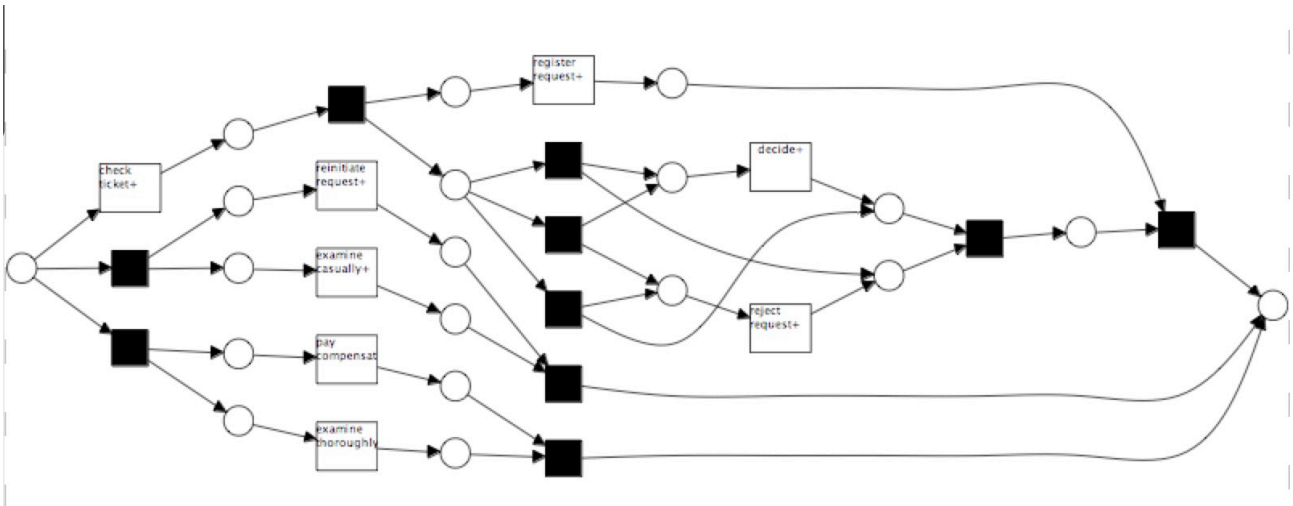

Figura 6.4. *Ejecución del Plugin validando un caso.*

Capítulo 6. Conclusiones y líneas futuras

*La **minería de datos** es un campo de las ciencias de la computación referido al proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos. Utiliza los métodos de la inteligencia artificial, aprendizaje automático, estadística y sistemas de bases de datos. El objetivo general del proceso de minería de datos consiste en extraer información de un conjunto de datos y transformarla en una estructura comprensible para su uso posterior. Además de la etapa de análisis en bruto, que involucra aspectos de bases de datos y de gestión de datos, de procesamiento de datos, del modelo y de las consideraciones de inferencia, de métricas de Intereses, de consideraciones de la Teoría de la complejidad computacional, de post-procesamiento de las estructuras descubiertas, de la visualización y de la actualización en línea.*

La minería de datos se basa en las bases de datos o logs y la minería de procesos en logs siendo ambas dos líneas diferentes. Pero con este trabajo se ha logrado demostrar que es posible combinar ambas técnicas y lograr una conjunción de diferentes técnicas con el fin de obtener mejor conocimiento y mejor utilidad sobre grandes cantidades de fuentes de datos e información existente que resultan inútiles por si solas.

En las pruebas se puede observar que el prototipo de plug-in desarrollado es capaz de combinar las secuencias frecuentes con el flujo de trabajo resultando de ello un único modelo que es capaz de aportar mayor conocimiento sobre la información y los datos. Los resultados obtenidos además de ser buenos, son una muestra de los beneficios que implica la combinación de las técnicas como es la capacidad de predecir workflows para un resultado deseado.

Resulta obvio, que aún queda mucho camino por recorrer. Se puede explotar aún más el potencial de las técnicas conocidas, se pueden mejorar, combinar o incluso crear nuevas.

Como línea futura se podría dotar a este plugin de la capacidad de configurar

determinados parámetros como por ejemplo el tamaño de la población, el score deseado, el número de generaciones, etc. Todos estos parámetros se pueden cambiar a mano antes de cada ejecución pero si se construye un menú de configuración previo, será posible definir y concretar aún más la búsqueda y la afinidad de la información para encontrar un flujo de trabajo aún mas acorde según el tipo y cantidades de datos con los que estemos tratando.

Capítulo 7.
Summary and Conclusions

Data mining is a field of computer science refers to the process that attempts to discover patterns in large volumes of data sets. Use the methods of artificial intelligence, machine learning, statistical and database systems. The overall objective of data mining process is to extract information from a data set into an understandable structure for later use. In addition to the analysis stage rough, involving aspects of databases and data management, data processing, model and inference considerations, metrics interests, considerations of computational complexity theory , post-processing of the discovered structures, visualization and online update.

Data mining is based on databases or logs and logs mining processes in both of two different lines. But this work has succeeded in demonstrating that it is possible to combine both techniques and achieve a combination of different techniques in order to obtain better knowledge and better useful information from large amounts of existing information sources that are useless by themselves.

In tests it can be seen that the prototype developed plug-in is able to combine frequent sequences with workflow resulting from this single model is able to provide greater knowledge of the information and data. The results besides being good, are an example of the benefits that involves combining techniques such as the ability to predict workflows to a desired result.

Obviously, much it remains to be done. You can further exploit the potential of the known techniques can be improved, combine or even create new ones.

As future line could provide this capability plug set certain parameters such as the population size, the desired score, the number of generations, etc. All these parameters can be changed by hand before each run but a previous configuration menu is built, it will be possible to define and specify further search and affinity of information to find a workflow even more according to the type and amounts of data with which we are dealing..

Capítulo 8. Presupuesto

El proyecto se ha desarrollado utilizando única y exclusivamente herramientas de software libre. Por lo que los costes del proyecto sólo se corresponden con los gastos de mano de obra. A continuación se muestra una tabla con el presupuesto fijado:

Referencia	*Cantidad/horas*	*Coste*	*Desglose*
Software	***8***	*0 €*	*Eclipse Luna* *Object Aid* *SVN Subversion* *Java 1.6* *Prom 6.1* *Weka 3.4* *ProcessTree* *GSP*
Ordenador	*1*	*356,34€*	
Mano de obra	*360*	*9000 €*	
Total	*9356,34 €*		

Tabla 8.1. *Tabla resumen del costo del presupuesto.*

JUSTIFICACIÓN DEL PRESUPUESTO

El proyecto se ha realizado en un total de 60 días, de los cuales se han dedicado una media de unas 4 horas/día. Se ha supuesto que el costo de la

mano de obra por hora es de unos 25 €/h para un ingeniero informático.

Calculando el coste de mano de obra obtenemos un precio de 7200€.

Se consiguió ahorrar costes ya que todo el proyecto se ha realizado con software y herramientas libres.

Respecto al coste de equipos como ordenadores se ha calculado la amortización correspondiente ya que se disponía de él desde un principio.

La amortización se calcula de la siguiente manera:

1359 euros x 26% = 356,34 euros

Apéndice A.
Objetos y diagramas

En este apartado tenemos todas las clases y objetos de las que se compone el grueso del plugin, en el se puede ver como se configura el proyecto a nivel de estructura

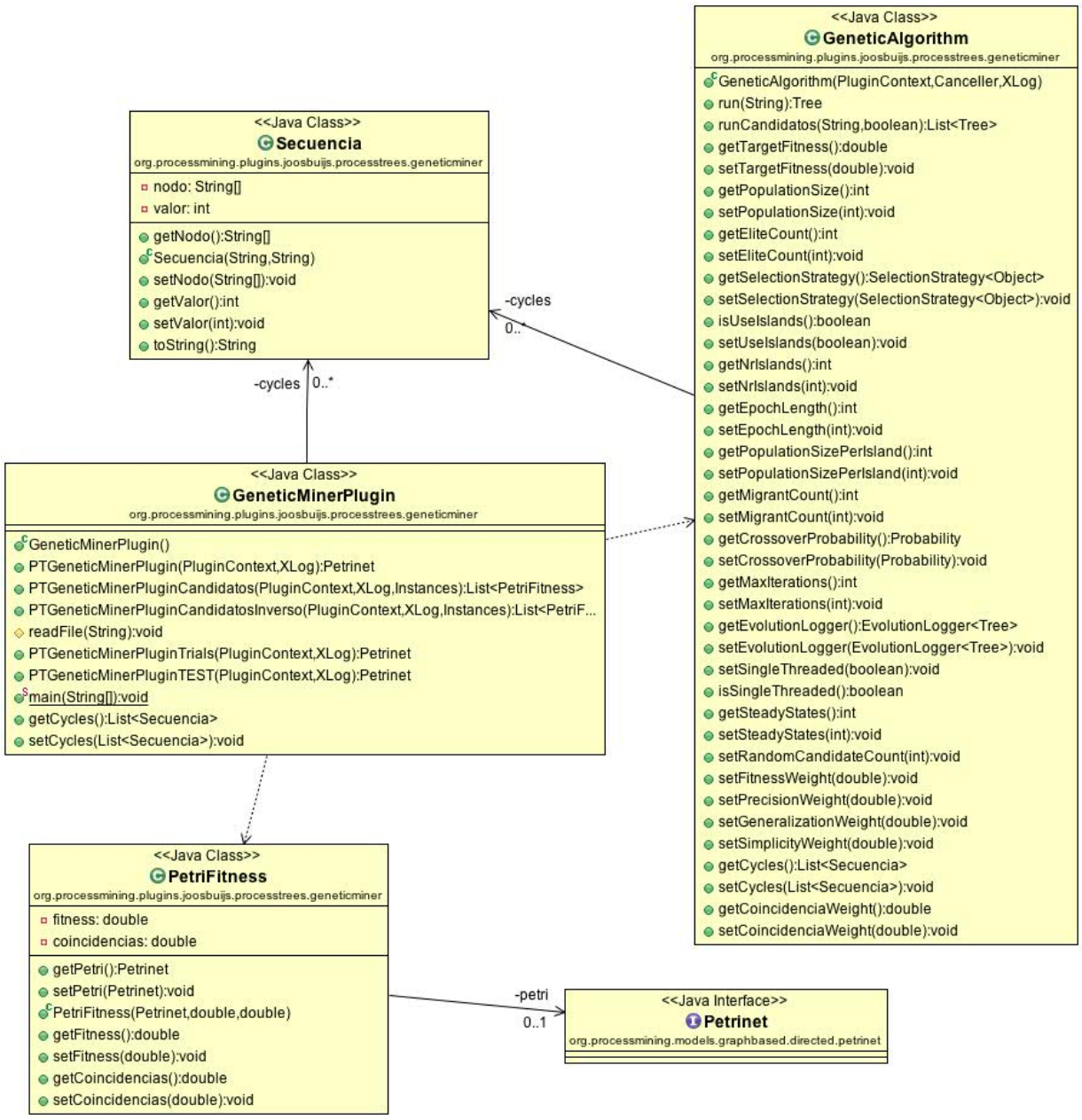

Figura A.1. *Diagrama de Clases de los objetos creados para guardar la información del plug-in, con sus métodos públicos y sus relaciones.*

Apéndice B.
Datasets de pruebas

Enlace a los ficheros fuentes para los datasets de pruebas

https://github.com/bonedark/TFG15/tree/master/resources

Bibliografía

[1] *Weka Official Page, Machine Learning Group at the University of Waikato.* http://www.cs.waikato.ac.nz/ml/weka/index.html.

[2] https://svn.win.tue.nl/trac/prom/wiki/setup/HowToBecomeAProMDevel oper .

[3] Aalst, Wil M. P. van der. *Process Mining Discovery, Conformance and Enhancement of Business Processes.* Editorial Springer, 2011.

[4] Agrawal, Ramakrishnan Srikant and Rakesh. *Mining Sequential Patterns: Generalizations and Performance Improvements.* IBM Almaden Research Center.

[5] «Data Mining: Mining sequence patterns in transactional databases.» http://www.cs.nyu.edu/courses/spring08/G22.3033-003/8timeseries.ppt .

[6] *Desarrollo Ágil con Kanban.* http://www.desarrolloweb.com/articulos/desarrollo-agil- kanban.html.

[7] H.M.W. (Eric) Verbeek, R. P. Jagadeesh Chandra Bose. *ProM6 Tutorial.* 2010.

[8] IEEE Task Force on Process Mining. Manifiesto sobre Minería de Procesos. http://www.win.tue.nl/ieeetfpm/lib/exe/fetch.php?media=shared:pmm-spanish-v1.pdf .

[9] Microsoft Developer Network. Data Mining. . http://msdn.microsoft.com/es- es/library/ms174949.aspx.

[10] «Process Street Genetic miner.» https://www.tue.nl/fileadmin/content/faculteiten/win/Onderzoek/IS/1_Research/Meetings/AIS_Meetings/thesisAISPresentation_20140515.pdf. JoosBuijs.

[11] «Prom6 Official Page, Process Mining Group, Eindhoven Technical University. © 2010: .» http://www.promtools.org/doku.php.

[12] Rendon, Maurizio. Memoria TFG, Escuela Técnica Superior de Ingeniería y Tecnología, Universidad de La Laguna, 2014.

[13] Salazar, Daniel Fernández del Castillo. Github. Repositorio del Plug-in. https://github.com/bonedark/TFG15.

[14] Sinnexus Business Intelligence, Data Mining. http://www.sinnexus.com/business_intelligence/datamining.aspx.

[15] Westergaard, Michael. ProM6 plugin development. https://westergaard.eu/2012/11/prom-6-plug-in-development-part-1-basics/ .

[16] «Wikipedia Minería de Procesos.» http://es.wikipedia.org/wiki/Miner%C3%ADa_de_procesos.